Project Title :

Voice Assistant

(By SEMESTER – VII of IV Year M.Sc.(CA & IT) 2023-24)

Submitted By :

Roll No	Name
4054	Prajapati Jaykumar Vinodbhai
4147	Prajapati Jaimin Kalpeshbhai
4174	Patel Yash Mukeshbhai

Date of Submission : 27 December, 2023

Submitted To :

K. S. School Of Business Management

M.Sc. – Computer Application and Information Technology

Table of Contents

Abstract

A Voice Assistant is one of the hot topics in the current world that are programs that listens to human’s verbal command and respond to them which makes it a human-computer/device interaction. In the current days, a voice assistant is everywhere which is a lot useful in these busy days. Nowadays, almost everyone in the current world is using voice assistant because it is everywhere starting from Google smartphone assistant which even 5 years old kids will know how to use because of the current world pandemic which makes them use smartphones till Amazon's Alexa which will be very useful to do works starting from entertaining the users till turning on and off the household products (Internet of Things). One of the greatest features is that it will be very useful to even physically challenged people, for example, people who are not able to walk use the Internet of Things (IoT) feature to operate the household products and maintain them. So, we tend to develop a voice assistant which will be very useful to the users same as the other voice assistants which are currently in the world.

CHAPTER – 1 INTRODUCTION

In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change Is performance. There is a drastic change in technology rather than advancement. In today’s world, we train our machines to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants.

Virtual assistants are software programs that help you ease your day-to-day tasks, such as showing weather reports, giving daily news, searching the internet etc. They can take commands by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana and Amazon's Alexa and this has been an inspiration for us to do this as a project. This system is designed to be used efficiently on desktops. Voice assistants are programs on digital devices that listen and respond to verbal commands. A user can say, “What's the weather?” and the voice assistant will answer with the weather report for that day and location.

This field of virtual assistants having speech recognition has seen some major advancements or innovations. This is mainly because of its demand in devices like smartwatches or fitness bands, speakers, Bluetooth earphones, mobile phones, laptop or desktop, television, etc. Almost all the digital devices which are coming nowadays are coming with voice assistants which help to control the device with speech recognition only. A new set of techniques is being developed constantly to improve the performance of voice automated search.

In today's technologically driven world, the integration of AI-powered voice assistants has become increasingly prevalent across various domains, from smart homes to personal devices and professional environments. The objective of this project is to delve into this burgeoning field and create an AI-based voice assistant capable of understanding natural language commands, executing tasks, and providing relevant information to users.

CHAPTER – 2 OVERVIEW

1. **Voice Assistant**

a. What is voice assistant

A voice assistant, also known as an intelligent personal assistant or a connected speaker, is a new type of device that is based on natural language speech recognition and is offered by popular companies like Apple, Amazon, and Google. We got inspired by that and created one our self.

b. Why do we need it

Usually, typing out and searching or doing day-to-day tasks becomes hectic. But our life does not need to be like that. One can ask for help to voice assistants. They let the users to perform a task using a speech command, as well as retrieve information via voice synthesis.

Following are the reasons to have a voice assistant.

- Minimal effort
  - It is easier to say a few words than type them.
- Eyes free
  - One can be as blind as a bat, but a voice assistant will always help you. Our ears are enough. One can also ask the bot about something while cooking at the same time.
- Fast response
  - Imagine how much time you must spend to find some information on a website? Or how many clicks do you need to make before you find the thing you need in a mobile application? Voice assistants do not generate such difficulties. One can ask a question and you have the answer.

c. Where to use it

Voice search has been a hot topic of discussion. Voice visibility will undoubtedly be a challenge. This is due to the lack of a visual interface for voice assistants. Users cannot see or interact with a voice interface unless it is linked to the Alexa or Google Assistant app. Search behaviour patterns will change dramatically as a result. Brands are currently undergoing a transformation in which touchpoints are transforming into listening points, and organic search will be the primary means of brand visibility. Advertising agencies are becoming more popular as voice search grows in popularity. Voice assistants will also continue to offer more individualized experiences as they get better at differentiating between voices. The number of people using voice assistants is expected to grow. According to the Voice Bot Smart Speaker Consumer Adoption Report 2018, almost ten percent of people who do not own a smart speaker plan to purchase one. If this holds true, the user base of smart speaker users will grow 50 percent, meaning a quarter of adults in the United States will own a smart speaker.

CHAPTER – 3 METHODOLOGY

EXISTING SYSTEM

Begin by providing an overview of the prevalent systems or technologies before the advent of AI voice assistants. Discuss the conventional methods or tools that people commonly used for tasks that your AI voice assistant now aims to improve or replace.

The existing systems presented several limitations that contributed to inefficiencies in user interaction and information retrieval:

- **Limited Interactivity:** Users were required to operate devices or systems through specific commands, limiting the natural flow of conversation and interaction.
- **Task Limitations:** Existing systems often struggled to perform complex tasks or provide dynamic, contextually relevant information based on user queries.
- **Complexity:** Some systems had intricate user interfaces, which posed challenges for users to navigate or operate efficiently.

Advancements in AI, natural language processing, and machine learning technologies laid the foundation for developing more intuitive and responsive systems. These advancements motivated the creation of an AI voice assistant to address the deficiencies of the existing systems.

The introduction of AI voice assistants signifies a significant transition from traditional systems to more sophisticated and user-friendly interfaces. The AI voice assistant aims to overcome the limitations of the previous systems by providing natural language interaction, enhanced task performance, personalized experiences, and faster access to information.

- **PROPOSED SYSTEM**

We are proposing a systemin an efficient way of implementing a Personal voice assistant, Speech Recognition library has many in-built functions, that will let the assistant understand the command given by user and the response will be sent back to user in voice, with Text to Speech functions. When assistant captures the voice command given by user, the under lying algorithms will convert the voice into text. And according to the keywords present in the text (command given by user), respective action will be performed by the assistant.

This is made possible with the functions present in different libraries. Also, the assistant was able to achieve all the functionalities with help of some API's. We had used these APIs for functionalities like performing extracting news from web sources, and for telling the weather. We will be sending a request, and through the API, we are getting the respective output. API's like NEWSAPI, are very helpful in performing things like calculations, making small web searches. And for getting the data from web. In this way, we can extract news from the web sources, and send them as input to a function for further purposes. Also, we have libraries like Random and many other libraries, each corresponding to a different technology. We used the library OS to implement Operating System related functionalities like Shutting down a system or restarting a system.

At the outset we make our program capable of using system voice with the help of sapi5 and pyttsx3. pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3. The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. Then we define the speak function to enable the program to speak the outputs.

- The proposed system will have the following functionality:
- The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.
- If the system is not able to gather information from the user input it will keep asking again to repeat till the desired number of times.
- The system can have both male and female voices according to user requirements.
- Features supported in the current version include playing music, texts, search on Wikipedia, or opening system installed applications, opening anything on the web browser, etc

- **OBJECTIVE**

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user "What can Ido for you?" and then responds to verbal input. Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding.

Provide a topic for research and continue with your tasks while the assistant does the research. Another difficult task is to remember test dates, birthdates, or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell assistant in advance about your tests and she reminds you well in advance so you can prepare for the test. 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 can speak around 150 during the same period. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

CODE: -

```
import pyttsx3
import speech_recognition as sr
import pyautogui
import requests
import pywhatkit
import cv2
import PyPDF2
import pygame
import speedtest
import json
import pyjokes
import platform
import psutil
import requests
import os
import time
import datetime
import webbrowser
import wikipedia
from bs4 import BeautifulSoup
from pywikihow import search_wikihow
from tkinter import filedialog
from plyer import notification

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
engine.setProperty('rate', 190)

def Speak(audio):
   print("      ")
   print(f"{audio}")
   engine.say(audio)
   engine.runAndWait()

def TakeCommand():
   r = sr.Recognizer()
   with sr.Microphone() as source:
      Speak("Listening...")
      r.pause_threshold = 1
```

```
        try:
            audio = r.listen(source)
            print("Recognizing.....")
            query = r.recognize_google(audio, language ='en')
            Speak(f"your command {query}\n")
            return query.lower()
        except sr.UnknownValueError:
            Speak("Sorry, I couldn't understand what you said. Could you please
repeat?")
            return ""

def greetMe():
    hour = int(datetime.datetime.now().hour)
    minute = int(datetime.datetime.now().minute)
    if (hour > 6 and hour < 12):
        Speak("Good Morning sir")
    elif (hour >= 12 and hour < 16):
        Speak("Good Afternoon sir")
    elif (hour >= 16 and hour < 19):
        Speak("Good evening sir")
    Speak(f"Time is {hour} Hour and {minute} minute")
    Speak("I am alpha. How my i help you")

if __name__ == "__main__":
    greetMe()
    try:
        while True:
            query = TakeCommand().lower()

            if 'exit alpha' in query or 'close alpha' in query or 'bye alpha' in query or 'stop
alpha' in query:
                Speak("Closing alpha. Goodbye!")
                break

            if 'alpha' in query or 'alfa' in query:
                if 'hello' in query:
                    Speak("hello sir. How may i help you.")

                #search on youube
                elif 'search on youtube' in query:
                    Speak("Ok sir, searching...")
                    query = query.replace("alpha", "")
                    query = query.replace("search on youtube", "")
                    web = "https://www.youtube.com/results?search_query=" + query
                    webbrowser.open(web)
                    Speak("Sir, here are the results")

                #wikipedia summary
                elif 'wikipedia summary' in query:
                    Speak("Searching on wikipedia....")
                    query = query.replace("alpha", "")
                    query = query.replace("wikipedia summary", "")
                    summary = wikipedia.summary(query, 3)
                    Speak(f"according to wikipedia {summary}")
                elif 'wikipedia' in query:
                    Speak("Searching on wikipedia....")
                    query = query.replace("alpha", "")
                    query = query.replace("wikipedia", "")
                    summary = wikipedia.summary(query)
                    Speak(f"according to wikipedia {summary}")

                # Screenshot
                elif "take screenshot" in query:
                    Speak('tell me a name for the file')
                    name = TakeCommand().lower()
                    time.sleep(3)
                    img = pyautogui.screenshot()
                    img.save(f"{name}.png")
                    Speak("screenshot saved")

                # IP address found
                elif "what is my ip address" in query:
                    Speak("Checking")
                    try:
                        ipAdd = requests.get('https://api.ipify.org').text
                        print(ipAdd)
                        Speak(f"your ip adress is {ipAdd}")
                        # Speak(ipAdd)
                    except Exception as e:
                        Speak("network is weak, please try again some time later")

                # Volume up
                elif "volume up" in query:
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")
                    pyautogui.press("volumeup")

                # Volume down
                elif "volume down" in query:
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")
                    pyautogui.press("volumedown")

                # mute
                elif "mute" in query:
                    pyautogui.press("volumemute")

                # Notepad open
                elif "open notepad and write your name" in query:
                    pyautogui.hotkey('win')
                    time.sleep(1)
                    pyautogui.write('notepad')
                    time.sleep(1)
                    pyautogui.press('enter')
                    time.sleep(1)
                    pyautogui.write("Alpha")

                # Notepad close
                elif "close notepad" in query:
                    os.system("taskkill /f /im notepad.exe")

                # open softwares - vs code, visual studio, photoshop, excel, word, cmd
                elif 'open' in query:
                    Speak('Ok sir opening...')
                    pyautogui.hotkey('win')
                    time.sleep(1)
                    query = query.replace("alpha", "")
                    query = query.replace("open", "")
                    pyautogui.write(query)
                    time.sleep(1)
                    pyautogui.press('enter')
                    time.sleep(1)

                #  close softwares vs code, visual studio, photoshop, excel, word, cmd
                elif 'close ' in query:
                    Speak('Ok sir closing...')
                    if 'vs code' in query:
                        pyautogui.hotkey('alt', 'f')
                        time.sleep(1)
                        pyautogui.press('x')

                    elif 'excel' in query:
                        os.system("TASKKILL /f /im EXCEL.EXE")
                    elif 'word' in query:
                        os.system("TASKKILL /f /im WINWORD.EXE")
                    elif 'powerpoint' in query:
                        os.system("TASKKILL /f /im POWERPNT.EXE")
                    elif 'this pc' in query:
                        pyautogui.hotkey('ctrl', 'w')
                    elif 'cmd' in query:
                        os.system("taskkill /f /im cmd.exe")
                    elif "paint" in query:
                        print('close paint')
                        os.system("TASKKILL /f /im MSPAINT.EXE")
```

```
            elif "window" in query:
                print('window closing')
                pyautogui.hotkey('alt','f4')

        elif " draw rectangle" in query:
            pyautogui.moveTo(515, 75, 1)
            pyautogui.rightClick()
            pyautogui.click()
            pyautogui.moveTo(965, 273, 1)
            pyautogui.dragRel(700, 300, 1)

        elif " draw rectangular spiral" in query:
            pyautogui.moveTo(100, 193, 1)
            pyautogui.rightClick()
            pyautogui.click()
            distance = 300
            while distance > 0:
                pyautogui.dragRel(distance, 0, 0.1, button="left")
                distance = distance - 10
                pyautogui.dragRel(0, distance, 0.1, button="left")
                pyautogui.dragRel(-distance, 0, 0.1, button="left")
                distance = distance - 10
                pyautogui.dragRel(0, -distance, 0.1, button="left")

        elif "drag yash" in query:
            pyautogui.moveTo(346, 55, 2)
            pyautogui.dragRel(1634, 41, 2)

        elif 'tell me a joke' in query:
            Speak(pyjokes.get_joke())

        elif 'system information' in query or 'hardware information' in query:
            system_info = f"System: {platform.system()} {platform.version()}\n"
            system_info += f"Processor: {platform.processor()}\n"
            system_info += f"Architecture: {platform.architecture()}\n"
            system_info += f"RAM: {psutil.virtual_memory().total / (1024 ** 3):.2f}
GB\n"
            system_info += f"CPU Cores: {psutil.cpu_count(logical=False)}\n"
            system_info += f"CPU Threads: {psutil.cpu_count(logical=True)}\n"
            print(system_info)
            Speak("Here is your system information.")
            Speak(system_info)
        elif 'find my location' in query:
            ipAdd = requests.get('https://api.ipify.org').text
            response = requests.get(f"https://ipinfo.io/{ipAdd}/json")
            data = response.json()
            Speak(f'IP Address:{data.get("ip")}')
            Speak(f'Location: {data.get("city")}')
            Speak(f'Region: {data.get("region")}')
            Speak(f'Country: {data.get("country")}')
            Speak(f'Location Coordinates: {data.get("loc")}')

        elif 'weather' in query:
            url = f'https://www.google.com/search?q=weather+{"Ahmedabad"}'
            response = requests.get(url)
            soup = BeautifulSoup(response.text, 'html.parser')
            # Extracting temperature and description
            temperature = soup.find('div', class_='BNeawe iBp4i AP7Wnd').text
            description = soup.find('div', class_='BNeawe tAd8D AP7Wnd').text
            Speak(f"Weather in Ahmedabad: {temperature}, {description}")

        elif "how" in query:
            query = query.replace("alpha", "")
            max_result = 1
            question_to = search_wikihow(query, max_result)
            assert len(question_to) == 1
            question_to[0].print()
            Speak(question_to[0].summary)

        elif 'battery' in query:
            battery = psutil.sensors_battery()
            percentage = battery.percent
            pluged_on = battery.power_plugged
            other = battery.secsleft
            Speak(f"sir our battery percentage is {percentage} and plug is
{pluged_on} ")

        elif 'switch the window' in query:
            pyautogui.keyDown("alt")
            pyautogui.press("tab")
            time.sleep(1)
            pyautogui.keyUp("alt")
        elif "speed test" in query:
            st = speedtest.Speedtest()
            download_speed = st.download() / 1000000  # Convert to Mbps
            upload_speed = st.upload() / 1000000  # Convert to Mbps
            Speak(f"Sir, we have {download_speed:.2f} Mbps download speed and
{upload_speed:.2f} Mbps upload speed.")

        elif "tell me news" in query:
            Speak("Which topic you want news")
            topic = TakeCommand()
            Speak("please wait sir , fetching the latest news")
            main_url = f"https://newsapi.org/v2/top-
headlines?country=in&category={topic}&apiKey=a15361ac0fab426aaa15b9275e754
68a"
            main_page = requests.get(main_url).json()
            articles = main_page['articles']
            head = []
            numberByNews = ["First","Second","third","fourth","fifth"]
            for ar in articles:
                head.append(ar["title"])
            for i in range(len(numberByNews)):
                Speak(f"todays {numberByNews[i]} news is : {head[i]}")

        elif "shutdown the system" in query:
            os.system("shutdown /s /t 5")

        elif "restart the system" in query:
            os.system("shutdown /r /t 5")

        elif "sleep the system" in query:
            print("sleeping")
            pyautogui.hotkey('win','x')
            time.sleep(1)
            pyautogui.press('u')
            time.sleep(1)
            pyautogui.press('s')

        elif 'read pdf' in query:
            path = filedialog.askopenfilename()
            pdf = open(path,'rb')
            pdfread = PyPDF2.PdfReader(pdf)
            pages = len(pdfread.pages)
```

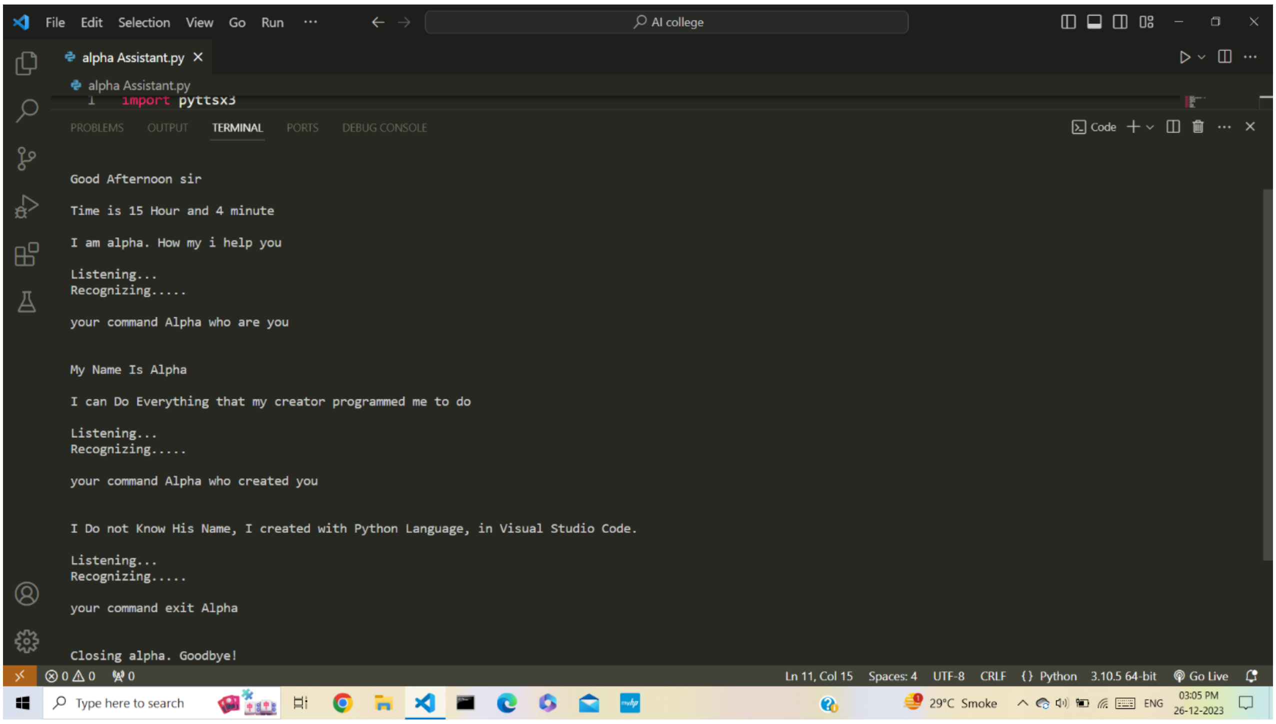
```
    Speak(f"Total pages in pdf are {pages}")
    print(f"Total pages in pdf are {pages}")
    Speak("Enter page number which i have to read")
    pag = int(input("Enter page number : "))
    pag = max(0, min(pag, pages - 1))
    pg = pdfread.pages[pag]
    text = pg.extract_text()
    Speak(text)

elif "schedule my day" in query:
    tasks = [] #Empty list
    file = open("tasks.txt","a")
    file.close()
    no_tasks = int(input("Enter the no. of tasks :- "))
    i = 0
    for i in range(no_tasks):
        Speak(f"Enter task {i+1}")
        task = TakeCommand().lower()
        tasks.append(task)
        file = open("tasks.txt","a")
        file.write(f"{i}. {tasks[i]}\n")
        file.close()

elif "tell my schedule" in query:
    if os.path.exists("tasks.txt"):
        print('your schedule')
        file = open("tasks.txt","r")
        content = file.read()
        file.close()
        notification.notify(
            title = "My schedule :-",
            message = content,
            timeout = 15
            )
    else:
        print("No schedule found.")

elif "clear my schedule" in query:
    file = open("tasks.txt","w")
    file.write(f"")
    file.close()
    Speak("Schedule is deleted")
```

```
elif 'set alarm' in query:
    Speak("OK sir set alarm of 10 second.")
    currentt = time.time()
    alarmt = time.time() + 5
    second = alarmt - currentt
    if second > 0:
        time.sleep(second)
    pygame.mixer.init()
    pygame.mixer.music.load("alarm_1.mpeg")
    pygame.mixer.music.play()
    print("Alarm! It's time!")

# chrome
# Whastsapp message send
elif 'send whatsapp message' in query:
    Speak("what is the message")
    message = TakeCommand()
    webbrowser.open('https://web.whatsapp.com/')
    time.sleep(17)
    pyautogui.click(325, 270)
    query = query.replace("alpha", "")
    query = query.replace("send whatsapp message", "")
    print(query)
    pyautogui.typewrite(query)
    time.sleep(5)
    pyautogui.click(294, 430)
    time.sleep(3)
    pyautogui.click(896, 972)
    pyautogui.typewrite(message)
    time.sleep(3)
    pyautogui.click(1845, 970)

# Google search
elif 'google search' in query:
    query = query.replace("alpha", "")
    query = query.replace("google search", "")
    pyautogui.hotkey('alt', 'd')
    pyautogui.write(f"{query}", 0.1)
    pyautogui.press('enter')

# Open website
```

```
elif 'open website' in query:
    Speak("Ok sir, opening.....")
    query = query.replace("alpha", "")
    query = query.replace("open website", "")
    query = query.replace(" ", "")
    website = 'https://' + query + '.com/'
    # 'https://stackoverflow.com/'
    print(website)
    webbrowser.open(website)
    Speak("sir " + query + " website opened")

elif 'open new window' in query:
    pyautogui.hotkey('ctrl', 'n')

elif 'open new tab' in query:
    pyautogui.hotkey('ctrl', 't')

elif 'maximize this window' in query:
    pyautogui.hotkey('alt', 'space')
    time.sleep(1)
    pyautogui.press('x')

elif 'open incognito window' in query:
    pyautogui.hotkey('ctrl','shift','n')

elif 'open history' in query:
    pyautogui.hotkey('ctrl', 'h')

elif 'open downloads' in query:
    pyautogui.hotkey('ctrl', 'j')

elif 'previous tab' in query:
    pyautogui.hotkey('ctrl', 'shift', 'tab')

elif 'next tab' in query:
    pyautogui.hotkey('ctrl', 'tab')

elif 'close tab' in query:
    pyautogui.hotkey('ctrl', 'w')

elif 'close chrome' in query:
    pyautogui.hotkey('ctrl', 'shift', 'w')
```

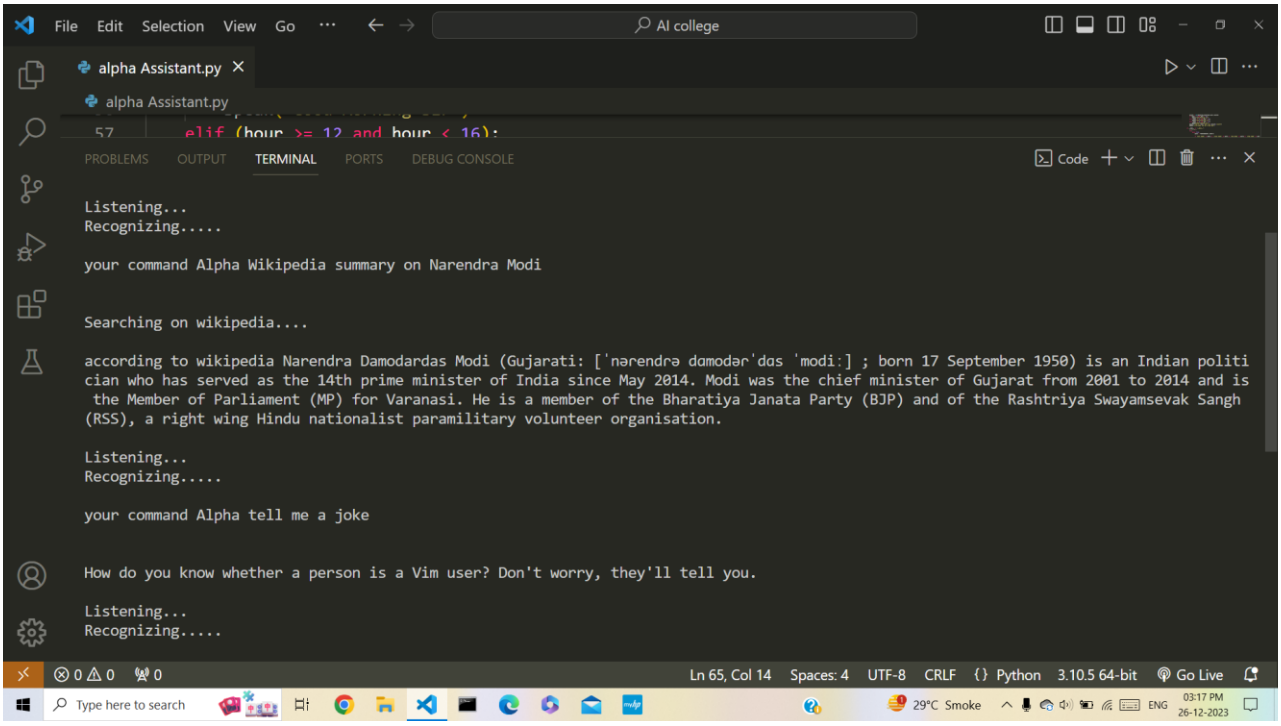
```
                elif "who are you" in query:
                    print('My Name Is Alpha')
                    Speak('My Name Is Alpha')
                    Speak('I can Do Everything that my creator programmed me to do')

                elif "who created you" in query:
                    Speak('I Do not Know His Name, I created with Python Language, in
Visual Studio Code.')
    except Exception as e:
        print(f"Error: {e}")
```

CHAPTER – 4 RESULT


File Edit Selection View Go Run ···
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 11, Col 15 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke
ENG
03:05 PM
26-12-2023

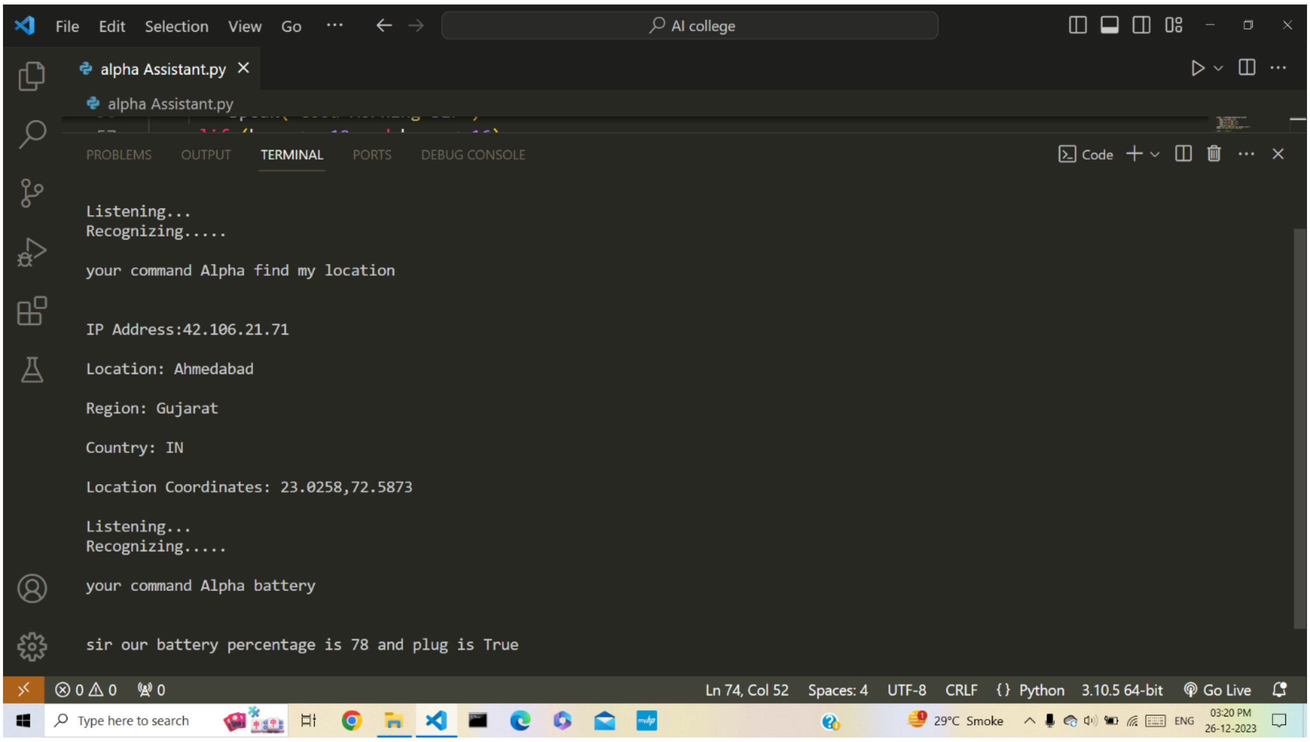
```
import pyttsx3
```

```
Good Afternoon sir

Time is 15 Hour and 4 minute

I am alpha. How my i help you

Listening...
Recognizing.....

your command Alpha who are you


My Name Is Alpha

I can Do Everything that my creator programmed me to do

Listening...
Recognizing.....

your command Alpha who created you


I Do not Know His Name, I created with Python Language, in Visual Studio Code.

Listening...
Recognizing.....

your command exit Alpha


Closing alpha. Goodbye!
```

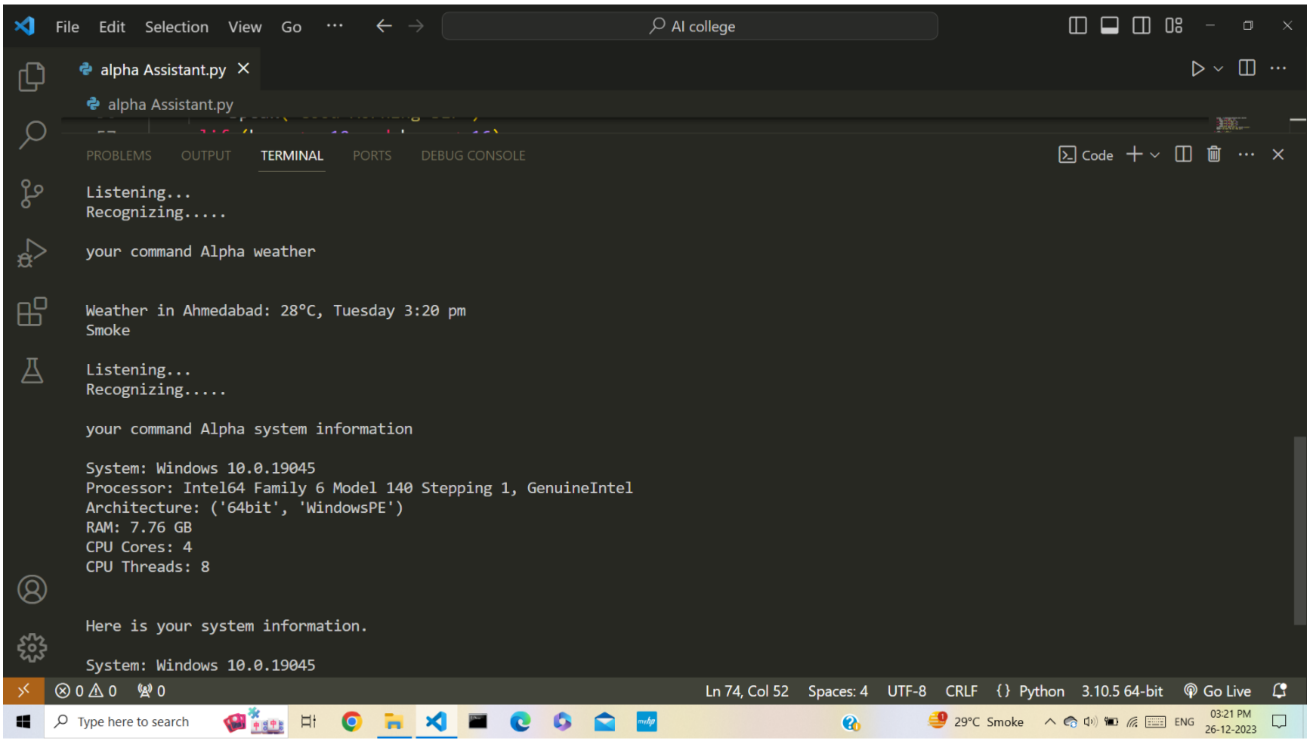

File Edit Selection View Go ···
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 65, Col 14 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke
ENG
03:17 PM
26-12-2023

```
57    elif (hour >= 12 and hour < 16):
```

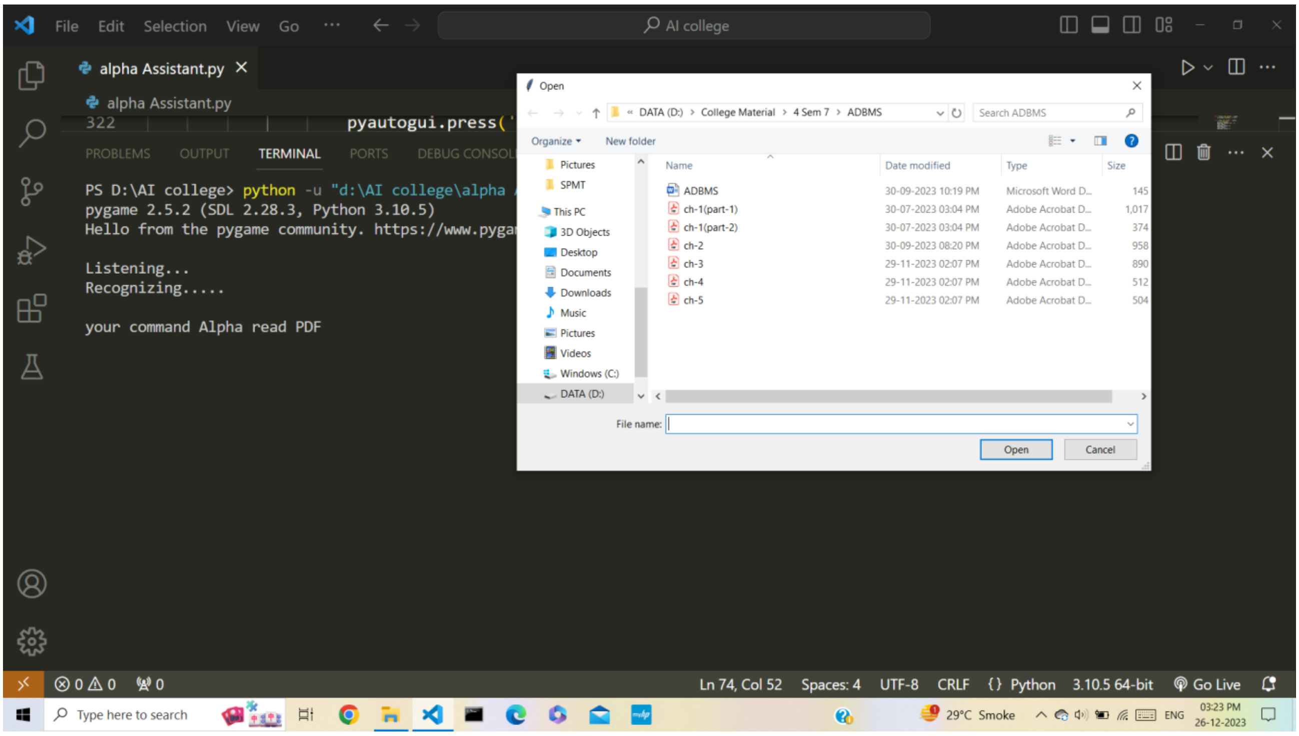
```
Listening...
Recognizing.....

your command Alpha Wikipedia summary on Narendra Modi


Searching on wikipedia....

according to wikipedia Narendra Damodardas Modi (Gujarati: [ˈnəɾendɾə dɑmodəɾˈdɑs ˈmodiː] ; born 17 September 1950) is an Indian politi
cian who has served as the 14th prime minister of India since May 2014. Modi was the chief minister of Gujarat from 2001 to 2014 and is
 the Member of Parliament (MP) for Varanasi. He is a member of the Bharatiya Janata Party (BJP) and of the Rashtriya Swayamsevak Sangh
(RSS), a right wing Hindu nationalist paramilitary volunteer organisation.

Listening...
Recognizing.....

your command Alpha tell me a joke


How do you know whether a person is a Vim user? Don't worry, they'll tell you.

Listening...
Recognizing.....
```


File Edit Selection View Go
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke ENG 03:20 PM 26-12-2023

```
Listening...
Recognizing.....

your command Alpha find my location


IP Address:42.106.21.71

Location: Ahmedabad

Region: Gujarat

Country: IN

Location Coordinates: 23.0258,72.5873

Listening...
Recognizing.....

your command Alpha battery


sir our battery percentage is 78 and plug is True
```


File Edit Selection View Go
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke ENG 03:21 PM 26-12-2023

```
Listening...
Recognizing.....

your command Alpha weather


Weather in Ahmedabad: 28°C, Tuesday 3:20 pm
Smoke

Listening...
Recognizing.....

your command Alpha system information

System: Windows 10.0.19045
Processor: Intel64 Family 6 Model 140 Stepping 1, GenuineIntel
Architecture: ('64bit', 'WindowsPE')
RAM: 7.76 GB
CPU Cores: 4
CPU Threads: 8


Here is your system information.

System: Windows 10.0.19045
```

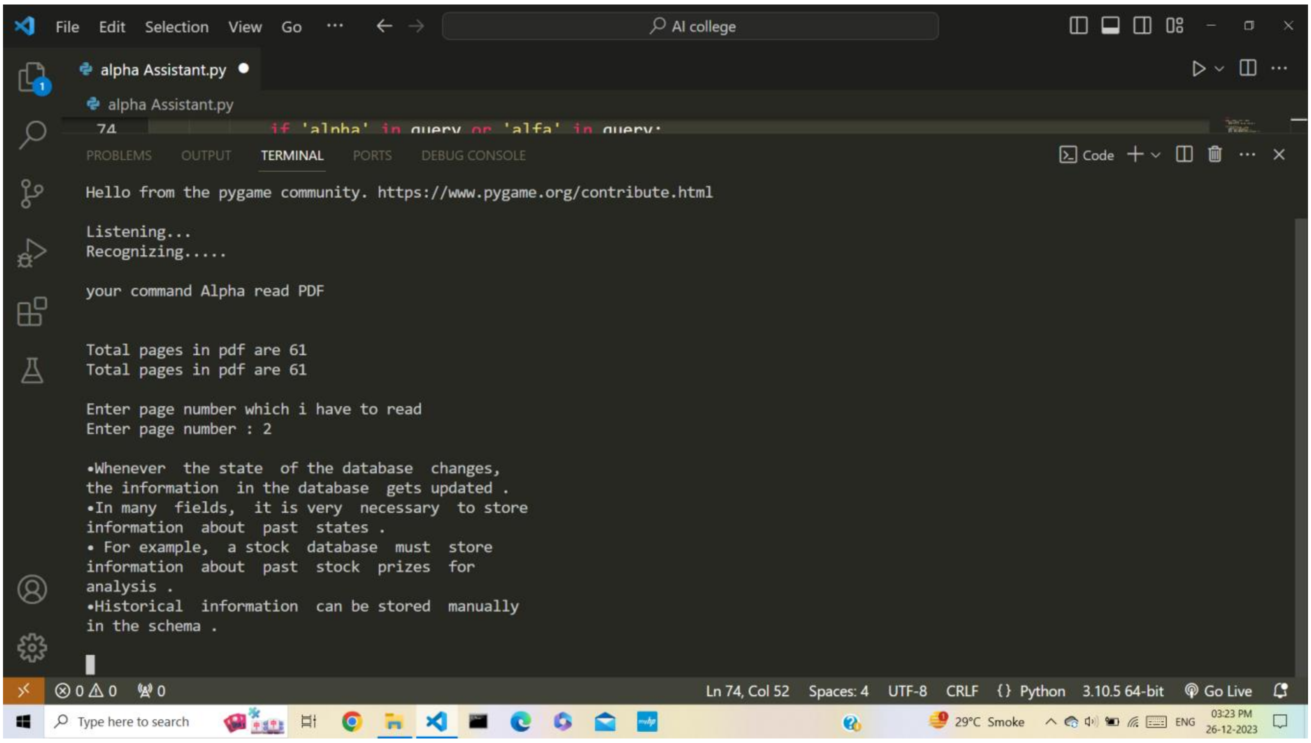


```
322                          pyautogui.press(
```

```
PS D:\AI college> python -u "d:\AI college\alpha
pygame 2.5.2 (SDL 2.28.3, Python 3.10.5)
Hello from the pygame community. https://www.pyga

Listening...
Recognizing.....

your command Alpha read PDF
```

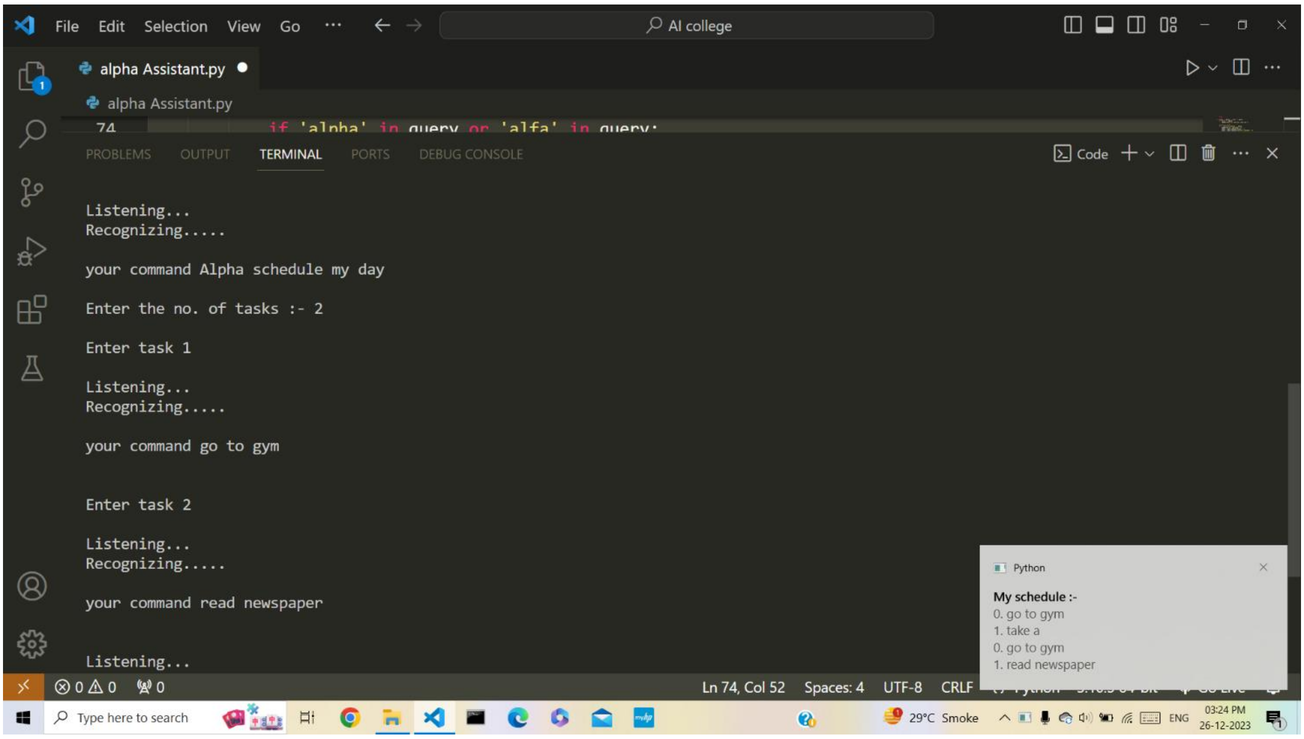
Name	Date modified	Type	Size
ADBMS	30-09-2023 10:19 PM	Microsoft Word D...	145
ch-1(part-1)	30-07-2023 03:04 PM	Adobe Acrobat D...	1,017
ch-1(part-2)	30-07-2023 03:04 PM	Adobe Acrobat D...	374
ch-2	30-09-2023 08:20 PM	Adobe Acrobat D...	958
ch-3	29-11-2023 02:07 PM	Adobe Acrobat D...	890
ch-4	29-11-2023 02:07 PM	Adobe Acrobat D...	512
ch-5	29-11-2023 02:07 PM	Adobe Acrobat D...	504

File name:



```
74              if 'alpha' in query or 'alfa' in query:
```

```
Hello from the pygame community. https://www.pygame.org/contribute.html

Listening...
Recognizing.....

your command Alpha read PDF


Total pages in pdf are 61
Total pages in pdf are 61

Enter page number which i have to read
Enter page number : 2

•Whenever  the state  of the database  changes,
the information  in the database  gets updated .
•In many  fields,  it is very  necessary  to store
information  about  past  states .
• For example,  a stock  database  must  store
information  about  past  stock  prizes  for
analysis .
•Historical  information  can be stored  manually
in the schema .
▌
```



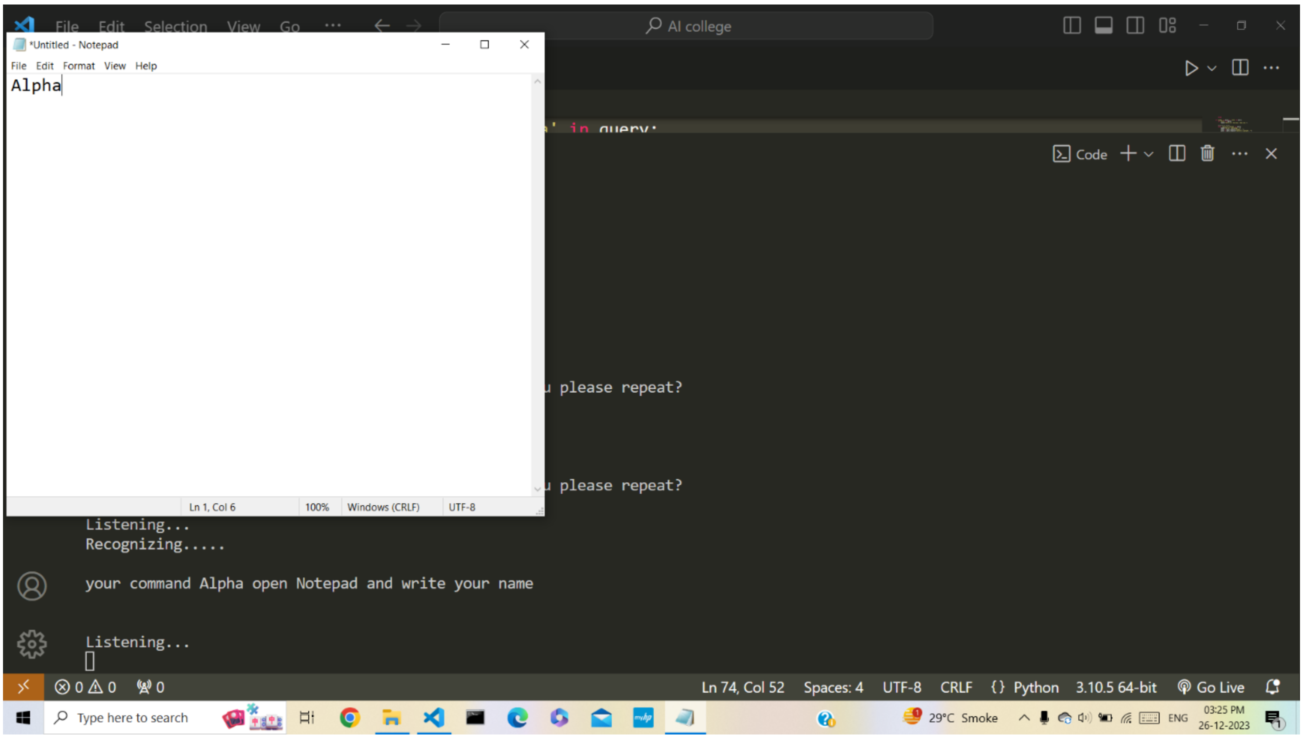
File Edit Selection View Go
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF
Type here to search
29°C Smoke
ENG
03:24 PM
26-12-2023

```
74        if 'alpha' in query or 'alfa' in query:
```

```
Listening...
Recognizing.....

your command Alpha schedule my day

Enter the no. of tasks :- 2

Enter task 1

Listening...
Recognizing.....

your command go to gym


Enter task 2

Listening...
Recognizing.....

your command read newspaper


Listening...
```

Python

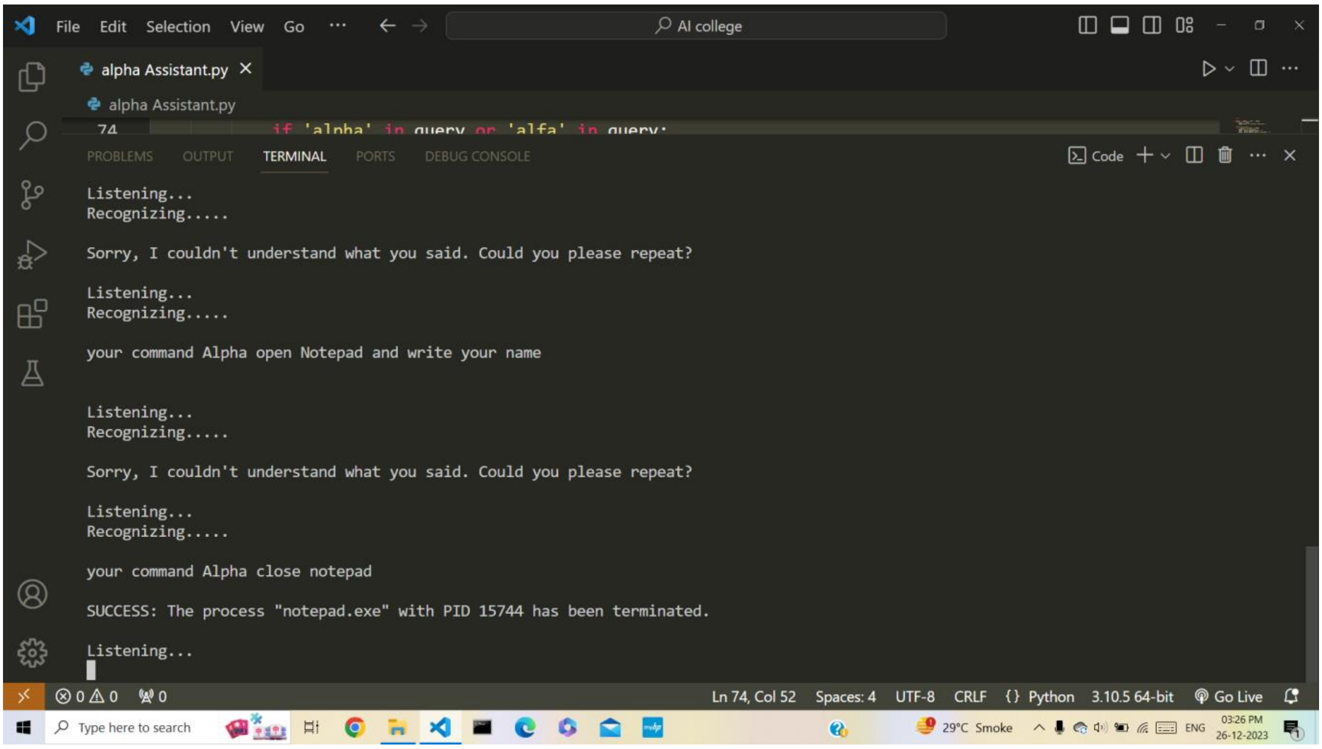
My schedule :-
0. go to gym
1. take a
0. go to gym
1. read newspaper



File Edit Selection View Go
AI college
*Untitled - Notepad
File Edit Format View Help
Ln 1, Col 6 100% Windows (CRLF) UTF-8
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke
ENG
03:25 PM
26-12-2023

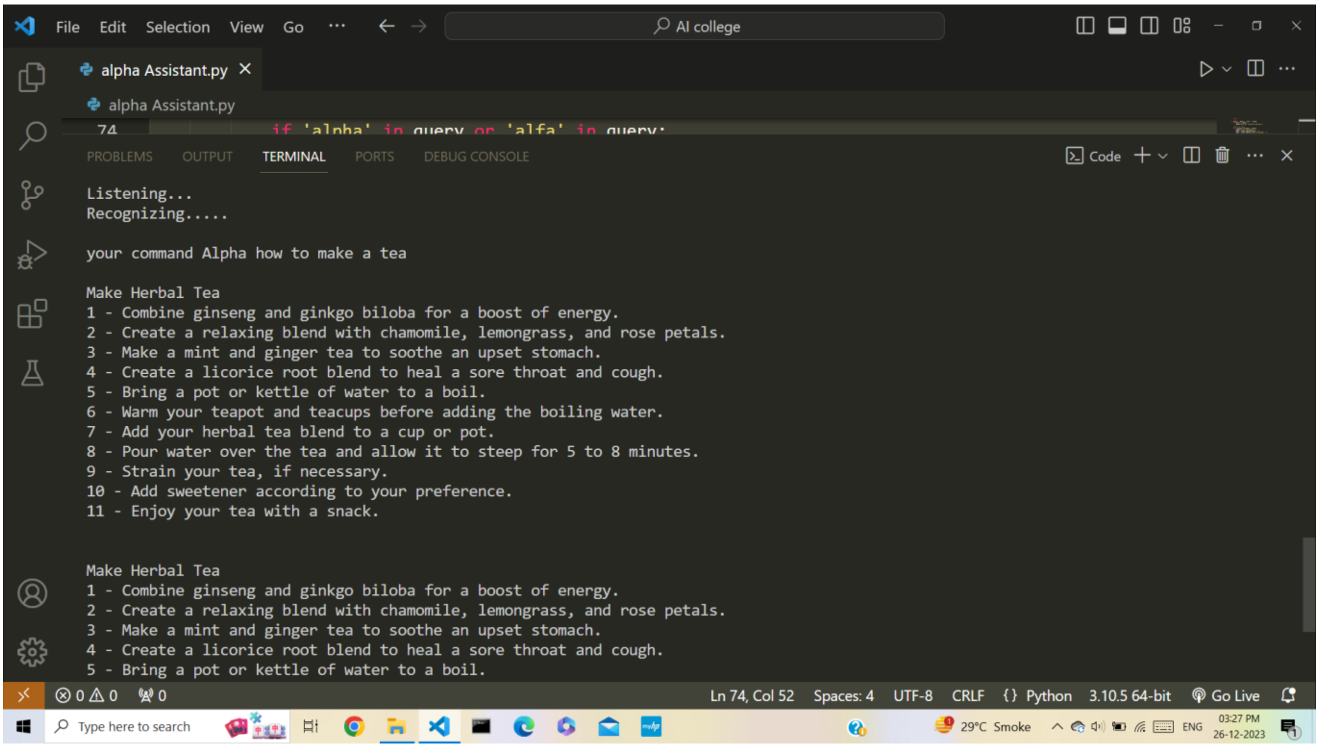
```
Alpha
```

```
u please repeat?

u please repeat?
Listening...
Recognizing.....

your command Alpha open Notepad and write your name


Listening...
```


File Edit Selection View Go ···
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke ENG 03:26 PM 26-12-2023

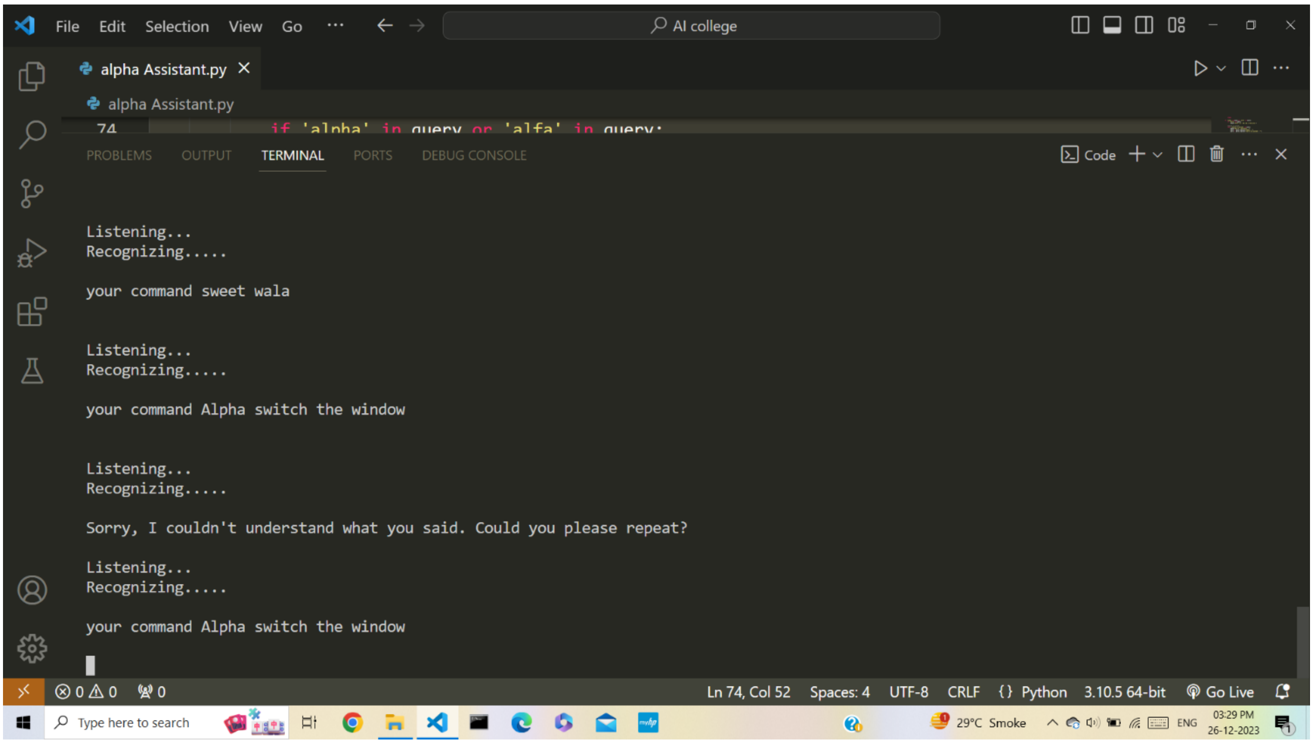
```
74            if 'alpha' in query or 'alfa' in query:
```

```
Listening...
Recognizing.....

Sorry, I couldn't understand what you said. Could you please repeat?

Listening...
Recognizing.....

your command Alpha open Notepad and write your name


Listening...
Recognizing.....

Sorry, I couldn't understand what you said. Could you please repeat?

Listening...
Recognizing.....

your command Alpha close notepad

SUCCESS: The process "notepad.exe" with PID 15744 has been terminated.

Listening...
```


File Edit Selection View Go ···
AI college
alpha Assistant.py
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke ENG 03:27 PM 26-12-2023

```
74            if 'alpha' in query or 'alfa' in query:
```

```
Listening...
Recognizing.....

your command Alpha how to make a tea

Make Herbal Tea
1 - Combine ginseng and ginkgo biloba for a boost of energy.
2 - Create a relaxing blend with chamomile, lemongrass, and rose petals.
3 - Make a mint and ginger tea to soothe an upset stomach.
4 - Create a licorice root blend to heal a sore throat and cough.
5 - Bring a pot or kettle of water to a boil.
6 - Warm your teapot and teacups before adding the boiling water.
7 - Add your herbal tea blend to a cup or pot.
8 - Pour water over the tea and allow it to steep for 5 to 8 minutes.
9 - Strain your tea, if necessary.
10 - Add sweetener according to your preference.
11 - Enjoy your tea with a snack.


Make Herbal Tea
1 - Combine ginseng and ginkgo biloba for a boost of energy.
2 - Create a relaxing blend with chamomile, lemongrass, and rose petals.
3 - Make a mint and ginger tea to soothe an upset stomach.
4 - Create a licorice root blend to heal a sore throat and cough.
5 - Bring a pot or kettle of water to a boil.
```


File Edit Selection View Go ···
AI college
alpha Assistant.py
alpha Assistant.py
74 if 'alpha' in query or 'alfa' in query:
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
Code
⊗ 0 ⚠ 0 0
Ln 74, Col 52 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 64-bit Go Live
Type here to search
29°C Smoke ENG 03:29 PM 26-12-2023

```
Listening...
Recognizing.....

your command sweet wala


Listening...
Recognizing.....

your command Alpha switch the window


Listening...
Recognizing.....

Sorry, I couldn't understand what you said. Could you please repeat?

Listening...
Recognizing.....

your command Alpha switch the window

▌
```

CHAPTER – 5 CONCLUSION

As stated before, "voice assistant is one of the biggest problem solvers" and you can see that in the proposals with the examples that it is in fact one of the biggest problem solvers of the current world. We can see that voice assistant is one of the major evolving artificial intelligences in the current world once again on seeing the proposal examples because at the past, the best feature which a voice assistant had was telling the date and searching the web and giving the results but now look at the functions that it can do so with this, we can say that it is an evolving software in the current world. The main idea is to develop the assistant even more advanced than it is now and make it the best ai in the world which will save an ample of time for its users. I would like to conclude with the statement that we will try our best and give one of the best voice assistants which we are able to.

REFERANCES

1. YouTube channels - www.youtube.com codewithharry.com
2. Artificial intelligence (AI), sometimes called machine intelligence. https://en.wikipedia.org/wiki/Artificial_intelligence
3. John Levis and Ruslan Suvorov, “Automatic Speech Recognition”.
4. GitHub - https://github.com/topics/ai-assistant
5. Wikipedia - https://www.wikipedia.org/