**AI Voice Assistant with ChatGPT Integration**

**Introduction**

The purpose of this task is to develop an AI-powered voice assistant capable of interacting with users through audio input and output. The system integrates multiple functionalities, including speech recognition, natural language processing with OpenAI's ChatGPT API, and text-to-speech (TTS) capabilities. The assistant processes spoken language, generates a response using ChatGPT, and then converts that response into a spoken audio message that is played back to the user. This report details the implementation steps and confirms the fulfillment of the core requirements.

**Task Overview**

The task involves the following essential functionalities:

1. **Recording the User's Voice Input**: The assistant records the user's voice input through the microphone.
2. **Converting the Audio Input to Text**: The recorded audio is processed using speech recognition to convert the speech into text.
3. **Sending the Text to the ChatGPT API**: The recognized text is sent to OpenAI's ChatGPT API for generating a response.
4. **Receiving the Response and Converting It to Audio**: The response from ChatGPT is then converted into speech using a text-to-speech engine (gTTS).
5. **Playing the Audio Response**: Finally, the audio response is played back to the user.

**Implementation Details**

1. **Recording the User's Voice Input**: The system utilizes the **Gradio** library to create a user interface that facilitates audio input from the microphone. When the user interacts with the interface, the microphone records the audio. Gradio’s gr.Audio component allows for easy integration of audio input directly from the user, facilitating a seamless recording process.
2. **Converting the Audio Input to Text**: The recorded audio is processed using the **SpeechRecognition** library. The audio file is fed into the speech recognizer, which uses the Google Speech-to-Text API to transcribe the audio into text. If the system successfully recognizes the speech, the text is returned for further processing. In cases where speech recognition fails, an error message is provided, ensuring proper feedback to the user.
3. **Sending the Text to the ChatGPT API**: Once the speech is converted to text, the text is sent to the **ChatGPT API** for processing. The request is made using the OpenAI Python client (openai library), where the text is provided as input, and the model generates a response. This response is returned as a string containing the assistant's answer to the user’s query. The integration ensures that the assistant provides a conversational experience by leveraging the power of natural language processing.
4. **Receiving the Response and Converting It to Audio**: After receiving the response from ChatGPT, the text is passed to the **gTTS (Google Text-to-Speech)** library. The system generates an audio file from the text response, which is saved as an MP3 file. The gTTS library offers high-quality text-to-speech synthesis, ensuring that the assistant's reply is presented clearly and in a natural-sounding voice.
5. **Playing the Audio Response**: The generated audio file is then played back to the user through the Gradio interface. The Gradio component gr.Audio is used to facilitate the playback of the audio response. The system ensures that the user can hear the assistant's reply after each interaction, completing the voice-based conversation loop.

**Code Implementation**

Here is the Python code that was implemented to achieve the desired functionality:

import gradio as gr

import speech\_recognition as sr

from gtts import gTTS

import os

import openai

# Your OpenAI API key

openai.api\_key = 'YOUR\_OPENAI\_API\_KEY'

# Function to convert audio to text using SpeechRecognition

def transcribe\_audio(audio\_file):

recognizer = sr.Recognizer()

audio\_data = sr.AudioFile(audio\_file)

with audio\_data as source:

audio = recognizer.record(source)

try:

user\_text = recognizer.recognize\_google(audio)

print("🔹 Extracted text:", user\_text) # Print the extracted text

return user\_text

except sr.UnknownValueError:

print("🔹 Couldn't understand the audio")

return "I didn't understand what you said"

except sr.RequestError:

print("🔹 Error connecting to Google API")

return "Error connecting to the API"

# Function to interact with the ChatGPT API

def chat\_with\_gpt(user\_text):

response = openai.Completion.create(

model="gpt-3.5-turbo",

messages=[{"role": "user", "content": user\_text}],

)

ai\_response = response['choices'][0]['message']['content']

print("🔹 ChatGPT response:", ai\_response) # Print the response from ChatGPT

return ai\_response

# Function to convert text to speech using gTTS

def text\_to\_speech(ai\_response):

tts = gTTS(text=ai\_response, lang='ar')

audio\_file = "response.mp3"

tts.save(audio\_file)

print("🔹 Audio saved at:", audio\_file) # Print the audio file name

return audio\_file

# Main function to record audio, convert it to text, and send the text to ChatGPT

def voice\_chat(audio):

# Convert audio to text

user\_text = transcribe\_audio(audio)

# Send the text to ChatGPT and get the response

ai\_response = chat\_with\_gpt(user\_text)

# Convert the response to audio

response\_audio = text\_to\_speech(ai\_response)

return user\_text, response\_audio

# User interface using Gradio

text\_reply = gr.Textbox(label="ChatGPT Text")

voice\_reply = gr.Audio(label="Audio Response")

gr.Interface(

fn=voice\_chat,

inputs=[gr.Audio(source="microphone", type="filepath")], # Audio input from the microphone

outputs=[text\_reply, voice\_reply], # Outputs are text and audio

title='AI Voice Assistant with ChatGPT AI',

live=True

).launch(debug=True)

**Conclusion**

The AI Voice Assistant with ChatGPT integration fulfills all the required functionalities. The following points summarize the system's capabilities:

1. **Voice Input Recording**: The system successfully records the user's voice using Gradio's microphone input functionality.
2. **Audio to Text Conversion**: Speech recognition via the SpeechRecognition library accurately transcribes the recorded audio into text.
3. **Text Processing with ChatGPT**: The transcribed text is sent to the ChatGPT API, which returns a conversational response based on the input.
4. **Text to Speech Conversion**: The text response from ChatGPT is converted into speech using the gTTS library, ensuring a smooth voice output.
5. **Audio Playback**: The generated audio response is played back to the user through the Gradio interface, completing the interaction.

In summary, the task has been successfully implemented, and the system functions as expected, providing a fully interactive voice assistant that leverages state-of-the-art AI technologies for natural language processing and speech synthesis.