

TALKING FRIDAY

BY
PRIYANSH



Important modules: -

```
import speech_recognition as sr  
import webbrowser  
import pytsxs3  
import musicLibrary  
import requests
```

Important module
for fetching data
from internet in our
case from news API

→ To Take Voice
Input

→ to open browser
with python
To convert text → Voice

→ Self made module to store
favourite music inside
a dictionary

```
engine = pytsxs3.init()  
newsapi = "Your-apihere"  
voices = engine.getProperty('voices')  
engine.setProperty('voice', voices[1].id)
```

→ to convert text to speech
an object to the class pytsxs3
to use its functions

→ to use news-api

[Voice1, Voice2, Voice3]

→ to fetch available Voices and assigning Voice to our Assistant

```
def speak(text):  
    engine.say(text)  
    engine.runAndWait()
```

Made with Goodnotes

A string

→ function created for our convenience
to make agent
talk whatever
we pass to it.

```

Astömg
def processCommand(command):
    print(command)
    if("open google" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://google.com")
    elif("open facebook" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://facebook.com")
    elif("open chat gpt" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://chatgpt.com/")
    elif("open github" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://github.com")
    elif("open linkedin" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://linkedin.com")
    elif("open youtube" in command.lower()):
        speak("opening " + command.split(" ")[1])
        webbrowser.open("https://youtube.com")
    elif("open portfolio" in command.lower()):
        speak("opening your " + command.split(" ")[1])
        webbrowser.open_new_tab("https://helpful-elf-967c0f.netlify.app/")
    elif command.lower().startswith("play"):
        try:
            song = command.lower().split(" ")[1]
            link = musicLibrary.music[song]
            webbrowser.open(link)
        except KeyError:
            speak("Sorry sir, I don't have this song in my library")
    elif "news" in command.lower():
        r = requests.get("https://newsapi.org/v2/top-headlines?country=us&apiKey=(newsapi)")
        if r.status_code == 200:
            data = r.json()
            articles = data.get('articles',[])
            for article in articles:
                speak(article['title'])

```

LinkedIn]

Command
"open linkedin" → (" ") → [open]
to derive element. Split by Space
At index = 1 of the list

Used as best practice to check if working within same program
or working with some other program

```

if __name__ == "__main__":
    speak("INITIALIZING FRIDAY") → Command At Very Beginning...
    # LISTENING FOR THE WAKE WORD JARVIS
    while True: → Infinite loop → making program never ending
        # Microphone se input lena
        r = sr.Recognizer() →
        print("listening...")
        try:
            with sr.Microphone() as source: → for opening mic
                # Google Speech Recognition API se text me convert karo
                print("processing...")
                audio = r.listen(source, timeout=2, phrase_time_limit=1) → for listening Audio
                command = r.recognize_google(audio, language='en-US') → for Converting Audio to Text
                if(command.lower() == "friday"):
                    speak("YA")
                    print("Give Command")
                    with sr.Microphone() as source: → Mic open this time to take command
                        audio = r.listen(source, timeout=5, phrase_time_limit=2) → Listen Audio command
                        command = r.recognize_google(audio, language='en-US')
                        processCommand(command)
        
```

object of Recognizer class from module sr

Used To (and except to handle errors And overcoming chances of exiting program because of errors...)

NOW LISTENING FOR COMMAND

wake statement

CONVERTING TO TEXT USING GOOGLE API...

PASSED TO process Command!!!

```
except sr.UnknownValueError:  
    print("Samajh nahi aaya, thoda clearly bolo.")  
except sr.WaitTimeoutError:  
    print("Timeout: Tumne kuch bola hi nahi.")  
except sr.RequestError as e:  
    print(f"Request error: {e}")
```

To HANDLE Error
we use except

Block

And Also it
makes our program
run smoothly...

```
engines/Windows/  
def close_process_with_grace(process_name, wait_seconds=5):  
    # Step 1: Gracefully close (taskkill without /F)  
    subprocess.run(["taskkill", "/IM", process_name], shell=True)  
    print(f"Sent close signal to {process_name}, waiting {wait_seconds} seconds...")  
  
    # Step 2: Wait for process to close  
    time.sleep(wait_seconds)  
  
    # Step 3: Force kill (taskkill with /F)  
    # This will ensure process is terminated if still running  
    subprocess.run(["taskkill", "/F", "/IM", process_name], shell=True)  
    print(f"Force kill command sent to {process_name} (if still running).")
```

To smoothly
shutdown
In window
Application

```
import subprocess  
import time
```

Module necessary to open
And close files...

To Add delay for smooth shutting
off of the Softwares like whatsapp
And Browser in our case...