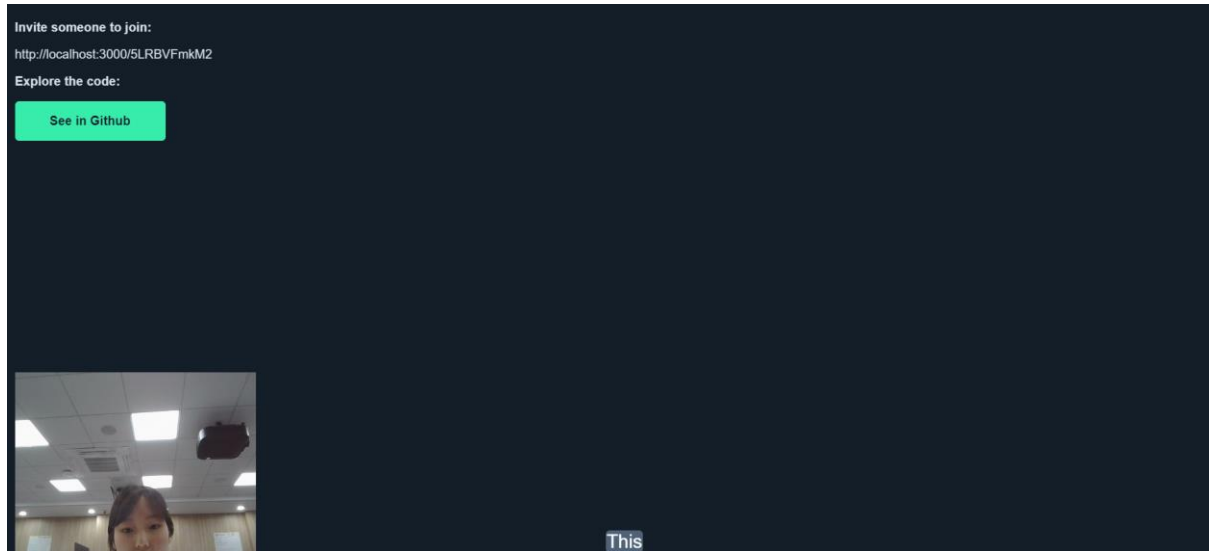


## [task – deepgram api의 video-chat 실행 및 분석]

프로젝트 주소 : <https://github.com/deepgram-devs/video-chat>

### [ 실행 결과 ]

Deepgram에서 제공하는 예제 프로젝트 video-chat을 실행해 본 결과.



실행 결과 발음 인식에는 큰 문제 없으나 속도에 약 2~3초간의 지연이 있음.

## [task – deepgram api의 video-chat 실행 및 분석]

### [ 코드 분석 ]

본 프로젝트는 Node.js의 express 환경에서 실시간 자막 기능을 구현하였음.

Server.js의 분석 결과는 다음과 같음.

1. .env 파일을 통해 API Key와 port와 같은 환경 변수를 관리함.
2. @deepgram/sdk패키지를 사용해 Deepgram API에 연결함.
3. Socket.io 패키지를 사용해 웹소켓 연결을 설정함.
4. *handle\_connection* 함수에서 클라이언트의 *join*이벤트를 처리해 그룹 콜을 설정함.
5. *setupWebRTCSignaling* 함수는 WebRTC signaling 메시지를 전달함
6. 실시간 음성 인식을 위해 *setupRealtimeTranscription* 함수가 호출됨. 해당 함수는 오디오 스트림을 Deepgram 서버로 전송하고, 처리된 텍스트 결과를 클라이언트로 보냄.
7. 서버는 웹소켓을 사용해 클라이언트와 통신하고 클라이언트는 웹소켓을 통해 오디오 스트림과 webRTC signaling을 서버로 전달함.
8. 서버는 웹 브라우저의 웹 소켓과 Deepgram API와의 통신을 담당함.

Video\_chat.js의 분석 결과는 다음과 같음

1. 웹 브라우저에서 비디오, 음성 텍스트를 표시할 수 있는 html 요소를 선택
2. Socket.IO를 사용해 웹소켓 연결 설정. 웹소켓을 통해 서버와 클라이언트가 실시간으로 통신함.
3. 사용자의 마이크와 카메라에 대한 액세스 권한을 요청하고, 사용 가능한 경우 로컬 비디오 스트림을 *localVideoNode*에 표시.
4. 방(Room)에 참여하고 서버와 연결된 Socket.IO 소켓을 생성
5. 그룹 콜에 참여한 사용자와 WebRTC통신을 설정. WebRTC를 사용해 피어 간 비디오 및 오디오 스트림을 전송.
6. 사용자의 음성을 실시간으로 텍스트로 변환해 표시하는 기능 설정. Deepgram API를 사용해 음성 인식 결과를 받아와 화면에 표시.
7. 사용자가 방을 나가면 피어 연결 해제.

## [task – deepgram api의 video-chat 실행 및 분석]

### [ 아키텍처 ]

본 예시는 다음과 같은 아키텍처를 가짐을 알 수 있음.

