Project Title: Voice-Activated AI Chatbot

Objective: Create a voice-activated AI chatbot using Python, capable of responding to a range o
commands, conducting online searches, fetching information from Wikipedia, interacting with sys
commands, and more. This project will demonstrate the integration of voice recognition, Natural
Language Processing (NLP), and simple automation using Python.

In [1]:
```
# 1. Setup & Imports (Colab-Compatible)
!pip install SpeechRecognition gTTS wikipedia-api
```

```
Requirement already satisfied: SpeechRecognition in g:\anaconda\lib\site-pack
(3.14.3)
Requirement already satisfied: gTTS in g:\anaconda\lib\site-packages (2.5.4)
Requirement already satisfied: wikipedia-api in g:\anaconda\lib\site-packages
(0.8.1)
Requirement already satisfied: typing-extensions in g:\anaconda\lib\site-pack
(from SpeechRecognition) (4.11.0)
Requirement already satisfied: requests<3,>=2.27 in g:\anaconda\lib\site-pack
(from gTTS) (2.32.3)
Requirement already satisfied: click<8.2,>=7.1 in g:\anaconda\lib\site-packag
(from gTTS) (8.1.7)
Requirement already satisfied: colorama in g:\anaconda\lib\site-packages (fro
click<8.2,>=7.1->gTTS) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in g:\anaconda\lib\si
packages (from requests<3,>=2.27->gTTS) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in g:\anaconda\lib\site-packages
requests<3,>=2.27->gTTS) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in g:\anaconda\lib\site-pac
(from requests<3,>=2.27->gTTS) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in g:\anaconda\lib\site-pac
(from requests<3,>=2.27->gTTS) (2025.4.26)
```

In [2]:
```
# 2. Initialize TTS (Text-to-Speech) with gTTS
from gtts import gTTS
from IPython.display import Audio

def speak(text):
    print(f"AI: {text}")
    tts = gTTS(text=text, lang='en')
    tts.save("response.mp3")
    return Audio("response.mp3", autoplay=True)
```

In [3]:
```
# 3. Simulated Voice Input (takeCommand())
def takeCommand():
    query = input("You (type your command): ")
    return query.lower()
```

```
In [4]: # 4. Greeting Function
        import datetime

        def wishMe():
            hour = datetime.datetime.now().hour
            if 0 <= hour < 12:
                return speak("Good Morning!")
            elif 12 <= hour < 18:
                return speak("Good Afternoon!")
            else:
                return speak("Good Evening!")
```

```
In [5]: # 5. Wikipedia Search
        import wikipediaapi

        wiki = wikipediaapi.Wikipedia(user_agent='your-user-agent', language='en'

        def searchWikipedia(query):
            topic = query.replace("wikipedia", "").strip()
            page = wiki.page(topic)
            if page.exists():
                return speak(page.summary[:500])
            else:
                return speak("Sorry, I couldn't find anything on that topic.")
```

```
In [6]: # 6. Web Commands
        import webbrowser

        def openWebsite(site):
            websites = {
                "google": "https://www.google.com",
                "youtube": "https://www.youtube.com",
                "github": "https://github.com",
            }
            if site in websites:
                webbrowser.open(websites[site])
                return speak(f"Opening {site}")
            else:
                return speak("Website not recognized.")
```

```
In [7]: # 7. Handle Commands Function
        def handleCommand(query):
            if "wikipedia" in query:
                return searchWikipedia(query)
            elif "open" in query:
                site = query.replace("open", "").strip()
                return openWebsite(site)
            elif "time" in query:
                strTime = datetime.datetime.now().strftime("%H:%M:%S")
                return speak(f"The time is {strTime}")
            elif "hello" in query or "hi" in query:
                return speak("Hello! How can I assist you?")
            elif "who are you" in query:
                return speak("I am your voice assistant chatbot built using Pytho
            elif "note" in query:
                return speak("Note functionality is not yet implemented.")
            elif "shutdown" in query or "restart" in query:
                 return speak("System command simulation: This feature is not ava
            else:
                return speak("Sorry, I didn't understand that command.")
```

```
In [ ]: # 8. Main Driver Code
        wishMe()
        while True:
            query = takeCommand()
            if query in ["exit", "quit", "bye"]:
                speak("Goodbye! Have a nice day.")
                break
            handleCommand(query)
```

```
AI: Good Morning!
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```