



# Seagull

## 그래서 세 달 동안 뭐했어?

# CONTENTS

01

## 팀 소개

- 간단한 팀 소개

003

02

## 결과물

- 결과물 소개  
- 시연영상  
- 전체 아키텍처

004

03

## 시행착오

- 프로젝트 회고

007

04

## 목표달성

- 목표달성 여부

011

05

## 느낀점

016

# 2022 스마일게이트 윈터 데브캠프

*Seagull* 팀 입니다.

Smilegate®

팀장

박성준

back-end

github

<https://github.com/tjdwns4537>

팀원

박정원

front-end

github

<https://github.com/jjjjjeongwon>

팀원

이효승

back-end

github

<https://github.com/hoos007>

팀원

이범수

back-end

github

<https://github.com/dldks321>

## 결과물

## eGuBa가 뭐야?

eGuBa는 한 방에 들어온 여러명의 사람들이 함께 영상을 시청하고 채팅을 칠 수 있는 영상 싱크, 채팅 서비스 웹사이트 입니다.

## 01. 함께보기

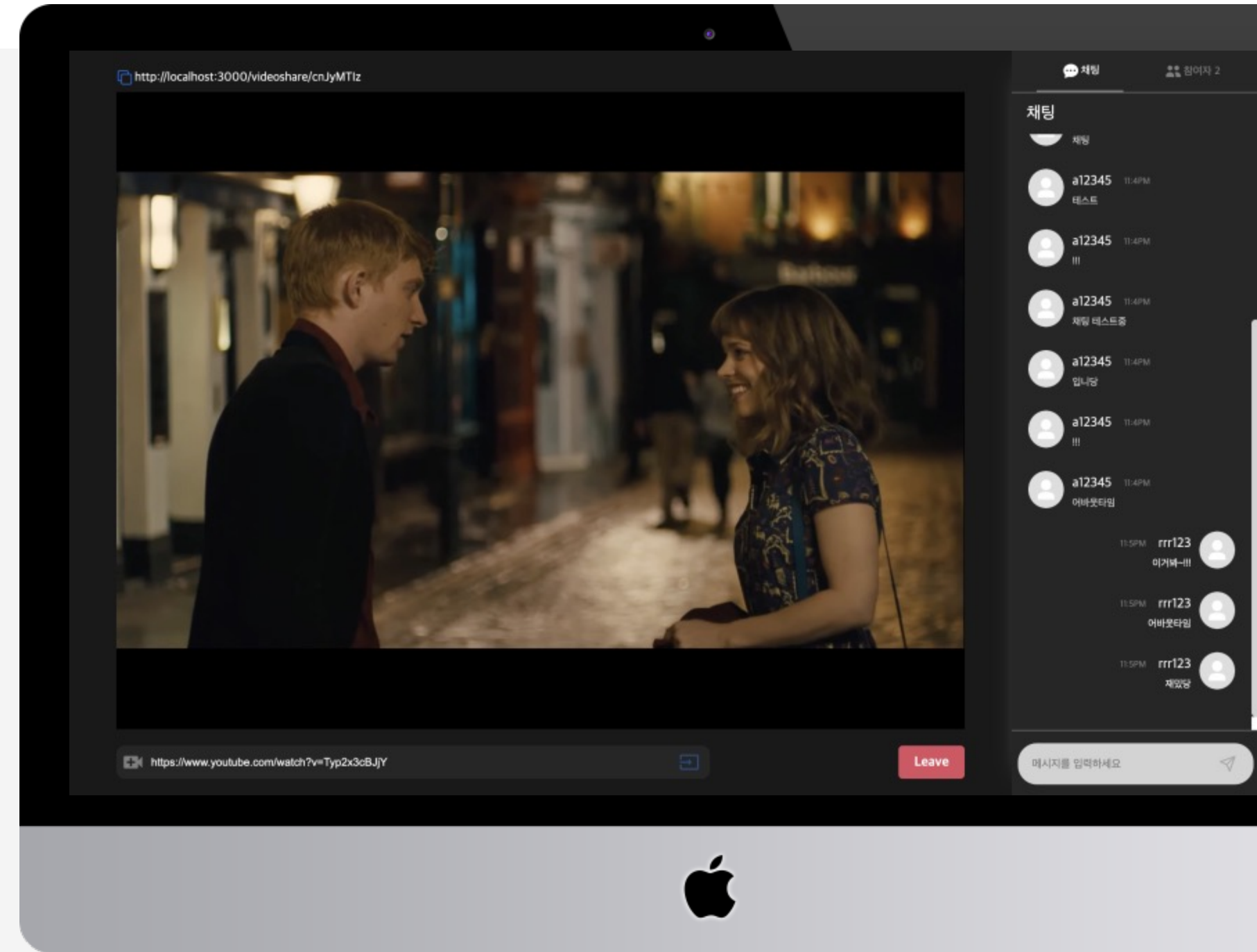
eGuBa는 방을 기준으로 모인 사용자들이 호스트가 재생하는 영상을 모두 같이 볼수있는 서비스를 제공합니다.

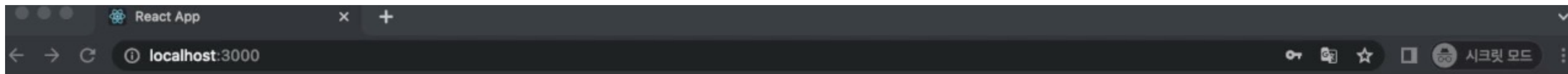
## 02. 실시간 채팅

eGuBa는 한 방에 모인 사람들이 실시간으로 채팅을 할 수 있는 서비스를 제공합니다.

## 03. 방 만들기

eGuBa는 사용자들이 모일 수 있는 사적인 방 서비스를 제공합니다.

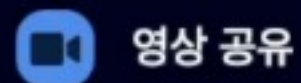




# eGuBa

[LOGIN](#) · [JOIN](#)

## 영상 공유를 위한 플랫폼



영상 공유



실시간 채팅



## eGuBa는 영상을 다같이 볼 수 있는 플랫폼입니다

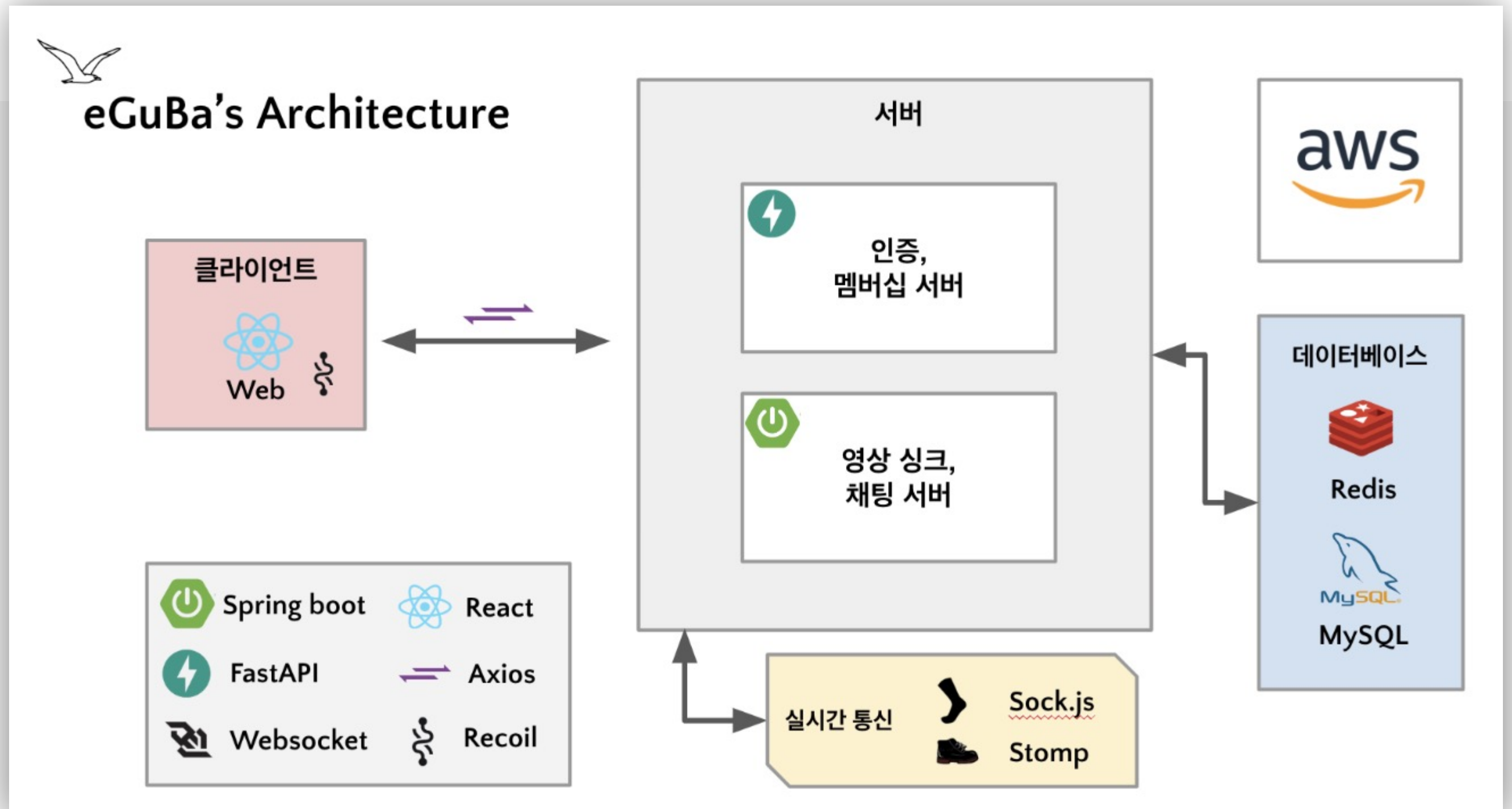
관심있는 유튜브 영상을 친구들과 함께 시청하고 공유방 내 채팅을 통해 실시간으로 소통해 보세요

[방만들기](#)[참여하기](#)

## 결과물

## 전체 아키텍처

eGuBa는 두개의 서버와 클라이언트로 이루어져 있고 실시간 통신을 위해 웹소켓을 이용했습니다.  
데이터베이스는 레디스와 MySQL을 사용하고 있습니다.





# 03

## 시행착오

### OVERVIEW

프로젝트개발중 발생한 여러 문제점과 분석 해결에 대한 챕터입니다.

## 시행착오

# 프로젝트 규모 축소

프로젝트를 진행하면서 발생했던 여러 시행착오들

## 문제상황. 프로젝트 규모를 축소하나 마냐에 대한 의견충돌 발생

중간발표 이후 본격적으로 구현작업에 들어가기 앞서 팀원 사이에서 프로젝트 규모를 축소하는 것이 어떤가 하는 의견이 있었습니다. 그것과 동시에 그대로 가자는 의견도 있었습니다.

## 해결. 회의를 통해 프로젝트 규모를 축소하기로 결정

회의를 통해 각자의 수준과 현재 가진 자원을 다시 점검했습니다. 그 결과 서버개발에 필요한 프레임워크를 처음 써보는 인원도 있었고 웹소켓 통신에 대해 잘 모르는 인원도 있었습니다. 초반에 아이디어를 정할때는 아이디어에 치중한 나머지 그 부분을 간과했습니다. 게다가 중간 발표때 프로젝트주제가 하나로 보이지 않는다는 피드백을 받아서 저희가 만들고 싶은 핵심부분만 두고 나머지는 과감하게 잘라냈습니다.

## 알게된 것. 새로운 아이디어도 중요하지만 완성하는것도 중요하다

프로그램을 만들때 무조건 좋은 아이디어만 있으면 되는 것이 아니고 나중에는 이게 기한내에 가능한지 현재 가진 자원은 어느정도인지도 잘 따져보고 스코프를 결정해야 한다는 것을 생각하게 되었습니다.



## 시행착오

## 영상 공유방식 선택

프로젝트를 진행하면서 발생했던 여러 시행착오들

### 문제상황. 영상을 모두가 함께 시청할 수 있도록 하는 기술을 구현

개발해야하는 서비스 내용중에 같은 방에 들어온 사람들이 서로 같은 영상을 보게 하는 기능을 구현해야 하는 상황이 발생했습니다. 여러가지 고민끝에 영상을 직접 웹소켓을 통해 공유하는 방식과 호스트의 조작 명령만 공유하고 각 클라이언트들의 영상 싱크만 맞추는 방식 두가지를 떠올렸고 그중 하나를 선택해야 하는 상황입니다. 우선 첫번째 방법은 확실하게 모든사람이 같은 영상을 볼 수 있다는 장점과 호스트 컴퓨터에 저장되어있는 영상파일도 공유할 수 있다는 장점이 있지만 영상파일을 직접 웹소켓 통신을 통해 내보내야 해서 많은 데이터 이동이 발생하는 문제가 있었습니다. 두번째 방법은 이동하는 데이터는 적지만 직접 다른 클라이언트들의 영상 싱크 관리를 해줘야하는데 그 과정에서 어쩔 수 없이 약간의 시간차이가 발생하는 단점이 있습니다.

### 해결. 영상을 직접 보내는 방법이 아니라 명령만 공유하고 싱크를 수시로 맞춰주는 방법을 선택

선택을 위해 서비스의 특징을 다시 검토했습니다. 저희의 서비스 내에서의 영상공유는 영상채팅을 위한 것이 아니고 영상 시청을 위한 것이었습니다. 따라서 실시간성보다는 재생품질이 중요했습니다. 그러기 위해서는 영상 화질에 제한이 있는 첫번째 방법은 좋지 않다고 판단했습니다. 저희는 두번째 방법을 선택했고 모든 사람이 같은 위치의 영상을 보게 하기위해 수시로 싱크를 맞출 수 있도록 로직을 구성하기로 했습니다.

### 알게된 것. 모두 가져갈 수 없으면 냉정하게 분석하고 빠르게 최선의 선택을 해야한다.

이 결정을 하면서 초반에는 같이 영상을 봐야하니까 당연히 실시간성이 중요하지 않을까 라는 생각을하고 첫번째 방법으로 진행하려고 했었습니다. 하지만 영상 품질에 제한이 있었고 실시간성과 영상 품질 사이에 많은 고민을 했습니다. 각각의 장점과 단점을 분석하고 내가 필요한 특징은 어떤것인가 고민하는 과정을 거치면서 한가지를 과감하게 포기하는것도 시간자원을 아끼고 좀더 최선의 선택을 하는것이구나 라는 것을 깨닫게 되었습니다. 만약 둘다 잡으려고 했다면 시간은 시간대로 지나가고 결국 손해를 보게될것이라고 생각합니다.

## 시행착오

## 영상 공유방 내 화면구성

프로젝트를 진행하면서 발생했던 여러 시행착오들

## 문제상황.

## 1. 영상을 집중해서 시청하기에 적합하지 않은 너무 밝은 배경의 밝기

밤에 어두운 환경에서 영상을 본 경험이 있던 팀원의 의견 "영상 배경화면이 너무 밝으니까 눈이 아프고 불편하더라!"

## 2. 채팅이 필요하지 않은 경우에도 화면의 약 20%를 차지하는 채팅창

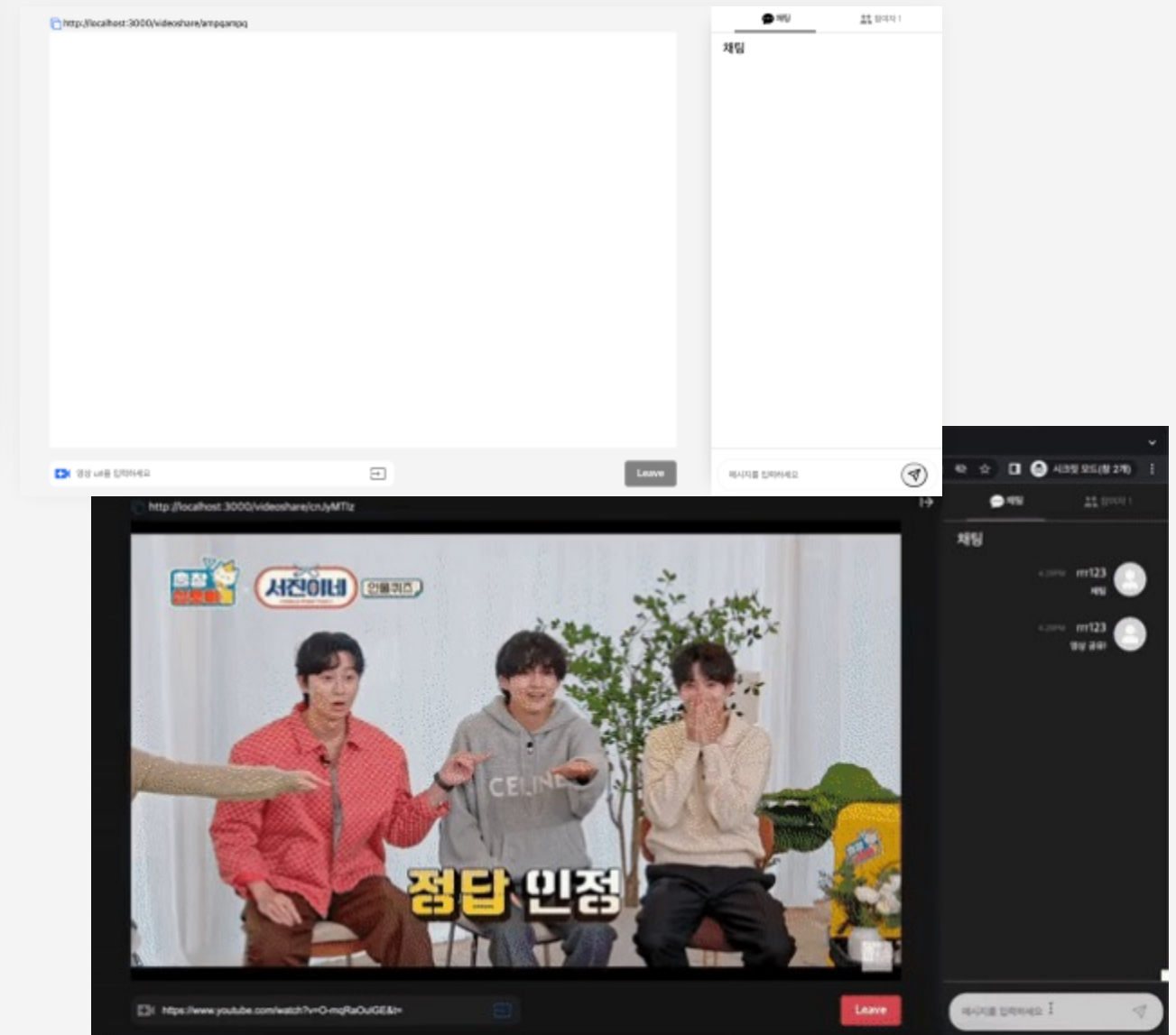
최종 시연영상을 찍고 팀원들과 확인하던 중 채팅창의 크기가 너무 크게 차지하다보니,  
막상 영상을 시청하는데에 집중이 안된다는 피드백

## 해결방안.

## 1. 영상 공유방을 어둡게 바꿔서 영상 시청에 집중할 수 있는 환경을 제공

## 2. 채팅창을 선택적으로 사용할 수 있도록 유동적인 뷰를 구현

## 알게된 것.

서비스를 이용하는 사용자의 입장에서 편리한 UI/UX를 고민하여 설계하는 것이  
실제로 이용하는 입장에서 크게 작용한다



# 04

## 목표달성

### OVERVIEW

프로젝트 초반에 정했던 목표달성여부와 그와 관련된 성취 내용을 보여주는 챕터

목표달성

## 박성준(BE)

PMP문서를 작성하며 정했던 목표를 달성했는지 여부와  
목표달성과 관련해 자신이 알게된 내용들

**01.** Stomp, Kafka 두 개로 채팅 서비스를 구현해봄으로써 각각의 장단점을 알 수 있었음

**02.** 방 링크에 따라 서로 다른 실시간 연결을 해주는 과정을 진행해봄으로써 브로드 캐스팅의 개념을 알 수 있었음

**03.** MySQL과 레디스의 사용 용도와 그 성능 차이를 알 수 있었음

**04.** 프론트와 직접 소통하며 웹 소켓 테스트를 해보면서 어떤 부분에서 트래픽이 발생할지에 대한 생각을 할 수 있었음

**목표달성**

# 박정원(FE)

PMP문서를 작성하며 정했던 목표를 달성했는지 여부와  
목표달성과 관련해 자신이 알게된 내용들

## 01. 생산성, 확장성을 확보할 수 있는 컴포넌트 설계하는 능력 기르기 (성공)

목표 달성을 위해 설계 단계에서 컴포넌트 분리에 집중한 결과,  
하나의 컴포넌트를 여러 페이지에서 재사용해볼 수 있었고 초반의 계획에서 수정되는 경우에도 유지보수에 용이함을 경험했습니다.

## 02. 협업 시 진행사항 공유를 위해 필요한 문서화 작업역량 기르기 (성공)

하루에 한번 개발 과정을 기록하지는 못했지만, 매주 개발일지 작성을 맡아 작성하고 코드리뷰 관련 내용을 꾸준히 정리하면서 문서화하는 습관을 들이고 문서화 작업 역량을 기르게 되었습니다.

## 03. 동적인 화면 구현 기술 익히기 (실패)

기본 기능 구현에 급급해 '모바일에서도 대응할 수 있는 반응형' 구현이라는 목표는 달성하지 못했지만 다양한 해상도로 이용하는 사용자의 입장에서 편리한 UI를 제공하기 위해 화면의 크기가 줄거나 늘었을 경우까지 생각하는 구현을 경험해봤다는 차원에서 목표에 근접해가는 방향으로 진행되었습니다.



## 목표달성

## 이효승(BE)

PMP문서를 작성하며 정했던 목표를 달성했는지 여부와  
목표달성과 관련해 자신이 알게된 내용들

### 01. 어떤 것이 좋고 나쁜지 판단하는 능력 기르기 - 나의 결정에 대한 근거 기록하기 (실패)

프로젝트 회의 진행중에 계속 따지걸고 왜 이렇게 생각하는지 질문하고 나의 생각을 말하면서 여러 생각을 듣고 같이 판단하는 기회를 많이 가졌습니다.  
그래서 세부목표달성은 실패했지만 근본적인 목표에 다가갈 수는 있었다고 생각합니다.

### 02. 서버 구조에 대한 전반적인 이해 - 구현 단계 이전에 스프링 프레임워크를 이용한 미니 프로젝트 1개 완성하기 (성공)

영상공유 로직을 만들기 위해서 간단한 웹소켓 서버를 만들었습니다. 이를 통해 스프링 프레임워크의 기본적인 사용법을 익히고 서버 프로그램의 간단한 구조와 웹소켓 통신을 서버에서 가능하게 하는 방법을 알게되었습니다.

### 03. 협업을 경험하고 노하우 알기 - 팀 워크플로우에 따른 단계별 결과 문서 내기 (실패)

세부목표를 달성하지는 못했지만 팀플경험이 많은 팀원들이 있어서 깃을 어떤식으로 사용하는지 api명세는 어떤식으로 만들어지는지 같은 여러 팀플관련 경험을 할 수 있었습니다.

**목표달성**

# 이범수(BE)

PMP문서를 작성하며 정했던 목표를 달성했는지 여부와  
목표달성과 관련해 자신이 알게된 내용들

**01. 프로그래머의 학습은 어떤 것을 배우기 위해 프로젝트를 하는 것이 아닌, 프로젝트를 진행하다가 부족한 점을 메꾸어 나가기 위해 틀을 학습하는 것이 중요하다는 것을 깨달음**

내가 특정한 개발환경을 사용하는 이유를 팀원들에게 설명하는 상황에 큰 도움이 되었음

**02. 반복문의 사용을 최소화하는 노력덕분에 순환복잡도를 최대한으로 줄일 수 있었음**

**03. 하지만 검증,기능 파트를 따로 분리하는 것에 실패하여 만족할만한 가독성을 갖추는 것에는 실패함**

# 05

## 느낀점

### OVERVIEW

프로젝트를 마무리하면서 느낀점을 보여주는 챗터

## 느낀점

## 최종 감상평

프로젝트를 마무리하면서 느낀점이 있다면



박성준

'어떤 방식으로 개발 공부를 해나가야 할 지에 대한 **방향성 확립**'

## 감상

- 스마일 게이트라는 IT회사에서 해주는 교육이 가장 도움이 많이 됐던거 같습니다.
- 학교나 인터넷 강의에서 볼 수 없는 보다 실무에 가까운 교육으로 어떤 방식으로 개발 공부를 해나가야 할 지 방향성을 잡을 수 있었습니다.



박정원

'무엇보다 가장 크게 얻어가는 **협업경험!**'

## 감상

- 팀 목표였던 '협업경험'이라는 목표를 가장 크게 달성한듯 합니다.
- 각자의 배경에서 공부하고 개발해온 팀원들이 모여 프로젝트를 통해 내 파트를 조금 더 쉽게 설명하고 설득하는 연습 그리고 소통하는 연습 등 협업경험 차원에서 많은걸 배웠습니다.



이효승

'**팀플레이의 가치**와 서버 개발자의 **기술범위**를 체험하는 좋은 시간이었습니다'

## 감상

- 다들 저보다 팀 프로젝트에대한 경험이 많다니 여러 노하우를 배울 수 있었고 또 진행하면서 서버 개발자들이 어떤 기술들을 사용하는지 간단하게 알 수 있었습니다.
- PMP문서를 작성하면서 개발자로서 성장하려면 어떤 생각을 해야하고 어떤식으로 행동해야하는지 많이 배웠습니다.



이범수

'**개발자로서 나아가야 할 방향성**을 잡을 수 있었습니다'

## 감상

- 단순히 특정 기술을 사용하는 것이 아닌 특정 기술을 언제 어떤식으로 사용해야 하는가에 대해서 배울 수 있었던 데브캠프였던것 같습니다.



*Seagull*

# THANK YOU.