



주성준

Back-End/Server Developer

2000. 09. 05

rdyjun00@gmail.com

010-6686-5836

github.com/rdyjun

rdyjun.github.io

함께 일할 때 더 강해지는 개발자입니다.

문제는 **여러 시각**에서 볼 때 더 선명해진다고 믿습니다.

실시간 매칭 구조의 문제를 해결하면서도 동료들과의 관점 차이를 토대로 더 나은 방식에 도달한 경험이 있습니다.

함께 성장하는 팀을 위해 지식을 공유하며, 사용자와 팀 모두에게 가치 있는 결과를 만드는 개발자가 되고자 합니다.

사용자 경험을 개선하는 개발자

실시간 처리가 중요한 환경에서 **확장성**과 **안정성**을 우선으로 설계하는 백엔드 개발자입니다.

대기열 기반 세션 매칭 구조 개선, 다중 인스턴스 환경 소켓 동기화, 무중단 배포 환경 구축 등

사용자 경험과 **운영 효율**을 높이기 위한 개선에 지속적으로 기여해왔습니다.

Experience

- 25. 03 ~ 26. 02 동양미래대학교 컴퓨터소프트웨어공학과 (학사과정)
- 24. 06 ~ 24. 12 네이버 부스트캠프 웹 · 모바일 9기
- 22. 03 ~ 24. 08 동양미래대학교 컴퓨터소프트웨어공학과 (전문학사과정)
- 20. 09 ~ 22. 03 군 복무
- 18. 11 ~ 20. 08 비개발 경력

Skills

- ⚡ Java / Node.js
- ⚡ Spring Boot / Nest.js / Express.js
- ⚡ MySQL / PostgreSQL / Redis
- ⚡ Docker / Nginx / AWS / OCI

동숲

2025. 03 ~

Spring Boot 3.4.3

Java 17

Nest.js 11.0.1

PostgreSQL 17

Redis

동양미래대학교 학생들의 교내 생활 편리성 증진을 목적으로 개발된 정보 제공 및 통합 커뮤니티 애플리케이션

GitHub: github.com/dongsooop

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

프로젝트 소개

누적 다운로드 수 290

스케줄링 기반 알림 서비스

- 평일 8-19시 사이 10분마다 업데이트되는 공지사항 내역을 각 사용자의 학과 소속에 따라 발송
- 평일 오전 8시 개인 일정과 학사 일정 통합 알림 및 요일별 시간표 알림 발송
- 알림 타입 기반 디바이스/사용자 별 개인 알림 제어센터 마련

교내 팀원 모집 커뮤니티

- 튜터링, 스터디, 프로젝트 카테고리에 맞는 팀원을 모집하고 참여
- FCM기반 지원 및 지원 결과 알림 발송

RAG 모델 및 GPT OSS를 활용한 챗봇 서비스

- 학교 질의에 대해 Ai의 실시간 응답
- RAG 모델이 실제 정보를 학습하고, GPT OSS가 자연어로 응답

실시간 이벤트 기반 과팅 서비스

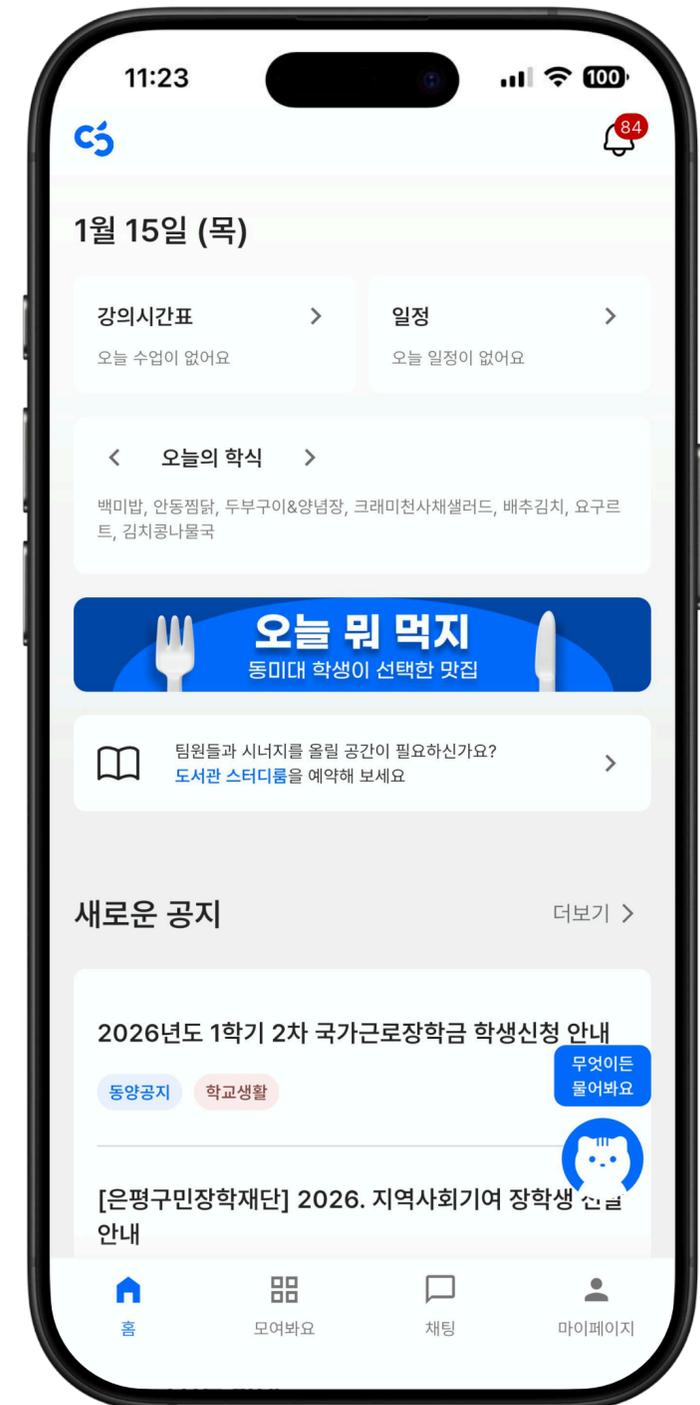
- 과팅 참여 시 대기열을 사용해 안전하게 세션 매칭
- 세션 종료 시점의 마음에 드는 사용자 선택 과정에서 서로를 선택한 사용자들은 1:1 채팅방 개설로 세션 이후에도 지속적인 채팅 지원

학교 주변 맛집 추천 커뮤니티

- 등록된 식당에 대해 사용자가 추천을 눌러 추천 수를 공유
- 앱 내 등록되지 않은 식당은 Kakao API를 통해 학교 주변 1km 이내 조회된 식당을 사용자가 등록 가능

Tesseract 모델을 활용한 이미지 분석 기반 시간표 자동 등록 서비스

- 학교 공식 서비스에서 제공하는 시간표 이미지 등록 시 시간표를 분석하여 데이터로 저장



동숲

2025. 03 ~

Spring Boot 3.4.3

Java 17

Nest.js 11.0.1

PostgreSQL 17

Redis

동양미래대학교 학생들의 교내 생활 편리성 증진을 목적으로 개발된 정보 제공 및 통합 커뮤니티 애플리케이션

GitHub: github.com/dongsooop

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

기술적 해결 과정

소셜 계정 연동 시 경쟁 조건

- 소셜 연동 중 회원 탈퇴 시 **탈퇴된 계정과 연동되는 문제**
- 동시에 동일한 소셜 계정 연동 시 중복 검증 후 저장 단계에서 실패하는 문제가 있었습니다.

SQL 의존적 + 락 적용

- 회원 탈퇴 문제를 방지하기 위해 회원에 락을 고려했고, 낙관적 락의 복잡도와 소셜 연동 및 회원 정보 수정이 빈번하지 않다는 특징에 따라 직관적인 **비관적 락**을 도입
- 저장되지 않은 행에 대해 락을 사용할 수 없기 때문에 불필요한 선 검증 구조를 **save() 실패 시 검증** 구조로 개선

과팅 스레드 점유

- 과팅 기능은 시나리오에 따라 메시지를 순차적으로 브로드캐스팅 하는 구조로 Spring Boot 서버에서는 **스레드 점유**로 이어질 수 있는 문제가 있었습니다.

Nest.js 프레임워크 사용

- **I/O 집약적**인 특징과, 스레드 점유를 방지하기 위해 **이벤트 루프** 기반 Nest.js를 도입했습니다.
- 유지보수성, 확장 가능성을 고려했을 때 일반 Node.js 나 Express.js보다 Nest.js로 일관된 구조에서 개발하는 것이 적합하다고 판단했습니다.

과팅 세션 입/퇴장 시 경쟁 조건 발생

- 여러 사용자가 동시에 매칭 요청을 보내면 세션 정원이 초과되었음에도 동일한 세션에 지정되는 **경쟁 조건**이 발생했습니다.

대기큐를 통한 동기적 처리

- **Redis Message Queue** 기반 **대기큐**를 구현하여 입장, 퇴장 이벤트를 순차적으로 처리하여 안정적으로 동시성을 보장하였습니다.

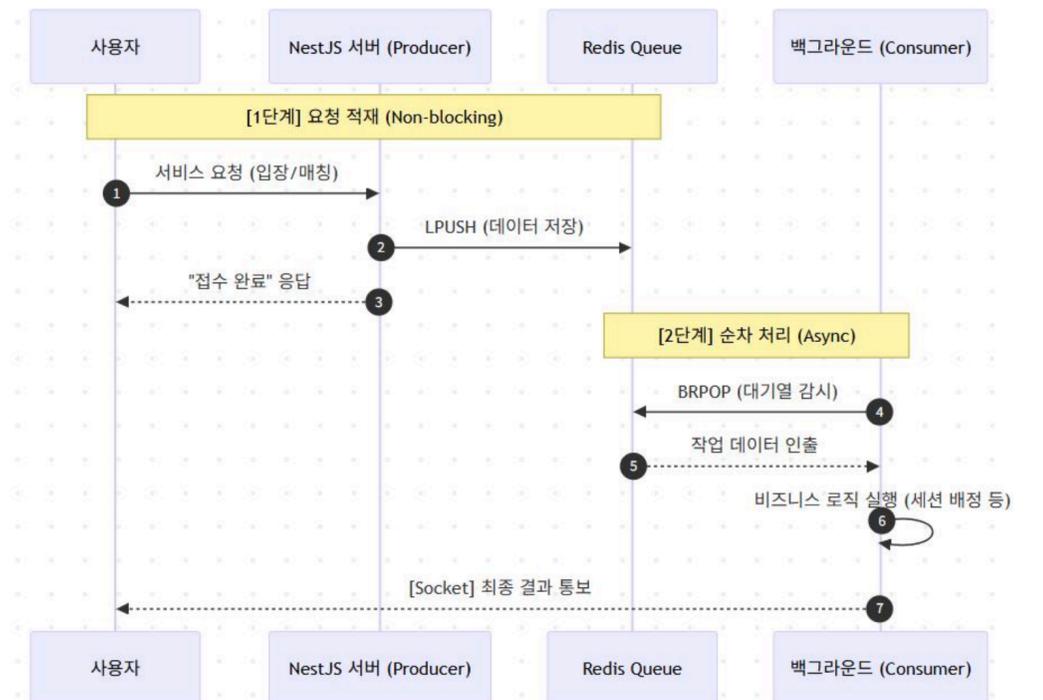
코드 일관성 부족

- 협업 시 코드 스타일이 일관되지 않거나, 누락된 예외 케이스로 인해 **피드백에 리소스를 많이 할당**해야 하는 문제가 발생했습니다.

PR Approve 필수 및 Ai 리뷰어 도입

- Pull Request를 통해 팀원의 **Approve**를 받아야만 main 브랜치에 병합하도록 팀원과 협의했습니다.
- 피드백에 할당되는 리소스를 줄이기 위해 GitHub에서 지원하는 **Copilot**을 통해 피드백을 받도록 했습니다.
- Copilot의 불필요한 피드백과 가독성 문제로 팀원과 회의를 통해 한글 지원 및 피드백 중요도에 따른 시각 요소가 추가된 Code Rabbit Ai를 도입하기로 결정했습니다.
 - **피드백 번역**과 중요도에 따른 **작업 우선순위**를 나누는 리소스가 줄었습니다.

Nest.js 대기큐 흐름



동숲

2025. 03 ~

Spring Boot 3.4.3

Java 17

Nest.js 11.0.1

PostgreSQL 17

Redis

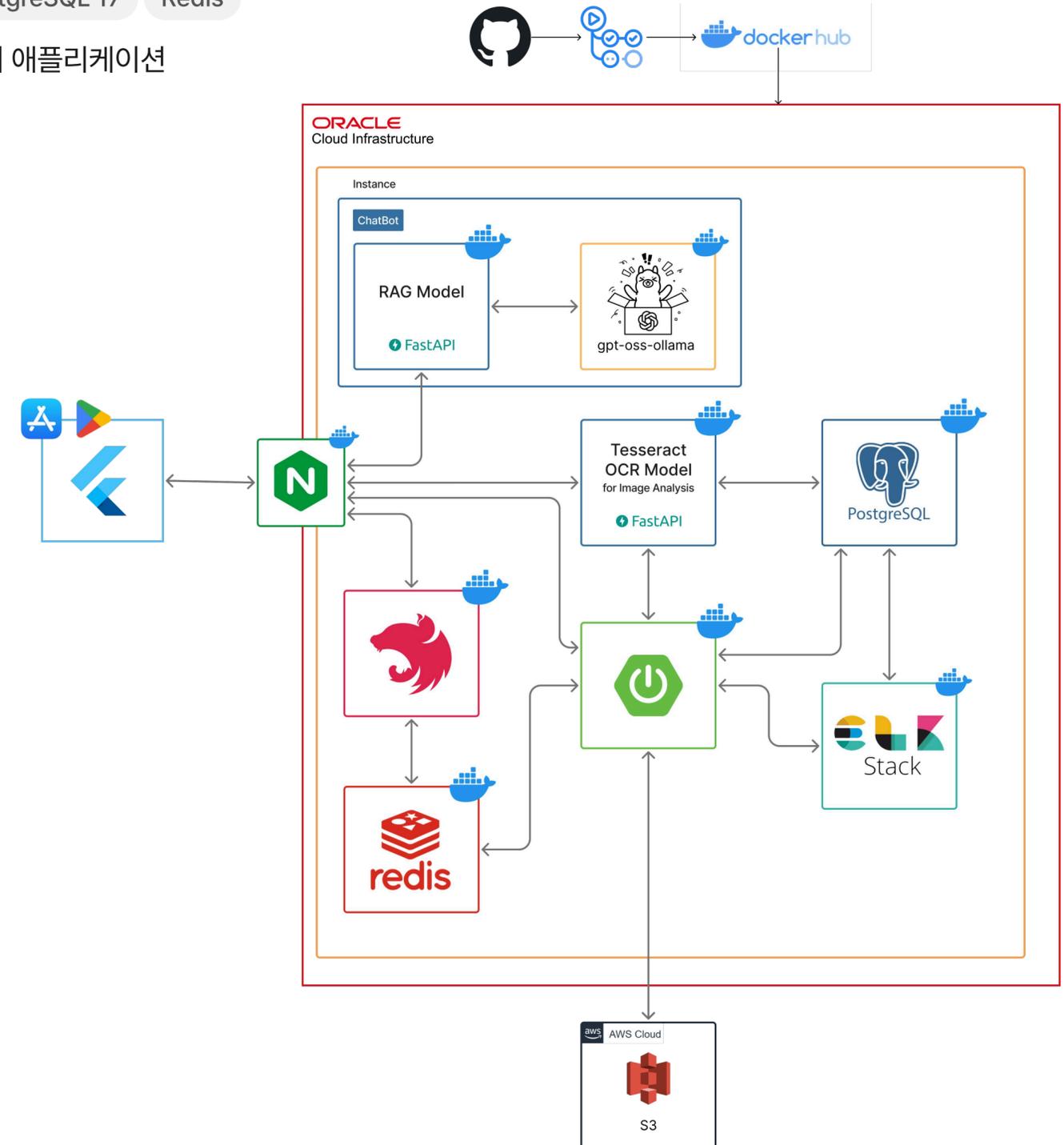
동양미래대학교 학생들의 교내 생활 편리성 증진을 목적으로 개발된 정보 제공 및 통합 커뮤니티 애플리케이션

GitHub: github.com/dongsooop

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

기술 선택 시스템 아키텍처

- 실시간 이벤트 처리를 핵심으로 하는 과팅 기능을 Nest.js로 도입
- Docker를 활용한 환경 설정 시간 단축
- Nginx 라우팅 및 Docker 컨테이너 기반 무중단 배포 도입
- CPU 집약적 작업과 동적 쿼리 기반 데이터 정합성을 위해 메인 서버로 Spring Boot 선택
- I/O 집약적인 작업 처리를 위해 과팅은 Nest.js로 분리
- 실시간 세션 진행을 위해 세션, 회원, 소켓 데이터를 Redis에 저장
 - 만료 기간 설정 및 세션 종료 시 메모리에 저장된 데이터 삭제
- 게시판 검색 기능을 위해 Elastic Search로 효율적인 다중 테이블의 행 조회
- 학사 데이터 학습을 위한 RAG 모델 컨테이너와 학습한 데이터 기반 자연어 응답을 위한 GPT OSS 컨테이너가 상호작용하여, 사용자 질의에 대한 응답 생성
- Tesseract Model로 사용자의 시간표 이미지 분석 및 데이터로 변환
- OS 친화적인 Flutter 기술을 사용하여 iOS, Flutter 개발



inear 2024. 10 ~ 2024. 12

Nest.js 10.4.6 MySQL Redis

오직 음악으로만 소통할 수 있는 공간을 필요로 하는 아티스트와 팬들을 위한 실시간 앨범 감상 서비스

GitHub: github.com/boostcampwm-2024/web18-inear

Refactor-GitHub: github.com/boostcampwm-2024/refactor-web18-inear

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

프로젝트 소개

실시간 음악 스트리밍 서비스

- 등록된 음원을 HLS 방식으로 사용자에게 스트리밍
- 앨범 내 음원을 순차적으로 재생

종료된 세션에 대한 앨범 수록곡 조회 게시판

- 세션이 끝난 앨범에 대해 앨범 수록곡 조회
- 댓글을 통해 앨범에 대한 개인의 감정을 공유

앨범 등록

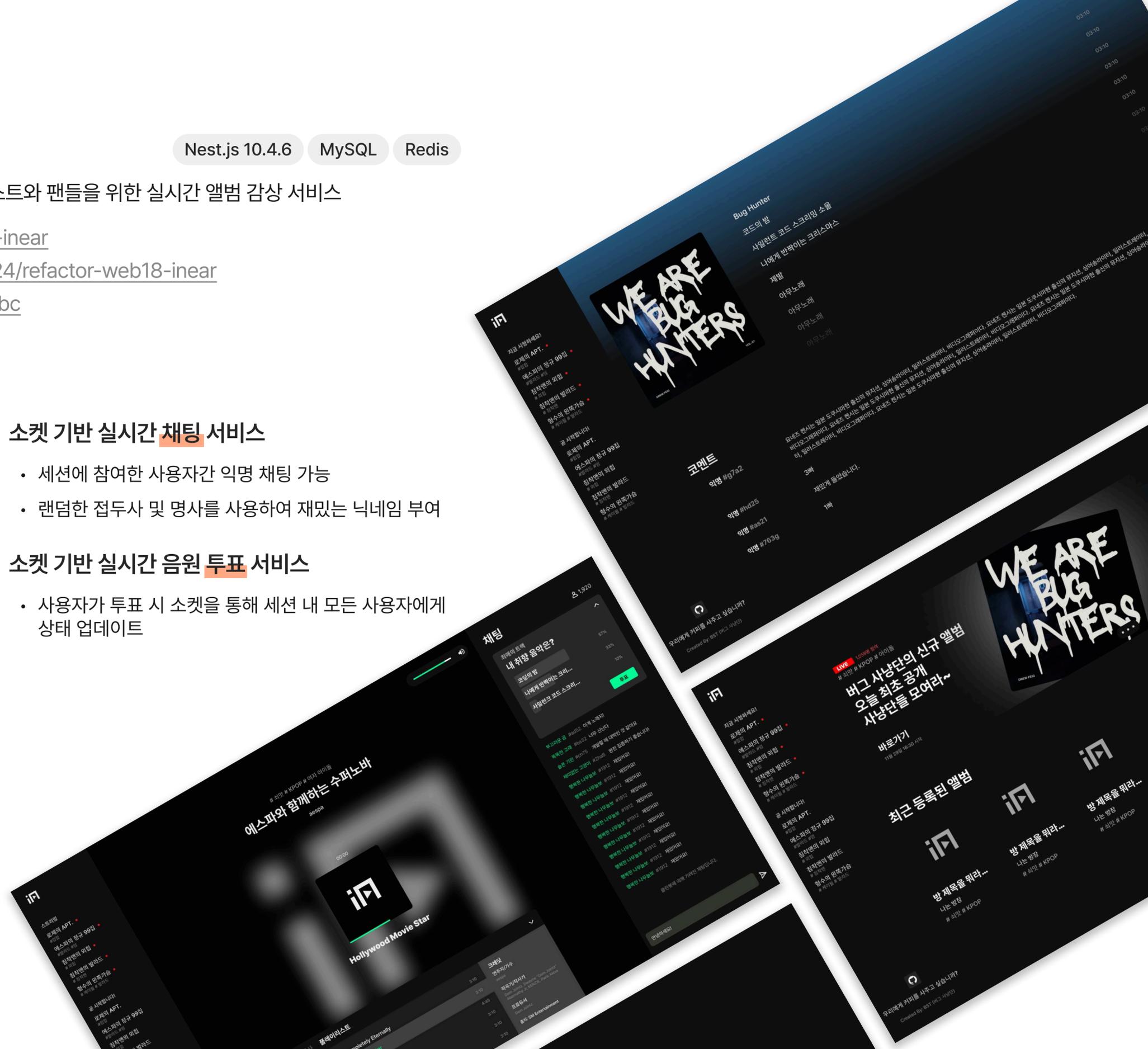
- 관리자와 컨택을 통해 앨범 단위의 음악 등록
- 앨범별 이미지 등록 서비스
- 원하는 스트리밍 시작 시간 등록 시 해당 시간 자동 스트리밍

소켓 기반 실시간 채팅 서비스

- 세션에 참여한 사용자간 익명 채팅 가능
- 랜덤한 접두사 및 명사를 사용하여 재밌는 닉네임 부여

소켓 기반 실시간 음원 투표 서비스

- 사용자가 투표 시 소켓을 통해 세션 내 모든 사용자에게 상태 업데이트



오직 음악으로만 소통할 수 있는 공간을 필요로 하는 아티스트와 팬들을 위한 실시간 앨범 감상 서비스

GitHub: github.com/boostcampwm-2024/web18-inear

Refactor-GitHub: github.com/boostcampwm-2024/refactor-web18-inear

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

기술적 해결 과정

Object Storage 과금

- 재생 목록에 따라 사용자별로 2초 단위 세그먼트를 Object Storage에 계속 요청하는 구조에서 과금이 발생하는 문제가 있었습니다.

CDN 도입

- 음원 세그먼트는 정적 파일이라는 특징에서 캐싱을 통해 요청 횟수를 줄일 수 있다고 판단했습니다.
- 서버에서 캐싱할 경우 요청 처리 과정에서 부하와 지연 문제를 고려해 CDN을 도입해 사용자 주변 지역에서 캐싱된 세그먼트를 지연 없이 수신하고, 서버 부하를 줄일 수 있었습니다.

스트리밍 구간 미일치

- 사용자가 진행중인 세션 도중에 입장할 경우 음원이 처음부터 재생되는 문제가 있었습니다.

재생 목록 실시간 편집

- 사용자가 접속한 시점에 맞춰 재생 목록을 실시간으로 편집하여 제공했습니다. 이미 지나간 구간은 제외하고 현재 시점부터 바로 재생되도록 가공해, 불필요한 대기 시간을 없앴습니다.

소켓 서버 귀속 문제

- 멀티 인스턴스 상황에서 소켓 서버가 각 인스턴스에 귀속되는 문제가 발생했습니다.

Redis 기반 소켓 동기화

- Redis-Adapter를 도입하여 각 인스턴스의 소켓을 redis 서버에 동기화했습니다. 통합 테스트를 통해 실제 두 개의 인스턴스에서 이벤트를 안정적으로 처리하는 것을 확인하였습니다.

업무 불투명

- 팀원 간 숙련도 차이로 인해 작업 속도가 상이했고, 이로 인해 전체적인 업무 진척도와 잔여 과업에 대한 가시성이 확보되지 않아 협업의 병목이 발생하는 문제가 있었습니다.

데일리 스크럼

- 매일 아침 데일리 스크럼을 통해 전일 업무 회고와 당일 과업을 공유했습니다. 팀원 간 일정을 세밀하게 조정하고 역할을 분배하여, 공백 없는 프로젝트 진행과 일정 준수를 이끌어냈습니다.

동시 배포로 인한 배포 지연

- 모노레포 구조로 백엔드 코드 배포 시 프론트엔드 코드도 재배포 되어 불필요하게 배포가 오래 걸리는 문제가 발생했습니다.

배포 파이프라인 분리

- GitHub Actions에서 git diff를 활용해 변경된 파일을 감지하고, 수정된 영역에 따라 배포 프로세스를 분리하여 배포 시간을 37.5% 단축했습니다.

용량 부족으로 인한 무중단 배포 충돌

- Docker Image 용량 증가로 인해 블루-그린을 위한 새 이미지를 다운 받을 수 없었습니다.

개선점

- 빌드 산출물만 이미지를 구성하도록 Docker Multi-Stage를 적용해, 빌드용 스테이지와 실행 스테이지를 분리하고 최종 이미지에는 실행 환경만 남도록 최적화해서 33% 경량화했습니다.

inear 2024. 10 ~ 2424. 12

Nest.js 10.4.6 MySQL Redis

오직 음악으로만 소통할 수 있는 공간을 필요로 하는 아티스트와 팬들을 위한 실시간 앨범 감상 서비스

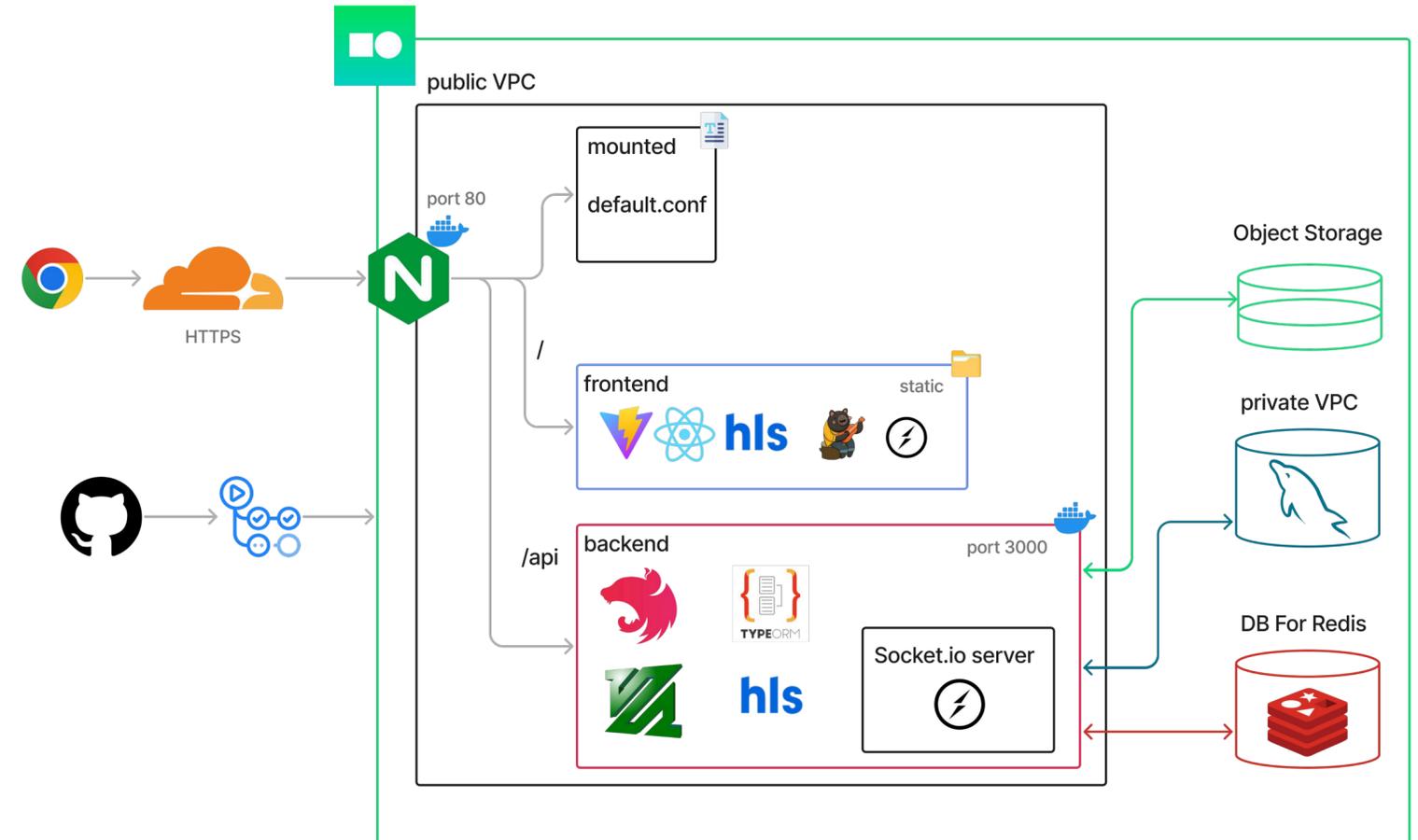
GitHub: github.com/boostcampwm-2024/web18-inear

Refactor-GitHub: github.com/boostcampwm-2024/refactor-web18-inear

ERD: dbdiagram.io/d/69282691a0c4ebcc2bf923bc

기술 선택 시스템 아키텍처

- 실시간 사용자 소통 서비스를 위해 Nest.js 도입
- Docker를 활용해 용량 부족 문제 및 빌드에 리소스를 Git Actions에 위임
- Nginx 라우팅 및 Docker 컨테이너 기반 무중단 배포 도입
- 앨범 이미지 및 음원 세그먼트 저장을 위한 Object Storage 도입
- 앨범 정보 저장을 위해 MySQL 도입
- 세션 입장 시 세션 진행도 및 정보를 받아오기 위해 Redis 도입
- 음악의 품질과 플랫폼(웹, 모바일)에 자유롭다는 특징에서 HLS 도입
- HLS 포맷에 맞는 음원 파싱을 위한 FFmpeg 도입



네이버 부스트캠프

2024. 06 ~ 2024. 12

Node.js

Typescript

TypeORM

웹·모바일 9기 | CS 기반 문제 해결 과정을 통한 지속 가능한 개발자로서의 성장 습관 습득

Site: boostcamp.connect.or.kr

GitHub: github.com/boostcampwm-2024

주요 활동

피어세션

- 매일 아침 미션 해결 과정을 공유하며 각자 설계한 구조의 특징을 분석하는 시간을 가졌습니다.
- 생각하지 못했던 구조를 간접적으로 경험해볼 수 있는 경험이었으며, 구현 과정에서 제대로 이해하지 못한 내용도 복기할 수 있는 시간으로 활용했습니다.

데모데이

- 프로젝트 시작 이후 매주 금요일마다 데모데이를 진행했고, 다른 팀의 진행상황을 참고할 수 있었기에 장황한 기획과 타겟층을 더 좁은 범위로 수정하고, MVP를 개발하는데 큰 도움이 되었습니다.

데일리 스크럼

- 프로젝트 기간 내 매일 아침마다 팀 내 진행상황을 공유하며, 개발 스프린트를 점검했습니다.
- 이슈나 새로운 학습정리 내용을 공유하여, 각자의 개발 과정에서의 빈틈을 최소화하고자 했습니다.

페어 프로그래밍

- 드라이버로서는 합의된 설계·방향성을 바탕으로 미션 구현에 집중해 코드 품질을 안정적으로 끌어올렸습니다.
- 내비게이터로서는 흐름과 설계 의도를 지속적으로 점검하고, 대안을 제시해 구현 방향을 빠르게 정렬함으로써 개발 속도와 효율을 높였습니다.

멘토링

- 미션 구현 과정에서 지속적으로 멘토링을 받아, 구조 개선과 코드 품질에 대한 피드백을 받았습니다.

주간 회고

- 매주 금요일마다 그룹원들과 한 주 회고를 진행했습니다. 회고에서는 각자의 참여 방식과 협업 태도를 구체적으로 피드백하고, 부족하다고 느낀 부분은 개선안이나 새로운 진행 방식을 함께 제안했습니다.
- 한 번은 제 **소극적인 참여**에 대해 스스로 회고하고 이후 회고와 작업 과정에서 **적극적으로 경청하며 질문을 늘리는 방식**으로 바로 행동을 바꿨습니다. 그 결과 참여도가 개선됐다는 긍정적인 피드백을 받았습니다.

데일리 스크럼 기록

프로젝트 노션 페이지 내 데일리 스크럼 페이지 중

Done	ToDo	Problem
<ul style="list-style-type: none">FE 현재 방 인원 수 표시메인 페이지 아래 상세 페이지의 목록게시판 댓글 CRUD(후 순위)앨범 상세지금 현재 에러채팅창에 !!!!!!!!!!!!!!! 많이 넣으면 넣은 만큼 화면 커징 - 채팅FE 투표 - 상준admin 테스트코드 마무리 - 윤서플레이리스트 정보 연동스트리밍 페이지 불륨 추가	<ul style="list-style-type: none">투표 중복 - 준Q/A<ul style="list-style-type: none">세부적인 에러 현상 - all기능 디테일한 부분 - 윤서대기화면 - 채기본 배너 - 준백그라운드 오류 - 온태최근 등록된 앨범 메인에는 6개만 - 6개만 올라주세요 온태관리자페이지 업로드 완료표시 - 채노래 글자면 메인페이지로 넘어가게 구현 - 채스트리밍페이지 앨범 데이터 안 바뀜 - 채시간 관점면 관리자페이지 관련에 글자수 제한 해두기 - 채채팅 글자수 제한 - 채FE 게시판발표 준비위키 및 링크의 정리	

주간 회고 피드백

boost course 멤버십 과정 (week4) 스터디 그룹 피드백

2개밖에 못 고르다니 아쉽네요. 성준님은 네 가지 부분에 있어서 다 잘하고 계신 것 같습니다. 사실 저번주 그룹회고 때 더 적극적으로 임해주시면 좋을 것 같다고 피드백드리려고 했습니다. 그런데 이미 성준님께서 인지하고 계시더라고요. 셀프 회고로 더 말을 많이 하겠다고 하셨고, 실제로 더 적극적으로 활동해주셨습니다. 특히 슬랙 그룹 DM에서 트러블 슈팅에 대해 이야기한 것, 정말 멋졌습니다. 멘토님이 계셔서 뭔가 말을 하기 망설였는데, 왜 안 쓰고 있었는지 조금 후회가 되네요. 성준님 덕분에 다음 그룹 활동에서는 슬랙 DM도 적극 활용해보고자 합니다. 감사합니다.