

Project Report: Virtual Assistant for Linux

Akshat Jain (401971879), Nikhil Katariya (401975324), Rohit Mehta (401324986)

May 2022

1 Introduction

When a technology becomes popular and the competition is fierce, open source alternatives are almost always available. These open software ideas can serve as a starting point for creating a customizable intelligent personal assistant. It takes time and effort to build a personal AI assistant, but there are benefits, such as customisation and data protection and well known software which already exist like google assistant or siri provides these functionalities for basic medical prescriber, daily schedule reminder, note writer, calculator, and search tool as in virtual assistant. So virtual assistant is a self-contained entity that offers administrative assistance to users as it offers users technical and social assistance. The virtual assistant can perform a variety of services ranging from scheduling appointments, making phone calls, arranging travel plans, and managing email accounts are all common responsibilities as well it includes secretarial work, customer care support, social media marketing, and web editing for a specific site or app. Virtual assistant services may be employed in any firm, and with proper training, the effectiveness of the work can be continually increased. There are a number of supporting technologies that enable the virtual assistant to function as a real-time assistant. Natural Language Processing (NLP), Artificial Intelligence (AI), and IoT-enabled technologies and apps, for example, serve as the foundation for virtual assistants. We used NLP in our project, which is a branch of computer science, artificial intelligence, and computational linguistics dealing with the interaction of computers and human (natural) languages, as well as programming computers to handle vast amounts of natural language text. It's a form of "software" that allows computers to read, analyse, understand, and derive valuable meaning from natural human languages. It is used to decipher the meaning and intent of text strings. In a word, NLP is a technique for assisting machines in comprehending human language.

2 Goal

Our voice assistance's major goal is to make people smarter by providing immediate and calculated outcomes. The voice help takes our vocal input through our microphone and translates it into computer comprehensible language, providing the user with the appropriate solutions and answers. This tool connects to the internet to deliver results that the user has questioned. The Natural Language Processing algorithm enables computer machines to communicate in a variety of ways using natural human language.

3 Related Work

Each virtual assistant developer employs his or her own development processes and approaches, which has an impact on the final product. The collection of traits that an assistant possesses is entirely dependent on the area of development the developer has prioritised. Because all systems are built on machine learning methods and employ massive amounts of data acquired from various sources and then trained on them, the source of this data, whether search systems, other information sources, or social networks, plays an essential role. These are some of the prior work in the open-source virtual assistant field.

3.1 Jarvis

It's a terminal-based personal assistant for Linux that's completely free and written in Python. When specified by a syntax that has been previously configured, the tool is responsible for conducting certain tasks. Jarvis gives you access to information about your computer, such as RAM, temperature, and processes, as well as your geographic position, allowing you to find restaurants, businesses, stadiums, and addresses near you. Similarly, we may use the program to download music, read news, search for photographs, and calculate routes, among other things. It enables the installation of new features and integration with third-party services. Instead of these features, the program currently lacks artificial intelligence integration, which should be a future goal for the developer if they want their application to be unique.

3.2 Dragonfire

It is open-source software written in Python by Mehmet Mert Yldran that functions as an Ubuntu virtual assistant. It integrates a number of technologies to make its users' lives simpler by doing highly exact searches, executing activities for you, and learning from the data that you supply. This can answer any of your queries since it has outstanding technology for searching for information on the internet, its dialogs are brief but precise, and it integrates well with other programs such as YodaQA and TeachableAI. The program is extremely intriguing in that it speaks to you fluently, and you can arrange it such that the answers to your queries are read to you by the tool.

3.3 Mycroft

Mycroft is an open-source artificial intelligence voice assistant that operates on a variety of platforms, including Linux. It's very advanced, and you can ask it to do things like play the news or set a timer. You won't have to bother about typing commands because Mycroft is speech-based. For example, if you want to set a timer for five minutes, all you must do is say, "Set a timer for five minutes." Mycroft will identify your voice and start your timer. As a result, it is quite effective at converting voice to text. It is also available in hardware devices that operate the assistant, comparable to the Amazon Echo.

Mycroft has more advanced features than other virtual assistants for Linux, such as Betty, which is a simple command-line-based assistant that is equivalent to Apple's Siri personal assistant and Google Voice Search.

4 Tools and Technology

1. IDE: Visual Studio Code
2. Programming Language: Python 3.10.4
3. Python Libraries:
 - (a) SpeechRecognition
 - (b) Pyttsx3
4. API: Wolfram alpha
5. Require Ubuntu 20.04 or lower

5 Methodology of Virtual Assistant

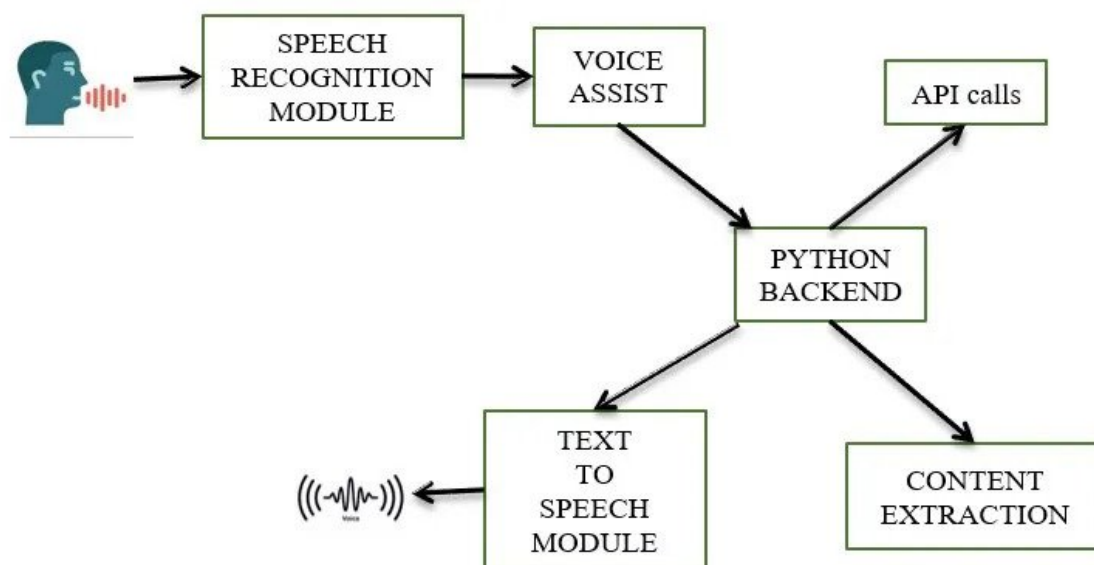


Figure 1: test

5.1 Speech Recognition Module

The process of translating spoken words into written texts is known as speech to text conversion. Although it is synonymous with speech recognition, the latter is used to refer to the broader process of speech comprehension. STT uses the same ideas and stages for speech recognition, but each phase uses a new combination of approaches.

5.2 API calls

API is an abbreviation for Application Programming Interface. An API is a software interface that allows two applications to communicate with one another. In other words, an API serves as a messenger, delivering your request to the provider and then returning the response to you. In this project, we use various APIs such as Wolfram Alpha for performing mathematical operations.

5.3 Interpret command

This is primarily concerned with Context extraction (CE). Context extraction (CE) is the process of extracting structured information from unstructured and/or semi-structured machine-readable documents automatically. In most cases, this activity involves using natural language processing to process human language texts (NLP). Recent developments in multimedia document processing, such as automatic annotation and content extraction from images/audio/video, could be viewed as context extraction TEST RESULTS.

5.4 Text-to-speech module

Text-To-Speech is a process that involves analysing, processing, and understanding incoming text before converting it to digital audio and speaking it. This is explained as follows: a TTS engine translates written text to a phonemic representation, which is subsequently converted to waveforms that can be output as sound.

1. Text Processing: The incoming text is analysed, normalised (to match the text and manage acronyms and abbreviations), and transcribed into phonetic or linguistic representation.
2. Speech Synthesis: Here the following techniques are used for this synthesis
 - (a) Articulator Synthesis
 - (b) Formant Synthesis
 - (c) Concatenative Synthesis

6 Feature and Skills

1. Asynchronous command execution speech recognition and interpretation
2. Configurable name for assistant on run time support
3. Assistant keep the commands history
4. Customization of voice-command
 - (a) The volume of the output audio
5. Greet User
6. Setting Alarm
7. Perform maths operations using Wolfram Alpha
8. Tell current Date and Time
9. Tell the current Location
10. Tell the top 5 news headlines
11. Make a note for the user
12. Open any software or app in the system

13. Show the system statistics
14. Tell the weather of any city
15. Open any website
16. Search anything on wikipedia
17. Can tell a Joke

7 Procedure

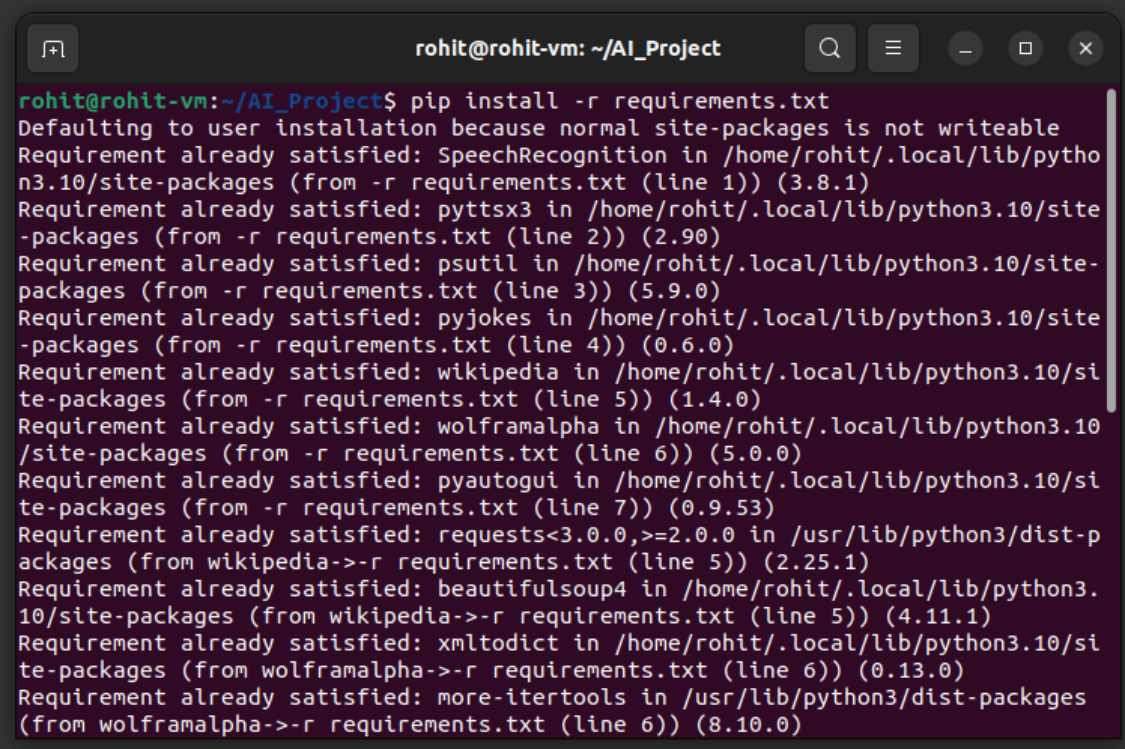
7.1 Install pip:

\$sudo apt install python3-pip

pip is the de facto and recommended package-management system written in Python and is used to install and manage software packages. We need pip to install some python packages and libraries that are required to run this project.

7.2 Install the libraries, packages, and dependencies:

There is a requirement text file that has the libraries and dependencies that are required to run this project such as SpeechRecognition, pyttsx3, pyjoke, etc.

A terminal window titled 'rohit@rohit-vm: ~/AI_Project' with a search icon, menu icon, and window controls. The terminal shows the command 'pip install -r requirements.txt' and its output. The output lists several packages already installed: SpeechRecognition (3.8.1), pyttsx3 (2.90), psutil (5.9.0), pyjokes (0.6.0), wikipedia (1.4.0), wolframalpha (5.0.0), pyautogui (0.9.53), requests (2.25.1), BeautifulSoup4 (4.11.1), xmltodict (0.13.0), and more-itertools (8.10.0).

```
rohit@rohit-vm:~/AI_Project$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: SpeechRecognition in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 1)) (3.8.1)
Requirement already satisfied: pyttsx3 in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 2)) (2.90)
Requirement already satisfied: psutil in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 3)) (5.9.0)
Requirement already satisfied: pyjokes in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 4)) (0.6.0)
Requirement already satisfied: wikipedia in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 5)) (1.4.0)
Requirement already satisfied: wolframalpha in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 6)) (5.0.0)
Requirement already satisfied: pyautogui in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 7)) (0.9.53)
Requirement already satisfied: requests<3.0.0,>=2.0.0 in /usr/lib/python3/dist-packages (from wikipedia->-r requirements.txt (line 5)) (2.25.1)
Requirement already satisfied: beautifulsoup4 in /home/rohit/.local/lib/python3.10/site-packages (from wikipedia->-r requirements.txt (line 5)) (4.11.1)
Requirement already satisfied: xmltodict in /home/rohit/.local/lib/python3.10/site-packages (from wolframalpha->-r requirements.txt (line 6)) (0.13.0)
Requirement already satisfied: more-itertools in /usr/lib/python3/dist-packages (from wolframalpha->-r requirements.txt (line 6)) (8.10.0)
```

Figure 2: test

7.3 Run the main file:

\$python3 main.py

When the main file is executed, the virtual assistant will greet the user and ask when he can do for the user. How the code works:

7.4 Project Structure:

VA - Contains the features and init file

features - Has all the code for various features of the VA

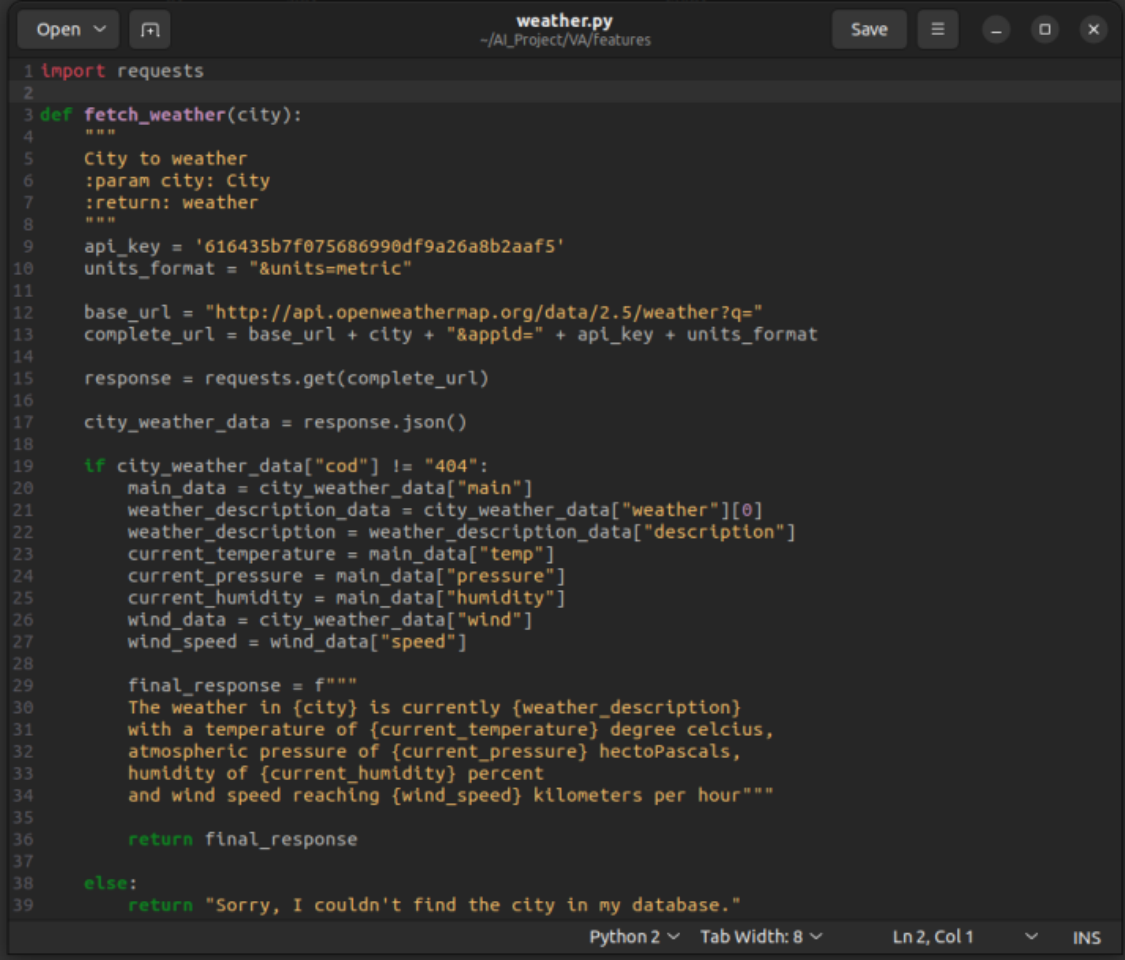
init.py - For providing packages

main.py - Main program file

readme.txt - readme file

requirements.txt - requirements file

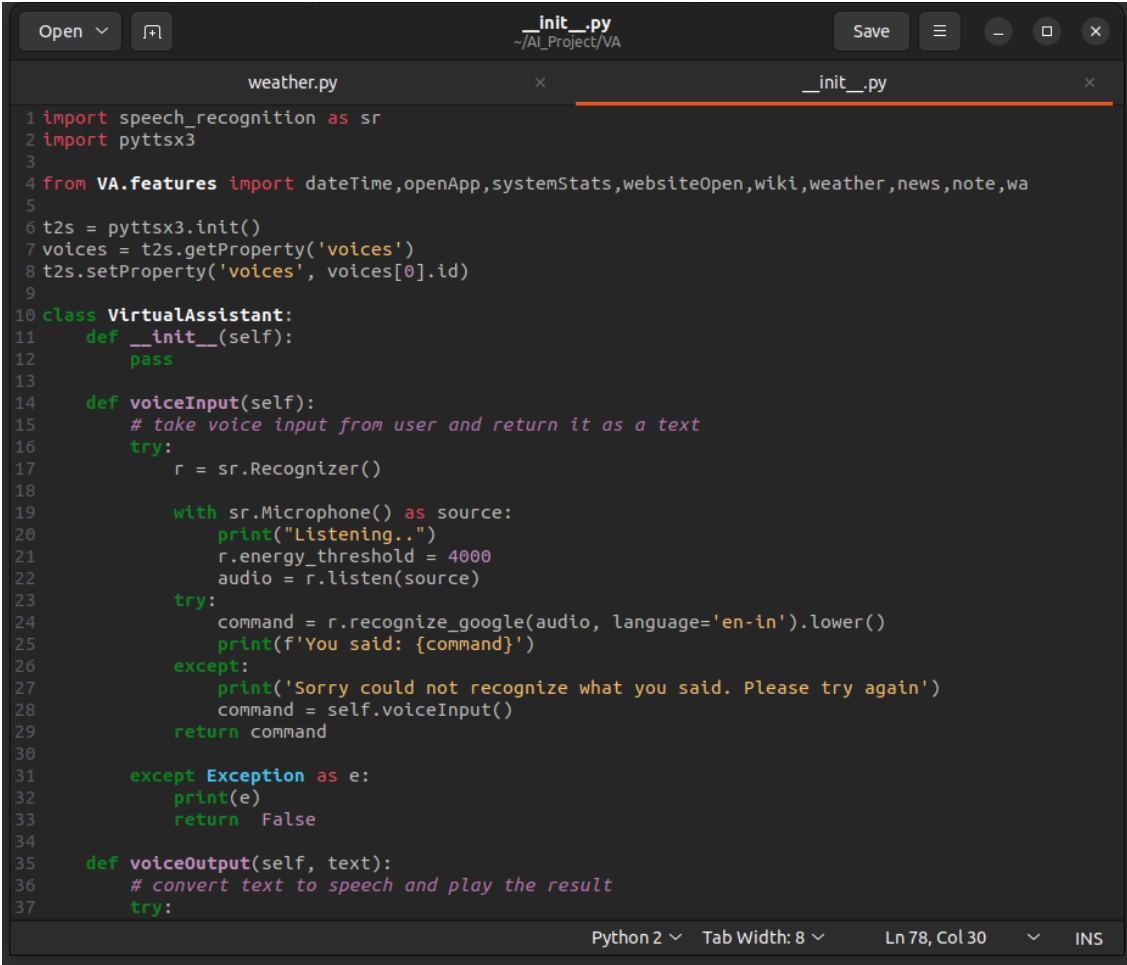
The virtual assistant can perform various operations that the user asks. These are called features of the assistant. For example: weather



```
1 import requests
2
3 def fetch_weather(city):
4     """
5     City to weather
6     :param city: City
7     :return: weather
8     """
9     api_key = '616435b7f075686990df9a26a8b2aaf5'
10    units_format = "&units=metric"
11
12    base_url = "http://api.openweathermap.org/data/2.5/weather?q="
13    complete_url = base_url + city + "&appid=" + api_key + units_format
14
15    response = requests.get(complete_url)
16
17    city_weather_data = response.json()
18
19    if city_weather_data["cod"] != "404":
20        main_data = city_weather_data["main"]
21        weather_description_data = city_weather_data["weather"][0]
22        weather_description = weather_description_data["description"]
23        current_temperature = main_data["temp"]
24        current_pressure = main_data["pressure"]
25        current_humidity = main_data["humidity"]
26        wind_data = city_weather_data["wind"]
27        wind_speed = wind_data["speed"]
28
29        final_response = f"""
30        The weather in {city} is currently {weather_description}
31        with a temperature of {current_temperature} degree celcius,
32        atmospheric pressure of {current_pressure} hectoPascals,
33        humidity of {current_humidity} percent
34        and wind speed reaching {wind_speed} kilometers per hour"""
35
36        return final_response
37
38    else:
39        return "Sorry, I couldn't find the city in my database."
```

Figure 3: test

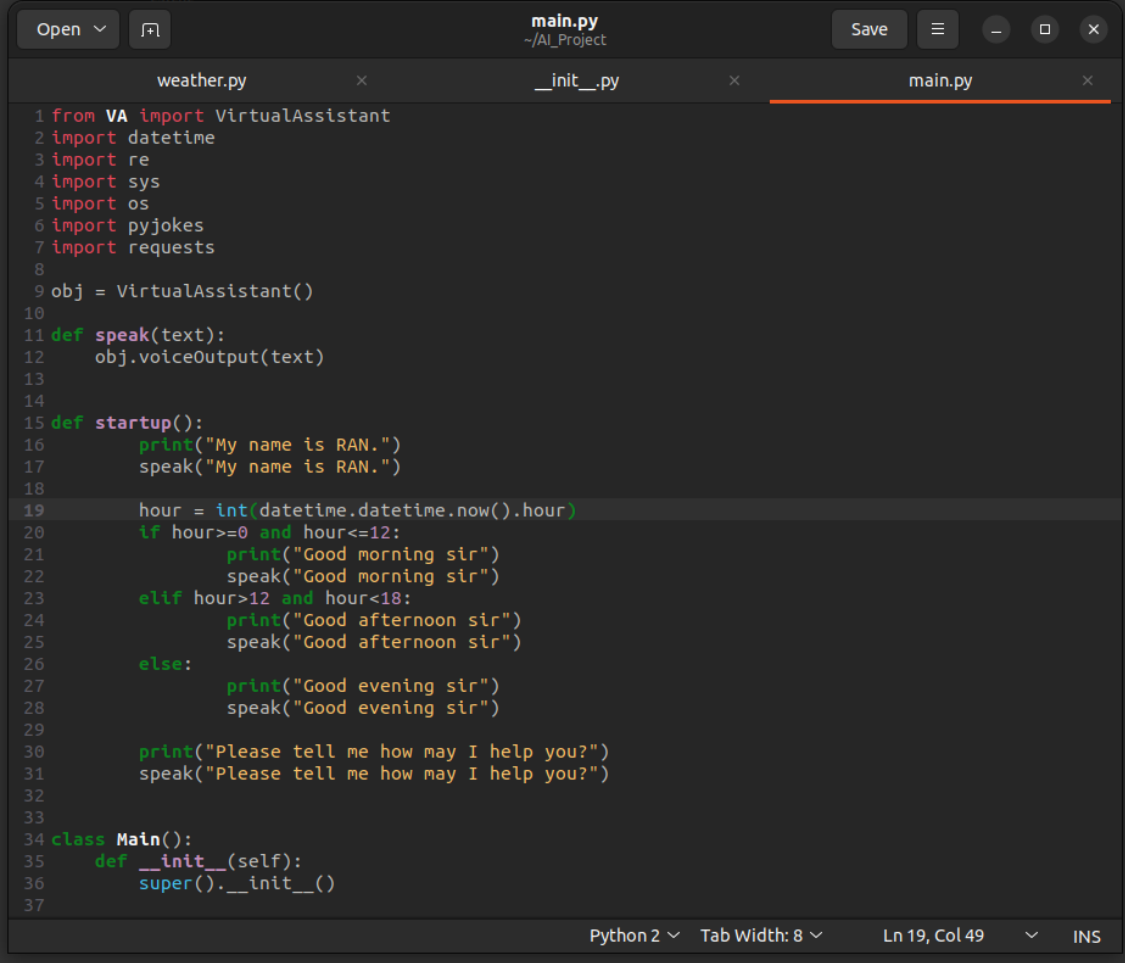
This is the weather.py file. Here, we've created a method where the program collects information about the weather and displays it as output.



```
1 import speech_recognition as sr
2 import pyttsx3
3
4 from VA.features import dateTime,openApp,systemStats,websiteOpen,wiki,weather,news,note,wa
5
6 t2s = pyttsx3.init()
7 voices = t2s.getProperty('voices')
8 t2s.setProperty('voices', voices[0].id)
9
10 class VirtualAssistant:
11     def __init__(self):
12         pass
13
14     def voiceInput(self):
15         # take voice input from user and return it as a text
16         try:
17             r = sr.Recognizer()
18
19             with sr.Microphone() as source:
20                 print("Listening..")
21                 r.energy_threshold = 4000
22                 audio = r.listen(source)
23
24             try:
25                 command = r.recognize_google(audio, language='en-in').lower()
26                 print(f'You said: {command}')
27             except:
28                 print('Sorry could not recognize what you said. Please try again')
29                 command = self.voiceInput()
30             return command
31
32     except Exception as e:
33         print(e)
34         return False
35
36     def voiceOutput(self, text):
37         # convert text to speech and play the result
38         try:
```

Figure 4: test

This is the `__init__.py` file. We use the init file to import packages into the main python file. In the init file, we've imported the speechRecognition and Python text to speech libraries. We've also created a method call to the weather file we've seen above. The reason we've done this is that we can call this method in the main file in another package.

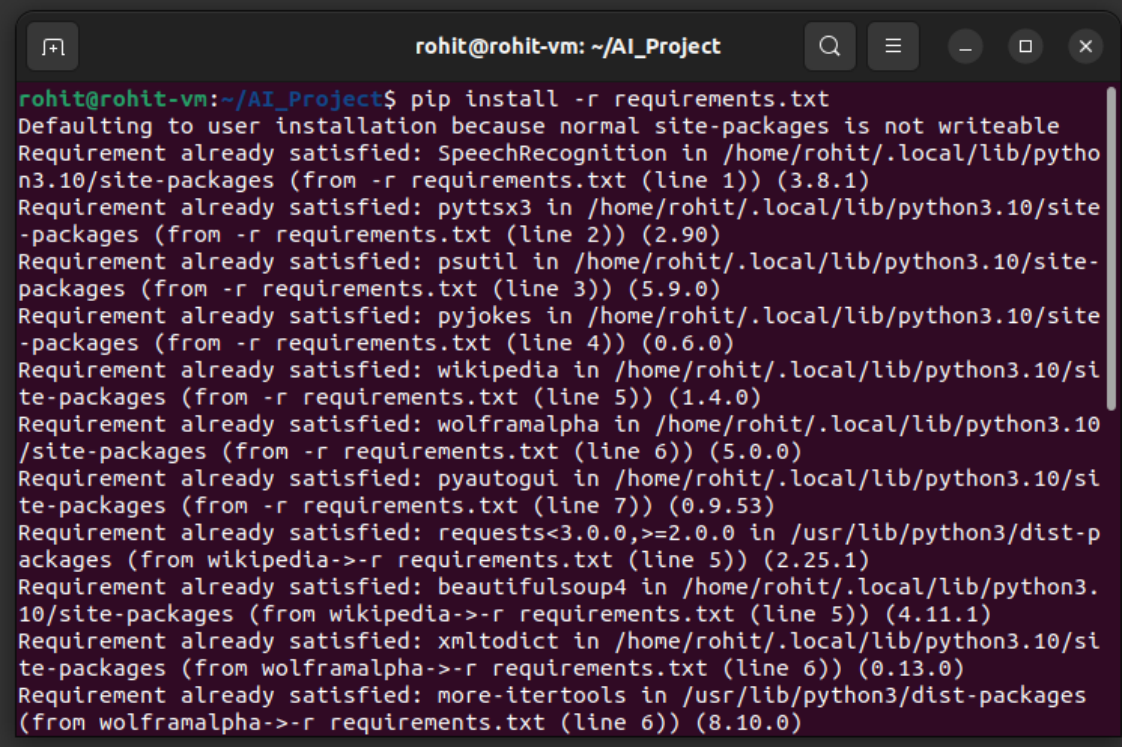


```
1 from VA import VirtualAssistant
2 import datetime
3 import re
4 import sys
5 import os
6 import pyjokes
7 import requests
8
9 obj = VirtualAssistant()
10
11 def speak(text):
12     obj.voiceOutput(text)
13
14
15 def startup():
16     print("My name is RAN.")
17     speak("My name is RAN.")
18
19     hour = int(datetime.datetime.now().hour)
20     if hour>=0 and hour<=12:
21         print("Good morning sir")
22         speak("Good morning sir")
23     elif hour>12 and hour<18:
24         print("Good afternoon sir")
25         speak("Good afternoon sir")
26     else:
27         print("Good evening sir")
28         speak("Good evening sir")
29
30     print("Please tell me how may I help you?")
31     speak("Please tell me how may I help you?")
32
33
34 class Main():
35     def __init__(self):
36         super().__init__()
37
```

Figure 5: test

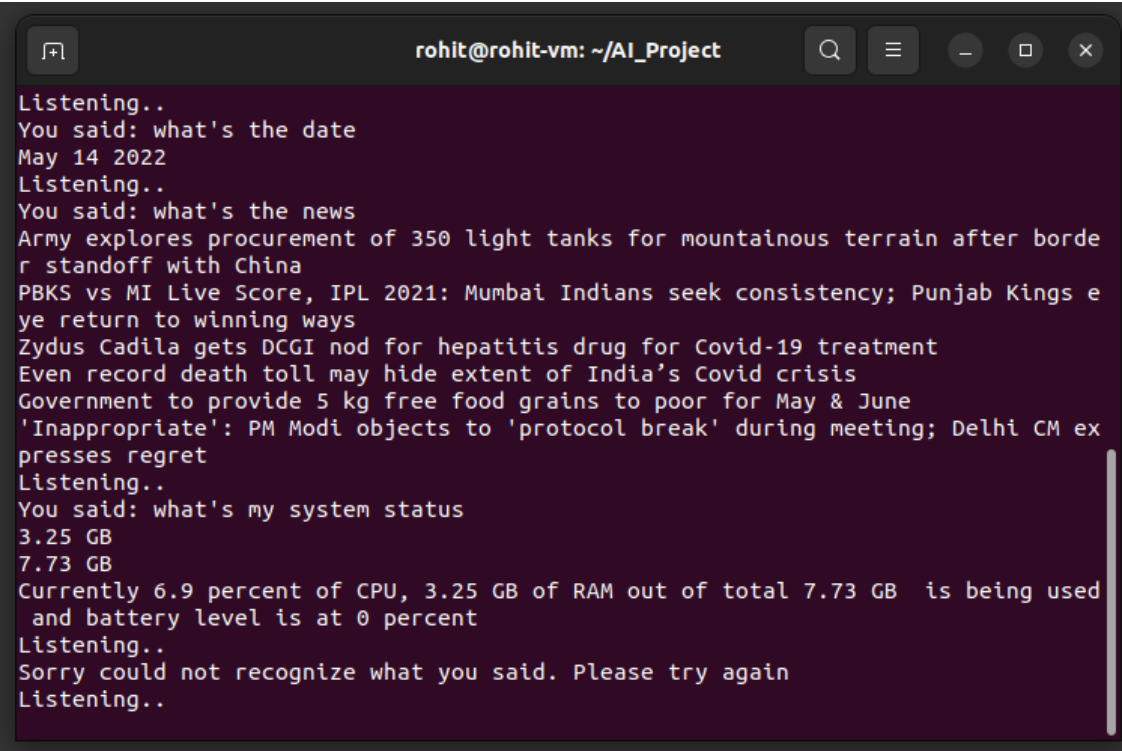
This is the main program file. Here we call the methods defined in the features directory for the appropriate user command. And the output is then provided in the form of speech.

8 Results



```
rohit@rohit-vm: ~/AI_Project
rohit@rohit-vm:~/AI_Project$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: SpeechRecognition in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 1)) (3.8.1)
Requirement already satisfied: pyttsx3 in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 2)) (2.90)
Requirement already satisfied: psutil in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 3)) (5.9.0)
Requirement already satisfied: pyjokes in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 4)) (0.6.0)
Requirement already satisfied: wikipedia in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 5)) (1.4.0)
Requirement already satisfied: wolframalpha in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 6)) (5.0.0)
Requirement already satisfied: pyautogui in /home/rohit/.local/lib/python3.10/site-packages (from -r requirements.txt (line 7)) (0.9.53)
Requirement already satisfied: requests<3.0.0,>=2.0.0 in /usr/lib/python3/dist-packages (from wikipedia->-r requirements.txt (line 5)) (2.25.1)
Requirement already satisfied: beautifulsoup4 in /home/rohit/.local/lib/python3.10/site-packages (from wikipedia->-r requirements.txt (line 5)) (4.11.1)
Requirement already satisfied: xmltodict in /home/rohit/.local/lib/python3.10/site-packages (from wolframalpha->-r requirements.txt (line 6)) (0.13.0)
Requirement already satisfied: more-itertools in /usr/lib/python3/dist-packages (from wolframalpha->-r requirements.txt (line 6)) (8.10.0)
```

Figure 6: test



```
rohit@rohit-vm: ~/AI_Project
Listening..
You said: what's the date
May 14 2022
Listening..
You said: what's the news
Army explores procurement of 350 light tanks for mountainous terrain after border standoff with China
PBKS vs MI Live Score, IPL 2021: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways
Zydus Cadila gets DCGI nod for hepatitis drug for Covid-19 treatment
Even record death toll may hide extent of India's Covid crisis
Government to provide 5 kg free food grains to poor for May & June
'Inappropriate': PM Modi objects to 'protocol break' during meeting; Delhi CM expresses regret
Listening..
You said: what's my system status
3.25 GB
7.73 GB
Currently 6.9 percent of CPU, 3.25 GB of RAM out of total 7.73 GB is being used and battery level is at 0 percent
Listening..
Sorry could not recognize what you said. Please try again
Listening..
```

Figure 7: test


```
rohit@rohit-vm: ~/AI_Project
ALSA lib pcm_oss.c:397:(snd_pcm_oss_open) Cannot open device /dev/dsp
ALSA lib confmisc.c:160:(snd_config_get_card) Invalid field card
ALSA lib pcm_usb_stream.c:482:(snd_pcm_usb_stream_open) Invalid card 'card'
ALSA lib confmisc.c:160:(snd_config_get_card) Invalid field card
ALSA lib pcm_usb_stream.c:482:(snd_pcm_usb_stream_open) Invalid card 'card'
ALSA lib pcm_dmix.c:1032:(snd_pcm_dmix_open) unable to open slave
Listening..
You said: what is 450 * 90 / 50
810
Listening..
You said: what's the weather in los angeles

The weather in angeles is currently clear sky
with a temperature of 18.68 degree celcius,
atmospheric pressure of 1017 hectoPascals,
humidity of 89 percent
and wind speed reaching 1.03 kilometers per hour
Listening..
You said: wikipedia differential equations
In mathematics, an equation is a formula that expresses the equality of two expressions, by connecting them with the equals sign =.
Listening..
You said: bye
rohit@rohit-vm:~/AI_Project$
```

Figure 8: test

```
rohit@rohit-vm: ~/AI_Project
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2664:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm_oss.c:397:(snd_pcm_oss_open) Cannot open device /dev/dsp
ALSA lib pcm_oss.c:397:(snd_pcm_oss_open) Cannot open device /dev/dsp
ALSA lib confmisc.c:160:(snd_config_get_card) Invalid field card
ALSA lib pcm_usb_stream.c:482:(snd_pcm_usb_stream_open) Invalid card 'card'
ALSA lib confmisc.c:160:(snd_config_get_card) Invalid field card
ALSA lib pcm_usb_stream.c:482:(snd_pcm_usb_stream_open) Invalid card 'card'
ALSA lib pcm_dmix.c:1032:(snd_pcm_dmix_open) unable to open slave
Listening..
You said: make a note
Listening..
You said: this is an artificial intelligence project
I've made a note of that
Listening..
You said: launch youtube.com
[Errno 2] No such file or directory: 'youtube.com'
True
Listening..
You said: bye
rohit@rohit-vm:~/AI_Project$
```

Figure 9: test

9 Conclusion

Voice-activated virtual assistants are an intriguing and valuable Artificial Intelligence concept. It is designed and developed to make the user's daily routine tasks much easier and hands-free, which saves time and is very convenient. As part of this project, which is a virtual assistant for Linux, we have implemented a number of features aimed at assisting a Linux-based user in his or her daily activities. We also learned how to use evaluation matrices to assess the performance of virtual assistants as well as user satisfaction. In the virtual assistant, we will include features such as telling the time and weather, setting reminders, performing calculations, and so on. Additionally, some features such as speech-to-text, text-to-speech, and so on. The assistant will execute these instructions and give output as the result of this execution and in speech.

10 Future Work

Our Virtual Assistant can perform various tasks that we have mentioned in the feature and skills sections, but our project still falls short of implementing security features such as Audio Recognition and OTP Verification, where we can perform Audio Recognition with previously saved authenticated user's audio files in Waveform (.WAV) Audio File Format. When the user records his audio for security purposes, his voice will be matched using the Resemblyzer and NumPy python libraries. The audio matching will be performed by converting the audio files into an array and then comparing them using the NumPy library. If the audio matches that of previously saved audio files, the virtual assistant grants the user access and greets him by calling his name. And If the user selects OTP verification for security check, the virtual assistant will send OTP to all authenticated users via email. The email IDs of all authenticated users will already be saved, and the SMTP server will be used to send emails. When the user enters the correct OTP, the virtual assistant grants the user access and greets him by calling his name. Our project also needs a suitable GUI, so we'll create one using the Tkinter Python library. The user interface may be created quickly and easily with this library. Since the user interface is the most effective approach for users to communicate with their virtual assistant and provides options for selecting the security check method and query response. The user can do a security check using any of the security check techniques.

References

- [1] Luigys toro," Jarvis: An Excellent Personal Assistant for Linux" on DesdeLinux
- [2] Luigys toro," Dragonfire: A virtual assistant for Ubuntu" on DesdeLinux
- [3] Senthil Kumar," Betty – The Siri-like Commandline Virtual Assistant For Linux" on OSTECHNIX on Published: August 17, 2017
- [4] William Elcock," Virtual Assistants for Linux You Can Use on Your Desktop," on June 28, 2020
- [5] Matthew Zajechowski," Automatic Speech Recognition (ASR) Software – An Introduction" at Usabilitygeek
- [6] G. Campagna, S. Xu, M. Moradshahi, R. Socher, and M. S. Lam, "Genie: A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands," PLDI '19, June 22's26, 2019, Phoenix, AZ, USA
- [7] Ayushi Trivedi, Navya Pant, Pinal Shah, Simran Sonik, and Supriya Agrawal," Speech to text and text to speech recognition systems-A review," IOSR Journal of Computer Engineering (IOSR-JCE) Mar.- Apr. 2018
- [8] Subhash S, Prajwal N Srivatsa, Siddesh S, Ullas A, Santhosh B," Artificial Intelligence-based Voice Assistant" IEEE Xplore 2020
- [9] Chanjun Park, Yeongwook Yang, Kinam Park, Heuseok Lim,"Decoding Strategies for Improving Low-Resource Machine Translation", in Electronics on 24 September 2020
- [10] Akash, Kaveri Mojad, Vaishnavee Dhage, and Yashika Saxena, "ALLY: An Intelligent Virtual Assistant using AI", IRJET, 2021 <https://www.irjet.net/archives/V8/i5/IRJET-V8I5856.pdf>