**Project Documentation**

**Project Title: AI Speech-to-Speech Chatbot Using Whisper and Gemini AI**

**Objective:**

The project allows users to interact with a voice-based AI assistant. The assistant transcribes voice input (speech), generates a response using Google's Gemini AI model, and outputs the response back as speech.

**Workflow of the Project**

The workflow involves the following steps:

1. **Audio Input (Recording):**
   * The program records live audio from the user through the microphone.
   * Recorded audio is saved as a .wav file.
2. **Speech-to-Text (Transcription):**
   * The recorded audio is fed to the **Whisper Model**, which converts speech to text.
3. **Text Processing (AI Response):**
   * The transcribed text is sent to Google's **Gemini 1.5 Flash** model, which processes the input and generates a relevant response.
4. **Text-to-Speech (TTS):**
   * The response text is converted back into speech using the **pyttsx3** library.
   * The response is played aloud to the user.
5. **Iteration (Loop):**
   * The program runs in a loop until the user decides to exit.

**Detailed Notes in Layman's Terms**

**Project Overview:**

This project is like a digital voice assistant that listens to your questions (like Siri or Alexa), understands them, and talks back with helpful answers.

**Components Used:**

1. **Whisper Model (from OpenAI):**
   * Converts your recorded voice into text.
   * Think of it as the "ears" of the assistant.
2. **Google Gemini AI (Gemini 1.5 Flash):**
   * Takes the converted text, understands the question/request, and generates a text-based response.
   * This is the "brain" of the assistant.
3. **Pyttsx3 Library:**
   * Converts the Gemini-generated response into speech and plays it aloud.
   * This is the "mouth" of the assistant.
4. **PyAudio Library:**
   * Records live audio input from the microphone.
   * Think of this as the "microphone" that listens.

**How Each Step Works:**

1. **Recording Audio:**
   * You speak into the microphone.
   * The PyAudio library records your voice for 5 seconds and saves it as a live\_audio.wav file.
2. **Transcribing Audio (Speech-to-Text):**
   * Whisper listens to the saved audio file and converts what you said into plain text.
   * Example: If you say "What is the weather today?", it outputs: What is the weather today?
3. **Processing the Text with Gemini AI:**
   * The program takes the transcribed text and sends it to the Gemini AI model.
   * Gemini processes the input and responds appropriately.
   * Example:  
     **Input:** "What is the weather today?"  
     **Output:** "I can't fetch live weather, but it's usually sunny around this time of year."
4. **Text-to-Speech:**
   * The program takes Gemini's response and uses pyttsx3 to "speak" it aloud.
   * You will hear the response as a voice.
5. **Loop & Exit Option:**
   * After completing the above steps, the program asks if you want to continue.
   * You can exit by typing n.

**5 W's (Interview Preparation)**

1. **What does this project do?**
   * This project transcribes live speech into text, processes it using Google's Gemini AI to generate a response, and converts that response back into speech.
2. **Why was this project built?**
   * To create an interactive voice-based AI assistant that can understand user queries and respond in a conversational manner, improving accessibility and user experience.
3. **Who is the target audience?**
   * Users who want a voice-based chatbot or interactive assistant for question-answering and information retrieval.
4. **Where can it be used?**
   * This project can be used in personal AI assistants, accessibility tools for individuals with typing difficulties, educational tools, and more.
5. **How does it work?**
   * The project works in the following steps:
     1. Record live speech.
     2. Convert speech to text using Whisper.
     3. Process the text input using Gemini AI.
     4. Convert the Gemini response back to speech using pyttsx3.
     5. Play the response aloud.

**Code Walkthrough Notes**

**Libraries Used:**

* Whisper: For speech-to-text transcription.
* pyttsx3: For text-to-speech conversion.
* PyAudio: For recording live audio.
* google.generativeai: To interact with Gemini 1.5 Flash.
* dotenv: To securely load API keys from a .env file.
* wave: To save the recorded audio in .wav format.

**Main Functions:**

1. **record\_audio(seconds)**
   * Records live audio for a given duration.
   * Saves the audio as a .wav file.
2. **transcribe\_audio(audio\_file)**
   * Uses Whisper to transcribe the recorded audio into text.
3. **get\_gpt\_response(text\_input)**
   * Sends the transcribed text to Gemini AI and retrieves the response.
4. **text\_to\_speech(text)**
   * Converts Gemini's response text into speech and plays it.
5. **main()**
   * Orchestrates the entire workflow in a loop.

**Example Workflow**

1. **User Action:** "Speak now!"  
   (You say: "What is 2 plus 2?")
2. **Speech-to-Text:**
   * Transcription: "What is 2 plus 2?"
3. **Gemini Response:**
   * "2 plus 2 equals 4."
4. **Text-to-Speech Output:**
   * Voice Response: "2 plus 2 equals 4."

**Improvements and Future Scope**

1. **Speech Duration Detection:**
   * Automatically stop recording when the user finishes speaking.
2. **Multiple Language Support:**
   * Add support for non-English speech recognition and response.
3. **Enhanced User Control:**
   * Allow voice commands to control the assistant (e.g., "Stop", "Repeat", etc.).
4. **Real-Time Processing:**
   * Reduce response latency for real-time applications.

**Whisper Model: Overview**

1. **What is Whisper?**
   * Whisper is a powerful **open-source automatic speech recognition (ASR)** model developed by **OpenAI**.
   * It is trained on **diverse audio data** to accurately transcribe speech into text.
2. **Why Whisper?**
   * It works well with various accents, noisy environments, and multilingual audio.
   * It is capable of handling **16 kHz audio** efficiently, making it ideal for real-time transcription.
3. **How does Whisper work?**
   * Whisper listens to an audio file (like the recorded live\_audio.wav) and breaks it into small audio chunks.
   * It processes these chunks and generates accurate text output.
   * **Example:** If you speak, "Hello, how are you?", Whisper outputs:  
     Hello, how are you?
4. **Versions of Whisper:**
   * The model comes in different sizes like tiny, base, small, medium, and large.
   * In this project, the **"base" version** is used for a balance between speed and accuracy.

**Gemini 1.5 Flash Model: Overview**

1. **What is Gemini?**
   * Gemini is a family of **Generative AI models** created by Google DeepMind.
   * **Gemini 1.5 Flash** is a lightweight and **faster variant** of the Gemini 1.5 series, optimized for quick responses.
2. **Why Gemini 1.5 Flash?**
   * It delivers fast and efficient **text generation** while maintaining accuracy.
   * It is ideal for **real-time applications** where low latency is critical.
   * It can handle a wide variety of tasks, including Q&A, summarization, reasoning, and conversational responses.
3. **Features of Gemini 1.5 Flash in This Project:**
   * **Input:** The transcribed text (e.g., "What is the capital of France?").
   * **Processing:** The model generates a coherent response (e.g., "The capital of France is Paris.").
   * **Output:** The response is sent back as plain text to be converted into speech.
   * Gemini's configuration includes:
     + **Temperature:** 1 (creativity in output).
     + **Top-p:** 0.95 (controls randomness).
     + **Max Tokens:** 8192 (maximum words it can generate).
4. **How Gemini Works in This Project:**
   * The transcribed speech (text input) is passed to the Gemini 1.5 Flash model via the **generate\_content** method.
   * Gemini processes the input and generates a concise, relevant response.
   * The response is then converted back into speech.

**Comparison between Whisper and Gemini:**

| **Aspect** | **Whisper** | **Gemini 1.5 Flash** |
| --- | --- | --- |
| **Purpose** | Speech-to-Text (ASR) | Text Generation / AI Response |
| **Developer** | OpenAI | Google DeepMind |
| **Input** | Audio File | Text Input |
| **Output** | Transcribed Text | Generated Text |
| **Usage in Project** | Converts audio into text | Processes text input to give responses |

By integrating **Whisper** for transcription and **Gemini 1.5 Flash** for text processing, this project combines cutting-edge AI tools to build a seamless voice assistant that listens, understands, and responds efficiently.