Task – Text to Speech and Speech to Speech

Day – 8

Code –

```
import os
import numpy as np
import sounddevice as sd
from scipy.io.wavfile import write
import pyttsx3
import speech_recognition as sr
from huggingface_hub import InferenceClient


# ✅ Hugging Face Token (keep secret)
HF_TOKEN = "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"  # Replace with your actual token
client = InferenceClient(token=HF_TOKEN) if HF_TOKEN else None
# ✅ Text-to-Speech Engine (offline)
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
#engine.setProperty('rate', 150)
#engine.setProperty('volume', 1.0)
def speak(text):
    print(" 🗣️ Speaking:", text)
    engine.say(text)
    engine.runAndWait()
def text_to_speech():
    text = input(" 🔤 Enter text to speak: ")
    speak(text)
```

```
def speech_to_text():
    print("🎙️ Recording for 5 seconds...")
    fs = 16000
    duration = 5
    recording = sd.rec(int(duration * fs), samplerate=fs, channels=1, dtype='float32')
    sd.wait()
    pcm = np.int16(recording * 32767)  # ✅ Corrected line
    write("input.wav", fs, pcm)
    recognizer = sr.Recognizer()
    with sr.AudioFile("input.wav") as source:
        audio = recognizer.record(source)
    try:
        text = recognizer.recognize_google(audio)
        print("📝 You said:", text)
        return text
    except sr.UnknownValueError:
        print("❌ Could not understand audio.")
        return ""
    except sr.RequestError:
        print("❌ Could not connect to recognition service.")
        return ""
def generate_reply(prompt):
    print("🤖 Asking Hugging Face...")
    try:
        response = client.text_generation(
            prompt=prompt,
            model="tiiuae/falcon-7b-instruct",  # Verified working
            max_new_tokens=50
        )
```

```
        # Response can be list or dict
        if isinstance(response, list):
            return response[0].get("generated_text", "")
        return response.get("generated_text", "")
    except Exception as e:
        print("❌ HF Error:", repr(e))
        return ""
def speech_to_speech():
    text = speech_to_text()
    if not text:
        return
    reply = generate_reply(text)
    if not reply:
        reply = "Sorry, I couldn't generate a response."
    print("🗣️ Reply:", reply)
    speak(reply)
def main():
    while True:
        print("\n🤖 Voice Assistant Menu")
        print("1. Text to Speech")
        print("2. Speech to Text")
        print("3. Speech to Speech")
        print("4. Exit")
        choice = input("Enter your choice: ")
        if choice == "1":
            text_to_speech()
        elif choice == "2":
            speech_to_text()
        elif choice == "3":
```

```
            speech_to_speech()
        elif choice == "4":
            print("👋 Exiting...")
            break
        else:
            print("❌ Invalid choice. Please enter 1–4.")
if __name__ == "__main__":
    main()
```

Output –