**VIRTUAL ASSISTANT WITH SENTIMENT ANALYSIS**

A Mini Project Report Submitted

By

**SAFOORA MASOOD (1604-19-737-007)**

**SADIA AFREEN (1604-19-737-010 )**



**INFORMATION TECHNOLOGY DEPARTMENT**

**MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Affiliated to Osmania University)**

Banjara Hills, Hyderabad - 500034

**2021-2022**

**MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Affiliated to Osmania University)**

Banjara Hills, Hyderabad - 500034

**DEPARTMENT OF INFORMATION TECHNOLOGY**

****

**CERTIFICATE**

It is certified that the work contained in the project report titled

**“VIRTUAL ASSISTANT WITH SENTIMENT ANALYSIS”** by

**SAFOORA MASOOD (1604-19-737-007)**

**SADIA AFREEN (1604-19-737-010)**

has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Project Supervisor Signature of Head

**Dr.SYEDA FOUZIA SAYEEDUNNISA Dr.MOUSMI AJAY CHAURASIA**

Information Technology Dept. Information Technology Dept.

MJ College of Engineering & Tech. MJ College of Engineering & Tech.

Hyderabad – 500034. Hyderabad – 500034

**External Examiner**

**DECLARATION**

We, **Safoora Masood** and **Sadia Afreen,** bearing Roll No. **1604-19-737-007** and **1604-19-737-010,** respectively.Here by declare that the project report entitled **“VIRTUAL ASSISTANT WITH SENTIMENT ANALYSIS”** is done as mini project during the Course work of VI Sem BE(IT) and is done under the guidance of Dr. S. Fouzia Sayeedunnisa, Associate Professor, Department of Information Technology, Muffakham Jah College of Engineering and Technology.

This is a record of bonafide work carried out by us in Muffakham Jah College of Engineering & Technology and the results embodied in this project have not been reproduced or copied from any source.

**By**

**SAFOORA MASOOD (1604-19-737-007)**

**SADIA AFREEN (1604-19-737-010)**

**Department of Information Technology,**

**Muffakham Jah College of Engineering and Technology,**

**Banjara Hills, Hyderabad - 500034.**

**ACKNOWLEDGEMENT**

We would like to take the opportunity to express our sincere thanks to our mini project coordinator Dr. S.Fouzia Sayeedunnisa, Associate Professor, Information Technology, MJ College of Engineering & Technology, for her invaluable support and guidance throughout our project work. Without her kind guidance & support this was not possible. We are grateful to her for her timely feedback which helped us track and schedule the process effectively. Her time, ideas and encouragement that she gave helped us complete our project efficiently.

We would also like to thank Dr.Mousmi Ajay Chaurasia, Head of Information Technology Department, for her encouragement and for providing an outstanding academic environment, also for providing the adequate facilities resources.

We sincerely acknowledge and thank Mr. T. Nooruddin, Lab Technician, MJ College of Engineering & Technology who provided timely guidance in the completion of this project work.

.

**ABSTRACT**

The project is meant to demonstrate the working of a virtual assistant with sentimental analysis. The back end of the project is being developed using Python. A virtual assistant uses advanced Artificial Intelligence (AI), natural language processing, and machine learning to extract information and complex data from conversations to understand them and process them accordingly.

Sentiment Analysis is a subcategory of Natural Language Understanding and Machine Learning – the technologies that give Conversational AI its ‘intelligence’ (NLU is what enables a virtual assistant to perceive and evaluate user information, while ML improves a virtual assistant’s performance over time based on past conversational data).

This project is aimed to enable instant service, render a personalized experience, allow multiple interactions with natural interface and ideally maps the user’s needs to offer services.

This project facilitates graphic designing, online marketing, social media management, social media ads, customer service, data entry, etc.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure No: | Figure Name | Page No: |
| 1.1 | Work Flow of Proposed System | 7 |
| 1.2 | System Architecture | 9 |
| 1.3 | Detailed Architecture Diagram | 9 |
| 1.4 | Flow Diagram | 10 |
| 1.5 | Application Working Diagram | 10 |
| 1.6 | Use Case Diagram | 11 |
| 1.7 | Component Diagram | 12 |

**INDEX**

**CONTENTS**

**1. INTRODUCTION**

[1.1 Problem Statement](#_1fob9te)

[1.2 Objectives](#_gjdgxs)

1.3 Applicability

**2. LITERATURE SURVEY**

[2.1 Related Research Papers](#_30j0zll)

2.2 Technology Used

2.2.1 Python

2.2.2 Libraries Used

**3. SYSTEM ANALYSIS**

* 1. Problems with Existing System

3.2 Proposed System

3.2.1 Procedure/ Working (Working of Proposed System)

* + - 1. Working Principles

3.3 Advantages of Proposed System

**4. SYSTEM DESIGN**

4.1 System Architecture

* 1. Flow Diagram

4.3 Application Working Diagram

4.4 Use Case Diagram

4.5 Component Diagram

**5. IMPLEMENTATION**

* 1. Code

5.2 Output Screen Shots

**6. SENTIMENT ANALYSIS**

**7. FUTURE SCOPE**

**8. CONCLUSION**

**9. REFERENCES**

**1 INTRODUCTION**

A virtual assistant is an application that understands voice commands and completes tasks for a user. Virtual assistants are available on most smartphones and tablets, traditional computers, and even standalone devices like the Amazon Echo and Google Home.

They combine specialized computer chips, microphones, and software that listen for specific spoken commands from you and can answer back in a voice that you select.

Virtual assistants like these can do everything from answer questions, tell jokes, play music, and control items in your home such as [lights](https://www.lifewire.com/smart-light-bulbs-4149561), thermostats, door locks, and smart home devices. They can respond to many voice commands, send text messages, make phone calls, and set up reminders. Anything you do on your phone, you can probably ask your virtual assistant to do for you.

Virtual assistants learn over time and get to know your habits and preferences, so they're always getting smarter. Using artificial intelligence (AI), they understand natural language, recognize faces, identify objects, and communicate with other smart devices and software.

**1.1** **Problem Statement**

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there’s no such virtual assistant available for the paradise of Developers i.e. Linux platform. This Software aims at developing a personal assistant for Linux-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

## **1.2 Objectives**

Overall goal and objective of a virtual assistant is to assist the end-user with day-to-day activities like general human conversation, searching queries in google, searching for videos, retrieving images, live weather conditions, the time and reminding the user about the scheduled events and tasks. Currently, the project aims to provide Virtual Assistant that would not only aid in their daily routine tasks but also help in automation of various activities.

One of the main advantages of voice searches is their rapidity. In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 during the same period of time15. In this respect, the ability of personal virtual assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

**1.3 Applicability**

Applicability The mass adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user’s life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous

# 2. LITERATURE SURVEY

## **2.1 Related Research Papers**

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner.

As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

* 1. **Technology Used**

**2.2.1 PYTHON**

Python is an OOPs (Object Oriented Programming) based, high-level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5 code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most of the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

**2.2.2 Libraries Used**

**Pyttsx**

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for textto-

speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

**Speech Recognition**

This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

**JSON**

JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data. JSON is often used when data is sent from a server to a web page. JSON is "self-describing" and easy to understand

**Requests**

Requests module allows you to send http requests using python. Requests is used for making GET and POST requests. It abstracts the complexities of making requests behind a beautiful, simple API

**Datetime**

Datetime package is used to showing Date and Time. This datetime module comes built-in with Python.

**Wikipedia**

We all know Wikipedia is a great and huge source of knowledge just like GeeksforGeeks or any other sources we have used the Wikipedia module in our project to get more information from Wikipedia or to perform a Wikipedia search. To install this Wikipedia module, use pip install Wikipedia.

**Web browser**

To perform Web Search. This module comes built-in with Python.

**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.

**Pyjokes**

Pyjokes is used for collecting Python Jokes over the Internet. Pyjokes is add in our project because it adds jokes in our project. It is very interesting. Pyjokes is the one-line joke which makes our project interesting.

**3. SYSTEM ANALYSIS**

**3.1 Problems with Existing System**

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner.

As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

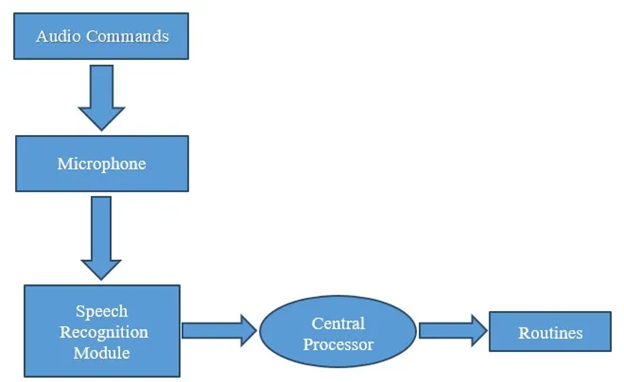
**3.2 Proposed System**

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching on Google without opening the browser and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. Jarvis is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions toper form any specific task. The IDE used in this project is Visual Studio Code. All the python files were created in VSCode and all the packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, etc.

I have created a live GUI for interacting with the JARVIS as it gives a design and interesting look while having the conversation. With the advancement JARVIS can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation.

**3.2.1 Working of Proposed System**

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer’s internal files, etc.



1.1 Work flow of proposed system

**3.2.1.1 Working Principles**

The working of Virtual Assistant uses following principles:

• **Natural Language Processing** : Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English. Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

**Automatic Speech Recognition :** To understand command according to user’s input.

**Artificial Intelligence :** To learn things from user and to store all information about behaviour and relations of user. The ability of a system to calculate, reason, perceive relationships and analogies, learn from experience, store and retrieve information from memory, solve problems, comprehend complex ideas, use natural language fluently, generalize, and adapt new situations.

**Inter Process Communication:** To get important information from other software applications.

**3.3 Advantages of Proposed System**

* It is quite easy to use as compared to other AI assistance.
* It can easily work with multiple commands. Usually, AI assistants are limited to a certain number of commands and fail to execute beyond that.
* JARVIS can be programmed with custom command option. Hence it does not stick to a single command type
* It is a secure system to involve in your daily life.
* It is an extremely helpful and useful system for disabled people.

**4. SYSTEM DESIGN**

**4.1 System Architecture**

The system architecture of this project shows the flow of the control through the system. The architecture diagram is as follows:

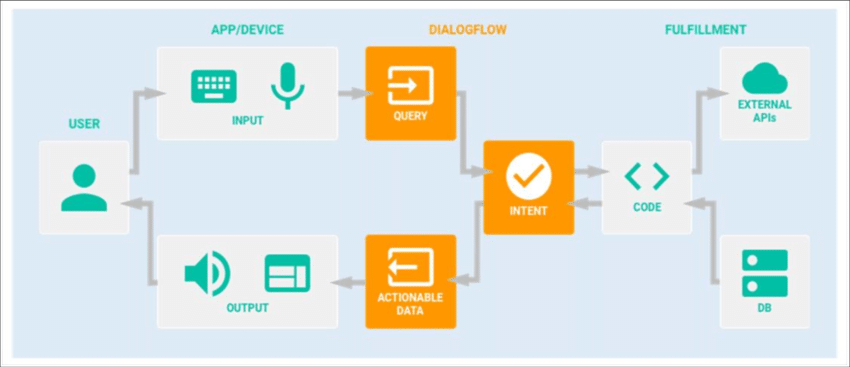
****

Fig.1.2 System Architecture

The architecture characterizes the system. User gives text or voice input. Voice commands are converted to text through Google speech API. Text input is simply stored in the database for further process. It is recognised and matched with the commands available in the database. Once the command is found, its respective task is executed as voice, text or through user interface as output. It can also be in the form of application execution.

The figure below shows the detailed architecture of a Virtual Assistant.

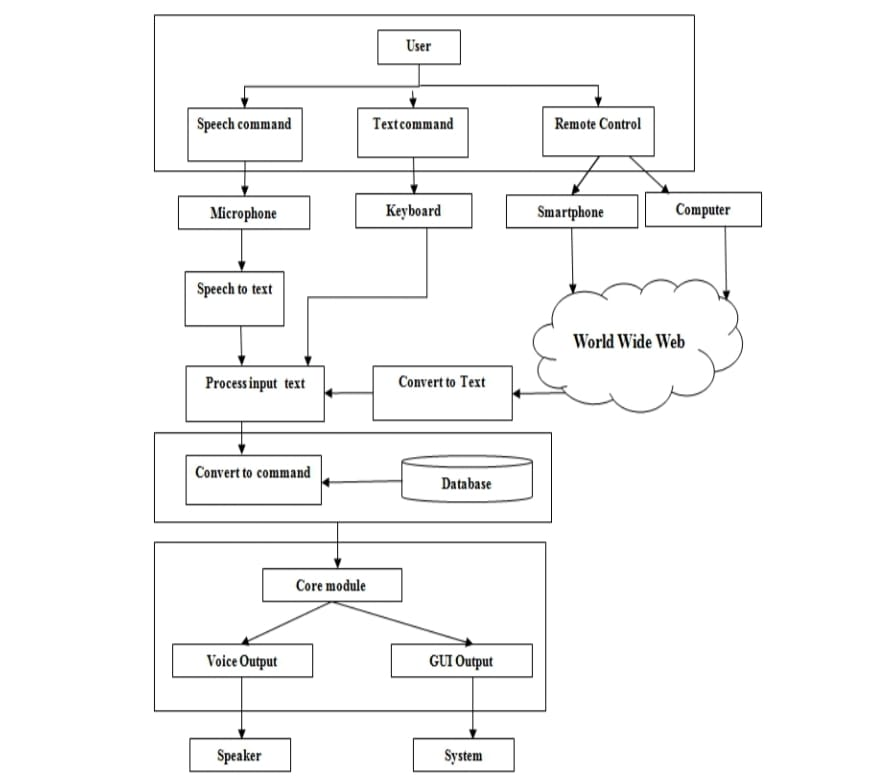


Fig. 1.3 System Architecture

**4.2 Flow Diagram**

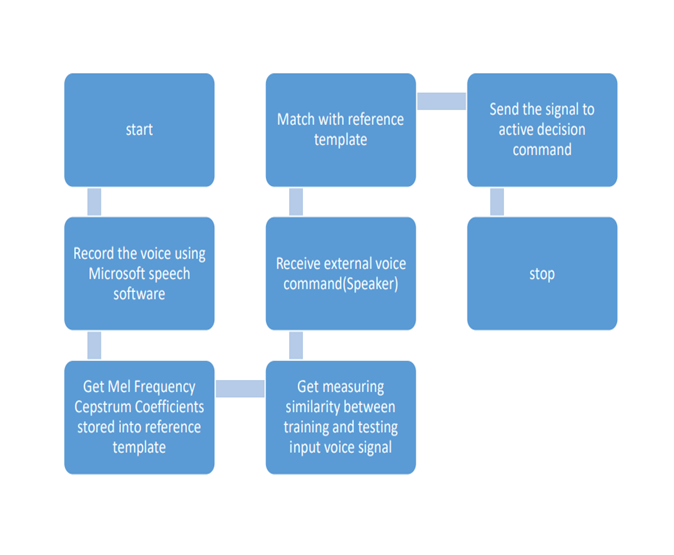
****

Fig. 1.4 Flow Diagram

**4.3 Application Working Diagram**

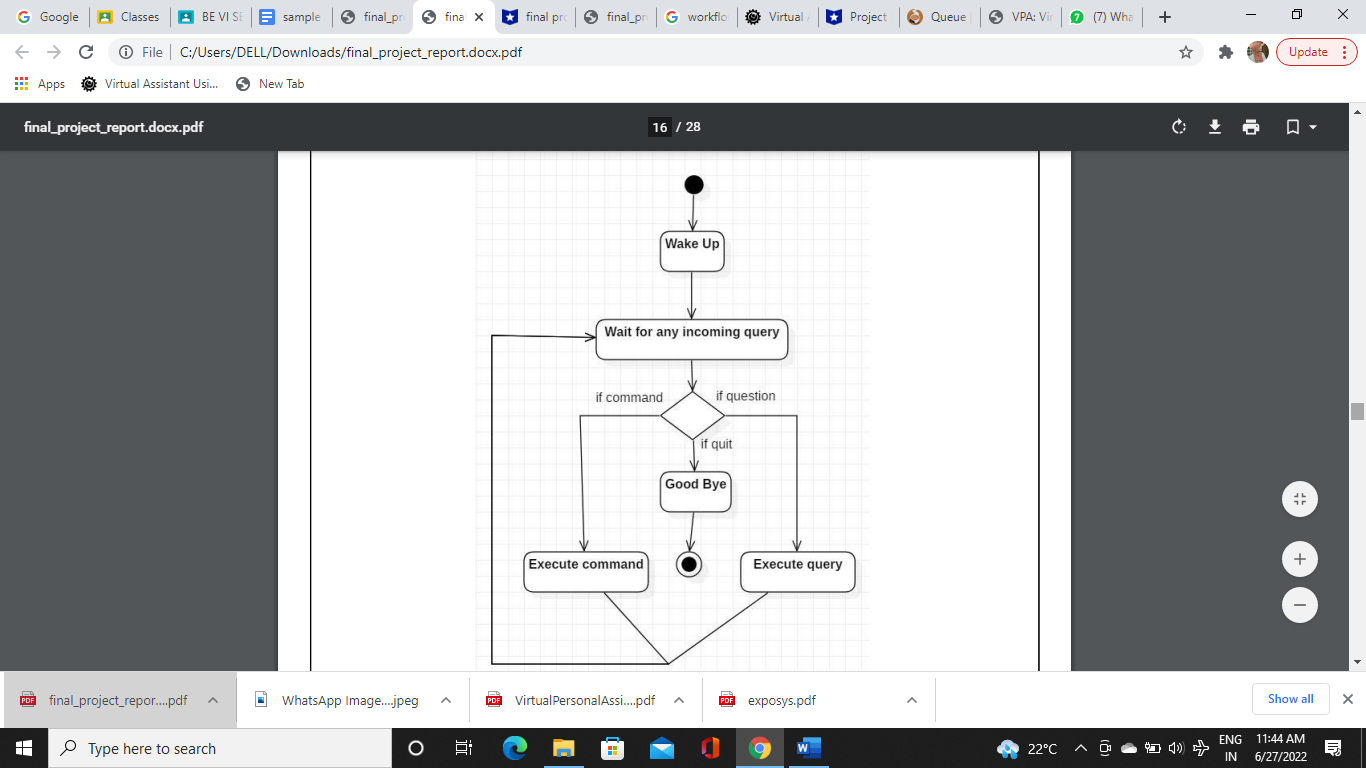
****

Fig. 1.5 Application working diagram

Initially, the system is in idle mode. As it receives any wake up cal it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

**4.4 Use Case Diagram**

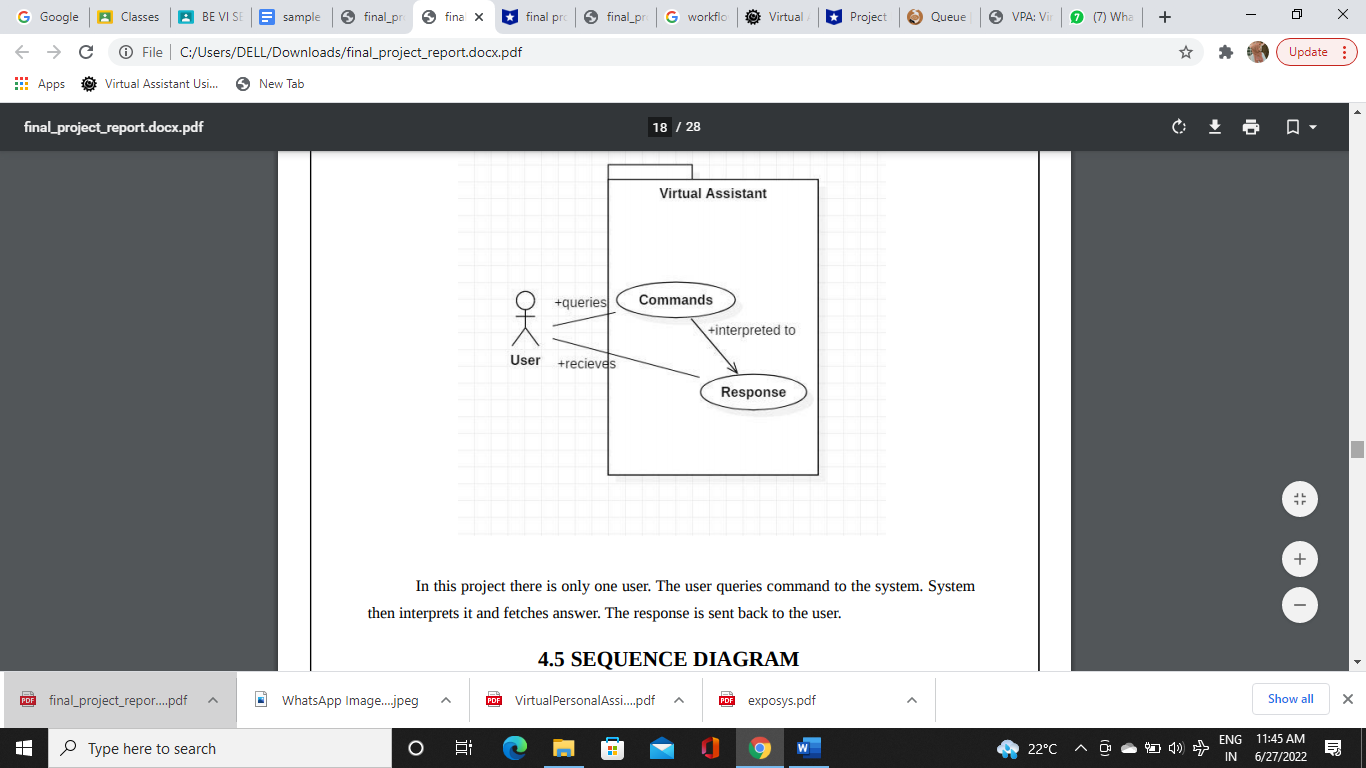
****

Fig. 1.6 Use Case Diagram

In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

**4.5 COMPOMENT DIAGRAM**

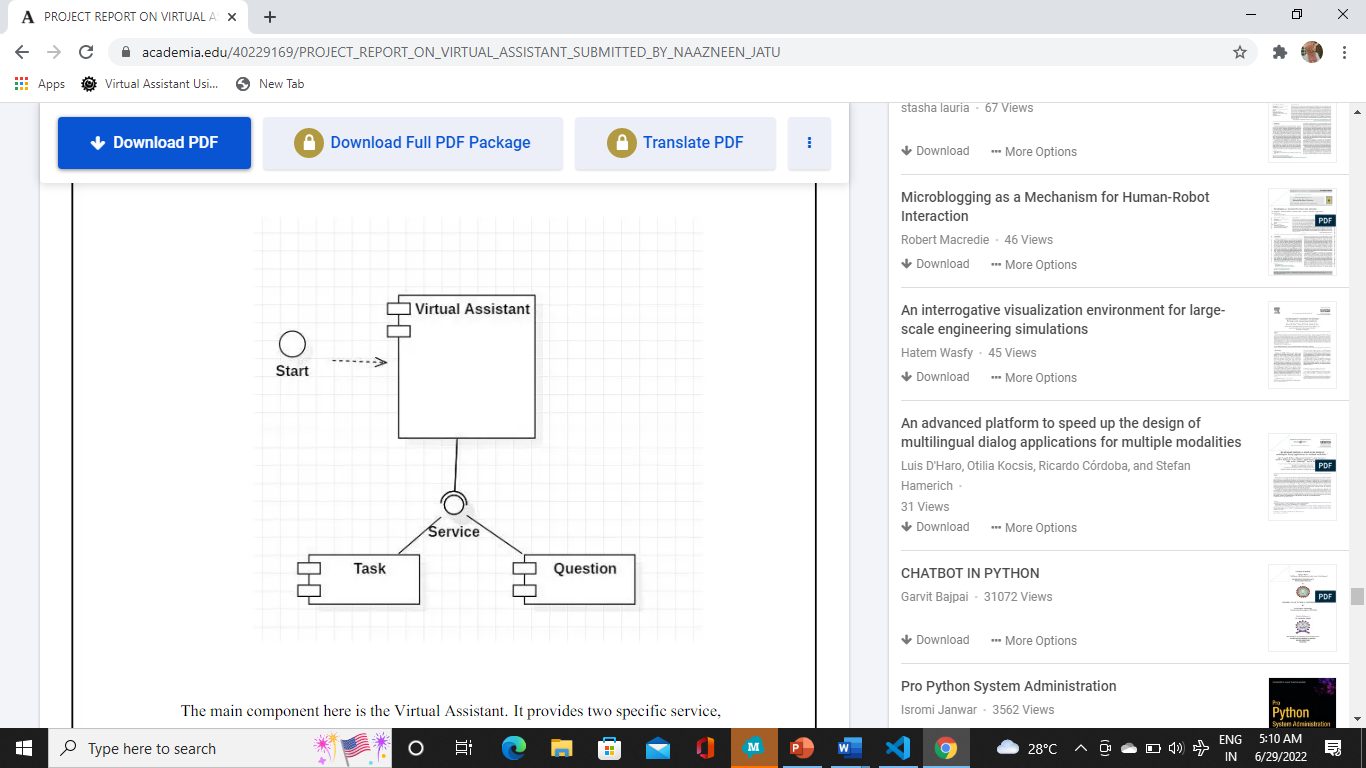


Fig. 1.7 Component Diagram

The main component here is the Virtual Assistant. It provides two specific service, executing Task or Answering your question.

**5. IMPLEMENTATION**

**5.1 Code**

**from cgitb import text**

**from email.mime import audio**

**import pyttsx3 #pip install pyttsx3**

**import speech\_recognition as sr #pip install speechRecognition**

**import datetime**

**import wikipedia #pip install wikipedia**

**import webbrowser**

**import os**

**import smtplib**

**import pyaudio**

**from newsapi import NewsApiClient**

**import sys**

**import random**

**import requests**

**from urllib.request import urlopen**

**import json**

**import config**

**from GoogleNews import GoogleNews**

**from textblob import TextBlob**

**from email.message import EmailMessage**

**import smtplib**

**import yagmail**

**import wolframalpha**

**sys.path.append('/usr/local/lib/python3.7/dist-packages/')**

**import pyjokes**

**import pywhatkit as pwk**

**newsapi = NewsApiClient(api\_key='API\_KEY')**

**googlenews=GoogleNews()**

**engine = pyttsx3.init('sapi5')**

**voices = engine.getProperty('voices')**

**# print(voices[1].id)**

**engine.setProperty('voice', voices[1].id)**

**def speak(audio):**

**engine.say(audio)**

**engine.runAndWait()**

**def wishMe():**

**hour = int(datetime.datetime.now().hour)**

**if hour>=0 and hour<12:**

**speak("Good Morning!")**

**print("Good Morning!")**

**elif hour>=12 and hour<18:**

**speak("Good Afternoon!")**

**print("Good Afternoon!")**

**else:**

**speak("Good Evening!")**

**print("Good Evening!")**

**speak("May I know your name Human?")**

**name=takeCommand()**

**speak('nice to meet you'+name)**

**speak('I am Jarvis, your virtual assistant. How may I help you?')**

**def takeCommand():**

**#It takes microphone input from the user and returns string output**

**r = sr.Recognizer()**

**with sr.Microphone() as source:**

**print("Listening...")**

**r.pause\_threshold = 1**

**audio = r.listen(source,timeout=10,phrase\_time\_limit=5)**

**r.adjust\_for\_ambient\_noise(source, duration=1)**

**try:**

**print('Recognizing...')**

**query = r.recognize\_google(audio, language="en-in")**

**print(f"User said: {query}\n")**

**except Exception as e:**

**#print(e)**

**print("Say that again please...")**

**return "None"**

**return query**

**class WeatherService(object):**

**API\_URL= "http://api.openweathermap.org/data/2.5/weather?q={}&APPID={}&units=metric";**

**API\_KEY = "3b31a7e394e41c3a30759dfde1a3383e";**

**def \_\_init\_\_(self):**

**pass**

**def weather():**

**city = "hyderabad"**

**res = requests.get(**

**f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid=16f0afad2fd9e18b7aee9582e8ce650b&units=metric").json()**

**temp1 = res["weather"][0]["description"]**

**temp2 = res["main"]["temp"]**

**speak(f"Temperature is {format(temp2)} degree Celsius \nWeather is {format(temp1)}")**

**def get\_random\_advice():**

**results = requests.get("https://api.adviceslip.com/advice").json()**

**return results['slip']['advice']**

**if \_\_name\_\_ == "\_\_main\_\_":**

**wishMe()**

**while True:**

**# if 1:**

**query = takeCommand().lower()**

**# Logic for executing tasks based on query**

**if 'wikipedia' in query:**

**speak('Searching Wikipedia...')**

**query = query.replace("wikipedia", "")**

**results = wikipedia.summary(query, sentences=2)**

**speak("According to Wikipedia")**

**print(results)**

**speak(results)**

**elif 'how are you' in query:**

**speak("I am good, how are you?")**

**print("i am good")**

**elif 'open youtube' in query:**

**webbrowser.open("youtube.com")**

**elif 'open google' in query:**

**webbrowser.open("google.com")**

**elif 'open stack overflow' in query:**

**webbrowser.open("stackoverflow.com")**

**elif 'play music' in query: path='C:\\Users\\DELL\\OneDrive\\Desktop\\jarvis.py\\audios'**

**files=os.listdir(path)**

**d=random.choice(files)**

**os.startfile(f"C:\\Users\\DELL\\OneDrive\\Desktop\\jarvis.py\\audios\\{d}")**

**elif 'the time' in query:**

**strTime = datetime.datetime.now().strftime("%H:%M:%S")**

**speak(f"The time is {strTime}")**

**print(f"The time is {strTime}")**

**elif 'open code' in query:**

**codePath= "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"**

**os.startfile(codePath)**

**elif "where is" in query:**

**listening = True**

**query = query.split(" ")**

**location\_url = "https://www.google.com/maps/place/" + str(query[2])**

**speak("Hold on, I will show you where " + query[2] + " is.")**

**webbrowser.open(location\_url)**

**elif 'how are you' in query:**

**speak("I am fine, Thank you")**

**speak("How are you, Sir")**

**elif 'song' in query:**

**song = query.replace('play ', '')**

**speak('playing ' + song)**

**pwk.playonyt("ami je tomar song", use\_api=False)**

**elif "advice" in query:**

**speak(f"Here's an advice for you all")**

**advice = get\_random\_advice()**

**speak(advice)**

**speak("For your convenience, I am printing it on the screen sir.")**

**print(advice)**

**elif ('weather' in query):**

**weather()**

**elif 'whatsapp' in query:**

**try:**

**# sending message to receiver**

**# using pywhatkit**

**pwk.sendwhatmsg("+917702515798","Hello, Jarvis here",22, 33)**

**print("All Messages Sent")**

**except:**

**print("An Unexpected Error!")**

**elif "calculate" in query:**

**app\_id = "Wolframalpha api id"**

**client = wolframalpha.Client(app\_id)**

**indx = query.lower().split().index('calculate')**

**query = query.split()[indx + 1:]**

**res = client.query(' '.join(query))**

**answer = next(res.results).text**

**print("The answer is " + answer)**

**speak("The answer is " + answer)**

**elif "write a note" in query:**

**speak("What should i write, sir")**

**note = takeCommand()**

**file = open('jarvis.txt', 'w')**

**speak("Sir, Should i include date and time")**

**snfm = takeCommand()**

**if 'yes' in snfm or 'sure' in snfm:**

**strTime = datetime.datetime.now().strftime("% H:% M:% S")**

**file.write(strTime)**

**file.write(" :- ")**

**file.write(note)**

**else:**

**file.write(note)**

**elif "show note" in query:**

**speak("Showing Notes")**

**file = open("jarvis.txt", "r")**

**print(file.read())**

**speak(file.read(6))**

**elif 'joke' in query:**

**speak(pyjokes.get\_joke())**

**print(pyjokes.get\_joke())**

**results=pyjokes.summary(query,sentences=3)**

**print(results)**

**elif 'who is the best' in query:**

**speak("undoubtedly, Sadia is the best")**

**elif 'text' in query:**

**speak('hi sir. sadia asked me to check on you? how do you feel?')**

**elif 'how are you' in query:**

**speak("I am good, how are you?")**

**elif 'news' in query:**

**news = webbrowser.open\_new\_tab ('https://timesofindia.indiatimes.com/home/headlines')**

**speak(news)**

**# Specify the query and number of returns')**

**speak('Here are some headlines from the Times of India,Happy reading')**

**elif 'whatsapp' in query:**

**try:**

**# sending message in Whatsapp in India so using Indian dial code (+91)**

**pwk.sendwhatmsg("+917386978652", "Hi sir, we share a common boss", 2, 13)**

**print("Message Sent!") #Prints success message in console**

**# error message**

**except:**

**print("Error in sending the message")**

**elif 'headlines' in query:**

**speak('Getting news for you...')**

**engine.runAndWait()**

**googlenews.get\_news('Todays news')**

**googlenews.result()**

**results=googlenews.gettext()**

**print(\*results[4:8],sep=',')**

**speak(results[4:8])**

**elif 'sentiment' in query:**

**r = sr.Recognizer()**

**with sr.Microphone() as source:**

**print("how do you feel?")**

**speak("how do you feel?")**

**audio=r.listen(source,timeout=15,phrase\_time\_limit=5)**

**print("ok")**

**try:**

**print("Text to speech:"+r.recognize\_google(audio))**

**except:**

**pass;**

**from textblob import TextBlob**

**r = sr.Recognizer()**

**blob=TextBlob(" " +r.recognize\_google(audio))**

**x=blob.sentiment.polarity**

**if x<0:**

**speak('you look sad dont worry, you will be fine')**

**print('you look sad dont worry, you will be fine')**

**elif x==0:**

**speak('thats alright')**

**print('thats alright')**

**elif x>0:**

**speak('you look happy so glad to hear this')**

**print('you look happy so glad to hear this')**

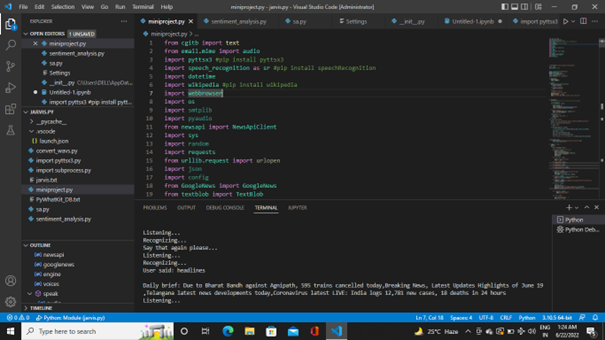
**elif x=<1:**

**speak('you look happy so glad to hear this')**

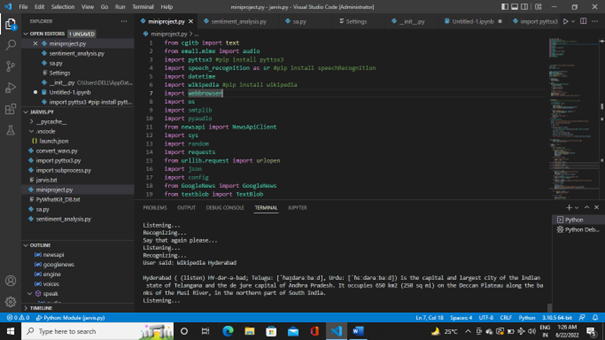
**print('you look happy glad to hear this')**

**5.2 Output**

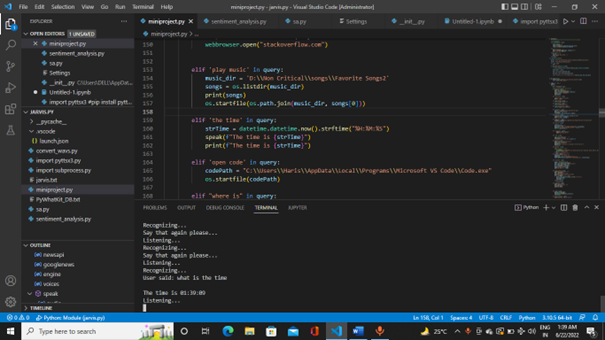
**5.2.1 Reads News Headlines**

****

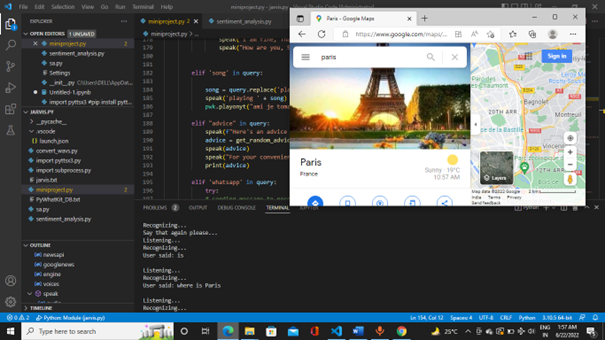
**5.2.2 Searches Wikipedia**

****

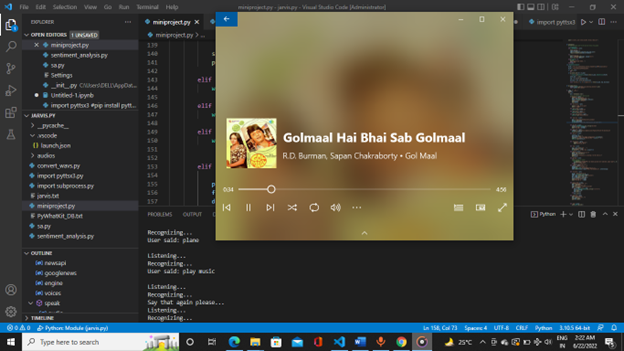
**5.2.3 Tells The Exact Time**

****

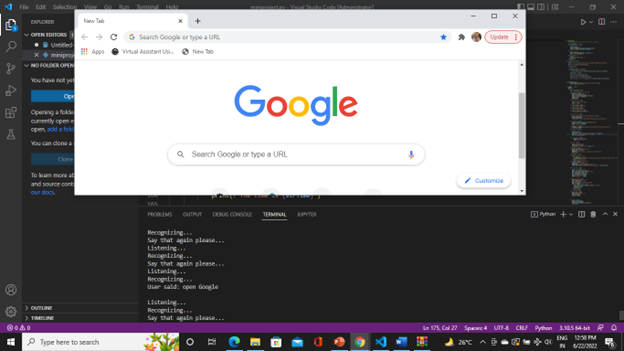
**5.2.4 Shows Location**

****

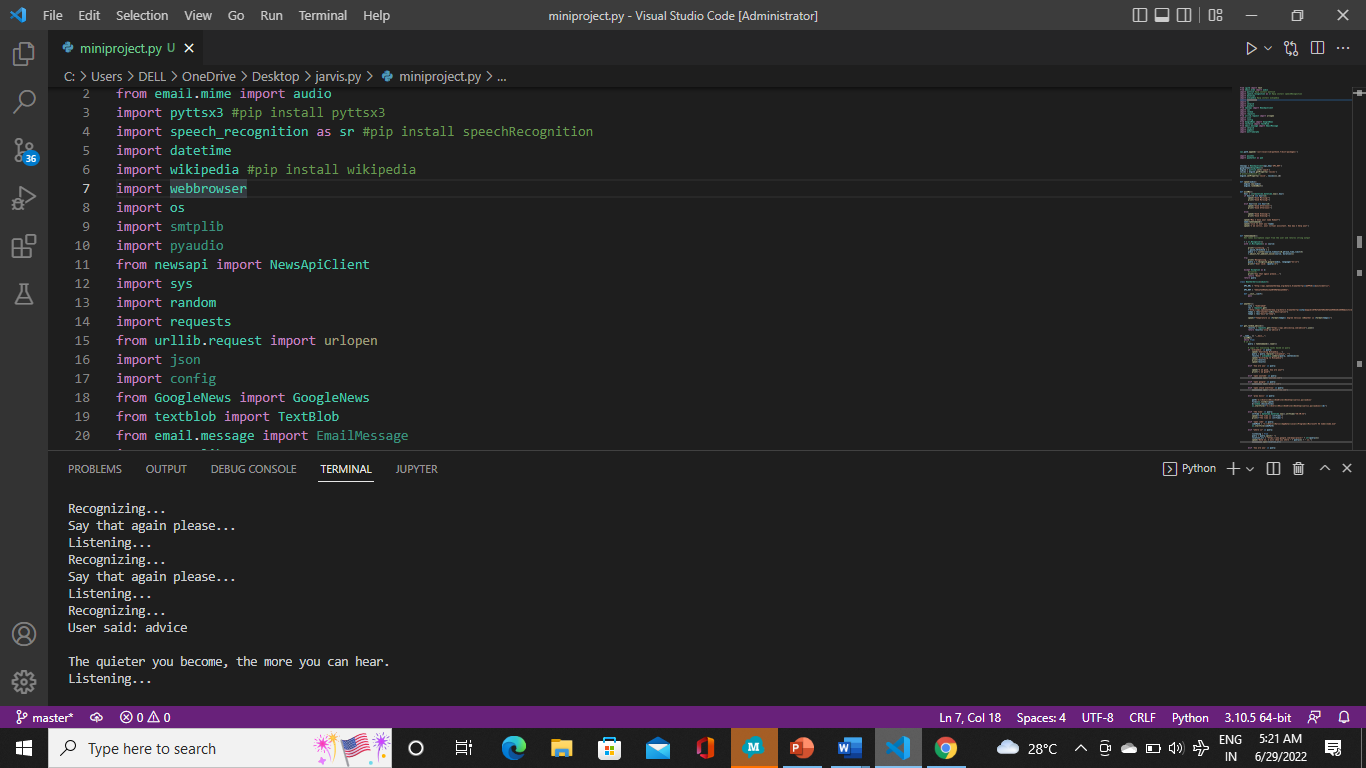
**5.2.5 Plays Music**

****

**5.2.6 Opens Google**

****

**5.2.7 Gives Advice**

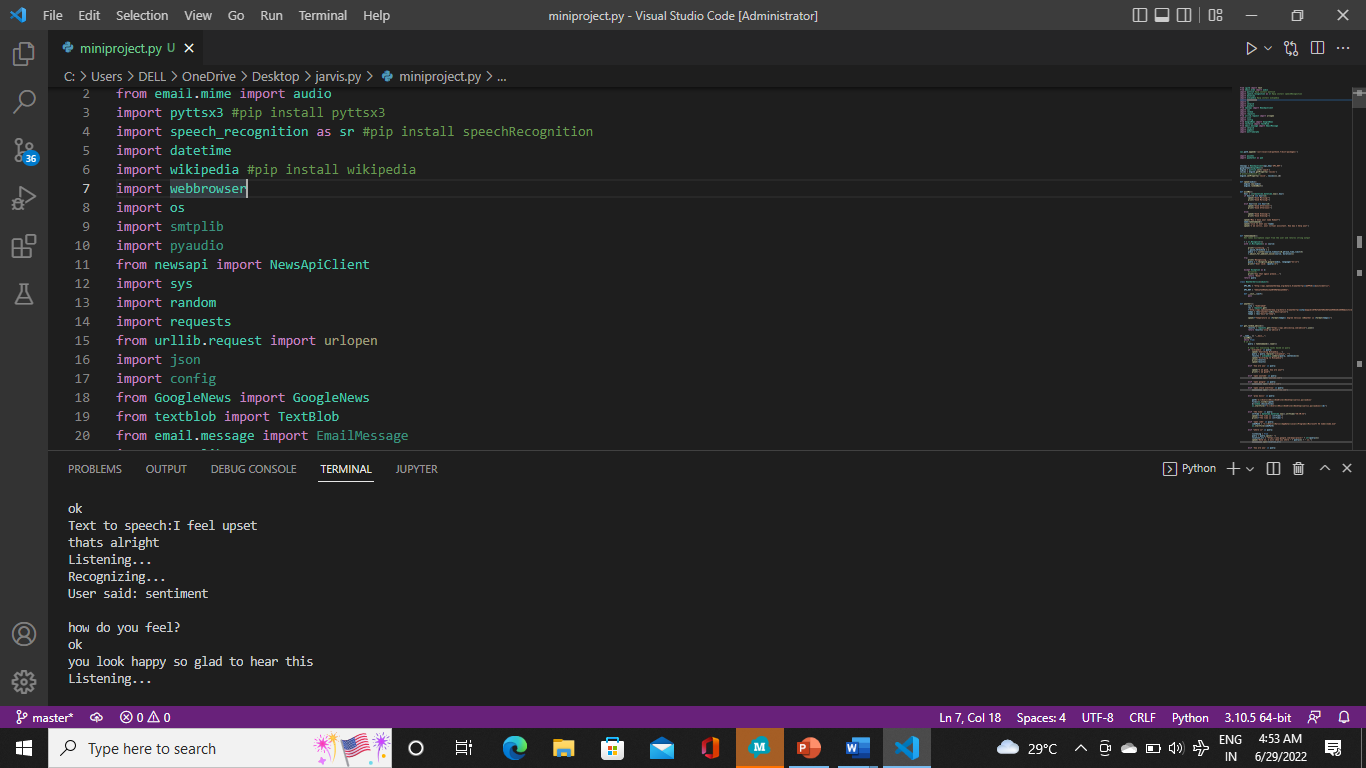


**6. SENTIMENT ANALYSIS**

Sentiment analysis is the act of using NLP to help us understand the opinion of the public in a particular context in natural language.

The evolution in NLP has been remarkable because we now have tools and packages that make solving problems with NLP very easy. One of those tools is Textblob.

* Textblob is a Python NLP library that uses a natural language toolkit (NLTK). It uses NLTK because it is simple, easy to deploy, will use up fewer resources, gives dependency parsing, and can be used even for small applications.
* Textblob can be used for complex analysis and working with textual data. When a sentence is passed into Textblob it gives polarity. Polarity is the output that lies between [-1,1], where -1 refers to negative sentiment and +1 refers to positive sentiment.



**7. FUTURE SCOPE**

To call any technology that makes our lives easier by one name is almost impossible. There are a variety of terms that refer to agents that can perform tasks or services for an individual, and they are almost interchangeable — but not quite. They differ mainly based on how we interact with the technology, the app, or a combination of both. Here are some basic definitions, similarities, and differences:

Intelligent Personal Assistant: This is software that can assist people with basic tasks,usually using natural language. Intelligent personal assistants can

go online and search for an answer to a user's question. Either text or voice can trigger an action.

**Automated Personal Assistant**: This term is synonymous with intelligent personal assistant.

**Smart Assistant**: This term usually refers to the types of physical items that can provide

various services by using smart speakers that listen for a wake word to become active and perform certain tasks. Amazon's Echo, Google's Home, and Apple's HomePod are types of smart assistants.

**Virtual Digital Assistants**: These are automated software applications or platforms that assist

the user by understanding natural language in either written or spoken form.

**Chatbot**: Text is the main way to get assistance from a chatbot. Chatbots can simulate a conversation with a human user. Many companies use them in the customer service sector to answer basic questions and connect with a live person if necessary.

**Voice Assistant**:

The key here is voice. A voice assistant is a digital assistant that uses voice

recognition, speech synthesis, and natural language processing (NLP) to provide a service through a particular application.

For the purpose of this discussion, the term voice assistant will be used interchangeably with the following related terms: intelligent personal assistant, automated personal assistant, smart assistant, and virtual digital assistant.

**8. CONCLUSION**

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. The future plans include integrating Jarvis with mobile using React Native to provide a synchronized experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.we used different types of Python APIs to get online service like Weather forecast, News report, Calculation, etc. which at the end becomes very useful features for the virtual assistant project. This assistant currently works online and performs basic tasks like weather updates, stream music, search Wikipedia, open desktop applications, etc. The functionality of the current system is limited to working online only.The accuracy of the devices can be increased using machine learning and categorizing the queries in particular result sets and using them further queries. The accuracy of the devices is increasing exponentially in the last decade. The devices can also be designed to accept commands in bilingual language and respond back in the same language queried by the user. The device can also be designed to help visually impaired people.

**9. REFERENCES**

· <https://codewithharry.com/videos/python-tutorials-for-absolute-beginners-120>

· <https://easychair.org/publications/preprint_open/j2GR>

· <https://www.freecodecamp.org/news/python-project-how-to-build-your-own-jarvis-using-python/>

· <https://extrudesign.com/virtual-assistant-using-python/>

· <https://www.youtube.com/watch?v=Lp9Ftuq2sVI&t=1708s>