Machine Learning-driven Personal Virtual Assistant

[1]Daneshwari N. Kori
Department of Information Science and Engineering
SDM College of Engineering and Technology
Dharwad, Karnataka, India

[2]Arati S. Nayak
Department of Information Science and Engineering
SDM College of Engineering and Technology
Dharwad, Karnataka, India

[3]Malathesh K
Department of Information Science and Engineering
SDM College of Engineering and Technology
Dharwad, Karnataka, India

---***---

Abstract - Modern society has brought greater sophistication and intelligence to everyday life. Some speech services, such as Google, Siri, etc., are already familiar to us. The fundamental objective of personal virtual assistant software is to help users complete their chores. Currently, our voice support system has features like automated Chrome, the ability to open social media websites in a web browser, the ability to know the time, the ability to request information from Wikipedia, the ability to send text messages via Whatsapp, etc. Using speech input, generating voice output, and screen text display, this project operates. Voice Help records voice input using our microphone and converts it into comprehensible computer language, delivering the answers and solutions the user needs. The World Wide Web is connected to this service in order to deliver the results the user has requested. The Machine Learning algorithm gives computers the ability to communicate in a variety of ways utilizing natural human language.

Keywords - Personal virtual assistant, Artificial Intelligence, Machine learning, Python.

I. INTRODUCTION

Virtual personal assistants (VPAs) are computer programme created to organically interact with humans to respond to questions, carry on discussions, and carry out different tasks. The two sorts of inputs that a VPA normally accepts are voice input (like Apple Siri) and text input (Google Assistant). The important thing to remember is that the end user should be able to communicate with the VPA using normal language, or in other words, as he would with a human, rather than needing to utilize a series of commands or a computer language. Even in the presence of these conventional input techniques, give certain consumers a better experience. The most recent advancements in ML technology can be directly related to the development of VPA. There are various definitions for machine learning. In a crude way, it's the capacity of a machine to reproduce a human being's cognitive behavior. the capacity of a machine, given specific inputs, to provide results that optimize specified success criteria Typically, algorithms is thought to be the field that deals with the development of these abilities. the capacity to get better at a certain task over time. Or, put another way, the capacity of a machine to advance autonomously in problem-solving.

II. LITERATURE REVIEW

[1] Machine learning-based intelligent personal assistant: The categorization of personal assistant agents usually employs a supervised machine learning technique called Support Vector Machine. The application will operate automatically and manage profiles without any help from a user. By enabling the user to keep track of everything, the application's Reminders—which are time, call, and location-based—will prevent the user from overlooking any crucial activities in their daily routine. To help users with daily activities, it would offer functionality such as profile management, reminders, and others.

[2] Correlational Analysis and Machine Learning Effects of Intelligent Virtual Assistants: Explores how these intelligent assistants' function, the AI approaches that are applied, their uses, their shortcomings, and how they may be improved.

[3] Deep learning and TensorFlow are used to develop a virtual assistant that can communicate via sign language: This article focuses on the technology and interface developed to allow deaf individuals to use a range of voice-activated virtual assistants with the aid of Sign Language.

[4] Future Virtual Personal Assistants (Google Home, Apple-Siri, Microsoft Cortana, Siri): The Next-Generation of Virtual Personal Assistants, a brand-new VPAs system created to have reasoned conversations with individuals, is introduced in this notion.

[5] Voice assistant powered by Artificial Intelligence (AI): The phrase "AI-based Voice assistants" refers to operating systems with in-built voices that can understand human speech. The voice assistant will capture microphone noises, transform them to text, and deliver them using GTTS (Google text to speech). The English-language audio file for the GTTS engine will be played using the play sound module of the Python computer language.

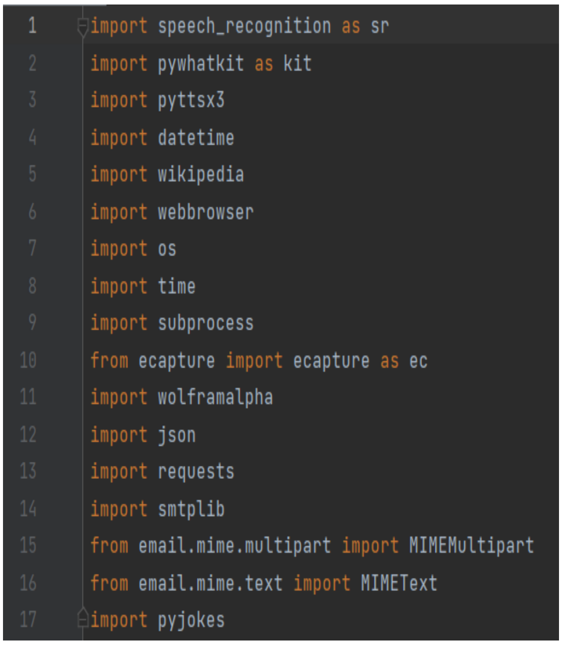
[6] Personal Virtual Assistant For Windows Using Python: Python is used to create a virtual personal assistant that is solely for Windows and accessible from any Windows explorer, including Windows 7, 8, and 10. Python is used as a programming language because it has a number of important libraries that are used to carry out tasks. Our personal virtual

assistant can identify and respond to user voice utilizing python installation packages.

[7] Desktop Assistant AI Using Python: Personal Assistant with Voice Recognition Intelligence, which uses Python to process user input in the form of voice or text and then provide the results to the user in a variety of ways, such as by dictating an action to be taken or the search result, uses voice recognition technology to recognize words in the user's speech.

[8] Python-based Voice Enabled Personal Assistant for Computers: A personal virtual assistant might even carry out some fundamental tasks, like opening apps, reading out the news, taking notes, etc., merely by speaking a voice command. Users can ask inquiries to them in the same way they would to a real person.

III. SYSTEM ARCHITECTURE



```
import speech_recognition as sr
import pywhatkit as kit
import pyttsx3
import datetime
import wikipedia
import webbrowser
import os
import time
import subprocess
from ecapture import ecapture as ec
import wolframalpha
import json
import requests
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import pyjokes
```

Fig. 1. Modules Imported to Implement the Personal Virtual Assistant

A. Imported Modules

i. Speech Recognition

A machine ability to hear and recognize spoken words is known as speech recognition. The uttered words can then be converted into text, a query can be posed, or a response can be given using Python's speech recognition software. Even some machinery may be set up to respond to these spoken phrases.

ii. pywhatkit

Python library PyWhatKit has several useful features. You don't need to perform any further setup because it is simple to use. It is currently one of the most used libraries for automating YouTube and WhatsApp.

The pywhatkit library has the characteristics listed below: WhatsApp message sending. YouTube video to be viewed. Google search is to be done.

iii. pyttsx3

The Python pyttsx3 package converts text into speech. It functions offline and is compatible with Python versions 2 and 3, in contrast to competing libraries. A reference to a pyttsx3. Engine instance can be obtained by an application using the pyttsx3.init() factory method.

iv. datetime

Datetime in Python combines dates and times. The traits of this class are similar to those of the different and date classes. These attributes include day, month, year, minute, second, microsecond, hour, and tzinfo.

v. wikipedia

Accessing and parsing Wikipedia data is made easier by the Python package Wikipedia. You can perform a search on Get information from a page, such as links and photos, using Wikipedia, article summaries, and more. Wikipedia covers the MediaWiki API so that you can concentrate on using it rather than obtaining it. Before printing Wikipedia, import it.

vi. webbrowser

Viewing web-based content is made possible through the webbrowser module, which provides a high-level user interface. The open() function from this module will typically produce the desired result.

vii. os

The OS module for Python includes tools for a variety of tasks, including adding and deleting folders (directories), reading their contents, changing and locating the directory that is currently open, and more. The os module must initially be imported before you may interact with the underlying operating system.

viii. subprocess

To run new applications and scripts by starting new processes, a Python module named subprocess is needed. It enables you to start up fresh applications right from a Python programme you're writing. Subprocess is a Python feature that allows you to run code from C or C++ programmes as well as code from other apps from a git repository.

ix. wolframalpha

Wolfram Alpha is an API that uses Wolfram's algorithms, knowledge base, and AI technologies to provide answers at the expert level. Because of the Wolfram Language, it is conceivable.

x. smtplib

Use the SMTP client session object specified by the smtplib module to send email to any internet-connected device that has an SMTP or ESMTP listener daemon.

xi. json

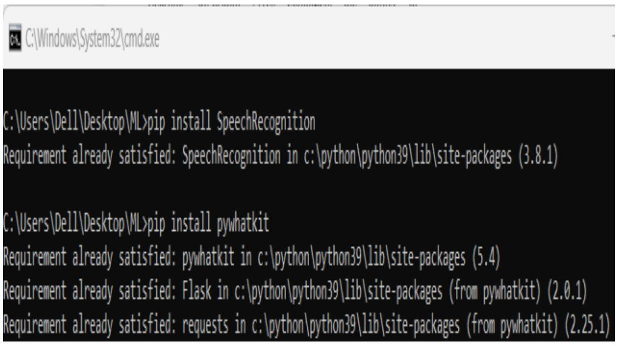
Data is routinely sent across a network as text using JavaScript Object Notation (JSON), a standardized format. It is used by numerous APIs and databases, and both humans and robots may readily access it. JSON uses name/value pairs to describe objects, much like a Python dictionary.

xii. pyjokes

Using the application Pyjokes, jokes in the Python language are compiled online. Because it contains jokes, Pyjokes is a part of our project. It really is fascinating. The one-line joke known as Pyjokes is what makes our project interesting.

xiii. requests

Sending HTTP requests is possible with Python's Requests module. To send GET and POST requests, use Queries. It hides the difficulties of submitting queries behind a clean, straightforward API.



```
C:\Users\Dell\Desktop\ML>pip install SpeechRecognition
Requirement already satisfied: SpeechRecognition in c:\python\python39\lib\site-packages (3.8.1)

C:\Users\Dell\Desktop\ML>pip install pywhatkit
Requirement already satisfied: pywhatkit in c:\python\python39\lib\site-packages (5.4)
Requirement already satisfied: Flask in c:\python\python39\lib\site-packages (from pywhatkit) (2.0.1)
Requirement already satisfied: requests in c:\python\python39\lib\site-packages (from pywhatkit) (2.25.1)
```

Fig. 2. Installation of packages

B. System Testing

Stages carried out to build the personal virtual assistant

Step 1: Begin or start.

Step 2: Install all necessary packages.

Step 3: Execute the code, such as python filename.py.

Step 4: Use voice commands to activate the virtual assistant.

Step 5: Does the command match as stated in the provided code? if so, go to step 6.

Step 6: Display the outcomes in accordance with the specified instruction.

Step 7: Exit if not.

Flow Chart

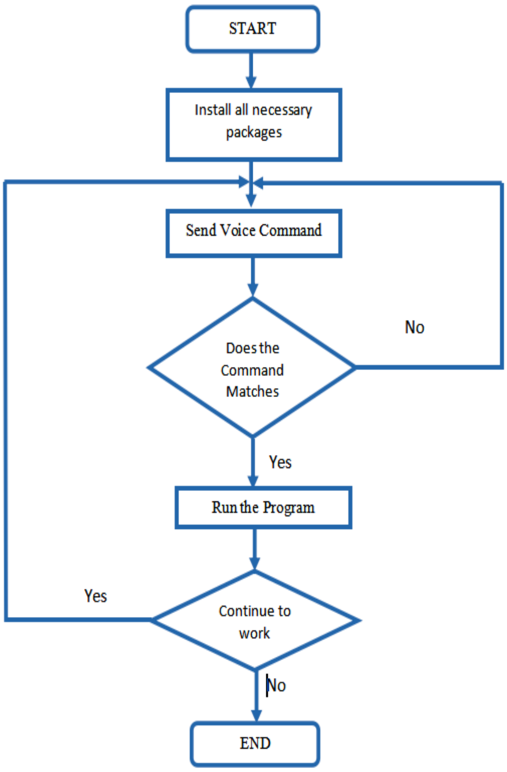


Fig. 3. Flow Chart for a virtual personal assistant [6]

Activity Diagram

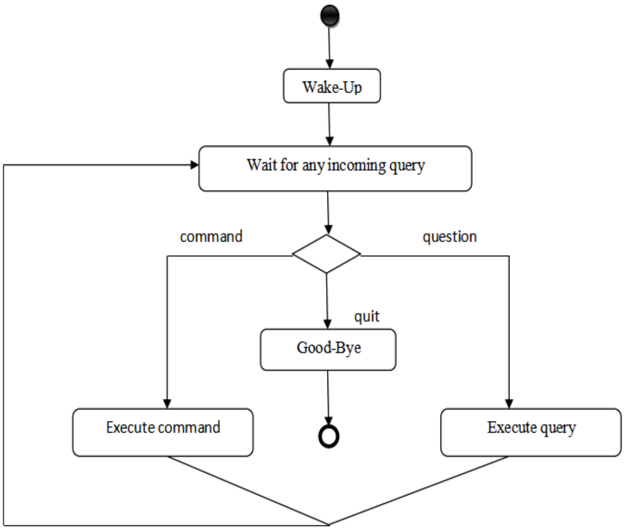


Fig. 4. Activity_Diagram

Initially, the system is in standby mode. It starts working as soon as it gets a wake-up call. The type of desired action—a task to be accomplished or a questionnaire—is determined. As a result, a specified action is taken. The system waits for another command until the issue has been fixed or the task has been completed. Unless a stop command is received, this loop runs endlessly. The moment that happens, it promptly goes back to sleep.

Block-Diagram

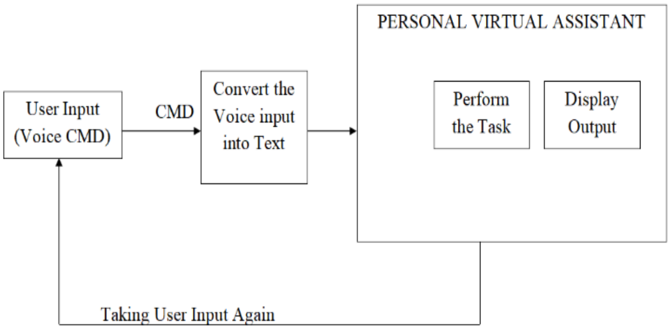


Fig. 5. Program Block Diagram for a virtual personal assistant [8]

a. **User Input (Voice CMD or Voice Command):** The assistant will first greet the user with respect to time before asking for a task and then listening to it.

b. **Concert the Voice I/P into Text:** The input command will be converted into text in this case with the aid of the speech recognition module.

c. **Personal Virtual Assistant**: Perform The Task and Display Output are the two sub-blocks that make up this assistant. Perform the Task: At this point, depending on the user's input, it will carry out the task, and Display Output: Here is the outcome of the assignment.

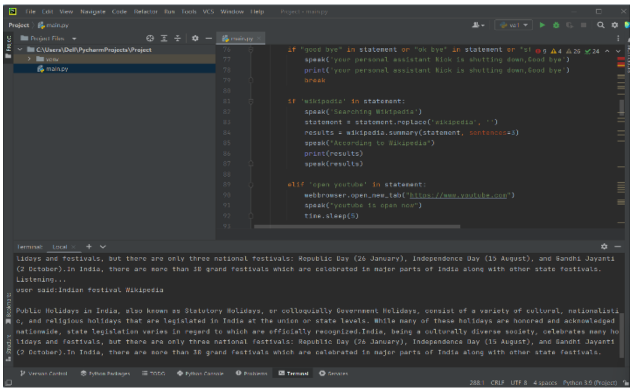
d. **Feedback:** The assistant will continue to ask for input and carry out tasks until instructed to "turn off" in a loop. The instruction to turn off will end the loop.

IV. METHODOLOGY

Voice and text input from users are converted into commands by virtual assistants. Natural language audio signals are transformed into executable commands or digital data that the programme may examine when a user requests their personal assistant to carry out a task. Then, to determine a suitable response, this data is compared to a data set from the software. With the help of a virtual assistant, you can control machinery. Speech recognition, Smtplib, Json, Wolfram Alpha, datetime, and other Python installation packages are used while creating virtual assistants. Transcribing audio into text is called speech recognition. Python has a Speech Recognition API that enables us to convert spoken or audio commands into text for subsequent processing. When a user gives a virtual assistant a command, it first listens for it, recognizes it, and then displays the user's command on the screen. In addition, it compares and analyses the commands that we have given in the code. Following the matching request, both text and speech output are done. The personal virtual assistant displays the results in accordance with the commands the user has provided.

V. RESULT AND DISCUSSION

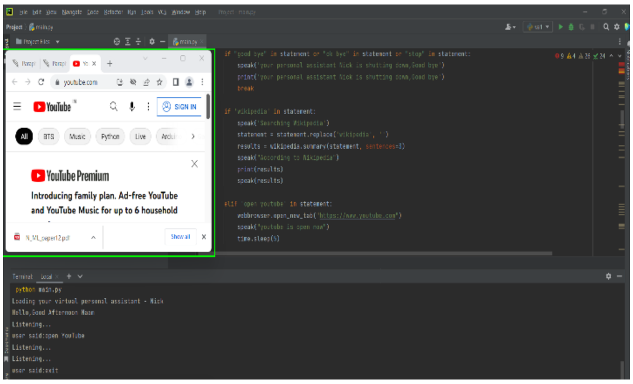
The required packages for the Python programming language have been set up, and the code was written using PyCharm's integrated development environment (IDE). Virtual assistant implementation includes questions like, "How are you?" and "What are you doing?" and exit to terminate the application. Wikipedia, a location map, the local weather, YouTube, Google, Gmail, Flipkart, Prime Video, Netflix, PyCharm, a PPT or a Word document etc can all be opened. Using a camera to take pictures and sending of Whatsapp messages.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Project main.py
Project Files
C:\Users\Dell\PycharmProjects\Project
venv
main.py
Terminal: Local
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
288:1 CRLF UTF 8 4 spaces Python 3.9 (Project)

```
        if "good bye" in statement or "ok bye" in statement or "st
            speak('your personal assistant Nick is shutting down,Good bye')
            print('your personal assistant Nick is shutting down,Good bye')
            break

        if 'wikipedia' in statement:
            speak('Searching Wikipedia')
            statement = statement.replace('wikipedia', '')
            results = wikipedia.summary(statement, sentences=3)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'open youtube' in statement:
            webbrowser.open_new_tab("https://www.youtube.com")
            speak("youtube is open now")
            time.sleep(5)
```

```
lidays and festivals, but there are only three national festivals: Republic Day (26 January), Independence Day (15 August), and Gandhi Jayanti
(2 October).In India, there are more than 38 grand festivals which are celebrated in major parts of India along with other state festivals.
Listening...
user said:Indian festival Wikipedia

Public Holidays in India, also known as Statutory Holidays, or colloquially Government Holidays, consist of a variety of cultural, nationalisti
c, and religious holidays that are legislated in India at the union or state levels. While many of these holidays are honored and acknowledged
nationwide, state legislation varies in regard to which are officially recognized.India, being a culturally diverse society, celebrates many ho
lidays and festivals, but there are only three national festivals: Republic Day (26 January), Independence Day (15 August), and Gandhi Jayanti
(2 October).In India, there are more than 38 grand festivals which are celebrated in major parts of India along with other state festivals.
```

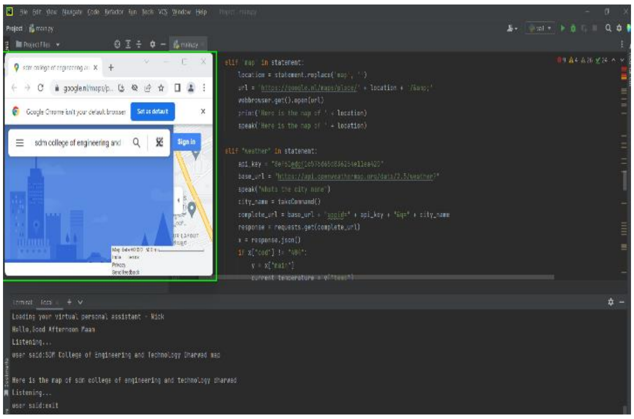
Fig. 6. Display for Wikipedia



```
if "good bye" in statement or "ok bye" in statement or "stop" in statement:
    speak('your personal assistant Nick is shutting down,Good bye')
    print('your personal assistant Nick is shutting down,Good bye')
    break

if 'wikipedia' in statement:
    speak('Searching Wikipedia')
    statement = statement.replace('wikipedia', '')
    results = wikipedia.summary(statement, sentences=3)
    speak("According to Wikipedia")
    print(results)
    speak(results)

elif 'open youtube' in statement:
    webbrowser.open_new_tab("https://www.youtube.com")
    speak("youtube is open now")
    time.sleep(5)
```

```
python main.py
Loading your virtual personal assistant - Nick
Hello,Good Afternoon Maam
Listening...
user said:open YouTube
Listening...
Listening...
user said:exit
```

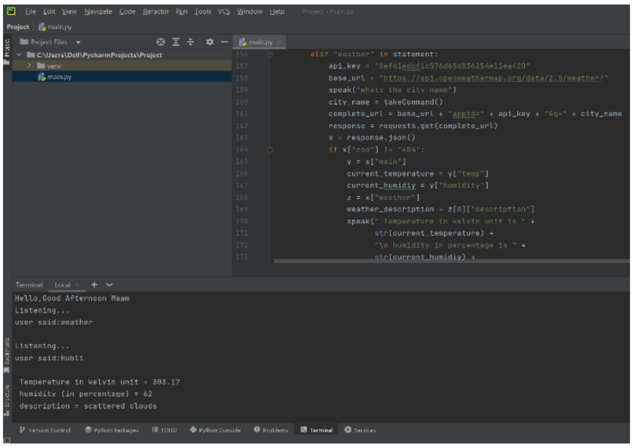
Fig. 7. The YouTube Output Screen


sdm college of engineering and
Google Chrome isn't your default browser
Set as default
Sign in
Privacy
Send feedback

```
elif 'map' in statement:
    location = statement.replace('map', '')
    url = 'https://google.nl/maps/place/' + location + '/&amp;'
    webbrowser.get().open(url)
    print('Here is the map of ' + location)
    speak('Here is the map of ' + location)

elif "weather" in statement:
    api_key = "8ef61edcf1c576d65d836254e11ea420"
    base_url = "https://api.openweathermap.org/data/2.5/weather?"
    speak("whats the city name")
    city_name = takeCommand()
    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()
    if x["cod"] != "404":
        y = x["main"]
        current_temperature = y["temp"]
```

```
Loading your virtual personal assistant - Nick
Hello,Good Afternoon Maam
Listening...
user said:SDM College of Engineering and Technology Dharwad map

Here is the map of sdm college of engineering and technology dharwad
Listening...
user said:exit
```

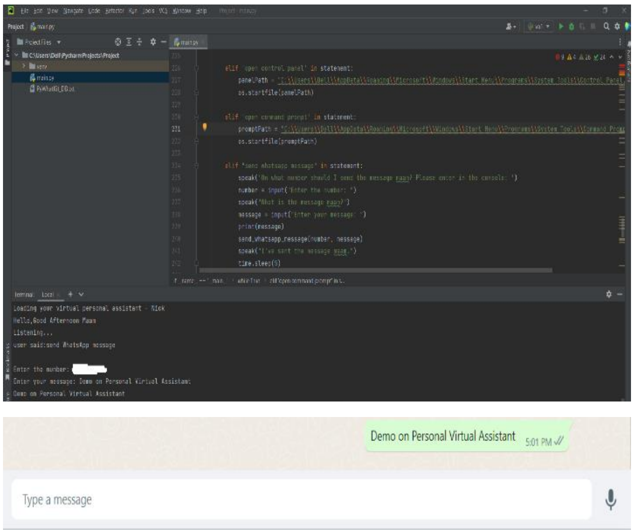
Fig. 8. Google Map Output Screen



```
elif "weather" in statement:
    api_key = "8ef61edcf1c576d65d836254e11ea420"
    base_url = "https://api.openweathermap.org/data/2.5/weather?"
    speak("whats the city name")
    city_name = takeCommand()
    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()
    if x["cod"] != "404":
        y = x["main"]
        current_temperature = y["temp"]
        current_humidiy = y["humidity"]
        z = x["weather"]
        weather_description = z[0]["description"]
        speak(" Temperature in kelvin unit is " +
              str(current_temperature) +
              "\n humidity in percentage is " +
              str(current_humidiy) +
```

```
Hello,Good Afternoon Maam
Listening...
user said:weather

Listening...
user said:Hubli

 Temperature in kelvin unit = 303.17
 humidity (in percentage) = 62
 description = scattered clouds
```

Fig. 9. Climatic Condition Output Screen



```
elif 'open control panel' in statement:
    panelPath = "C:\\Users\\Dell\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\System Tools\\Control Panel
    os.startfile(panelPath)

elif 'open command prompt' in statement:
    promptPath = "C:\\Users\\Dell\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\System Tools\\Command Prom
    os.startfile(promptPath)

elif "send whatsapp message" in statement:
    speak('On what number should I send the message maam? Please enter in the console: ')
    number = input('Enter the number: ')
    speak('What is the message maam?')
    message = input('Enter your message: ')
    print(message)
    send_whatsapp_message(number, message)
    speak("I've sent the message maam.")
    time.sleep(5)
```

```
Loading your virtual personal assistant - Nick
Hello,Good Afternoon Maam
Listening...
user said:send WhatsApp message

Enter the number: [redacted]
Enter your message: Demo on Personal Virtual Assistant
Demo on Personal Virtual Assistant
```

Demo on Personal Virtual Assistant 5:01 PM

Type a message

Fig. 10. Whatsup Message Output Screen

VI. CONCLUSION

Hands-free application today, it is highly helpful in everyday life. It is an incredibly easy application. Voice assistants are useful in a wide range of contexts, such as education, everyday life, home appliances, etc. Humans' lives are made easier by virtual assistants. The freedom to only hire a virtual assistant for the services they require. Using a virtual personal assistant to manage or organize your schedule is a good idea. Because they are more movable, dependable, and always accessible. The reliability of virtual personal assistants exceeds that of human personal assistants. Our virtual assistant will learn more about you and inform you of suggestions as well as take commands.

VII. REFERENCES

[1] Prajyot Mane; Shubham Sonone; Nachiket Gaikwad; Jyoti Ramteke, "Smart personal assistant using machine learning", 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), no. 4, 01-02 August, 01-02 August, 2017.

[2] Mohd Aijaj Khan; Anubhav Tripathi; Aaradhya Dixit; Manish Dixit, "Correlative Analysis and Impact of Intelligent Virtual Assistants on Machine Learning", 2019 11th International Conference on Computational Intelligence and Communication Networks (CICN), no. 7, 03-04 January, Honolulu, HI, USA, 2019.

[3] Dipanshu Someshwar; Dharmik Bhanushali; Vismay Chaudhari; Swati Nadkarni, "Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)), no. 6, 115-17 July, Coimbatore, India, 2020.

[4] Veton Këpuska; Gamal Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)", 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), no. 5, 08-10 January, Las Vegas, NV, USA, 2018.

[5] S Subhash; Prajwal N Srivatsa; S Siddesh; A Ullas; B Santhosh, "Artificial Intelligence-based Voice Assistant", 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), no. 4, 27-28 July, London, UK, 2020.

[6] Anjali Fapal; Trupti Kanade; Bharati Janrao; Mrunalini Kamble; Megha Raule, "Personal Virtual Assistant For Windows Using Python", International Research Journal of Modernization in Engineering Technology and Science, no. 7, volume 3, Issue 7, July-2021.

[7] Abeed Sayyed; AshpakShaikh; AshishSancheti; Swikar Sangamnere; Jayant H Bhangale, "Desktop Assistant AI Using Python", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), no. 9, volume 6, Issue 2, June 2021.

[8] V. Geetha; C.K.Gomathy; Kottamasu Manasa Sri Vardhan; Nukala Pavan Kumar, "The Voice Enabled Personal Assistant for Pc using Python", International Journal of Engineering and Advanced Technology (IJEAT), no. 4, volume-10, Issue-4, April 2021.