AIM:

To develop a **real-time speech-to-text transcription system** using the **microphone** in Python that converts live speech input into text and displays it instantly.

ALGORITHM:

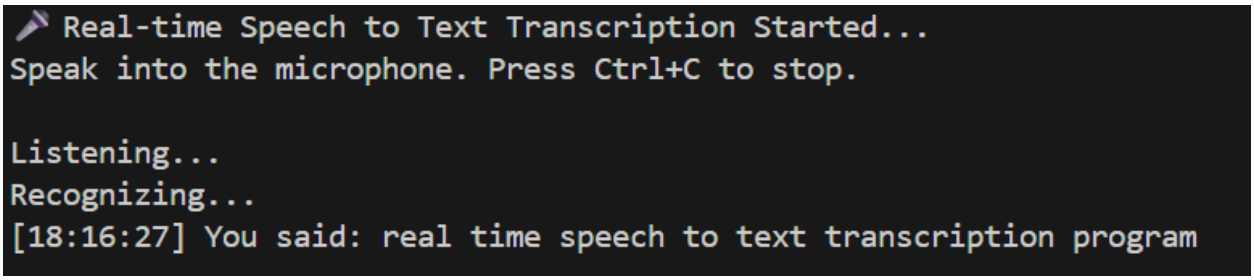
1. **Start** the program.
2. **Import** the necessary libraries (speech_recognition, datetime).
3. **Create a recognizer** object using sr.Recognizer().
4. **Access the microphone** using sr.Microphone() and adjust for ambient noise.
5. Continuously **listen to the user's speech input**.
6. Convert the **audio to text** using recognize_google() API.
7. **Display and store** the recognized text with timestamps.
8. Stop the recognition when the user interrupts (Ctrl+C or Stop condition).
9. **End** the program.

CODE:

```
import speech_recognition as sr
from datetime import datetime
def real_time_speech_to_text():
    recognizer = sr.Recognizer()
    print("Real-time Speech to Text Transcription Started...")
    print("Speak into the microphone. Press Ctrl+C to stop.\n")
    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source)
        while True:
            try:
                print("Listening...")
                audio = recognizer.listen(source, timeout=5, phrase_time_limit=8)
```

```
            print("Recognizing...")
            # Convert speech to text using Google Web Speech API
            text = recognizer.recognize_google(audio, language='en-IN')
            timestamp = datetime.now().strftime("%H:%M:%S")
            print(f"[{timestamp}] You said: {text}\n")
            # Optional: Save the result to a text file
            with open("transcript.txt", "a") as f:
                f.write(f"[{timestamp}] {text}\n")
        except sr.WaitTimeoutError:
            print(" Timeout: No speech detected, listening again...\n")
            continue
        except sr.UnknownValueError:
            print("Could not understand audio. Try again...\n")
        except KeyboardInterrupt:
            print("\n Transcription stopped by user.")
            break
        except Exception as e:
            print(f" Error: {e}")
            break
if __name__ == "__main__":
    real_time_speech_to_text()
```

OUTPUT:



RESULT:

Thus, the program was implemented successfully.