Project Report

On

" P Y B O T "

Submitted for partial fulfillment of requirement for the Diploma

in Information Technology

By

Siddhesh S. Dharmadhikari-18IF018

Yashwardhan T. Ramchaware-18IF048

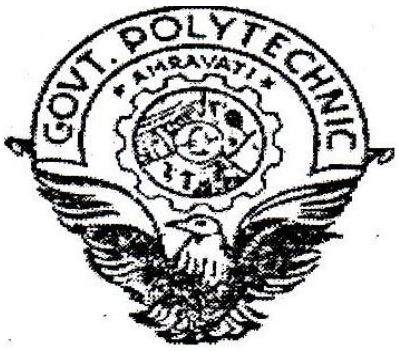
Atharva R. Nawathe-18IF045

Vedant P. Shah-18IF052

Shreyash K. Nakod-18IF043

Under the Guidance of

Prof. R.R. Bhoge



Department of Information Technology,

Government Polytechnic Amravati

(An Autonomous Institute of Government of Maharashtra)

2020-2021

Department of Information Technology

GOVERNMENT POLYTECHNIC AMRAVATI

(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that the Project titled

'PYBOT'

is submitted

By

Siddhesh S. Dharmadhikari-18IF018

Yashwardhan T. Ramchaware-18IF048

Atharva R. Nawathe-18IF045

Vedant P. Shah-18IF052

Shreyash K. Nakod-18IF043

In the partial fulfillment of the Diploma Course in

Information Technology

*During the academic year **2020-2021.***

(Name & Sign of the guide)	*Head,* *Dept. of Information Technology*	*(Name & Sign of the External Examiner)*	*Principal* *Government Polytechnic, Amravati*

ACKNOWLEDEMENT

It gives us immense pleasure in submitting the project report on topic **"PYBOT"** to our guide **Prof. R. R. Bhoge** who was a constant source of guidance and inspiration throughout developing the project and its report.

I am also very thankful to our Head of Department **Dr. P.P. Karde** and all the staff members of Information Technology Department, who have indirectly guided and helped me in development of this project.

We also express our sincere gratitude to our honorable principal **Dr.R.P.Mogre** sir for providing us necessary facilities required during development of project.

Thanking You!

Siddhesh S. Dharmadhikari-(18IF018)

Yashwardhan T. Ramchaware-(18IF048)

Atharva R. Nawathe-(18IF045)

Vedant P. Shah-(18IF052)

Shreyash K. Nakod-(18IF043)

INSTITUTE'S VISION

To be a vibrant technical institute of global reputes contributing towards the needs of industries & societies.

INSTITUE'S MISSION

- To develop competent diploma engineers suitable for contemporary industrial environment.
- To inculcate socially accepted ethics & values among budding engineers.
- To nurture innovations and entrepreneurship.
- To produce engineers with psychomotor & cognitive skills committed to lifelong learning.

DEPARTMENT'S VISION

To ensure excellent education environment with the technical capability in the field of IT engineering to serve vibrant industry and society.

DEPARTMENT'S MISSION

- Facilitate students to learn the basics of IT Engineering entailing the technical gain.
- Train students with technical skills with rational capacity to meet the requirements of industry with technological aspect.
- Motivate the students for an advance knowledge in IT Engineering and other value added programs for their holistic development.
- Imparting ethical values, leadership and social value in students which transform them in good human being.
- Provide an academic environment and consultancy services to the industry and society in the area of IT & Computer Engineering.

INDEX

ABSTRACT

A personal voice assistant is the software that can perform task and provide different services to the individual as per the individual's dictated commands. This is done through a synchronous process involving recognition of speech patterns and then, responding via synthetic speech.

Through these assistants a user can automate tasks ranging from but not limited to internet surfing, tasks management and media playback. As the technology is developing day by day people are becoming more dependent on it, one of the mostly used platform is computer.

We all want to make the use of computers more comfortable, the traditional way to give a command to the computer is through keyboard and mouse or trackpad but a more convenient way is to give the command through voice.

Giving input through voice is not only beneficial for the normal people but also for those who are visually impaired who are not able to give the input by using a keyboard. For this purpose, there is a need of a voice assistant which can not only take command through voice but also execute the desired instructions and give output either in the form of voice or any other means.

This technology is very much popular these days, there are tremendous upgradations going on in this particular technology we have seen many examples like the assistants Alexa of Amazon, Cortana of Microsoft, Siri of Apple and so on, this number of bots and assistants goes on increasing.

Some of the above examples are not just assistants or bots but the AI implemented assistants or bot, but as we all know the first step towards making an intelligent bot/assistant is making a task completing assistant which can do variety of tasks which the maximum number of users does on regular basis.

Our this project is based on that variety doing of all tasks, we have implemented bunch of functionalities which can help the end-users to complete their regular basis variety of tasks with ease and automation. The presented report will give all the details of features and functionalities of our developed assistant/bot named 'Pybot'.

INTRODUCTION

Today the development of any type of bot systems that are able to organize a natural human-machine interaction (through voice, communication, gestures, etc.) are gaining in popularity. The usage of virtual assistants is expanding rapidly after 2017, more and more products are coming into the market.

Due to advancement in the technology many different features are being added in the mobile phone and desktops. To use them with more convenient and fun way we require a means of input which is faster and reliable at the same time.

In our project we use voice command to input the data into the system for that the microphone is used which convert acoustic energy into electrical energy. After taking the input there is a requirement to understand the audio signal for this google API is used. Different companies like google, apple use different API's for this purpose.

Recently I came in interaction with a bot on a discord server from which I got really impressed by seeing what type of tasks it can able to do for the end-users so that there would not be any kind of disturbance between the user and his other important work. It was interactive, polite to talk with and listens and perform the commands or tasks you give it. It was like having a personal assistant which will do your not so important daily basis tasks by just commanding.

The work on creating and improving such personalized assistants has been going on for a long time in various industries. These systems are constantly improving and improving, go beyond personal computers and have already firmly established themselves in various mobile devices and gadgets.

One of the most popular voice assistants are Siri, from Apple, Amazon Echo, which responds to the name of Alexa from Amazon, Cortana from Microsoft, Google Assistant from Google.

I worked on similar technology to make an easy to use bot which can be able to work anywhere as it is in a .py file, the independence of the file just makes it very useful to any kind of end-user. I kept it in this format to implement this file anywhere, on any operating system or on any kind of device as user desires, which eventually shows the flexibility of

the project. Various changes to the code can be made according to the user's requirement as it is in a python file. It can be uploaded to the official site of python as a module so that it can be implemented or modified by anyone desired to make a similar kind of bot.

These voice assistants work as your companion which performs your day by day task with minimum efforts and also help the user to function better by giving daily updates. It was after the recognition of importance of voice commands in day to day life that we have aimed to develop a personal assistant for desktop which will do every work from playing music to surfing on internet.

This bot contains plenty of features and functionalities which can make anybody's working, surfing on internet, accessing different files, accessing information from the internet easy and interactive. I tried to apply some new features which inbuilt assistants like Cortana won't be able to do so that this bot can stand out from other similar bots and assistants.

OBJECTIVES

The objectives of this project are-

- To achieve Speech-to-Text.
- Utilization of this project should be beneficial for human beings.
- Interactive and Easy to use for everyone.
- Do almost every regular basis task.
- Increase in accuracy of speech recognition.
- Time saving.
- Speed and automation of performing tasks.

LITERATURE SURVEY

- Desktop Voice Assistant for visually Impaired by Ankush Yadav, Aman Singh.
- AI based Voice Assistant Using Python by Ria Umahiya, Deepak Shende.

In the above mentioned projects, they made an assistant which performs some basic functionalities by commanding to it. They made this project using python but they have created an executable file of it which set limited boundaries to the project. There project does consist of 2-3 functionalities which makes user easy to work. They divided the project into two parts backend and frontend, backend containing all the functionalities and processing tasks and frontend containing the graphical user interface which is confronted by the user during execution of project. There project needs compulsory connection of the internet.

Their Project's Functionalities:

- Opens and access data from specific websites only.
- Interactive with Users.
- Was able to launch some installed applications of system.
- Can surf through some specific websites only.

The project we developed is similar to them but actually much more updated and loaded with more functionalities. The flexibility or we can say independence to the type of system or application is present in our project. Implementation of our project can be done anywhere it does not need to be installed manually. Our project is a single file which contains all the functionalities in it and can be executed or implemented anywhere. Our project too need compulsory need of internet connection as it contains the voice recognizer provided by Google. The number of upgradations future scope and they recommended were fulfilled by our project in terms of functionalities and flexibility.

Our Project's Additional Functionalities:

- Can search any website user wants by just commanding.
- It can access latest or top trending 10 Technology based News-Headlines.
- It can tell Jokes.
- Can open any video on You-tube directly by just commanding.
- Can search any file stored on computer hard disk and open it.
- Can open various applications.
- It can type for user on notepad while user dictate the sentences.
- Can find the IP address of the system.
- Can tell Temperature of any location from worldwide.
- Can switch windows on computer.
- Can make the system sleep, shutdown and restart by commanding.
- It can search for anything on google search engine by just commanding.
- It can search on Wikipedia for some particular information.

PROBLEM DEFINITION

Previous similar developed projects were having some problems which we have overcome in this project, previous development on this type of project faced the speed, latency issues.

In those projects, the functionalities were numbered i.e. very few because of the older versions of the python packages. Because of such issues the utilization of such projects were very less, people were using it rarely.

Some of the main issues are listed below which caused very much problem in utilization of such bots/projects. People was preferring manual work rather than the bots working really slow for them.

- Listening of commands was not accurate.
- Speech Recognition was not accurate.
- Not able to properly interpret descriptive text.
- Slow interaction with users.
- Not able to access local files stored on the system.
- The functionalities of such projects were numbered.

PROPOSED SYSTEM

Basically we proposed a system which can do almost all regular basis common tasks by just giving voice commands. The system we proposed is able to perform Speech-to-Text and vice a versa which implemented various features in our project.

The regular basis common tasks may include surfing on the internet, finding about some specific information on internet or on Wikipedia, getting news, playing videos offline as well as online, switching of tabs and applications while multitasking, system controls such as shutting-down, restarting and sleeping of system, launching some built-in applications, searching different files on the system easily and opening it, closing the running applications and so on. All of these tasks gets performed by just giving the voice commands to an interactive bot/assistant.

The system is proposed by our team aiming to develop this project beneficial and seamlessly easy to use for all the technical and non-technical people.

TOOLS & PLATFORM

Project is developed in **Python** language, so we have used the **Anaconda Navigator** which provides different IDEs of python, from which we have used the **Spyder** application for development and execution.

We have developed and executed this project on **Windows** platform. Since it is developed in python language it **Platform Independent** with some slight changes in code according to the system's platform.

PROJECT TYPE

It is an **Application** which can be implemented at different level of stages of the system according to user's requirements.

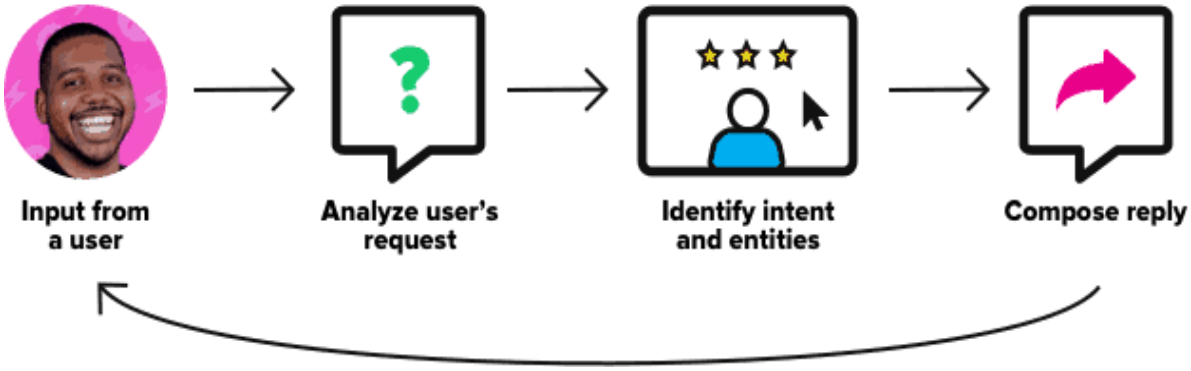
HARDWARE REQUIREMENTS FOR DEVELOPER & END-USER

Only a **Computer System** is needed for this project for developing by developer and executing by end-user.

SOFTWARE REQUIREMENTS FOR DEVELOPER & END-USER

Only **Python and some Libraries** are needed to be installed for developing by developer and executing by end-user.

METHODOLOGY & MODULE DESCRIPTION



1. PYTTSX3

Pyttsx3 is an open-source text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3. This library is essential to my project because it provides the necessary functions which are really useful for our project as it implies the interaction process of the bot. This library is basically a text-to-speech library which converts the written text into a spoken speech.

In addition, this module has been tested and is known to work on the following systems:

- SAPI5 on Windows XP, Windows Vista, and Windows 8, 8.1, 10
- NSSpeechSynthesizer on Mac OS X 10.5 (Leopard) and 10.6 (Snow Leopard)
- eSpeak on Ubuntu Desktop Edition 8.10 (Intrepid), 9.04 (Jaunty), and 9.10 (Karmic)

Installation of Package:

The proper way to install this module is by cloning the repository and installing via the setup Python file.

pip install pyttsx3

Above command is used to install this particular package.

Text to speech

The text-to-speech features for this module are based on languages installed in your operating system.

By default, it should come together with the language pack during the installation of the operating system. You need to install the language pack manually if you intend to use other languages.

For Windows user, head over to the Language setting.

If you encountered errors related to a missing driver, you can try to pass in the driver's name. There are four available built-in drivers:

- dummy.py — Test driver that does nothing. You can use the structure as reference to implement your own driver.
- espeak.py — Driver for other platforms, such as Ubuntu
- nsss.py — Driver for MacOS
- sapi5.py — Driver for Windows operating system.

Properties

The engine object has the following variables that can be fine-tuned:

- **rate** — Integer speech rate in words per minute. The base value is 200.
- **voice** — String identifier of the active voice
- **volume** — Floating point volume in the range of 0.0 to 1.0 inclusive
- **voices** — List of pyttsx.voice.Voice descriptor objects

You can set them via the **setProperty function.**

Voice property

The voice property has the following variables:

- age — Integer age of the voice in years. Defaults to None if unknown.
- gender — String gender of the voice: male, female, or neutral. Defaults to None if unknown.

- id — String identifier of the voice. Used to set the active voice via pyttsx3.engine.Engine.setPropertyValue(). This attribute is always defined.
- languages — List of string languages supported by this voice. Defaults to an empty list of unknown.
- name — Human-readable name of the voice. Defaults to None if unknown.

Available inbuilt Voices:

- David (English male)
- Zira (English female)
- Haruka (Japanese)
- Huihui (Chinese)

The functionalities of this library provides vast options and variations in speed, toning and type of speech of the bot. The purpose of interaction between user and bot is got covered in our project because of this library. I found out about this library at pypi.org while researching about our idea of this project, I found the necessary documents of this library on the same site which helped me to learn and utilize this library in my project.

2. Speech_recognition

Speech Recognition is an important feature in our project and in several applications used such as home automation, artificial intelligence, etc. It is a library for performing speech recognition, with support for several engines and APIs, online:

- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text

We have used the **Google Speech Recognition** from above listed apis in our project. The time it takes to recognize a sentence or statement is reduced by using this api, the ambient noise filtration is also supported in this package so that the commands passed by the user should get focused only, not other noises in background. To use all the functionalities of this library **some prerequisites** are needed they are as follows:

- **Python** 2.6, 2.7, or 3.3+ (required)
- **PyAudio** 0.2.11+ (required only if you need to use microphone input, Microphone)
- **Google API Client Library for Python** (required only if you need to use the Google Cloud Speech API, recognizer_instance.recognize_google_cloud)

These above prerequisites are really easy to download and install on any computer system. The features it provides are uncommendable, the speed, accuracy consistency gets upgraded with each update of these packages.

PyAudio (for microphone users)

PyAudio is required if and only if you want to use microphone input (Microphone). PyAudio version 0.2.11+ is required, as earlier versions have known memory management bugs when recording from microphones in certain situations.

If not installed, everything in the library will still work, except attempting to instantiate a Microphone object will raise an AttributeError.

The installation instructions on the PyAudio website are quite good - for convenience, they are summarized below:

On Windows, install PyAudio using Pip: execute **pip install pyaudio** in a terminal.

On other POSIX-based systems, install the portaudio19-dev and python-all-dev (or python3-all-dev if using Python 3) packages (or their closest equivalents) using a package manager of your choice, and then install PyAudio using Pip: **pip install pyaudio** (replace pip with pip3 if using Python 3).

PyAudio wheel packages for common 64-bit Python versions on Windows and Linux are included for convenience, under the third-party/ directory in the repository root. To install, simply run pip install wheel followed by pip

install/thirdparty/WHEEL_FILENAME (replace pip with pip3 if using Python 3) in the repository root directory.

Google API Client Library for Python (for Google Cloud Speech API users)

Google API Client Library for Python is required if and only if you want to use the Google Cloud Speech API (recognizer_instance.recognize_google_cloud).

If not installed, everything in the library will still work, except calling recognizer_instance.recognize_google_cloud will raise an RequestError.

According to the official installation instructions, the recommended way to install this is using Pip: execute pip install google-api-python-client (replace pip with pip3 if using Python 3).

Alternatively, you can perform the installation completely offline from the source archives under the/third-party/Source code for Google API Client Library for Python and its dependencies/ directory.

Installation of Package:

First, make sure you have all the requirements listed in the "Prerequisites" section.

The easiest way to install this is using **pip install SpeechRecognition.**

Otherwise, download the source distribution from PyPI, and extract the archive.

In the folder, run python setup.py install.

3. Requests

The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc). This package is basically used to get the specific data from http webpages without actually visiting/displaying them on screen.

This package is used in our to project to fetch various information/data from different webpages into our project. We used it to fetch latest technology based

news, we used it to fetch the exact temperature of which ever city user commands. There are various methods including in this package some of them are:

1. get(url, params, args)- Sends a GET request to the specified url.
2. post(url, data, json, args)- Sends a POST request to the specified url.
3. put(url, data, args)- Sends a PUT request to the specified url.
4. request(method, url, args)- Sends a request of the specified method to the specified url.

We have used the **get()** and **request()** methods to fetch the required data. Because of this package our project is able to get the data dynamically whenever the user wants it, our project is able to fetch the data, process or use it further and generates the output accordingly.

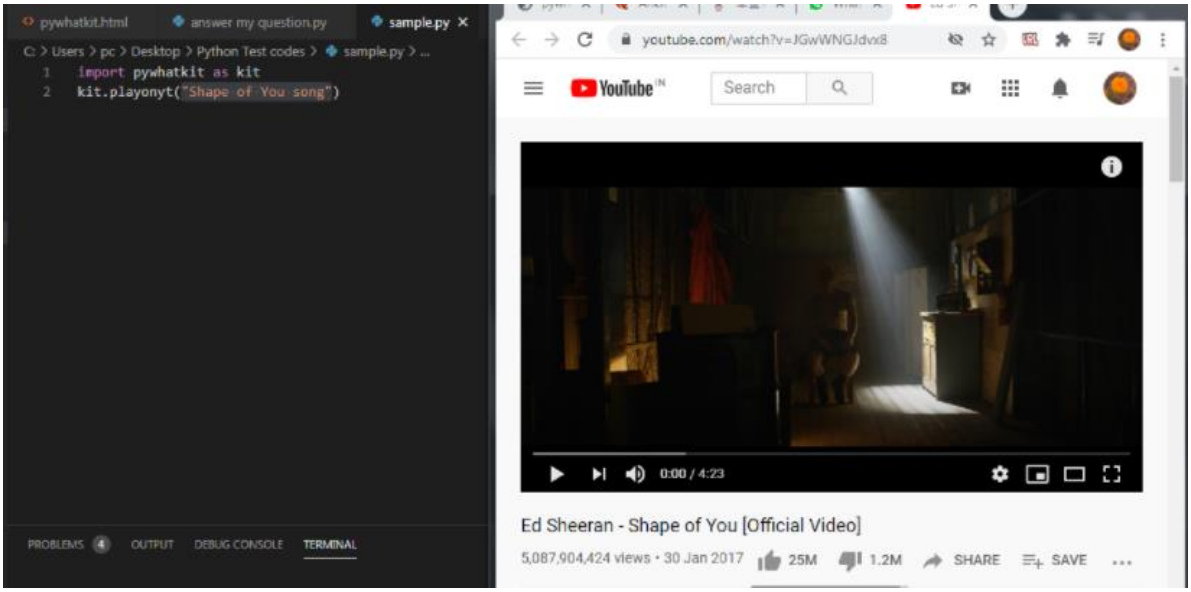
4. Pywhatkit

PyWhatKit is a Python library with various helpful features. It is an easy to use library which does not requires you to do some additional setup. This library was basically developed to automate the Whatsapp messages sending but in our project it is used to automate the browsing and surfing through internet. Before this package the automation was implemented using selenium package but this pywhatkit package replaced it for all the basic websites automation for searching, surfing and browsing.

This package does include variety of functions which automate the variety of websites. Some of them are shown and implemented in our project too, they are as follows:

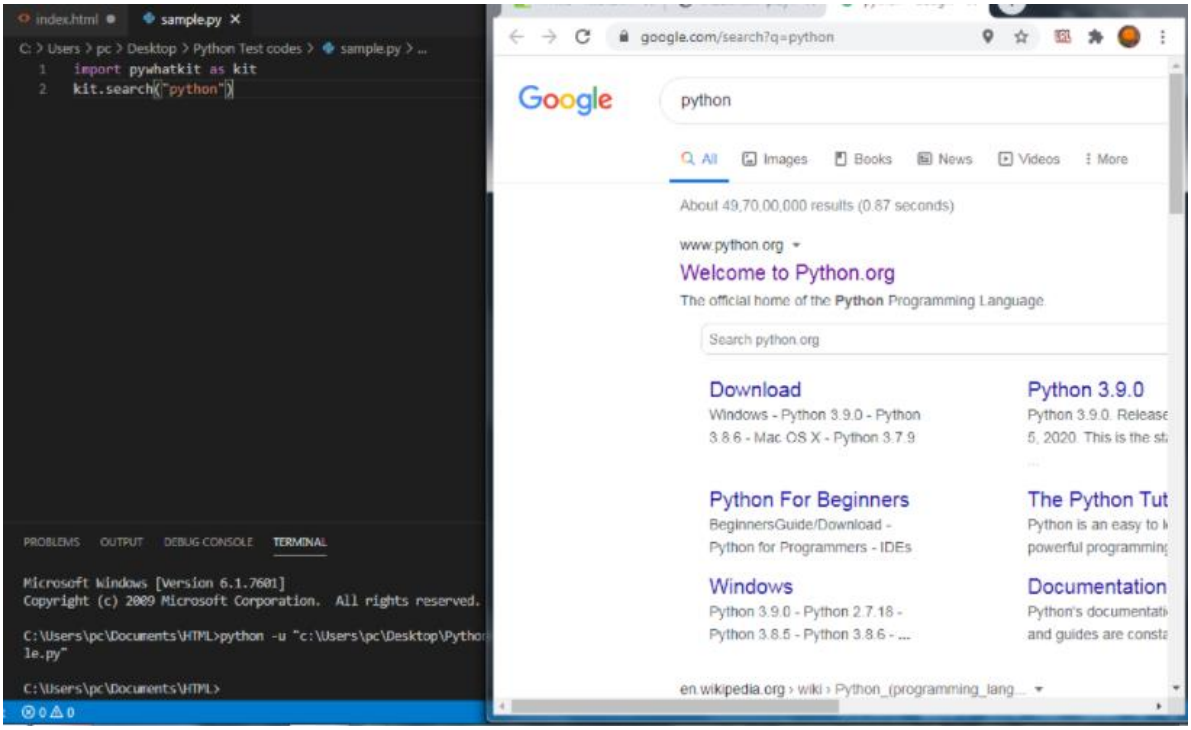
1) kit.playonyt()

This function can be used to search and play a particular video on YouTube by passing an argument to the function, like "Shape of You song" passing a string (name of video) is compulsory.



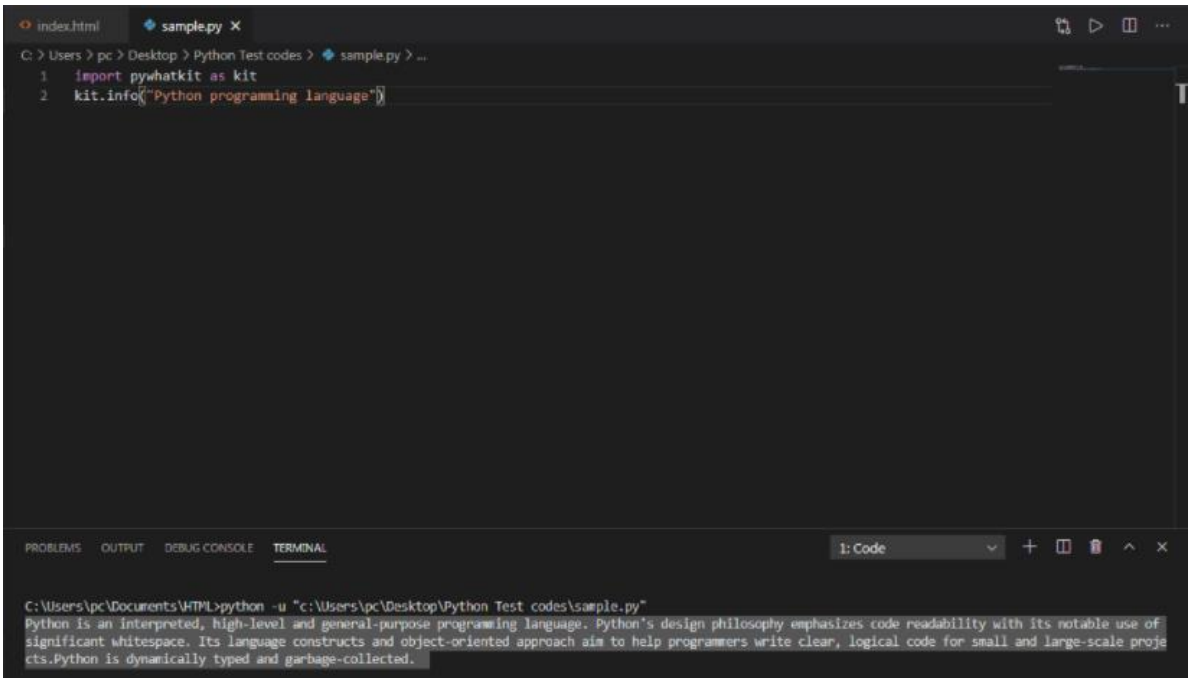
2) kit.search()

This function is used to search any term on official Google website. Like in the below example the searched term is 'python', this searched term (topic) is compulsory to pass as an argument.



3) kit.info()

This function can be used to fetch information about any topic from the official Wikipedia webpage. Like in the below example the argument passed 'Python Programming Language' is the compulsory argument to pass each and every time this function get used.



This package does have various functions, we have mentioned and used some of them as mentioned above. These functions are used to automate the web searching by just giving a command to the bot we developed.

5. Pyautogui

PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse & keyboard. Windows has no dependencies for this package. This package is mainly used to control the keyboard and mouse.

It lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be as simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3.

PyAutoGUI has several features:

- Moving the mouse and clicking or typing in the windows of other applications.
- Sending keystrokes to applications (for example, to type anything).
- Take screenshots, and given an image (for example, of a button or checkbox), find it on the screen.
- Locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently)
- Display message boxes for user interaction while your GUI automation script runs.

Keyboard and Mouse Control using Pyautogui

The x, y coordinates used by PyAutoGUI has the 0, 0 origin coordinates in the top left corner of the screen. The x coordinates increase going to the right (just as in mathematics) but the y coordinates increase going down (the opposite of mathematics). On a screen that is 1920 x 1080 pixels in size, coordinates 0, 0 are for the top left while 1919, 1079 is for the bottom right.

Currently, PyAutoGUI only works on the primary monitor.
PyAutoGUI isn't reliable for the screen of a second monitor (the mouse functions may or may not work on multi-monitor setups depending on your operating system and version).

All keyboard presses done by PyAutoGUI are sent to the window that currently has focus, as if you had pressed the physical keyboard key.
The three major operating systems (Windows, macOS, and Linux) each have different ways to programmatically control the mouse and keyboard. This can often involve confusing, obscure, and deeply technical details. The job of PyAutoGUI is to hide all of this complexity behind a simple API.

In our project functions of this package is used at various places. Whenever we needed any type of control of mouse and keyboard we used this package's function. Using this package, we were able to add various features in our bot namely some of them are- we can able to **switch the window** by just commanding to the bot telling him again and again can make him switch multiple windows and the user can work easily and efficiently. We developed and applied a feature because of which the user can **dictate** the bot and the bot will type/write as per the dictation given by the user, till now this functionality is applied on notepad application only.

6. Bs4 (BeautifulSoup)

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

For example, we have found some webpages that display data relevant to your research, such as date or address information, but that do not provide any way of downloading the data directly. Beautiful Soup helps you pull particular content from a webpage, remove the HTML markup, and save the information. It is a tool for web scraping that helps you clean up and parse the documents you have pulled down from the web.

The Beautiful Soup documentation will give you a sense of variety of things that the Beautiful Soup library will help with, from isolating titles and links, to extracting all of the text from the html tags, to altering the HTML within the document you're working with.

7. Datetime

In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.

Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

The datetime classes are categorize into 6 main classes –

date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
datetime – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
tzinfo – It provides time zone information objects.
timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

8. Time

Python has a module named time to handle detailed time-related tasks. This module consists of various functions of time conversions and types each and every function returns result in detail as compared to datetime module mentioned above. It is a built-in package, no need for additional installation.

In our project this module is used in a feature of reminding or say alarming, this module's functions are used to convert the time into minutes and set reminder according to the minutes commanded by the end-user.

9. Sys

The **sys module** in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment.

It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter. This is a built-in package so no need for any installation.

This module is used to exit from the execution of program and to stop everything like continuous listening, recognizing and so on.

10. Os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The **os** and **os.path modules** include many functions to interact with the file system.

These modules can be said to be the main modules in our project because of their functionalities, their functions are used when some application user wants to open, their functions are used when user wants to close some applications, their functions are used when user searches for a specific file, in this way the vast amount of functionalities of these modules are utilized in our project according to the project's requirements and needs. It can be used to open files, applications and almost everything locally stored or installed on the system.

Using these modules in our project, we were able to develop a feature which no one was able implement before i.e. file searching, any kind of file can be searched in any

partition of the disks we just have to mention the filename with its specific extension and then the disk partition name (C,D,E,F) then the file will get searched in that particular mentioned partition, if the file got found it will simply just open the file along with its exact location written in the prompt so that user can copy from there for future-use, if the file does not found then the bot will simply tells the user that the specified file does not exist.

11. Pyjokes

One line jokes for programmers (jokes as a service), this package provides varieties of technical jokes just to lighten the mood, if the end-user is a technical person he would understand and enjoy the jokes. This package has to be installed before implementation by using the **pip install pyjokes** command in the interpreter.

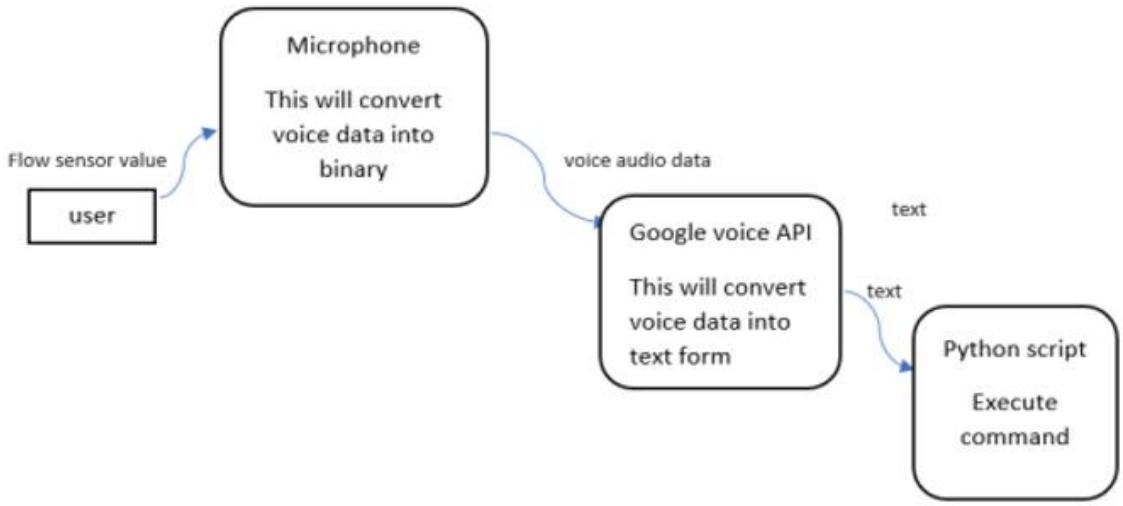
This package was implemented in this project because some of the similar projects have this feature in them with non-technical jokes but as far as we researched there is not a python package developed to tell a non-technical joke, so instead we just implemented this package.

12. Wikipedia

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. This package has to be installed first before implementation by commanding **pip install wikipedia** in the interpreter.

Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the Media-Wiki API so you can focus on using Wikipedia data, not getting it.

DESIGN



Data-Flow Diagram

The system design consists of:

1. Taking the input as speech patterns through microphone.

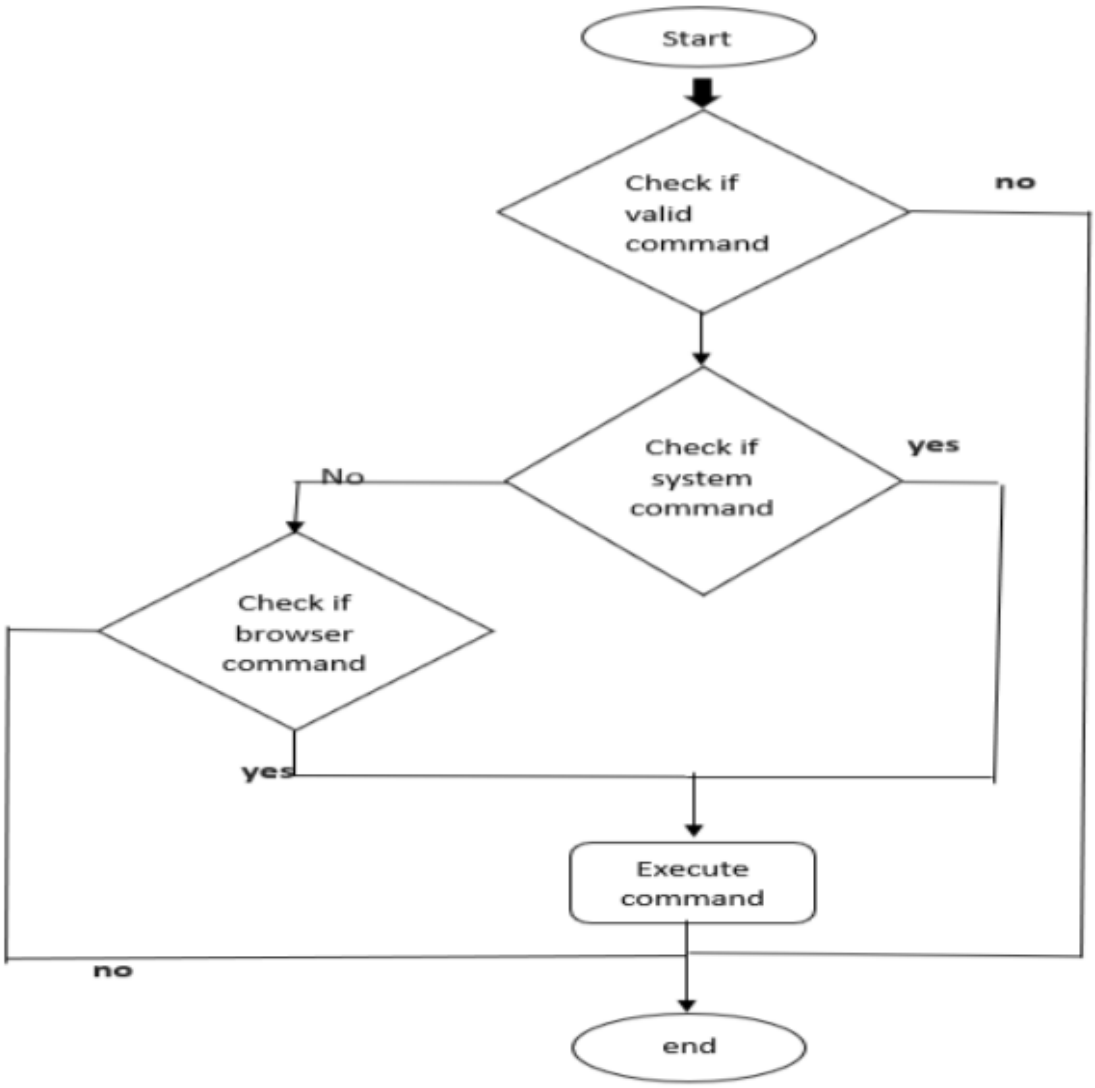
2. Audio data recognition and conversion into text.

3. Comparing the input with predefined commands

4. Giving the desired output

As you can see in the above data flow diagram, the working of our project starts from the **user's input speech** through the **Microphone** then it goes to the **Google Voice API** which converts the speech to the text format and then the control goes to the python program other functionalities are developed. All the controls are placed in the same program but the data flows as shown in the above diagram.

According to the user's command the function will perform, each function is distinguished with specific key word and feature, the user has to take care that the key word will get spoked while commanding to the bot, the key words will get written in the IDE as you run the program, it will get displayed every time the program gets executed so that the user won't get confused and do not have to remember it all the time just read them and speak accordingly the task will get performed according to that command. In this way the user will get his desired output by just commanding/speaking to the bot. The listening task is always on it only stops when the user commands to stop or when the user manually stops the program.



Flow Chart

As we can see in the above flowchart, the bot will ask for some command, when the user gives the command it will check if the command is valid or not. If the command is invalid it will ask again and take the command again, this keeps going on till the user gives the valid command.

When user gave the valid command the bot will check which type of command is it and perform the task accordingly. If the command is internet based, then it will open the necessary related applications if user required them or perform some of the functionality without opening any application just by fetching the data from internet according to the command.

If the command is related with the local system like accessing files, launching applications and so on then it will work according to that.

Everything is based on the command user gives if it is valid or not it will work according to that.

EXPERIMENTAL WORK

All the functionalities present in this project were an experiment by the programmers of this project, right from the bot's greeting function up to the exiting from the code everything was an experiment.

Basically this project was made from scratch, we researched about this topic on the internet found some of the similar ones after that we came to know about the different packages which can be used in this project.

We started research on basic packages first, after finding and implementing all the basic functionalities of a bot then the project team started to search about the features to be added so that the bot can become more interactive with the user.

While researching about the project we came to know about their problems which we have mentioned above to solve those problems was itself an Experimental task for our team. We went thoroughly through each and every package used before their upgradations till now their changes and everything.

Later on while completing this project we checked whether we overcame those mentioned problems or not, actually we did, we solved each and every main problem along with various additional Experimental functionalities implemented in our project.

The implementation of **stored file searching functionality** was a huge task to complete because no other bot or assistant was able to do it even the artificial intelligence based assistants too.

We have done a lot research for that and then came to know about the module **OS** which have the functionalities like a command prompt, some commands can be passed and the system will give output according to those commands, it can be used to open files, applications and almost everything locally stored or installed on the system. Such functionalities were used and we developed a code to search any type of file in the complete system and open, locate it without any error or problem.

The **dictation on the notepad** functionality also comes under the experimental work, we have to go through so many modules to find about that, later on we found the **Pyautogui module** which can control the keyboard, so we implemented the dictation functionality successfully and later on tested it thoroughly, by testing we are able to say that it is little bit unstable because the complete function is based on the recognition, if the bot cannot recognize exactly the output will be given wrong.

In this way many of the functionalities and features are complete experimental by our team, since this were experiments we also took care of the system and data security because sometimes the experiments can cause really big problems and sometimes loss of data and irregularities in the system.

WORKING OF FEATURES

1) It will launch notepad by saying **"open notepad"** in a sentence.

It will ask user if he/she want to dictate him to write or not.

If user said ‘yes’ to dictation mode then it will start writing on notepad till the user says **"stop writing"**.

If user said ‘no’ then it will ask is there any other work to do for him?

2) It will increase volume by 1 if user says **"increase volume"** or **"volume up"** in a sentence.

Same is for decreasing volume by 1 if user says **"decrease volume"** or **"volume down"** in a sentence.

Same is to mute the volume completely, user has to say **"volume mute"** or **"mute"** in a sentence.

3) It will tell and shows the current day and date by user saying **"day"** or **"date"** in sentence.

4) It will search any type of file the user has to say **"search file"** in a sentence.

It will ask to enter the complete filename with extension, then it will ask to enter disk-partition name (i.e. C, D, E, F, G)

It will scan the entered partition for file and if found tells the exact location and opens the file. If not found it will ask to enter another partition of disk.

5) It will launch command prompt if user says **"open command prompt"** in a sentence.

6) It can launch system camera to click photos by saying **"open camera"** in a sentence.

7) It can tell temperature of a any city/region in the world by saying **"temperature"** in a sentence.

It will ask temperature of which city? then the user tells name of the city and he will tell the temperature.

8) It can tell the system's exact IP Address, user just have to say **"ip address"** in a sentence.

9) It can fetch any information from Wikipedia, user just have to say **"[subject] wikipedia"** in a sentence.

The above mentioned [subject] is the topic or name or anything of which user needs information from Wikipedia.

10) It can load any website on the system's default web browser, user just have to say **"open website"** in a sentence.

Then it will ask which website to open and the user will say the specific website and it will be launched.

11) It can close any running application which is displayed on screen, user just have to say **"close application"** in a sentence.

12) It can search and open google website with results according to users query, user just have to say **"open google"** in a sentence.

It will ask user what to search on google, user just have to say the query which he/she want to search on google and the related results will be displayed on screen.

13) It can close the current using tab of browser, user just have to say **"close or kill current tab"** in a sentence.

14) It can play any You-tube video directly by just saying **"video on youtube"** in a sentence.

It will ask which video to play? user will answer with the name of the desired video.

15) It can set reminder, user just have to say **"timer"** or **"stopwatch"** in a sentence.

It will ask for how many minutes? user have to give answer in minutes and the bot will remind user after specified minutes by saying "Sir I am reminding you, your set time is finished".

16) It will be in standby mode (just listening and recognizing) by saying **"wait"** in a sentence.

User have to say **"ok listen"** in a sentence to deactivate the standby mode and to give more tasks.

17) It can tell jokes, user just have to say **"joke"** in a sentence.

18) It can tell top 10 trending headlines of news, user just have to say **"news"** in a sentence.

19) It can switch windows (applications) by just saying **"change window"** in a sentence.

20) It can shutdown, restart, sleep the computer system by saying **"shut down the system", "restart the system", "sleep the system"** respectively.

21) The program will get terminated by saying **"stop listening"** in a sentence.

ADVANTAGES OF PYBOT

- **Easy to work with.**
  This project makes our daily tasks so easy to work so that the user can focus on some other important tasks.

- **Dictation Mode.**
  This bot having feature called dictation mode, helps the user to take notes by just giving voice commands and getting every word typed on the notepad.

- **Automation of Tasks.**
  This project automates almost every task, which will be very beneficial to the user. It can search and select the appropriate results automatically for the query you passed through voice command.

- **Time Saving.**
  This bot helps to save time of the end-user efficiently so that the user can work on other important tasks rather than doing the regular tasks manually.

- **Variety of Features.**
  This project provides many variety of features which will be useful for the user in different fields of work.

- **Accurate Search-Results.**
  The searching features of this bot whether it is online search or offline search, they are accurate always but user have to provide input properly so that the output will be generated accurately.

- **Online Surfing.**
  This bot can surf online for the videos, results user asked which eventually saves the time of user and accurate results are provided.

- **Information from Wikipedia.**
  We all know how important Wikipedia is in our lives, it provides us information about everything, this bot is having a feature which will retrieve information from official Wikipedia and paste it on the output screen while reading it for the user.

LIMITATIONS OF PYBOT

- **Internet Connection.**
  This project needs a constant speed of internet each and every time, whether the work is online or offline it needs the internet connection compulsorily.

- **Single-tasking while Dictation mode.**
  This bot is having a feature called dictation mode, when dictation mode starts it will write each and every word except ‘stop writing’ on notepad whether you give other feature command it will write that too on the notepad. So it performs only dictation when dictation mode is started.

- **Wikipedia Results.**
  The result of information retrieved from the Wikipedia is specifically of 5 lines only so that the user will the idea about his topic but to study deeply it has to be searched by the other way.

- **Speech Recognition.**
  The speech recognition is under development which sometimes causes problems, sometimes recognition is not accurate and the resultant output changes because of that.

- **Slower in some cases.**
  Some of the features really take lot of time for example searching a file in ‘C’ drive.

RESULT & DISCUSSION

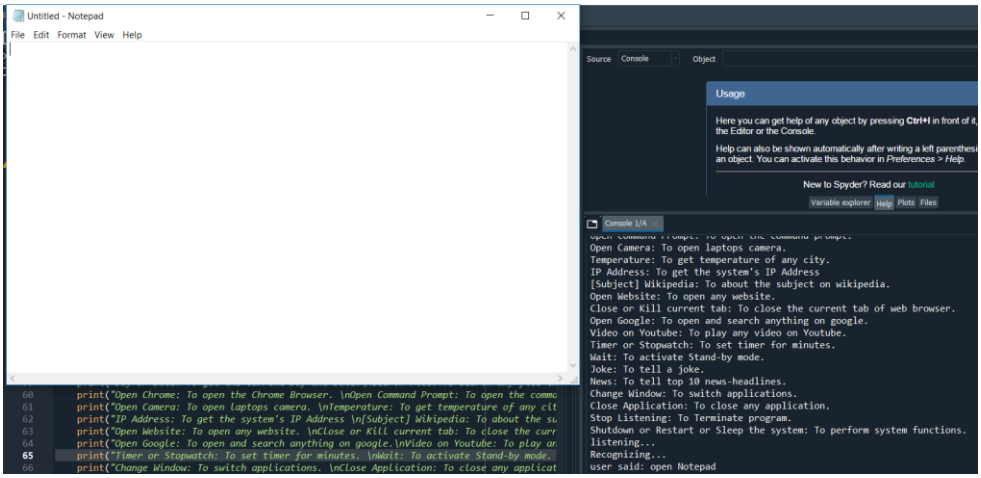
The starting of our project that is starting the execution of our project begins with the following-

```
good afternoon, its 03:37 PM
i am pybot sir. please tell me how may i help you
Speak one of the following words to use specific feature:
open notepad: To open Notepad.
Volume up or down or mute: For Volume Controls.
Day or Date: To get the current Day and Date.
Search File: To Search any file in system.
Open Chrome: To open the Chrome Browser.
Open Command Prompt: To open the command prompt.
Open Camera: To open laptops camera.
Temperature: To get temperature of any city.
IP Address: To get the system's IP Address
[Subject] Wikipedia: To about the subject on wikipedia.
Open Website: To open any website.
Close or Kill current tab: To close the current tab of web browser.
Open Google: To open and search anything on google.
Video on Youtube: To play any video on Youtube.
Timer or Stopwatch: To set timer for minutes.
 Wait: To activate Stand-by mode.
Joke: To tell a joke.
 News: To tell top 10 news-headlines.
Chnage Window: To switch applications.
 Close Application: To close any application.
Stop Listening: To Terminate program.
Shutdown or Restart or Sleep the system: To perform system functions.
listening...
```

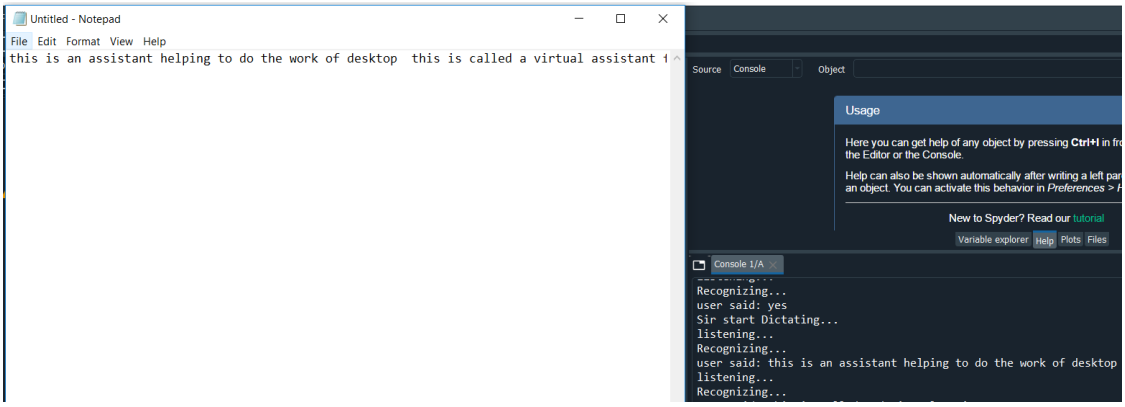
It begins with wishing the user according to the execution time of program. As we can see in the above the program is executed in the noon so, the bot wishes ‘Good Afternoon’ and tells the current time, introduces and ask how to help, and prints the keywords which user has to use, to use various features like a user manual just then starts listening to the user’s commands.

We are sharing the output of each feature of our project in the following-

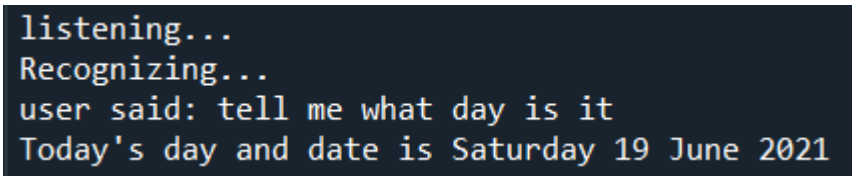
Opening The Notepad-



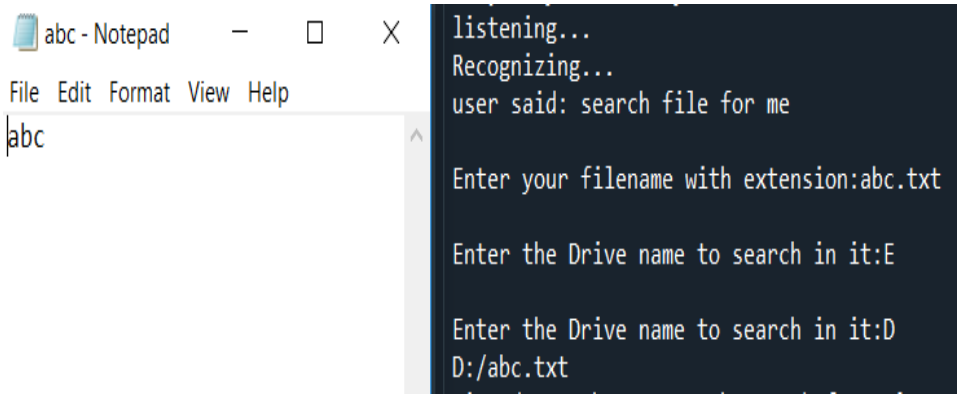
Dictating to write on Notepad-



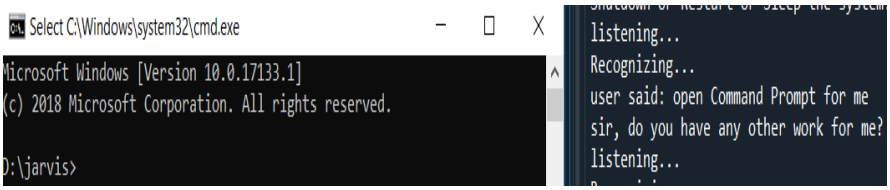
After asking what day is it-



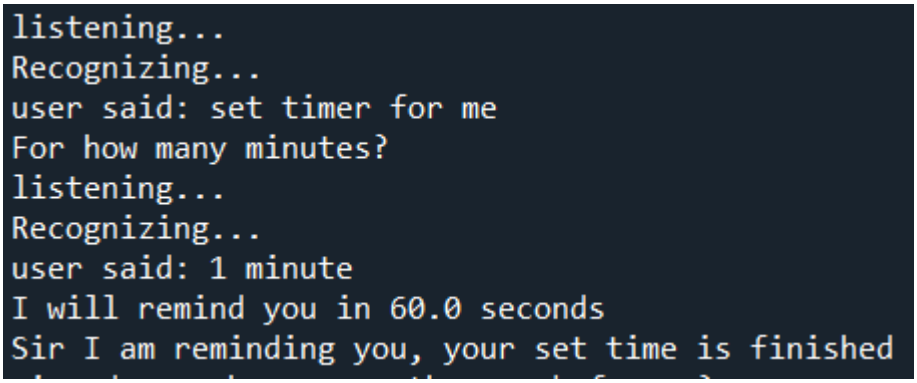
Asking to Search File-



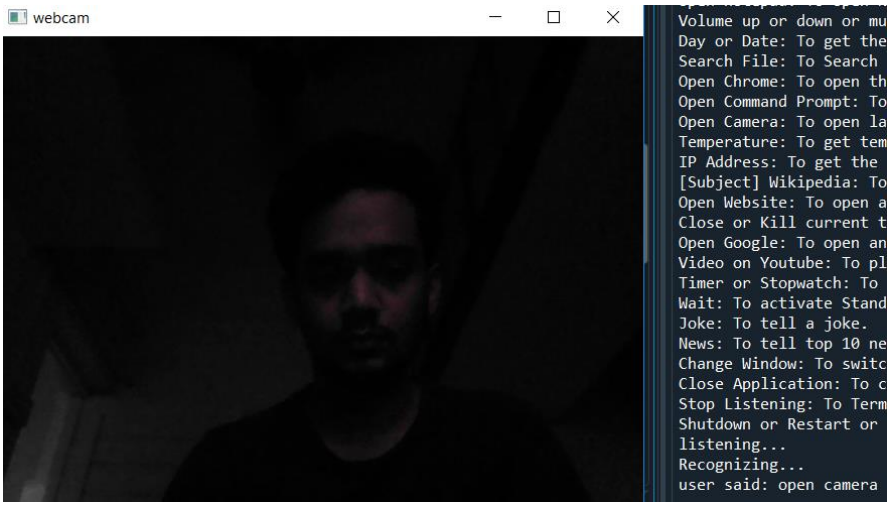
Asking to Open Command Prompt-



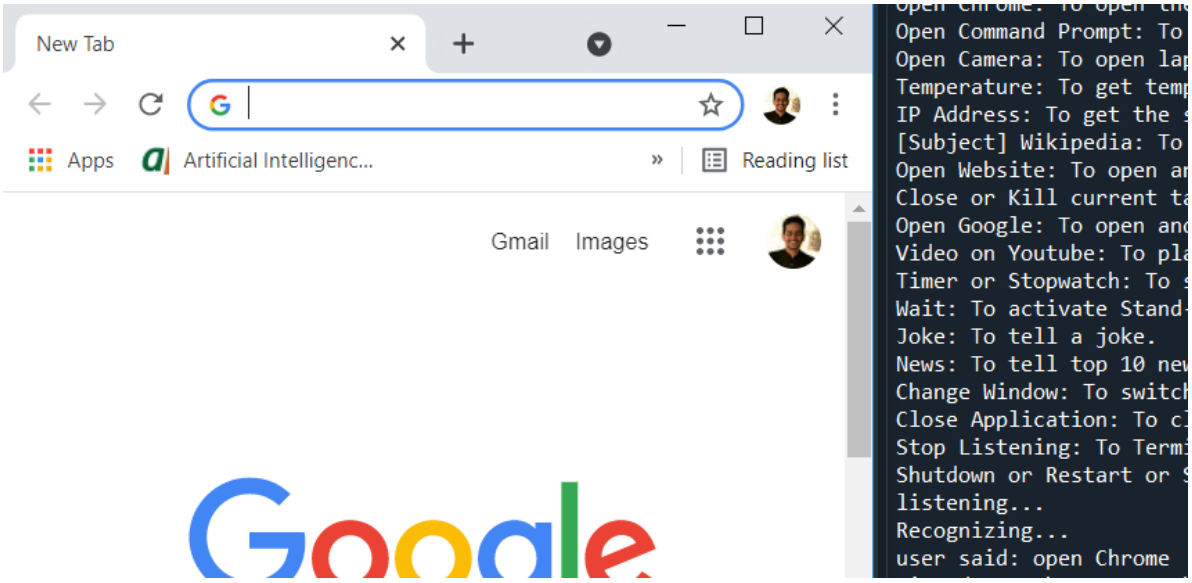
Asking to Set Timer-



Asking to Open Camera-



Opening Chrome Browser-



Asking the Temperature of City-

```
listening...
Recognizing...
user said: tell me temperature
sir,temperature of which city?
listening...
Recognizing...
user said: Amravati
current Temperature in Amravati is 32°C
```

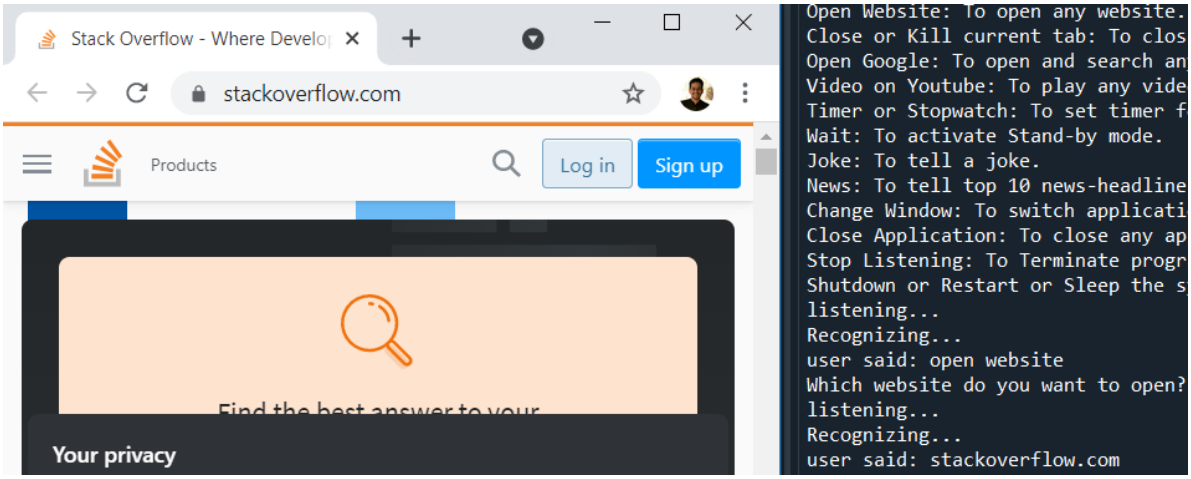
Asking for the IP Address-

```
listening...
Recognizing...
user said: tell me my system IP address
your IP address is 103.109.136.239
```

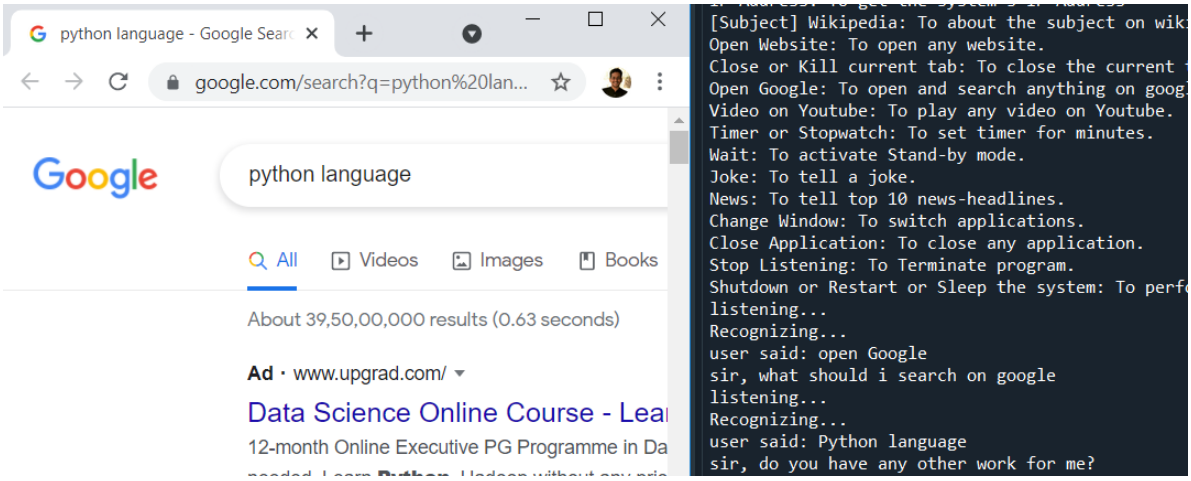
Searching about Virtual Assistant on Wikipedia-

```
listening...
Recognizing...
user said: virtual assistant Wikipedia
searching wikipedia....
according to wikipedia
An intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a software agent
that can perform tasks or services for an individual based on commands or questions. The term
"chatbot" is sometimes used to refer to virtual assistants generally or specifically accessed by
online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some
virtual assistants are able to interpret human speech and respond via synthesized voices. Users can
ask their assistants questions, control home automation devices and media playback via voice, and
manage other basic tasks such as email, to-do lists, and calendars with verbal (spoken?) commands.
```

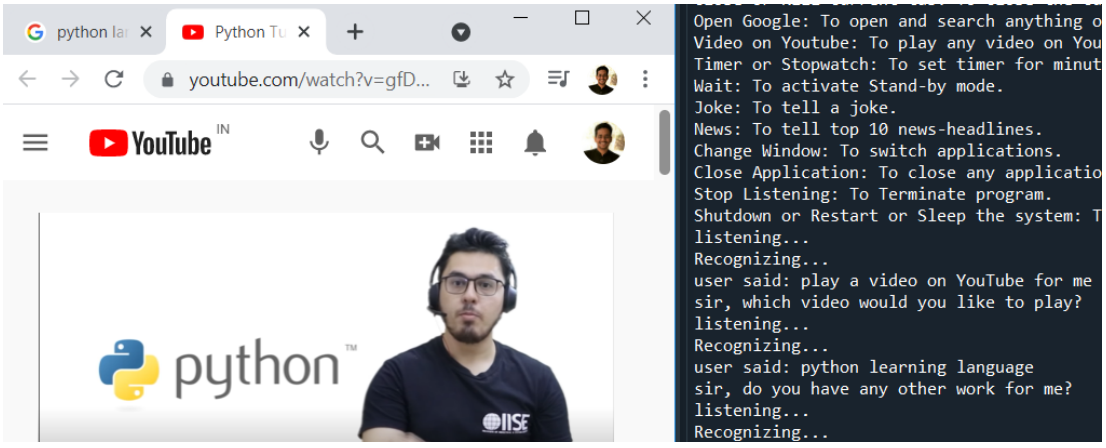

Opening any website-



Opening Google and searching about Python Language-



Asking to play video on You-tube-



Asking to close above running web-browser-

```
listening...
Recognizing...
user said: close application
Closing..
```

Asking to tell some jokes-

```
listening...
Recognizing...
user said: tell me a joke
['hip', 'hip'] (hip hip array!)
sir, do you have any other work for me?
listening...
Recognizing...
user said: tell me another joke
There are two ways to write error-free programs; only the third one works.
```

Asking to tell News Headlines-

```
listening...
Recognizing...
user said: tell me some news headline
please wait sir, fetching the latest news
today's first news is: Investors say Eindhoven poised to become Netherlands' No. 2 tech hub
today's second news is: Daily Crunch: Spotify and Ford make acquisitions
today's third news is: Extra Crunch roundup: influencer marketing 101, spotting future unicorns,
Apple AirTags teardown
today's fourth news is: Perspectives on tackling Big Tech's market power
today's fifth news is: Mediflash is a freelancer marketplace for health professionals
today's sixth news is: Owning the paycheck is the key to fintech success
today's seventh news is: Tiger Global in talks to back BharatPe at $2.5 billion valuation
today's eighth news is: UK's ICO warns over 'big data' surveillance threat of live facial recognition
in public
today's ninth news is: Ford acquires Electriphi as it prepares to woo EV fleet customers
today's tenth news is: Delivery service Gopuff acquires rideOS for $115 million-plus
```

Stand-By mode and Termination of program-

```
listening...
Recognizing...
user said: wait
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
listening...
Recognizing...
user said: ok listen
sir, do you have any other work for me?
listening...
Recognizing...
user said: no stop listening
okay sir, have a good day.

In [13]:
```

As we have seen features of this project in the above outputs. The bot/assistant keeps asking the user do you need help with something after each task given to him is completed unless the user says ‘wait’ and activate stand-by mode or user terminates the program.

CONCLUSION

In this project we discussed about the development, implementation and execution of our Pybot. To develop this project, we have done research and used the available open source modules and libraries because of which the near future updates will get easily available without any cost to pay. The implementation of this code file can be done anywhere according to user's requirements.

This type of technology started developing many years ago, each time new and advanced features gets added, sometimes they are free to use and sometimes not. The interactivity, ease of use, automation of daily basis tasks increases the popularity of such technology. The addition of features is unending but they can be implemented according to the user's requirements. This technology leads to other developing technologies such as Cloud Computing, Artificial Intelligence and so on to improve and implement advanced and new things.

FUTURE SCOPE

As I mentioned before this technology is upgrading every day, there can be so many more functionalities to implement in the future. New advanced technologies can be developed using this project, some can implement **AI** [Artificial Intelligence] in it then this assistant/bot can also work like other AI based assistants may be even better than them.

The development in this technology is unending because the functions and features for the human benefits cannot be measured as such. Some other technologies combination can make even better utilization of assistants and can be more and more beneficial for mankind. Some of the combinations can be seen nowadays like **Cloud computing** is attached or applied or implemented with this assistant technology so that it can store a large amount of informative data and provide it to the user whenever user is in need of it.

The utilization of this particular project can be done in any way according to the user's requirements for example, this project file can be executed in the background each and every time the computer system powers on so that the bot/assistant can be used right from the start of the computer use.

This project can also work like the backend of an application of desktop assistant, the User Interface should be developed and attached to this project to develop a complete application for the computer system.

More functionalities can be added in this project as I have said before the functionalities of this project are never ending so more and more feature and functions can be applied in the future, more accuracy of speech recognition and speed of doing tasks can be improved with the upgrades in the implemented packages and modules, the speed of recognizing can also be improved in the future updates.

Multiple languages can be implemented in this project, so that the bot/assistant can understand in different types of languages and even communicate with the user in different languages too. By applying this feature, the language barrier will be removed and users with no English language skills can also use this bot for their computer use.

Some industry based features can be applied in this project if required like for example Resume-Analyzer which analyzes the trainee resumes with the idle resume, those who are close to the score of the idle one can be the right candidate for the company, in this way some similar functionalities can be implemented in this project for benefits of mankind.

This project can be developed for different types of devices, now it is developed only for computer systems but in future it can be developed and implemented in mobile phones and special devices can also be made like the Alexa and Google Home are having, so that using AI and IOT they can control the house appliances accordingly.

These are some developments which can be implemented in the near future on this type of project.

REFERENCES

- AI based Voice Assistant Using Python by Deepak Shende, Ria Umahiya, Monika Raghorte, Aishwarya Bhisikar, Anup Bhange from K.D.K. College of Engineering, Nagpur. [https://www.jetir.org/view?paper=JETIR1902381]

- Desktop Voice Assistant for Visually Impaired by Ankush Yadav, Aman Singh, Aniket Sharma, Ankur Sindhu, Umang Rastogi from Meerut Institute of Engineering and Technology. [https://www.ijrte.org/wp-content/uploads/papers/v9i2/A2753059120.pdf]

- Virtual Assistant on Wikipedia. [https://en.wikipedia.org/wiki/Virtual_assistant]

- Libraries and modules research on official Pypi website. [https://pypi.org/project]