1. 어떻게 webRTC와 연결을 해 줄까?

* 웹 알티씨 안에서 연결을 위한 URL이 필요함?
* 오디오 메시지 : 비정규화
* 알림 -> 메일로 참석자와 호스트에게 리마인드 할 수 있또록
* <word count> 를 사용하면 키워드를 뽑아낼 수 있을 것
* 수정하는 화면 필요할 듯?
* 채팅 필요할 듯.
* 실시간에 각각 다녀오는게 훨씬 낫다. 발화가 발생할 때마다 -> 처리 중 일 때는 모션 보여주기 return 돼도 모션 보여주기
* 시간 or 순서 -> 시간이 기준인 게 낫지 않을까? 대화의 텀이 길어진다면 중간에 텀이 있었다는 것을 알 수 있으니까. 누가 언제 무슨 얘기를 한 것이 대화에 있어서 중요한 것 같다. 시간을 어떻게 구별 할 것인가??? -> 둘다 쓰는 것
* Order만 수정할 때 쓰고, 타임 스탬프는 찍어주는 것이 중요함.
* 서버에 도달하는 시간을 타임스탬프에 찍히는 게 맞지 않을까요?? 나머지는 사용자가 회의록을 보고 수정을 할텐데 -> 그럼 이걸 쪼갤 수 없나요? (요구사항이 생길 수도 있다.)
* 나중에 유저가 많아지면 테이블이 필요할 듯.
* 실시간 성을 위해 성능을 높이려면 채팅으로 실시간 해보기 ?
* 브라우저에서 서버에 보낼 수 있는 요청이 최대 6개. 오디오 파일 보내는 것도 시간 걸리고, 채팅도 시간 걸리고, 그럼 동작들이 계속 밀린다. Timeout이 난다. 채팅은 웹 소켓으로 해라. 스톰프랑 웹 소켓으로 채팅을 구현하고, 오디오 파일도 보내고 나서 우리 서버에서 다시 API에 태워야하는데 오디오 전송은 전송대로 하시고, 얘는 return을 받아오고 끝내야하지 않을까. 오디오를 보내고 얘를 연결 유지된 상태로 서버에서 다시 API를 보내고 return을 받으려면 더 오래 걸리지 않을까?
* 보내고, 오디오 전송으로 끝내고, return을 받고, update가 되는 발화가 모인 창의 상태 ( 웹 소켓, 레디스) 를 활용하여 그것의 변화를 받아보는 작업이 완료 된 것의 처리도
* 따로 처리가 완료되면 처리하고,
* 프앤 – 서버 : 웹 소켓 , 서버 : 웹 소켓 + 상태를 기록하기 위한 ( 레디스 스트림 ? )
* 새로 들어온 사람과, 도중에 나가는 사람. 복구 로직이 있어야 유저 입장에서 부드럽다고 느낄 수 있다. 그게 들어가기 시작하면 난이도가 많이 올라갈 것.
* 단순히 API 호출로 하면 조금만 말이 많아져도 밀리는 현상이 발생할 것이다.
* 일단은 API로 구현을 하고, 채팅 같은 것을 웹 소켓으로 구현하고, 버전 업을 하면서 프로젝트를 진행하는 것이 좋을 것이다. 기록으로 과정을 남겨두자.
* 처음엔 이렇게 만들었고 -> 유저가 많아지니 이런 문제 발생 -> 이런 것을 도입 -> 해결 로 포트폴리오를 만들면 맛있을 것.