```
pip install openai-whisper pyaudio SpeechRecognition spacy
python -m spacy download en_core_web_sm
```

```
Collecting SpeechRecognition
  Using cached speechrecognition-3.14.3-py3-none-any.whl.metadata (30
kB)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.11/dist-packages (from SpeechRecognition)
(4.13.2)
Using cached speechrecognition-3.14.3-py3-none-any.whl (32.9 MB)
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.14.3
```

```
import speech_recognition as sr
import whisper
import spacy
import json

# Load Whisper model (base or tiny for speed, medium/large for
accuracy)
model = whisper.load_model("base")

# Load spaCy English NLP model
nlp = spacy.load("en_core_web_sm")

def transcribe_audio_whisper(filename):
    result = model.transcribe(filename)
    return result["text"]

def record_audio(filename="audio.wav"):
    recognizer = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening for command...")
        audio_data = recognizer.listen(source)
        with open(filename, "wb") as f:
            f.write(audio_data.get_wav_data())
    return filename

def parse_command(command_text):
    doc = nlp(command_text.lower())
    json_response = {
        "intent": None,
        "entities": {}
    }

    if "add" in command_text and "lead" in command_text:
        json_response["intent"] = "add_lead"
        for ent in doc.ents:
            if ent.label_ == "ORG":
                json_response["entities"]["company"] = ent.text
```

```
    elif "delete" in command_text and "lead" in command_text:
        json_response["intent"] = "delete_lead"
        for ent in doc.ents:
            if ent.label_ == "ORG":
                json_response["entities"]["company"] = ent.text
    else:
        json_response["intent"] = "unknown"

    return json_response

def main():
    # Step 1: Record and transcribe
    audio_file = record_audio()
    command_text = transcribe_audio_whisper(audio_file)
    print(f"Recognized Text: {command_text}")

    # Step 2: Parse
    parsed_result = parse_command(command_text)
    print("Parsed JSON:")
    print(json.dumps(parsed_result, indent=4))

if __name__ == "__main__":
    main()
```