**팀 미 팅 회 의 록**

|  |  |  |  |
| --- | --- | --- | --- |
| **팀명** | YouHi | **차수** | 3차 |
| **일 시** | 2020년 4월 9일 금요일 18시 00분 – 20시 30분 ( 2시간 30분 ) | | |
| **장 소** | 각자 집 | | |
| **참석자** | 이태훈, 이인평, 김성수, 이주형, 김민재 | | |
| **불참자** |  | | |
| **안 건** | 소켓 통신을 이용한 백 엔드 구현 | | |
| **회의내용** | 1. 다중 접속자에 대한 소켓 서버의 작동 가능성  - 웹을 이용하는 다중 접속자들이 이용 가능한 소켓 서버를 구현해야 한다.  2. 동일한 파일에 대한 독립적인 중복 실행  - 다중 접속자들로 인한 동일한 파일의 중복 실행이 존재할 것이다.  - Javascript child\_process 모듈의 exec 함수를 통한 Python 파일의 독립적인 중복 실행이 가능한 것을 확인했다.  3. AWS Lambda 함수의 소켓 통신  - AWS Lambda 함수의 동영상 전처리가 끝난 후 소켓 서버에게 이벤트를 전달할 수 있어야 한다.  4. 동영상의 주체 확인  - 웹을 통해 동영상을 업로드한 주체와 AWS Lambda 함수가 전처리를 완료한 동영상의 주체가 동일한지 파악한 후 웹의 Filter 버튼을 활성화시켜야 한다. | | |
| **결과물** | 멀티 소켓 서버를 구현하여 다중 접속자들이 이용 가능하도록 하였다.  AWS Lambda 함수에서 동영상 전처리가 끝나면 멀티 소켓 서버에게 이벤트를 전달하는 Lambda 함수 내의 소켓 클라이언트를 구현하였다.  소켓 통신에서 client가 가지는 고유한 값인 clientID와 멀티 소켓 서버에서 client에게 부여하는 고유 번호를 통해 웹에 영상을 업로드한 주체와 AWS Lambda 함수가 전처리한 영상의 주체가 동일한지 판단할 수 있었다. | | |