

```
pip install SpeechRecognition pytttsx3 googlesearch-python pyaudio
```

Requirement already satisfied: SpeechRecognition in c:\users\admin\anaconda3\lib\site-packages (3.14.2)

Requirement already satisfied: pytttsx3 in c:\users\admin\anaconda3\lib\site-packages (2.98)

Requirement already satisfied: googlesearch-python in c:\users\admin\anaconda3\lib\site-packages (1.3.0)

Requirement already satisfied: pyaudio in c:\users\admin\anaconda3\lib\site-packages (0.2.14)

Requirement already satisfied: typing-extensions in c:\users\admin\anaconda3\lib\site-packages (from SpeechRecognition) (4.11.0)

Requirement already satisfied: comtypes in c:\users\admin\anaconda3\lib\site-packages (from pytttsx3) (1.4.10)

Requirement already satisfied: pypiwin32 in c:\users\admin\anaconda3\lib\site-packages (from pytttsx3) (223)

Requirement already satisfied: pywin32 in c:\users\admin\anaconda3\lib\site-packages (from pytttsx3) (305.1)

Requirement already satisfied: beautifulsoup4>=4.9 in c:\users\admin\anaconda3\lib\site-packages (from googlesearch-python) (4.12.3)

Requirement already satisfied: requests>=2.20 in c:\users\admin\anaconda3\lib\site-packages (from googlesearch-python) (2.32.3)

Requirement already satisfied: soupsieve>1.2 in c:\users\admin\anaconda3\lib\site-packages (from beautifulsoup4>=4.9->googlesearch-python) (2.5)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\admin\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (2.2.3)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests>=2.20->googlesearch-python) (2025.1.31)

Note: you may need to restart the kernel to use updated packages.

```
import speech_recognition as sr
```

```
import pytttsx3
```

```
from googlesearch import search
```

```
def recognize_speech_from_mic(recognizer, microphone):
```

```
    """  
    Capture speech from the microphone and convert it to text.  
    Returns a dictionary with 'success', 'error', and 'transcription'  
    keys.  
    """
```

```
    if not isinstance(recognizer, sr.Recognizer):
```

```

        raise TypeError("`recognizer` must be an instance of
Recognizer")
    if not isinstance(microphone, sr.Microphone):
        raise TypeError("`microphone` must be an instance of
Microphone")

    # Adjust recognizer sensitivity to ambient noise and record audio
    with microphone as source:
        print("Adjusting for ambient noise, please wait...")
        recognizer.adjust_for_ambient_noise(source)
        print("Listening... Please speak your search query.")
        audio = recognizer.listen(source)

    response = {
        "success": True,
        "error": None,
        "transcription": None
    }

    try:
        # Using Google Web Speech API to recognize audio
        response["transcription"] = recognizer.recognize_google(audio)
    except sr.RequestError:
        # API was unreachable or unresponsive
        response["success"] = False
        response["error"] = "API unavailable/unresponsive"
    except sr.UnknownValueError:
        # Speech was unintelligible
        response["error"] = "Unable to recognize speech"

    return response

def speak_text(text):
    """
    Use pyttsx3 text-to-speech to say the given text aloud.
    """
    engine = pyttsx3.init()
    engine.setProperty('rate', 150) # Speech rate
    engine.say(text)
    engine.runAndWait()

def perform_search(query, num_results=5):
    """
    Perform a Google search using the 'googlesearch' library.
    Return a list of URLs for the top results.
    """
    print(f"Searching for: {query}")
    try:
        search_results = []
        for url in search(query, num_results=num_results, lang='en'):

```

```

        search_results.append(url)
    return search_results
except Exception as e:
    print(f"Error during search: {e}")
    return []

def main():
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    print("Welcome to the Speech-Based Search Engine!")
    print("Please speak your search query after the prompt.\n")

    speech = recognize_speech_from_mic(recognizer, microphone)

    if not speech["success"]:
        print("I didn't catch that. What did you say?")
        speak_text("I didn't catch that. Please try again.")
        return

    if speech["error"]:
        print(f"ERROR: {speech['error']}")
        speak_text("Sorry, there was an error recognizing your speech.")
        return

    query = speech["transcription"]
    print(f"You said: {query}")
    speak_text(f"You said: {query}. Searching now.")

    results = perform_search(query)

    if not results:
        print("No results found or error occurred during search.")
        speak_text("Sorry, I could not find any results.")
        return

    print("\nTop search results:")
    for i, url in enumerate(results, start=1):
        print(f"{i}. {url}")
        # Optionally read out each URL (can be commented out if not
        # desired)
        # speak_text(f"Result {i}: {url}")

    speak_text(f"I found {len(results)} results for your query. Check
    your screen for details.")

if __name__ == "__main__":
    main()

```

Welcome to the Speech-Based Search Engine!
Please speak your search query after the prompt.

Adjusting for ambient noise, please wait...
Listening... Please speak your search query.
You said: hello hello
Searching for: hello hello

Top search results:

1. <https://www.youtube.com/watch?v=fN1Cyr0ZK9M>
2. <https://www.youtube.com/watch?v=wKXXT244yFI>
3. [/search?num=7](#)
4. <https://supersimple.com/song/hello-hello/>
5. <https://www.youtube.com/watch?v=dNBC8jQFgHo>