

# **Artificial Intelligence**

Voice Assistant

3<sup>rd</sup> Semester

Session 2023-2027

BS in Artificial Intelligence



**SUPERIOR UNIVERSITY**

Department of Software Engineering

The Superior University Lahore

Fall 2024

## Artificial Intelligence

Type(Nature of project)		Development	
Area of specialization		Artificial Intelligence	
Individual Project			
Roll no.	Name	Email Id	Signature
068	Mahak Farhan	su92-bsaim-f23-068@superior.edu.pk	Mahak

# Artificial Intelligence

## Table of Contents

Chapter 1.....	5
Introduction.....	5
Chapter 2.....	8
Tools and Technologies.....	8
Chapter 3.....	12
Implementation Code.....	12
Chapter 4.....	18
Result.....	18

# Artificial Intelligence

## List of Figures

Figure 1: Greetings.....	19
Figure 2: Opening Notepad.....	20
Figure 3: Wikipedia search.....	21
Figure 4: Playing on Youtube.....	22
Figure 5: Opening Calculator.....	23
Figure 6: Opening Setting.....	24
Figure 7: Searching from google.....	25
Figure 8: Opening Google.....	26
Figure 9: Opening Whatsapp.....	27
Figure 10: Telling Date.....	28
Figure 11: Telling time.....	29

# **Artificial Intelligence**

## **Chapter 1**

### **Introduction**

### **Voice assistant**

## Artificial Intelligence

Voice assistants are now an integral part of our lives. They make easier and faster the completion of certain tasks by issuing commands in voice. From setting up an alarm to searching something on the Internet; anything that needs to be done becomes easier and faster. This report is on designing a simple voice assistant using Python.

We do this by using popular Python libraries **os**, **pywhatkit**, **datetime**, **wikipedia**, **speech\_recognition**, and **pyttsx3**. We could develop such a program that would listen to what is being said, understand it, and respond in a friendly manner by using all the libraries.

The most pressing reason for having a widely used voice assistant is the saving of time and efforts, since it does away with manual

typing or clicking, thereby allowing the user to open applications or ask questions through voices. The project is going to be a voice assistant that can do, for example, tell the time, search on Wikipedia, play YouTube, and open applications. Such a project is perfect for Python because of ease of use and many libraries executing different functions.

Each library we use plays an important role in building the assistant. The **speech\_recognition** library helps the assistant listen to and understand voice commands by interpreting speech into text. The **pyttsx3** library enables the assistant to respond through the voice, hence making it conversational with the user. For any activity on the computer, we make use of the **os** library that is helpful in opening files or running programs. The **pywhatkit** library is used for playing videos on YouTube. The **datetime** library provides the current date and time, and the **wikipedia** library helps answer questions by finding information from the web.

## **Artificial Intelligence**

The goal of this project was to build a voice assistant that is easy to use, which can be used simply for simple tasks. By combining these libraries, the assistant will understand the voice coming from the users and process commands, giving useful responses.

In summary, the creation of voice assistants with Python is a magnificent tool for learning programming and artificial intelligence. It justifies the combination of fun and functionality in how technology creates simple solutions for everyday challenges. With the support of libraries like os, pywhatkit, datetime, wikipedia, speech\_recognition, and pyttsx3, this project develops a low-tech, user-friendly assistant that listens, understands, and responds. This is a small step toward a better understanding of tomorrow's voice-based technology and how it can improve our lives.

# **Artificial Intelligence**

## **Chapter 2**

### **Tools and Technologies**



# Artificial Intelligence

Tools and technologies that are used in this project are:

## **1. Python(Programming Language):**

Python is one such versatile and beginner-friendly language which I used in building this project. With features that are both easy to use and rich in libraries, it's exactly what developers use for developing intelligent, interactive systems.

## **2. Libraries:**

The following are the libraries which I used in the project:

### **a) os:**

os is the "hands" of the assistant enabling it to open up applications, manage files and execute system-level commands.

### **b) Pywhatkit:**

This gives the assistant the ability to automatically search web pages, or even play YouTube videos with just a single voice command.

### **c) datetime:**

It can provide the current date and time.

### **d) Wikipedia:**

It can provide short and reliable information from a large database.

### **e) speech\_recognition:**

## Artificial Intelligence

It can recognize speech from microphone and then give it to Google API to convert it into text.

### **f) Pyttsx3:**

It offers the assistant a vocal functionality whereby the assistant can communicate audibly, changing tone, pitch, and speed. And also convert text to speech.

### **3. Tools for Development:**

This is built in a Python-compatible Integrated Development Environment such as VS Code, which was fully equipped with debugging and testing capabilities to ensure smooth-running operations.

### **4. Audio Equipment:**

It records the voice of the user as the input medium and has speakers to deliver the audible response by the assistant.

### **5. Internet Connectivity:**

Voice assistant relies on internet connectivity to retrieve real-time information such as answering a question from Wikipedia or playing a YouTube video; it enables the assistant to provide dynamic, up-to-date answers.

### **6. About Laptop:**

This laptop that is used to make this project is Dell 7<sup>th</sup> generation, device name: Desktop-lnueqnd, processor: Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz ,2.40 GHz, installed RAM: 16.0 GB (15.9 GB usable), system type: 64-bit operating system, x64-based processor, edition: Windows 11 pro, version: 22H2. Total number of line space

## **Artificial Intelligence**

taken by this project is 109. And total memory taken by this project is 1-2 KB per function.

# **Artificial Intelligence**

## **Chapter 3**

### **Implementation Code**

## Artificial Intelligence

```
import pywhatkit as kit
import speech_recognition as sr
import pyttsx3
import wikipedia
from datetime import datetime
import os

p = pyttsx3.init()
rate = p.getProperty('rate') #Here rate is a instance and we are getting property
which is builed in pyttsx3 and we are getting rate property just to change the
speed of voice.
p.setProperty('rate',200) #Here we are setting the speed of voice to 200
def speak(text):
    p.say(text) #Say is a built in function it will say what the text we will pass in
this function.
    p.runAndWait()

def listen():
    recognizer = sr.Recognizer() #Here Recognizer() class creates an object that
helps us to retrieve information from a source
    with sr.Microphone() as source:
        print("Listening...")
        try:
```

## Artificial Intelligence

audio = recognizer.listen(source, timeout = 5) #This listen function listens to what we say and captures in our microphone and saves the command in the variable.

```
user_order = recognizer.recognize_google(audio)
```

```
print(user_order)
```

```
return user_order.lower()
```

```
except sr.UnknownValueError:
```

```
    speak("Sorry, I didn't understand what you said. Please say it again")
```

```
    return ""
```

```
def perform_task(user_order):
```

```
    if "how are you" in user_order:
```

```
        speak("I am doing great")
```

```
        speak("How may i help you")
```

```
    elif "time" in user_order:
```

```
        time = datetime.now().strftime("%I:%M")
```

```
        print(f"The current time is: {time}.")
```

```
        speak(f"The current time is {time}.")
```

```
        speak("Anyother thing you want to do")
```

```
    elif "date" in user_order:
```

## Artificial Intelligence

```
date = datetime.now().strftime("%B %d")
print(f"Today's date is: {date}.")
speak(f"Today's date is {date}.")
speak("Anyother thing you want to do")
```

```
elif "search" in user_order:
    speak("What you are searching for?")
    query = listen()
    if query:
        kit.search(query)
        speak(f"Searching for {query}.")
        speak("Anyother thing you want to do")
```

```
elif "play" in user_order:
    speak("Which video should I play?")
    video = listen()
    if video:
        kit.playonyt(video)
        speak(f"Playing {video} on YouTube.")
        speak("Anyother thing you want to do")
```

```
elif "give me some lines on the topic" in user_order:
    speak("What topic should I search on Wikipedia?")
```

## Artificial Intelligence

```
topic = listen()
if topic:
    search = wikipedia.summary(topic, sentences=2)
    print(f"According to Wikipedia, {search}")
    speak(f"According to Wikipedia, {search}")
speak("Anyother thing you want to do")

elif "open" in user_order:
    if "notepad" in user_order:
        os.system("notepad")
        speak("Opening Notepad.")
        speak("Anyother thing you want to do")
    elif "google" in user_order or "browser" in user_order:
        os.system("start chrome")
        speak("Opening Google.")
        speak("Anyother thing you want to do")
    elif "calculator" in user_order:
        os.system("start calc")
        speak("Opening calculator")
        speak("Anyother thing you want to do")
    elif "whatsapp" in user_order:
        os.system("start whatsapp://")
        speak("Opening whatsapp")
```



## Artificial Intelligence

```
    speak("Anyother thing you want to do")
```

```
elif "settings" in user_order:
```

```
    os.system("start ms-settings:")
```

```
    speak("Opening settings")
```

```
    speak("Anyother thing you want to do")
```

```
else:
```

```
    speak("Sorry, I can't open that application.")
```

```
    speak("Anyother thing you want to do")
```

```
elif "exit" in user_order or "quit" in user_order or "no thank you" in user_order:
```

```
    speak("OK, Let me know if you need further help.")
```

```
    speak("Have a good day!")
```

```
    exit()
```

```
else:
```

```
    speak("I didn't understand that command. Can you try again?")
```

```
# Main loop
```

```
speak("Hello! I am your voice assistant. How are you?")
```

```
while True:
```

```
    user_command = listen()
```

```
    if user_command:
```

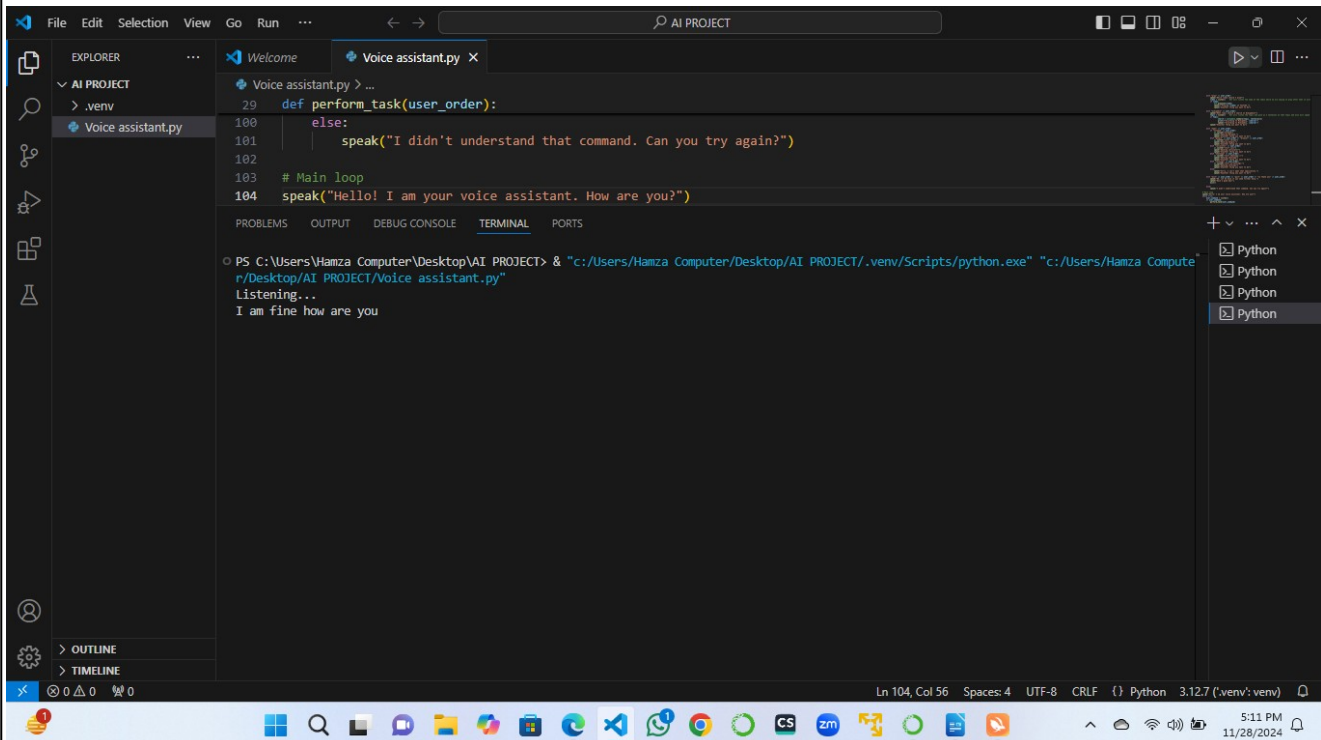
```
        perform_task(user_command)
```

# **Artificial Intelligence**

## **Chapter 4**

## **Results**

# Artificial Intelligence



```
File Edit Selection View Go Run ... AI PROJECT
Voice assistant.py X
Voice assistant.py > ...
29 def perform_task(user_order):
100     else:
101         speak("I didn't understand that command. Can you try again?")
102
103 # Main loop
104 speak("Hello! I am your voice assistant. How are you?")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
o PS C:\Users\Hamza Computer\Desktop\AI PROJECT> & "c:/Users/Hamza Computer/Desktop/AI PROJECT/.venv/Scripts/python.exe" "c:/Users/Hamza Computer/Desktop/AI PROJECT/Voice assistant.py"
Listening...
I am fine how are you
```

Figure 1: Greetings

First of all it will speak “I am your voice assistant. How are you”. After that it will listen what I will say. Here I say “I am fine. How are you”. It will speak “I am doing great. How may I help you”. This whole procedure is done by the libraries named “pyttsx3 and speech recognition”. pyttsx3 library is used to convert text to speech. Speech recognition library is used to convert speech to text. It is printing what I am saying.

# Artificial Intelligence

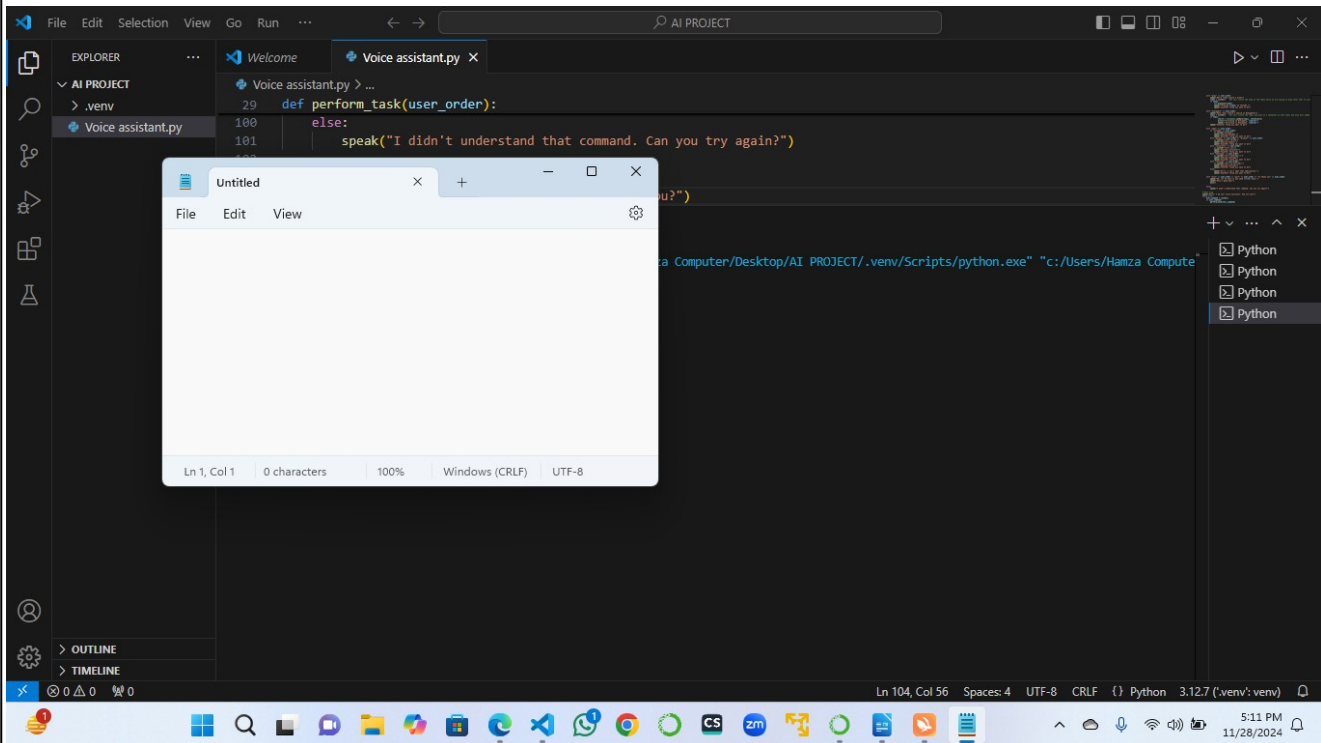
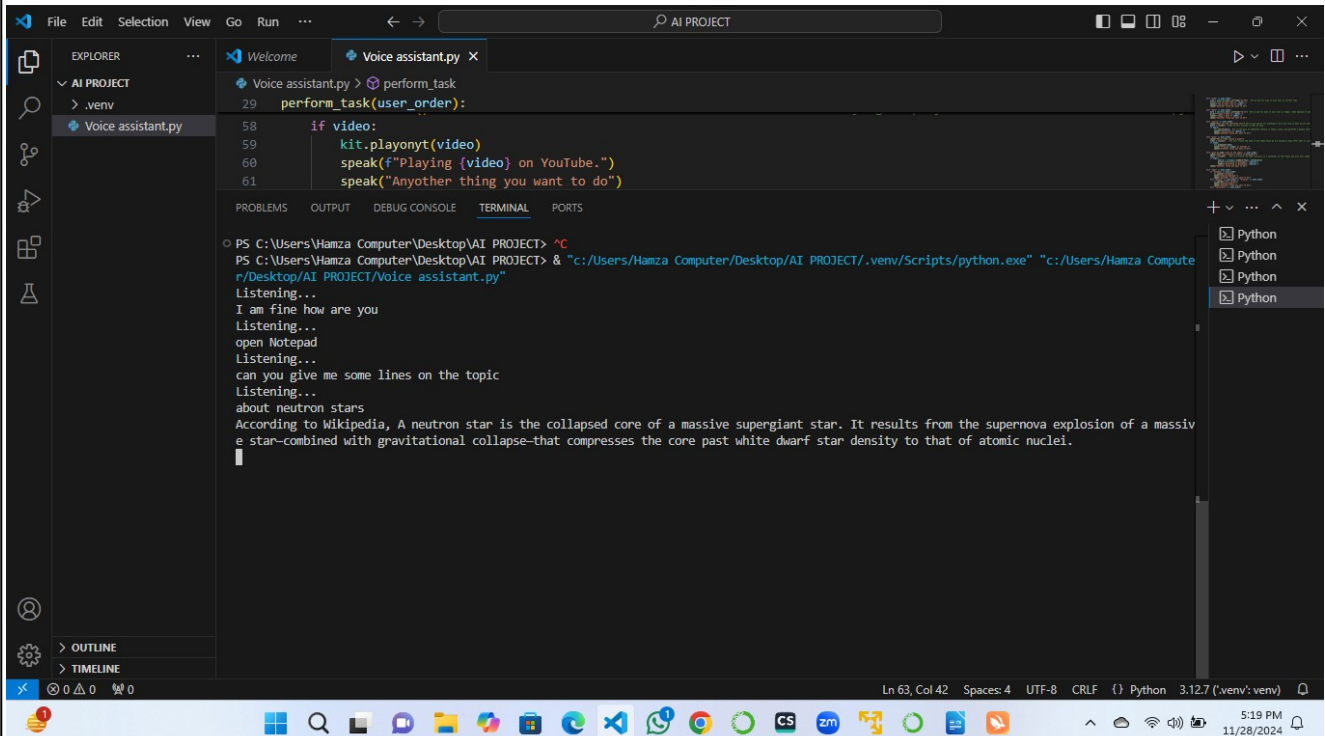


Figure 2: Opening Notepad

After that it will listen to what I am saying. Here I am saying it to open notepad so it is opening notepad for me. This procedure is done by the help of “os” library. The “os” library is linked with the operating system of the system so it can open the applications of the system.

# Artificial Intelligence



The screenshot shows a Visual Studio Code editor window with a file named 'Voice assistant.py' open. The code in the editor is as follows:

```
29 perform_task(user_order):  
58     if video:  
59         kit.playonyt(video)  
60         speak(f"Playing {video} on YouTube.")  
61         speak("Anyother thing you want to do")
```

The terminal window at the bottom shows the execution of the script. The prompt is 'PS C:\Users\Hamza Computer\Desktop\AI PROJECT> ^C'. The command executed is 'PS C:\Users\Hamza Computer\Desktop\AI PROJECT> & "c:/Users/Hamza Computer/Desktop/AI PROJECT/.venv/Scripts/python.exe" "c:/Users/Hamza Computer/Desktop/AI PROJECT/Voice assistant.py"'. The output of the script is:

```
Listening...  
I am fine how are you  
Listening...  
open Notepad  
Listening...  
can you give me some lines on the topic  
Listening...  
about neutron stars  
According to Wikipedia, A neutron star is the collapsed core of a massive supergiant star. It results from the supernova explosion of a massive star-combined with gravitational collapse-that compresses the core past white dwarf star density to that of atomic nuclei.
```

The status bar at the bottom of the editor indicates the current line and column as 'Ln 63, Col 42', the encoding as 'UTF-8', the line ending as 'CRLF', the language as 'Python', and the version as '3.12.7 (.venv:venv)'.

Figure 3: Wikipedia search

Here I am saying it to “search something from Wikipedia”. It will give me two sentences on that topic because I set it to two sentences. And it will also read that sentences. This procedure is done by “Wikipedia” library. It can provide true short information from a large database.

# Artificial Intelligence

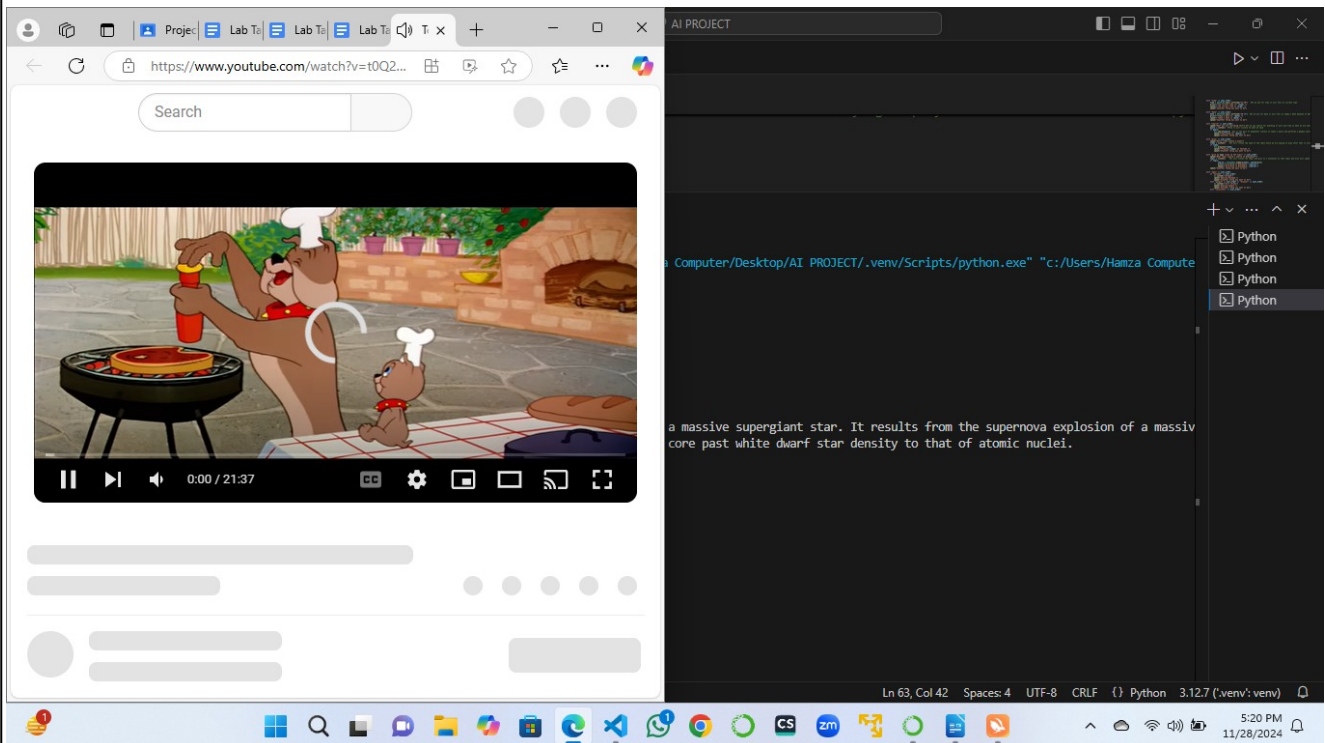


Figure 4: Playing on Youtube

Here I am saying it to play a video for me on Youtube. After that it will ask me that “which video should I play”. So the video I am saying to it is “Tom and Jerry”. It will play the cartoon on Youtube. This procedure is done by the “pywhatkit” library.

# Artificial Intelligence

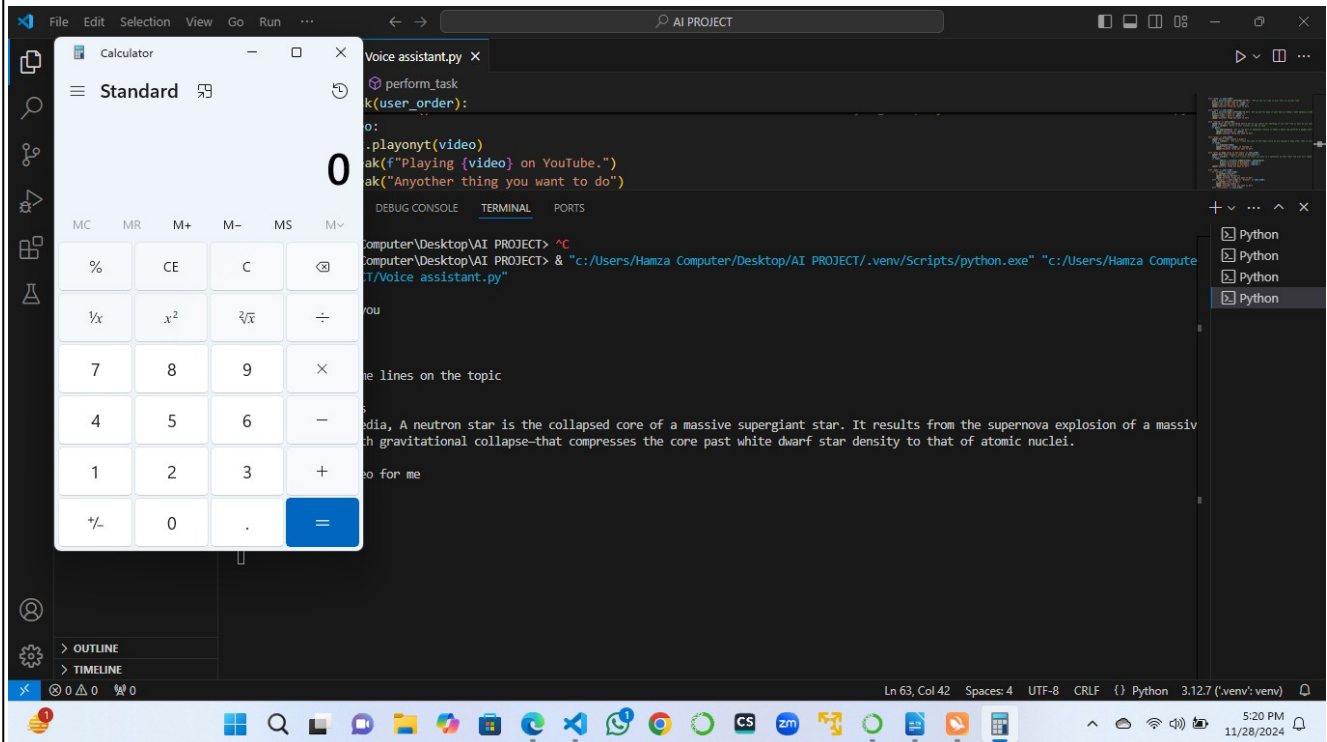


Figure 5: Opening Calculator

Here I am saying it to open Calculator. So it will open it for me with the help of “os” library.

## Artificial Intelligence

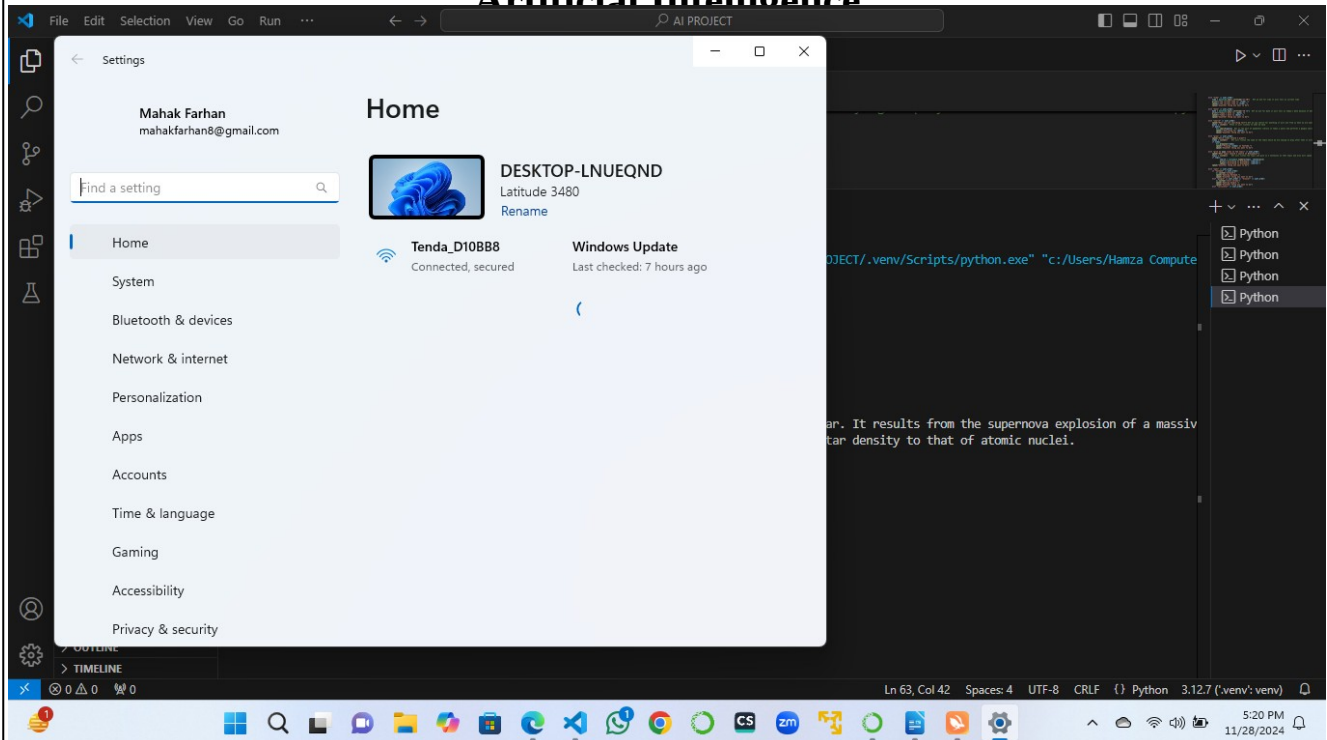


Figure 6: Opening Setting

Here I am saying it to open settings. So it will open it for me with the help of “os” library.



# Artificial Intelligence

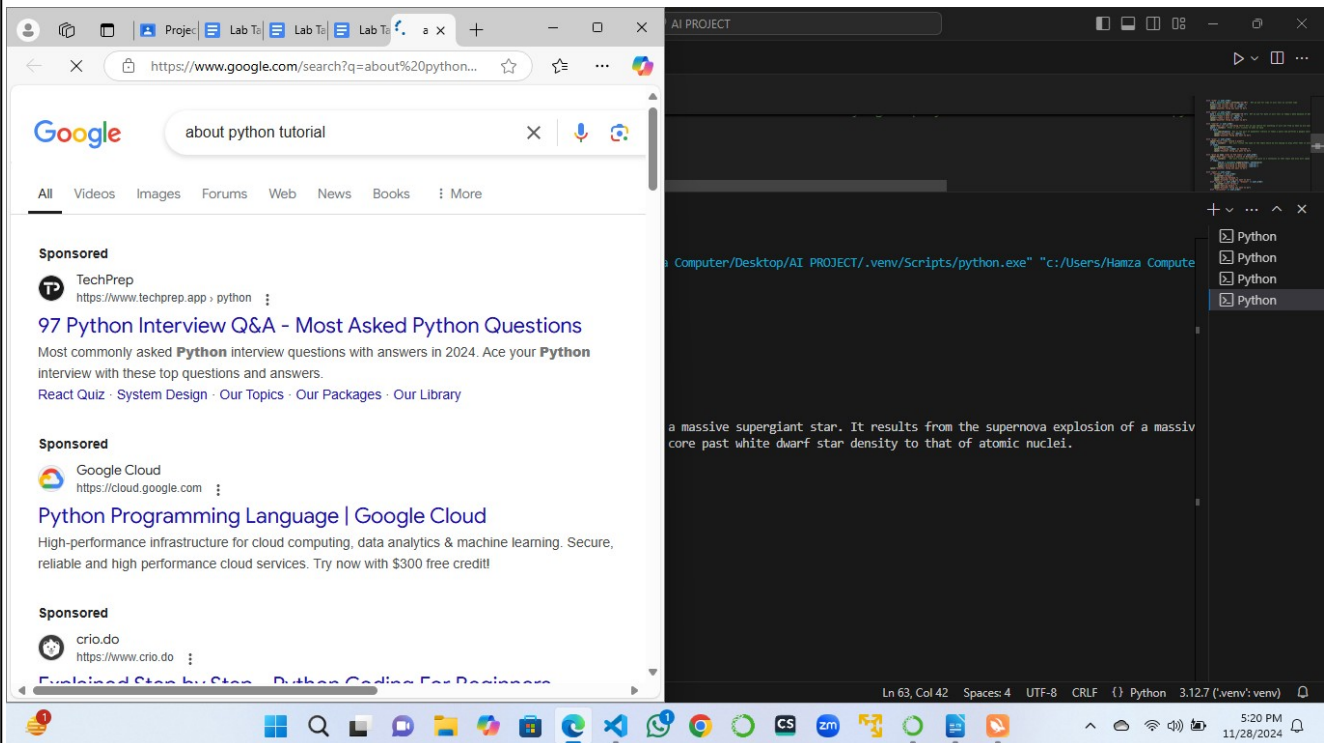


Figure 7: Searching from google

Here I am saying it to search something from google. So it will go to the google and search that topic on google. This procedure is also done by “pywhatkit” library. Because this library can also search web pages.

# Artificial Intelligence

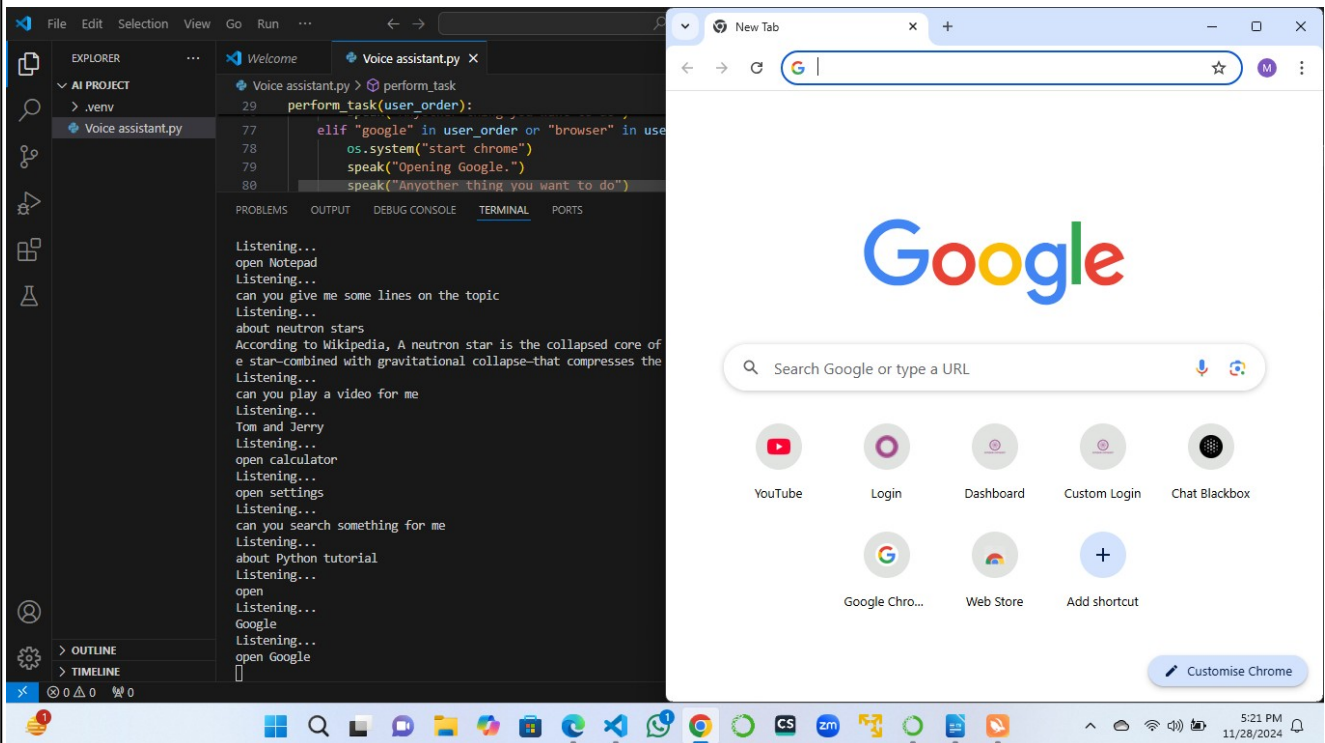


Figure 8: Opening Google

Here I am saying it to open Google. So it will open it for me with the help of “os” library.

# Artificial Intelligence

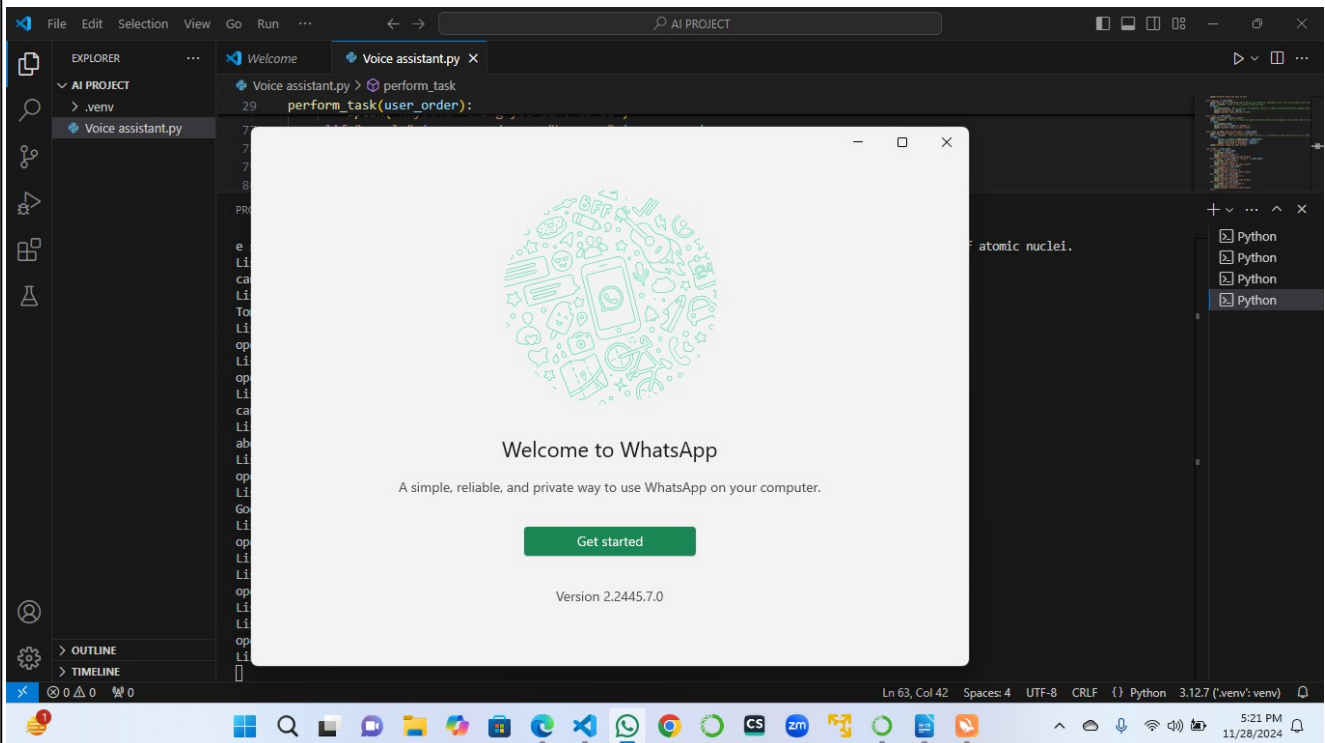


Figure 9: Opening Whatsapp

Here I am saying it to open Whatsapp. So it will open it with the help of “os” library.

# Artificial Intelligence

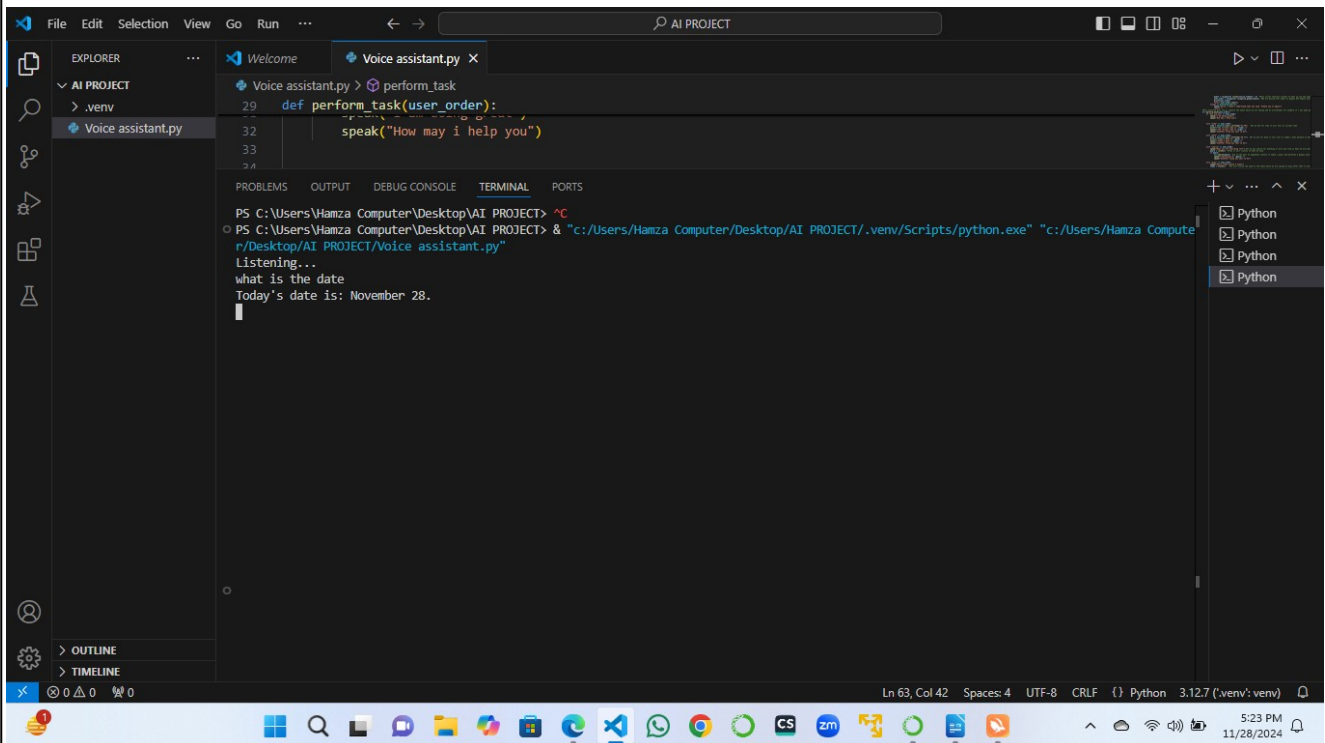
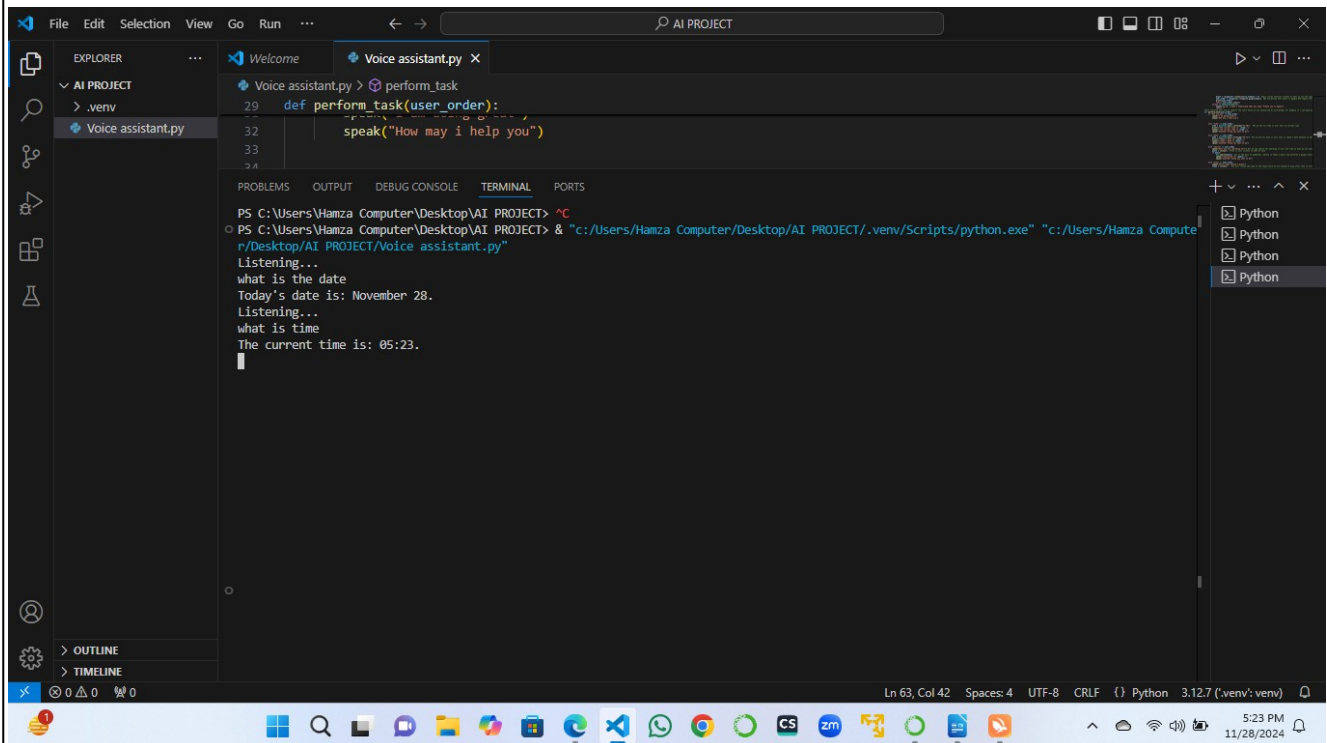


Figure 10: Telling Date

Here I am saying it to tell me today's date. So it is giving me today's date with the help of "datetime" library.

# Artificial Intelligence



The screenshot shows a Visual Studio Code editor window titled 'AI PROJECT'. The Explorer sidebar on the left shows a file named 'Voice assistant.py'. The editor area displays the following Python code:

```
def perform_task(user_order):  
    speak("How may i help you")
```

The TERMINAL panel at the bottom shows the execution of the script. The command prompt is at 'C:\Users\Hamza Computer\Desktop\AI PROJECT'. The script runs, listening for input. The user enters 'what is the date', and the script responds 'Today's date is: November 28.'. The user then enters 'what is time', and the script responds 'The current time is: 05:23.'. The status bar at the bottom indicates the file is at 'Ln 63, Col 42' and is using 'Python 3.12.7'.

Figure 11: Telling time

Here I am saying it to tell me current time. So it will tell me current time with the help of “datetime” library.