Alex - the AI Assistant

ABSTRACT

Python is a brand-new language, thus you may simply script the Voice wizard in Python as you are aware. Wizard commands may be handled in accordance with user needs. The process of turning voice into text is called speech recognition. Alexa, Siri, and other voice assistants often utilize this. Python offers a voice recognition API that can translate spoken words into text. It was fun having you as my helper. With just one voice command, you may launch your preferred IDE, search Google without using a browser, send emails without typing a word, and do a number of other commonplace operations. Tasks are simpler to do than ever before. In the present situation, technological advancements have made it possible for us to execute any work as well or better than we do. I came to see that his AI technology might minimize human labour and save time across the board after starting this job.

These are this project's characteristics:
1. You can email someone.
2. Use WhatsApp to send SMS.
3. You may launch Notepad, your preferred IDE, Command Prompt, etc.
4 You can search Wikipedia.
5. You may use your web browser to access websites like Google, YouTube, and others.
6. A weather prediction is available.

Here, a basic question is raised. How does AI work? The virtual assistant I created is not an artificial intelligence; rather, it resembles the result of a number of instructions. But at its core, AI devices are designed to execute human activities as effectively as or perhaps more effectively than people. Although my virtual assistant is not a very excellent example of AI, it does exist.

1. Introduction

Machines' capacity to think like humans is shown by the deployment of artificial intelligence in such machines. Usually, human involvement is required for computer systems. Since Python is a relatively new language, you can easily write voice assistants in it. The user's requirements may determine how the wizard's instructions are handled. Alexa, Siri, etc. all use voice recognition. To translate voice to text, Python includes an API called Speech Recognition. Making my own helper was a fun project. Numerous more routine operations, include launching his preferred IDE with a single voice command, searching Google without opening a browser, sending emails without typing a word, etc. Now, doing it is simple. In the present situation, technological advancements enable us to accomplish any activity with an efficacy equal to or higher than our own. Through working on this project, I came to see how the idea of AI can decrease human effort and save time across the board.

The voice assistant employs artificial intelligence, therefore the outcomes are very precise and effective. Task completion with the aid of an assistant takes less time and effort than without it. He completely abandoned the idea of typing, pretended to be someone else we were speaking to, and gave us a mission to do. The performance of an assistant is superior to that of a human assistant in all tasks. It might be stated to be more productive and successful. The libraries and packages that were utilized to build this wizard are time-saving and complexity-focused.

Its capabilities include the ability to write emails, send SMS over WhatsApp, open command prompts, your preferred IDEs, notepads, etc., do Wikipedia searches, and access websites like Google and YouTube. The weather prediction and reminders for certain workstations may both be shown via a web browser. able to carry on some simple talks.

The Visual Studio Code IDE is the technology and tool used to construct this project, and I made all of the .py files in Visual Studio Code. Also included in my project were the following modules and libraries: pyttsx3, speech recognition, datetime, wikipedia, smtplib, pywhatkit, pyjokes, pyautogui, etc.

1.1 Present System

Many voice assistants now in use, including Alexa, Siri, Google Assistant, and Cortana, employ speech processing and speech recognition techniques. It responds to human orders as necessary and carries out its specialized tasks in a highly effective and efficient manner.

As a consequence of using artificial intelligence, these voice assistants provide very precise and effective solutions. These wizards facilitate job completion while minimizing human effort and time. They no longer require us to type, pose as someone else while we are speaking to them, or do any other activity. may be argued to be more successful and effective. The methods utilized to build these wizards concentrate on reducing time complexity.

These assistants can only be used with an online connection and need an account (such as a Google account for Google Assistant, a Microsoft account for Cortana, etc.). They also need an internet connection to function. increase. Many gadgets, including speakers, computers, and phones, have them built in.

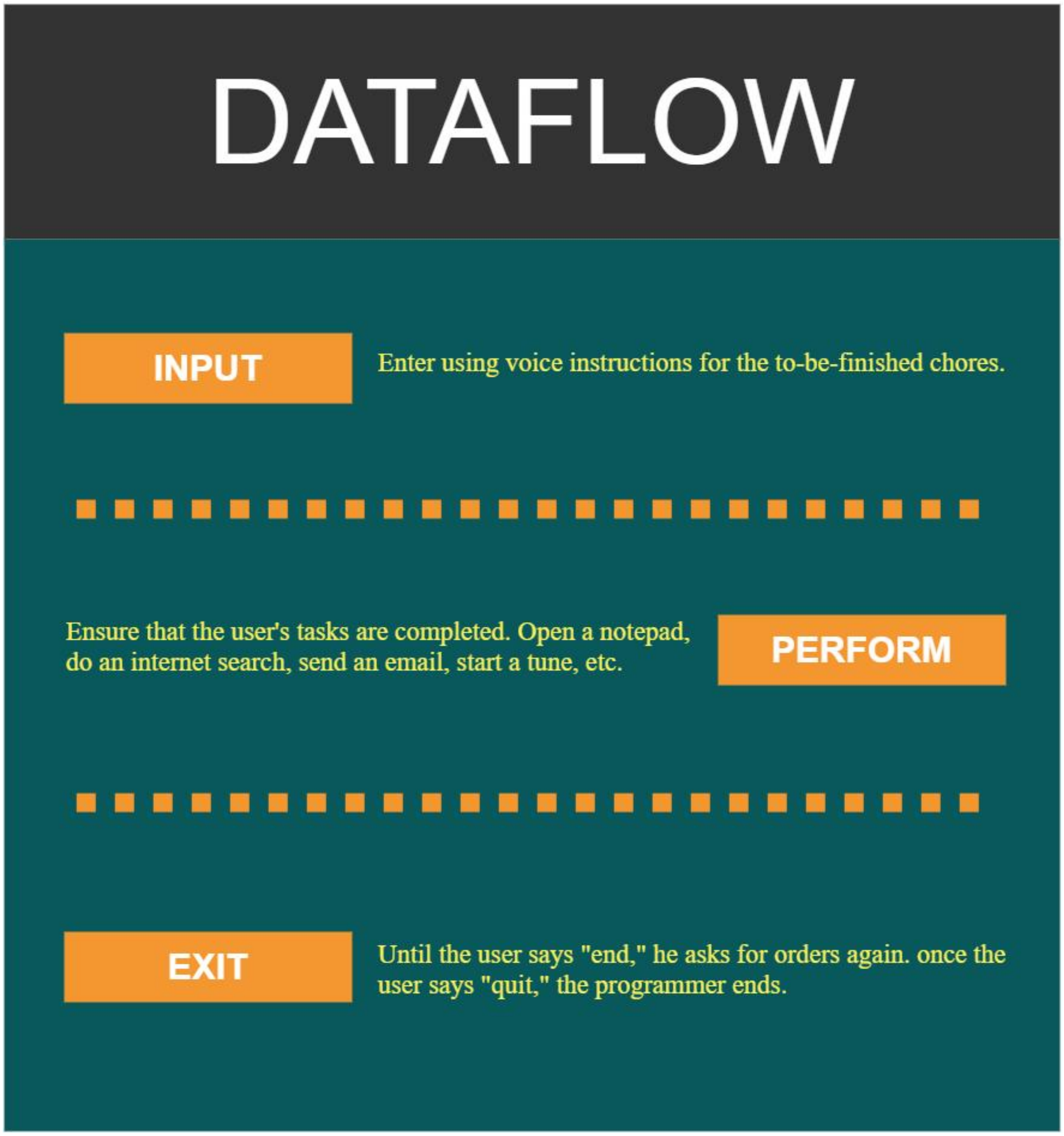
1.2 Proposed System

Making my own helper was a fun project. Numerous more routine operations, include launching his preferred IDE with a single voice command, searching Google without opening a browser, sending emails without typing a word, etc. Now, doing it is simple. ALEX, in contrast to other conventional voice assistants, is built into the desktop and does not need users to register an account in order to use it. It uses an internet connection to send commands to users to carry out certain activities. I don't need it.

This project makes use of Visual Studio Code as its IDE. Visual Studio Code was used to create all Python files, and this IDE made it simple to install all necessary packages. In this project, the following modules and libraries were utilized: As an example, consider the following: pyttsx3, voice recognition, date, wikipedia, smtplib, pywhatkit, pyjokes, pyautogui, etc.

With the use of advancements, ALEX may now do any work as effectively as us or perhaps more so. Through working on this project, I came to see how the idea of AI can decrease human effort and save time across the board. This project includes features like the ability to send emails, WhatsApp messaging, open command prompts, your preferred IDEs, notepads, etc., do Wikipedia and Google searches, and more. You may use websites like YouTube, check the weather on your browser, and read reminders on your preferred desktop. able to carry on some simple talks.

2. Data Flow



DATAFLOW

INPUT

Enter using voice instructions for the to-be-finished chores.

Ensure that the user's tasks are completed. Open a notepad, do an internet search, send an email, start a tune, etc.

PERFORM

EXIT

Until the user says "end," he asks for orders again. once the user says "quit," the programmer ends.

Figure 1: Data Flow for ALEX

The system was created using the relevant Python tools and the artificial intelligence paradigm. To complete your assignment, Python offers a wide variety of libraries and packages. To get the date and time, for instance, use datetime library. The Software Details part of this report contains information on these packages.

Only user input is used as data for this project. Whatever the user says, the wizard will carry out the job as instructed. User input consists of a list of possible tasks written in human language rather than any actual data. H. English, I demand execution.

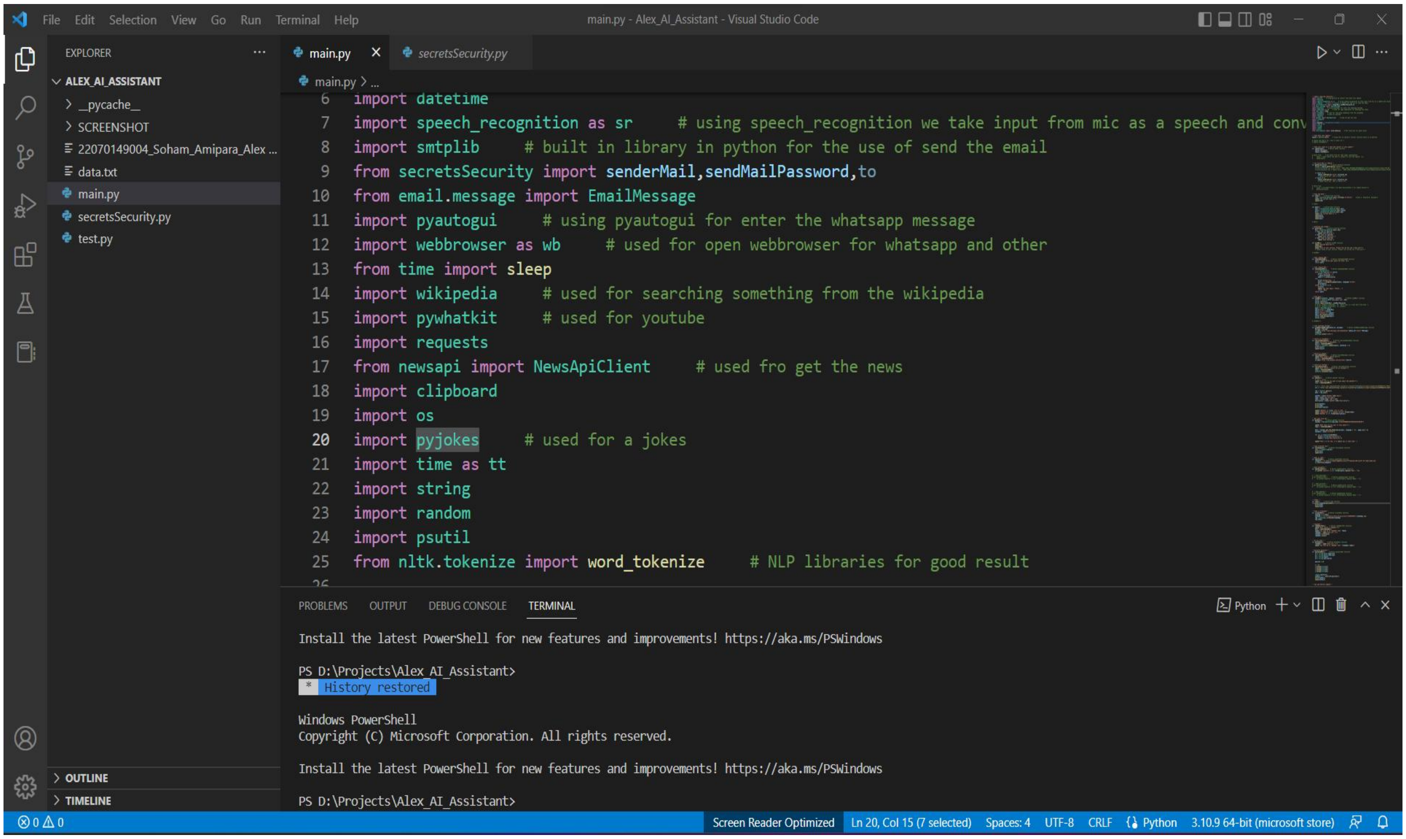
3. Software Details

This project makes use of Visual Studio Code as its IDE. Visual Studio Code was used to create all Python files, and this IDE made it simple to install all necessary packages. B. pyttsx3, voice recognition, datetime, wikipedia, smtplib, pywhatkit, pyjokes, pyautogui, etc. were among the modules and libraries used in this project.

3.1 Visual Studio Code

Microsoft created the source code editor Visual Studio Code, sometimes referred to as VS Code, for use on Windows, Linux, and macOS. Among the features are debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and built-in Git compatibility.

For several programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust, Fortran, and others, Visual Studio Code serves as a source code editor. Its foundation is the Electron framework, which is used to create Node.js web apps that employ the Blink layout engine. The editor module used by Azure DevOps in Visual Studio Code is known as "Monaco" (formerly known as Visual Studio Online and Visual Studio Team Services).



```python
import datetime
import speech_recognition as sr     # using speech_recognition we take input from mic as a speech and conv
import smtplib     # built in library in python for the use of send the email
from secretsSecurity import senderMail,sendMailPassword,to
from email.message import EmailMessage
import pyautogui     # using pyautogui for enter the whatsapp message
import webbrowser as wb     # used for open webbrowser for whatsapp and other
from time import sleep
import wikipedia     # used for searching something from the wikipedia
import pywhatkit     # used for youtube
import requests
from newsapi import NewsApiClient     # used fro get the news
import clipboard
import os
import pyjokes     # used for a jokes
import time as tt
import string
import random
import psutil
from nltk.tokenize import word_tokenize     # NLP libraries for good result
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Projects\Alex_AI_Assistant>
 *  History restored

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Projects\Alex_AI_Assistant>
```

Figure 2: Visual Studio Code

3.2 Python Libraries

ALEX made use of the following Python libraries:

3.2.1 pyttsx3: This text-to-speech Python library does the conversion.

3.2.2 SpeechRecognition: A Python script for text-to-speech conversion.

3.2.3 pywhatkit : Is a Python package that has extra functionality for delivering WhatsApp messages at certain times.

3.2.4 Datetime: The current date and time are available from this library.

3.2.5 Wikipedia: Wikipedia search functionality in Python.

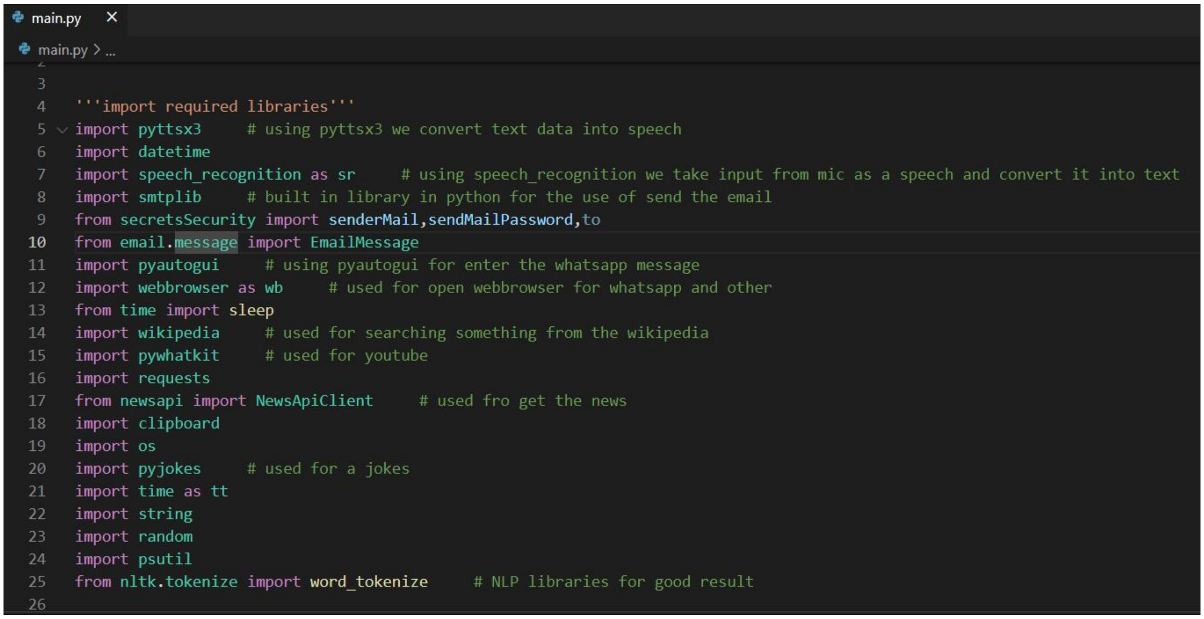
3.2.6 Smtplib: A straightforward email transfer protocol that enables the sending and receiving of emails across email servers.

3.2.7 Pyjokes: A Python package that contains a tonne of funny jokes.

3.2.8 Webbrowser: Gives consumers a user interface for reading web-based content.

3.2.9 Pyautogui: Is a graphical user interface Python library.

3.2.10 os: Represents characteristics of the operating system.



```python

'''import required libraries'''
import pyttsx3     # using pyttsx3 we convert text data into speech
import datetime
import speech_recognition as sr     # using speech_recognition we take input from mic as a speech and convert it into text
import smtplib     # built in library in python for the use of send the email
from secretsSecurity import senderMail,sendMailPassword,to
from email.message import EmailMessage
import pyautogui     # using pyautogui for enter the whatsapp message
import webbrowser as wb     # used for open webbrowser for whatsapp and other
from time import sleep
import wikipedia     # used for searching something from the wikipedia
import pywhatkit     # used for youtube
import requests
from newsapi import NewsApiClient     # used fro get the news
import clipboard
import os
import pyjokes     # used for a jokes
import time as tt
import string
import random
import psutil
from nltk.tokenize import word_tokenize     # NLP libraries for good result

```

Figure 3: Python Libraries

4. Implementation Work Details

ALEX, is a voice assistant that enables you to launch your preferred IDE or carry out other common desktop operations with with a single voice command. ALEX, in contrast to other conventional voice assistants, is built into the desktop and does not need users to register an account in order to use it. It uses an internet connection to send commands to users to carry out certain activities. I don't need it.

4.1 Real Life Application

4.1.1 Save Time:

With the help of the desktop voice assistant ALEX, you can carry out speech searches, manage voice-activated devices, and carry out a variety of other activities.

4.1.2 Conversational interaction:

Conversational interactions employ the necessary Python modules or libraries automatically to finish jobs quickly. As a result, each user who assigns a job has the impression that the assignment is being given to a human assistant who will engage in dialogue with the user in order to gather input and provide the necessary output a task that has been accomplished.

4.1.3 Reactive nature:

The Desktop Assistant responds. In other words, it reacts appropriately and comprehends human language and the user-provided context. H. In English, a language that people can comprehend. Users therefore arrive at their decisions in an educated and wise manner.

4.1.4 Multitasking:

Multitasking could be its main use. You may ask for more instructions one at a time till the user turns it off.

4.1.5 No Trigger phase:

without a start step, only asks for the command to be executed after hearing the user's answer.

4.2 Functions

4.1.1 takeCommand():

Using the user's microphone, this function accepts instructions and outputs them as strings.

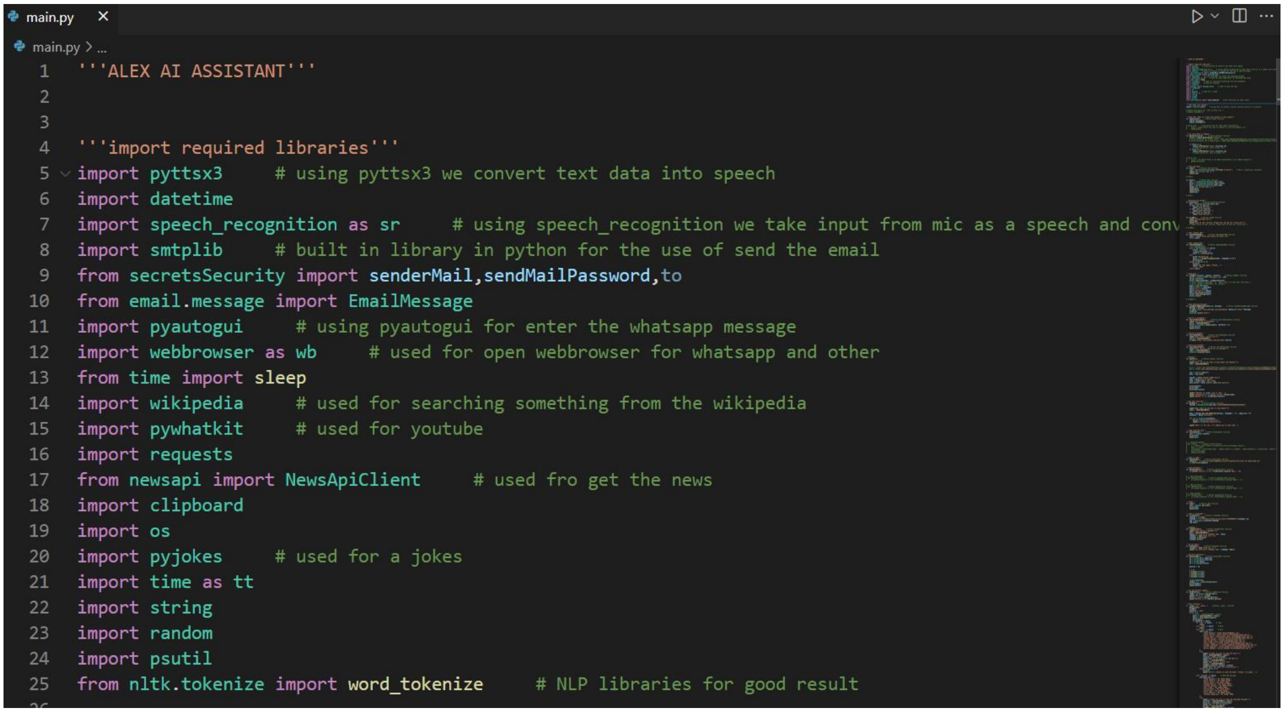
4.1.2 wishMe():

According on the time of day, this feature welcomes visitors with phrases like "Good Morning," "Good Afternoon," and "Good Evening."

4.1.3 __main__():

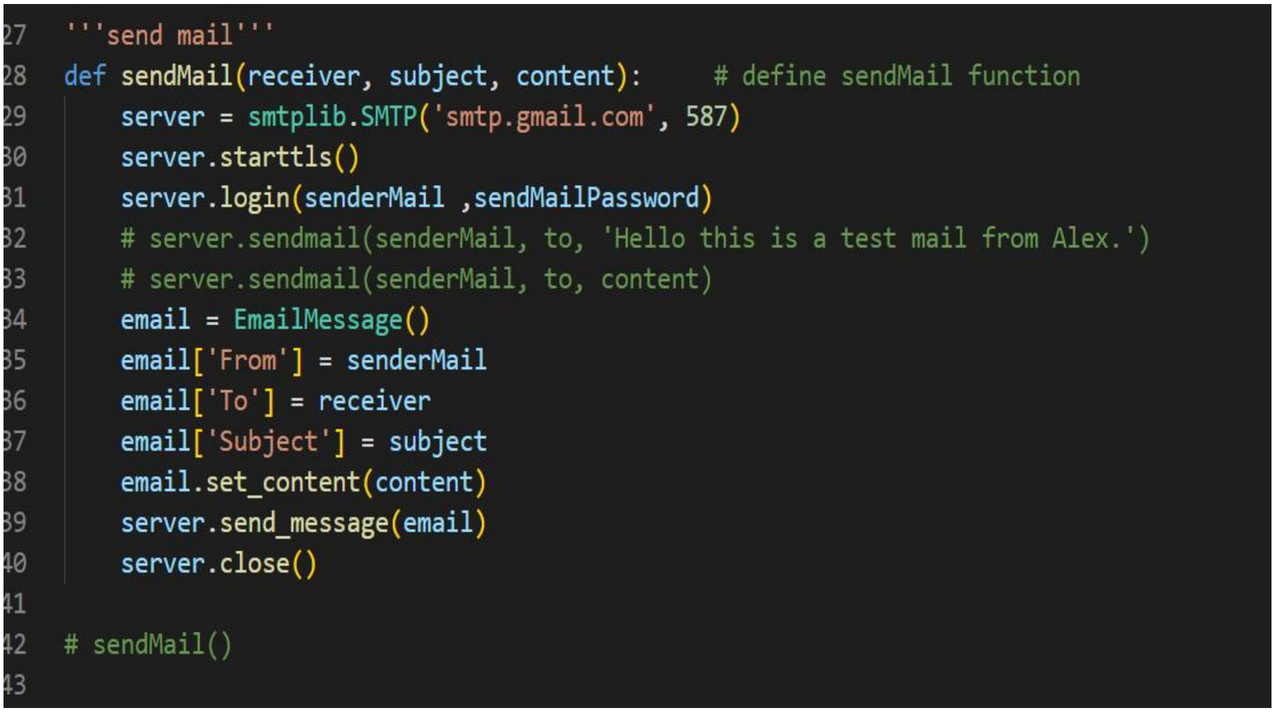
For circumstances like "Open Google", "Open Notepad", "Search Wikipedia", and "Open Command. prompt", etc., this method provides all essential task definitions, such as sendEmail() and news().

5. Input/Output Screenshot



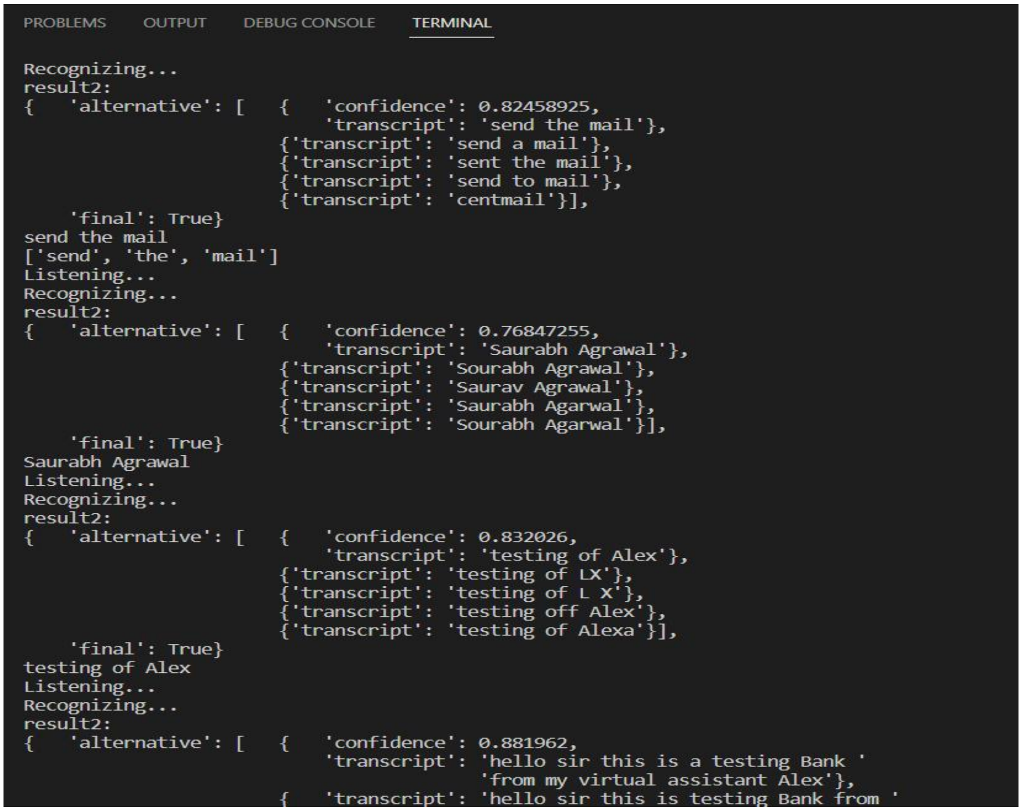
```python
'''ALEX AI ASSISTANT'''


'''import required libraries'''
import pyttsx3     # using pyttsx3 we convert text data into speech
import datetime
import speech_recognition as sr     # using speech_recognition we take input from mic as a speech and conv
import smtplib     # built in library in python for the use of send the email
from secretsSecurity import senderMail,sendMailPassword,to
from email.message import EmailMessage
import pyautogui     # using pyautogui for enter the whatsapp message
import webbrowser as wb     # used for open webbrowser for whatsapp and other
from time import sleep
import wikipedia     # used for searching something from the wikipedia
import pywhatkit     # used for youtube
import requests
from newsapi import NewsApiClient     # used fro get the news
import clipboard
import os
import pyjokes     # used for a jokes
import time as tt
import string
import random
import psutil
from nltk.tokenize import word_tokenize     # NLP libraries for good result
```

Figure 4: Import Required Libraries



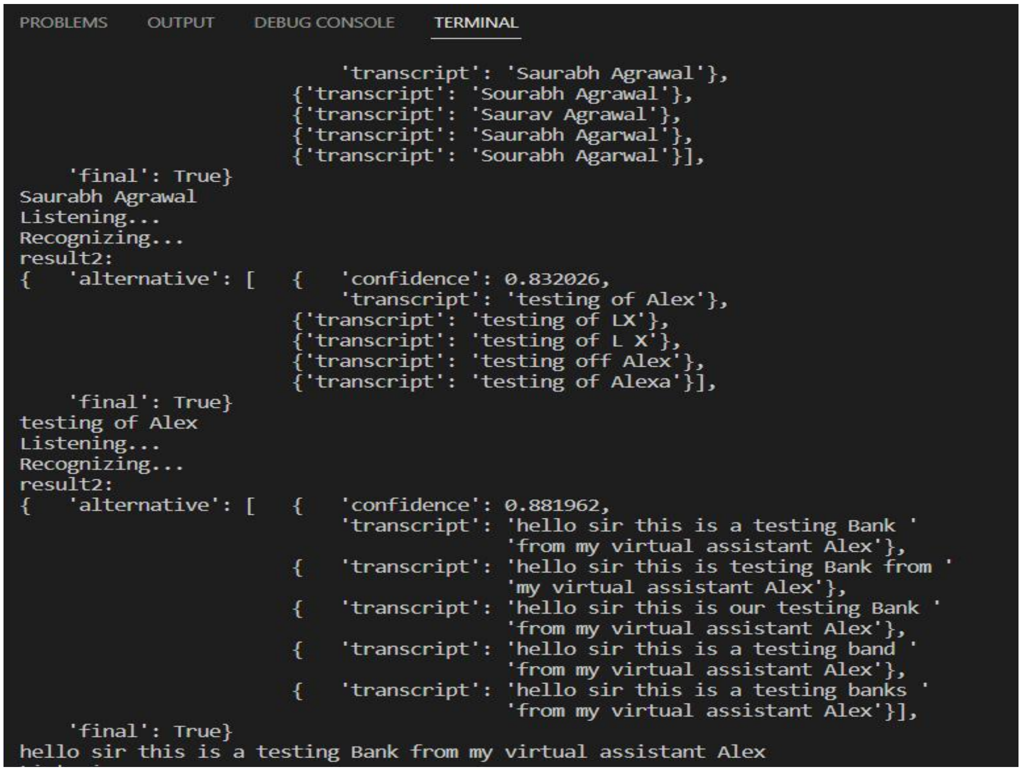
```python
'''send mail'''
def sendMail(receiver, subject, content):     # define sendMail function
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(senderMail ,sendMailPassword)
    # server.sendmail(senderMail, to, 'Hello this is a test mail from Alex.')
    # server.sendmail(senderMail, to, content)
    email = EmailMessage()
    email['From'] = senderMail
    email['To'] = receiver
    email['Subject'] = subject
    email.set_content(content)
    server.send_message(email)
    server.close()

# sendMail()

```

Figure 5: Send Mail Function



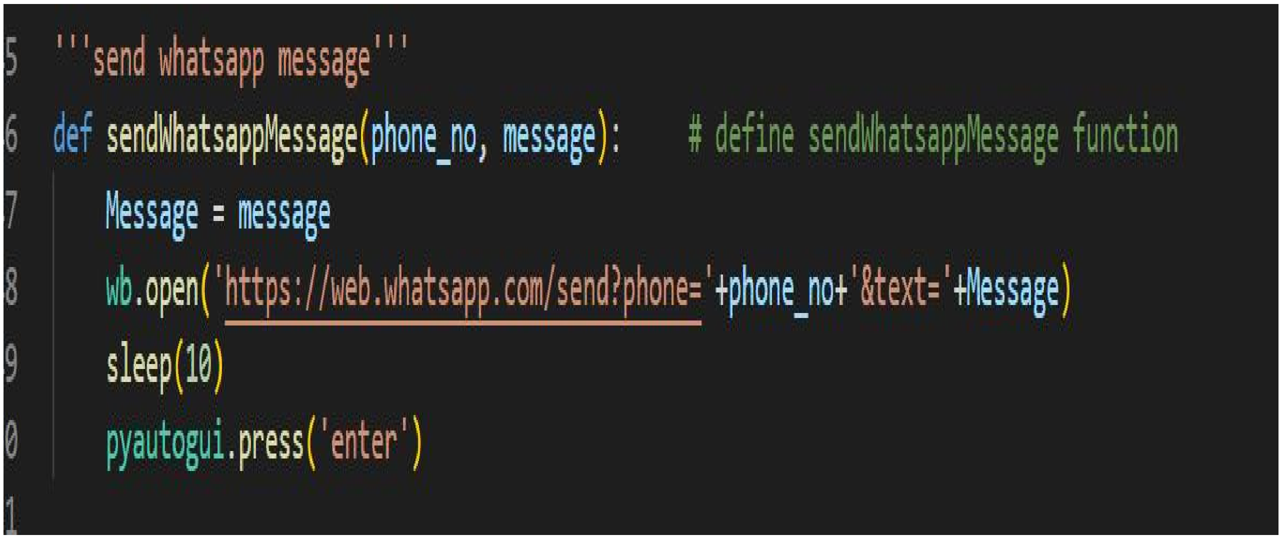
```
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.82458925,
                           'transcript': 'send the mail'},
                       {'transcript': 'send a mail'},
                       {'transcript': 'sent the mail'},
                       {'transcript': 'send to mail'},
                       {'transcript': 'centmail'}],
    'final': True}
send the mail
['send', 'the', 'mail']
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.76847255,
                           'transcript': 'Saurabh Agrawal'},
                       {'transcript': 'Sourabh Agrawal'},
                       {'transcript': 'Saurav Agrawal'},
                       {'transcript': 'Saurabh Agarwal'},
                       {'transcript': 'Sourabh Agarwal'}],
    'final': True}
Saurabh Agrawal
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.832026,
                           'transcript': 'testing of Alex'},
                       {'transcript': 'testing of LX'},
                       {'transcript': 'testing of L X'},
                       {'transcript': 'testing off Alex'},
                       {'transcript': 'testing of Alexa'}],
    'final': True}
testing of Alex
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.881962,
                           'transcript': 'hello sir this is a testing Bank '
                                         'from my virtual assistant Alex'},
                       {   'transcript': 'hello sir this is testing Bank from '
```

Figure 6: Output of send mail using Alex AI Assistant



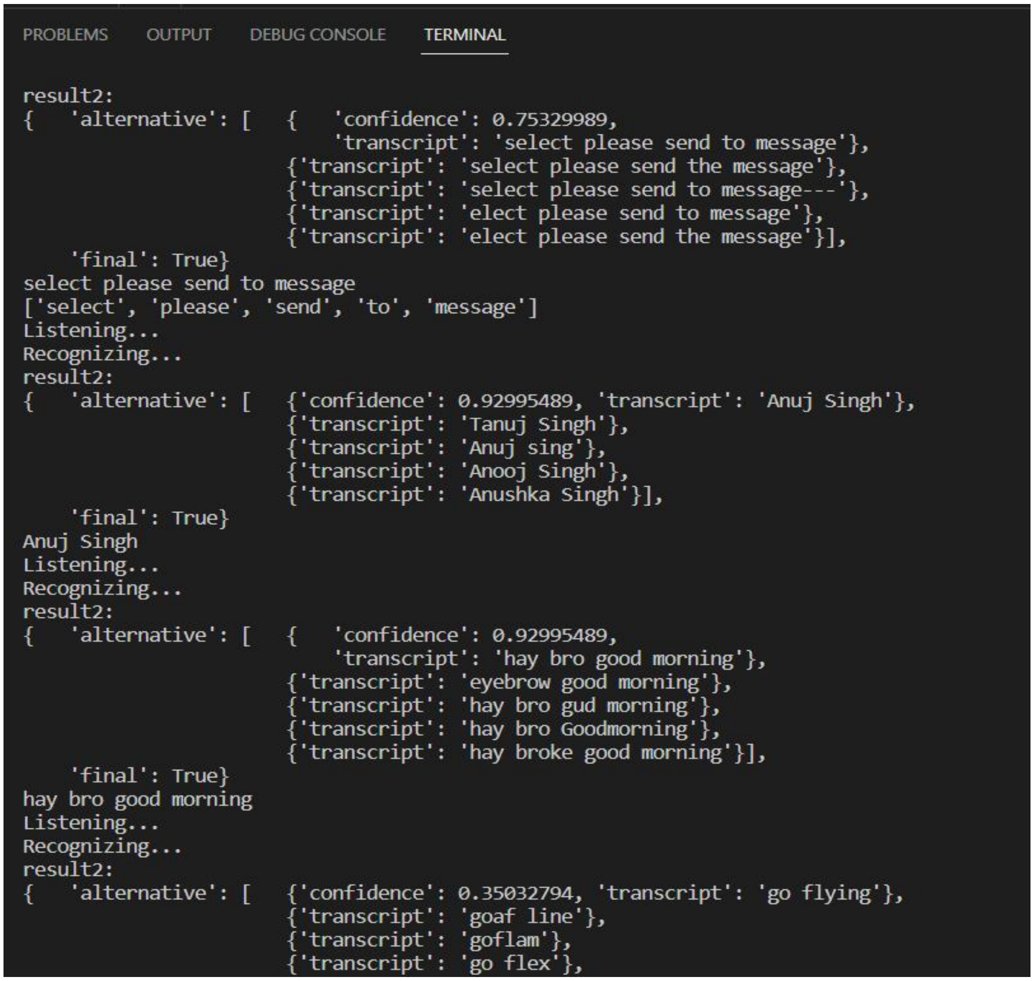
```
                           'transcript': 'Saurabh Agrawal'},
                       {'transcript': 'Sourabh Agrawal'},
                       {'transcript': 'Saurav Agrawal'},
                       {'transcript': 'Saurabh Agarwal'},
                       {'transcript': 'Sourabh Agarwal'}],
    'final': True}
Saurabh Agrawal
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.832026,
                           'transcript': 'testing of Alex'},
                       {'transcript': 'testing of LX'},
                       {'transcript': 'testing of L X'},
                       {'transcript': 'testing off Alex'},
                       {'transcript': 'testing of Alexa'}],
    'final': True}
testing of Alex
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.881962,
                           'transcript': 'hello sir this is a testing Bank '
                                         'from my virtual assistant Alex'},
                       {   'transcript': 'hello sir this is testing Bank from '
                                         'my virtual assistant Alex'},
                       {   'transcript': 'hello sir this is our testing Bank '
                                         'from my virtual assistant Alex'},
                       {   'transcript': 'hello sir this is a testing band '
                                         'from my virtual assistant Alex'},
                       {   'transcript': 'hello sir this is a testing banks '
                                         'from my virtual assistant Alex'}],
    'final': True}
hello sir this is a testing Bank from my virtual assistant Alex
```

Figure 7: The results of sending message using Alex's AI Assistant



```
'''send whatsapp message'''
def sendWhatsappMessage(phone_no, message):     # define sendWhatsappMessage function
    Message = message
    wb.open('https://web.whatsapp.com/send?phone='+phone_no+'&text='+Message)
    sleep(10)
    pyautogui.press('enter')
```

Figure 8: Send WhatsApp Message Function



```
result2:
{   'alternative': [   {   'confidence': 0.75329989,
                           'transcript': 'select please send to message'},
                       {'transcript': 'select please send the message'},
                       {'transcript': 'select please send to message---'},
                       {'transcript': 'elect please send to message'},
                       {'transcript': 'elect please send the message'}],
    'final': True}
select please send to message
['select', 'please', 'send', 'to', 'message']
Listening...
Recognizing...
result2:
{   'alternative': [   {'confidence': 0.92995489, 'transcript': 'Anuj Singh'},
                       {'transcript': 'Tanuj Singh'},
                       {'transcript': 'Anuj sing'},
                       {'transcript': 'Anooj Singh'},
                       {'transcript': 'Anushka Singh'}],
    'final': True}
Anuj Singh
Listening...
Recognizing...
result2:
{   'alternative': [   {   'confidence': 0.92995489,
                           'transcript': 'hay bro good morning'},
                       {'transcript': 'eyebrow good morning'},
                       {'transcript': 'hay bro gud morning'},
                       {'transcript': 'hay bro Goodmorning'},
                       {'transcript': 'hay broke good morning'}],
    'final': True}
hay bro good morning
Listening...
Recognizing...
result2:
{   'alternative': [   {'confidence': 0.35032794, 'transcript': 'go flying'},
                       {'transcript': 'goaf line'},
                       {'transcript': 'goflam'},
                       {'transcript': 'go flex'},
```

Figure 9: Output of send WhatsApp message using Alex AI Assistant

6. System Testing

To ascertain if the requirements are satisfied, system testing is done on a fully integrated system. System checks conducted by ALEX Desktop Assistant concentrate on the following four factors:

6.1 Functionality

Additionally, it tests the system's functioning by seeing whether it completes the job it was designed to do. The system has successfully passed that specific functional test if you have tested and used each function to ensure that it can do the required task. For instance, once the user instructed ALEX to "Open Google," ALEX inquired as to what to look for on the search engine. When the user asked, "What is Python?" ALEX accessed Google and looked for the information needed.

6.2 Usability

The ease of the programme and how user-friendly it is for the user to use, as well as how it responds to each question that the user asks, are used to assess a system's usability.

Any work may be completed more quickly and easily since Python's core modules and libraries are used automatically and in a conversational manner. Because of the verbal contact required to provide input and get the intended output a job completed any user who gives it a task feels as if they are delivering it to a human helper.

The desktop assistant is reactive, meaning it understands human language extremely well, as well as the context that the user provides, and responds in a manner that is human intelligible, such as English. As a result, the user reacts in a wise and educated manner.

Multitasking may be the primary usage. Until the user disables them, consecutive commands may be requested one after the other. It doesn't need an initiation phase; instead, it asks for a command, listens to the user's answer, and then does the requested action.

6.3 Security

The primary emphasis of security testing is on risks and vulnerabilities. Data leaking is not a concern while using ALEX since it is a local desktop programme. When a user signs in, the system-specific software is enabled.

6.4 Stability

A system is considered to be stable if its outputs are confined and specific to its restricted inputs. A system's stability relies on its outputs. If a system functions at both poles of its function, it is said to be stable.

7. Conclusion

As a voice assistant that saves consumers time via interactive engagement, efficacy, and efficiency, ALEX is unquestionably extremely helpful. But while working on this project, we found significant drawbacks and areas for future development, which are detailed below:

7.1 Limitations

- There is a security issue somewhere. In this project, spoken orders are not encrypted.
- Background noise can cause issues.
- Accents may lead to misunderstandings and inaccuracies in communication.
- ALEX cannot be summoned at any moment from outside, unlike other conventional assistants that may be accessed by uttering "OK Google!" or the Google Assistant.

7.2 Future Scope

- Allow ALEX to grow in its knowledge and acquire new talents on its own.
- Applications for ALEX may also be created on Android.
- ALEX Produce additional voice terminals.
- For security, voice instructions may be encrypted.