2024.01.16. 팀 미팅 회의록

< 차별화 부분 >

* 입문자를 위한 상세한 설명 제공
* 아이디어 발산 기법의 구체적 제공

< 미팅 후, 내부 회의 후 결론 >

* DB관련: mongoDB, MySQL 사용 결정
* MSA는 초기 구현에서는 사용X, 추가구현에서 고민하기
* 뒤로가기 기능 등은 구현 관련 추가 자료조사 후, 더 고민 예정

< 참고도서 링크 >

[가상 면접 사례로 배우는 대규모 시스템 설계 기초 | 알렉스 쉬 | 인사이트 - 교보ebook (kyobobook.co.kr)](https://ebook-product.kyobobook.co.kr/dig/epd/ebook/E000003160871)

< 기술스택 관련 질의응답 >

1. 백: 저장하는 형태를 실시간으로 할 경우 DB를 쓰는 방향?

- 서로 실시간 편집을 위한 / WebSocket / NodeCast를 할 수 있는 장치가 필요하다 (수정사항에 대해서)

- 에디슨을 중간에? / 최종적으로 저장하는 장소를 mongoDB

- 충돌이 날 수 있지만 불가결하다. 줄일 수 있는 방향으로 개발해야한다.

- 어떤 기준으로 해야하는 가를 고민해야 한다.

- merge를 하는 알고리즘? TimeStamp를 활용하여 먼저 도달한 순서대로 수정을 한다?

- mongoDB 쓰는걸로!

1. 프론트

- 프론트 TypeScript 쓰는 것을 권장, 생산성 향상에 도움

- 잔기술을 쓸 수는 없지만 훨씬 안정적이다. (안정감)

- Redux 나쁘지않다.

- StoryBook = 컴포넌트 관리. 용이하다 공부추천

1. 그 외

- docker 필수

- ufw의 예외? iptype으로 관리하여 방화벽이 열려있으니 utf는 다시 방화벽을 닫는 용도

-unit test를 거친 후에야 merge를 할 수 있게끔

-CI/CD 해킹 조심!

< MSA 활용 관련 질의응답 >

* 어디에 부하가 생길지 생각해서 시도해보는 것도 괜찮을 듯
* 기능을 단위로 or 유저를 단위로 => 나누어볼 수 있음
* 우리 프로젝트에서는 사용하기 다소 애매함, 하나로 개발하고 전환 고려 추천

< 실시간, 편집기능 관련 >

* 뒤로가기 기능과 관련하여 고민할 필요 있음
* 크기, 좌표, 마지막 수정시간, 마지막 수정사람 => 상속 받는걸로 하면 어떨까
* json을 통째로 옮긴다고 생각하면 안될 것 같다.
* 타임스탬프에 맞춰 순서대로 처리해보자
* 이벤트 발생 이후 0.5초 ~ 이밴트 종료 이후 0.5초까지의 작업 넣기 등 정책적으로 정할 필요가 있음