

Meeting Assistant POC

This is a proof-of-concept (POC) for a meeting assistant application that can capture system audio and microphone input, display audio waveforms, and perform real-time transcription using Whisper AI.

Features

- System audio capture using virtual audio devices (e.g., BlackHole)
- Microphone input capture
- Audio visualization with waveform display
- Real-time transcription using OpenAI's Whisper API
- Simple, intuitive user interface

Prerequisites

Before running this application, you'll need:

1. **Node.js and npm** - Version 14 or higher
2. **macOS** - This POC is designed for macOS
3. **BlackHole** - Virtual audio device for system audio capture
4. **OpenAI API Key** - For Whisper transcription

Setting Up BlackHole for Audio Capture

BlackHole is an open-source virtual audio driver that allows you to capture system audio on macOS.

Installation

1. Download BlackHole from Existential Audio or install via Homebrew:

```
brew install blackhole-2ch
```
2. Create a Multi-Output Device to route audio to both your speakers and BlackHole:
 - Open "Audio MIDI Setup" (search in Spotlight)
 - Click the "+" button in the bottom left corner and select "Create Multi-Output Device"
 - Check both your regular output device (e.g., speakers or headphones) and "BlackHole 2ch"
 - Optionally, rename this device to something like "Meeting Assistant Output"
3. Set the Multi-Output Device as your system's default output:
 - In System Preferences > Sound > Output, select your newly created Multi-Output Device

- Alternatively, click the volume icon in the menu bar while holding Option key, then select the Multi-Output Device

Now, any audio played on your system will be routed to both your speakers/headphones and the BlackHole virtual device, which our application can capture.

Installation

1. Clone or download this repository
2. Navigate to the project directory
3. Install dependencies:


```
npm install
```
4. Create a `.env` file based on the provided `.env.example`:


```
cp .env.example .env
```
5. Edit the `.env` file to add your OpenAI API key and configure other settings

Running the Application

Start the application with:

```
npm start
```

Using the Application

1. Select the appropriate audio device from the dropdown:
 - Choose “BlackHole 2ch” to capture system audio
 - Choose your microphone to capture microphone input
2. Click “Start Recording” to begin capturing audio
3. Play audio from a meeting or speak into your microphone
4. Click “Stop Recording” when finished
5. Click “Transcribe” to send the audio to OpenAI’s Whisper API for transcription
6. View the transcription results in the text area

Known Limitations

This POC has several limitations:

- Audio format conversion is simplified and may not handle all edge cases
- The waveform visualization is basic and may not accurately represent complex audio

- Transcription is performed after recording rather than in real-time streaming
- No speaker diarization (identifying different speakers)
- Limited error handling and recovery
- No persistent storage of transcriptions

Troubleshooting

No Audio Capture

- Ensure BlackHole is properly installed and configured
- Verify that your Multi-Output Device is set as the system's default output
- Check that you've selected "BlackHole 2ch" in the application's device dropdown
- Make sure your system is actually playing audio during recording

Transcription Issues

- Verify your OpenAI API key is correct in the `.env` file
- Check your internet connection
- Ensure the audio quality is sufficient for transcription
- For non-English content, consider specifying the language in the transcription settings

Next Steps

Future development could include:

- Real-time streaming transcription
- Speaker diarization
- Integration with meeting platforms (Teams, Google Meet, etc.)
- Improved audio processing and noise reduction
- Summarization of meeting content using OpenAI's GPT models
- iOS companion app integration

License

This project is licensed under the ISC License.