

지역 광주

팀명 codeRunner



이름 박가인, 장소명



Smilegate®

CONTENTS



- 
- 
- 01. 프로젝트 소개
 - 02. 역할 소개 및 이슈 해결 과정
 - 03. 프로젝트 리뷰



1. 프로젝트 소개



Slack Clone Project

프로젝트 소개



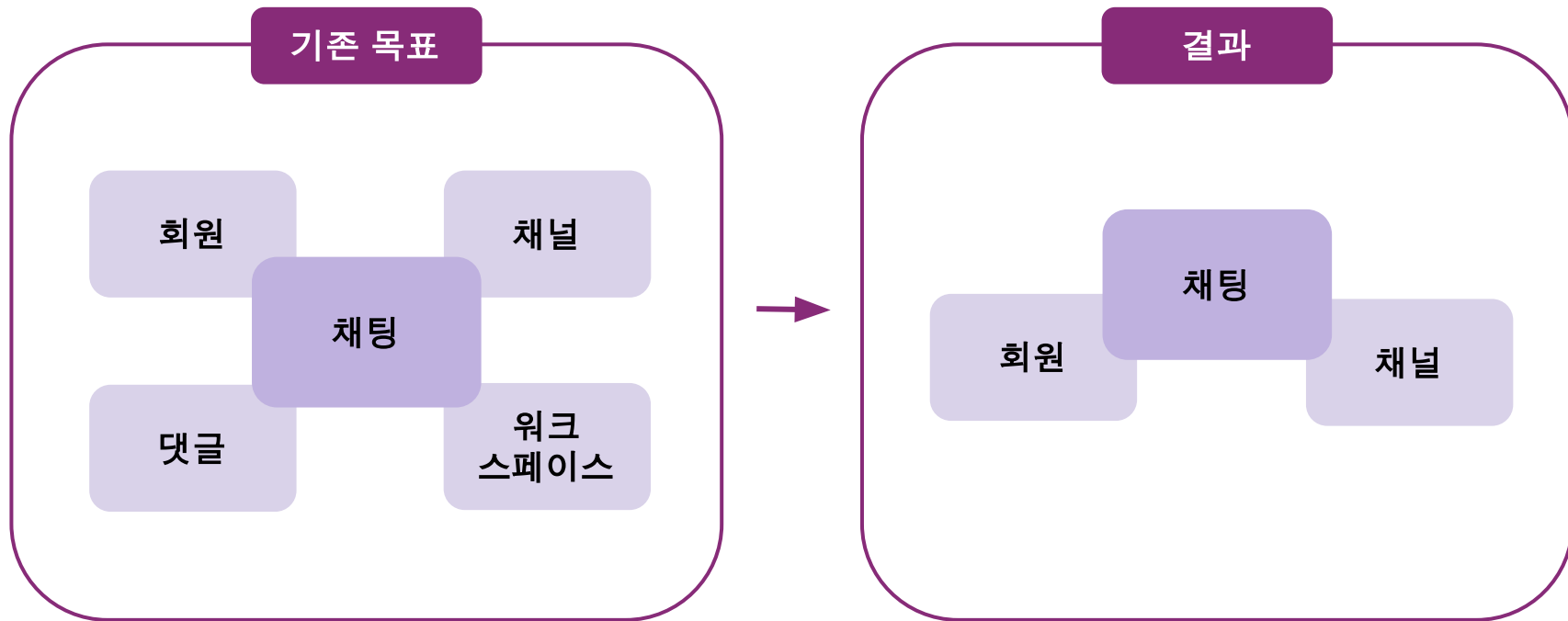
선정 이유

1. 팀원 모두에게 익숙한 플랫폼
2. 한 프로젝트에서 다양한 경험 가능 (EX. Socket 통신, MSA 구조)



Slack Clone Project – 주요 기능

프로젝트 소개



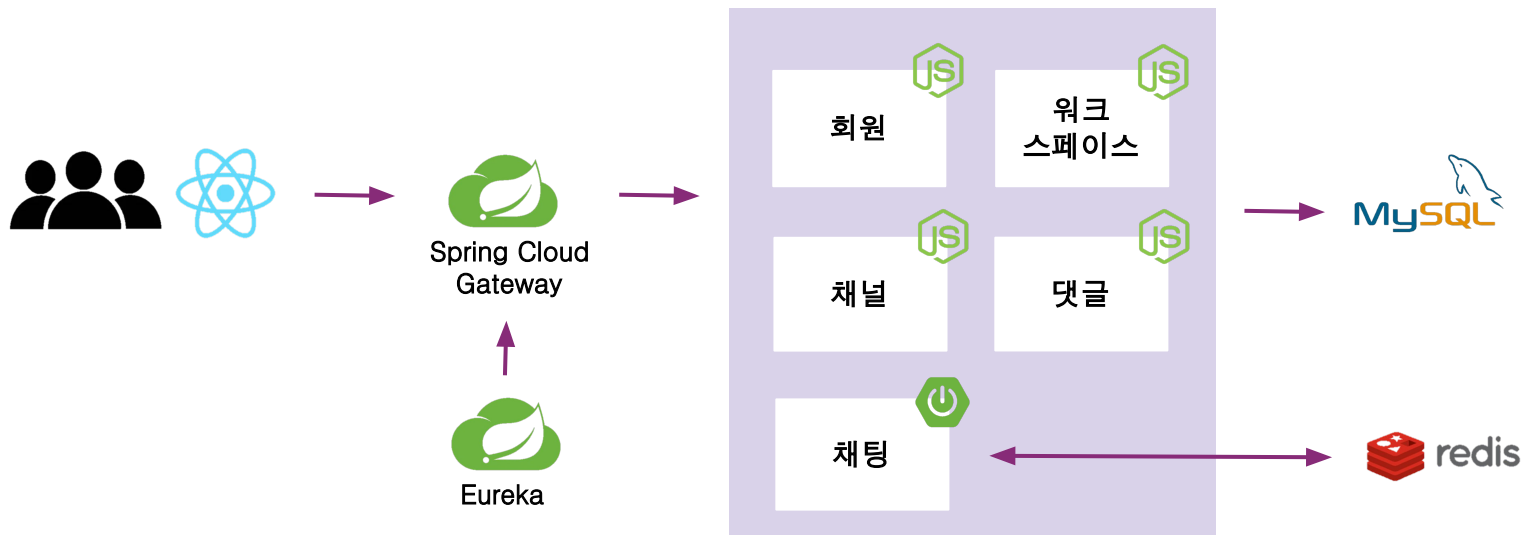


Slack Clone Project – 아키텍처

프로젝트 소개



기존 목표



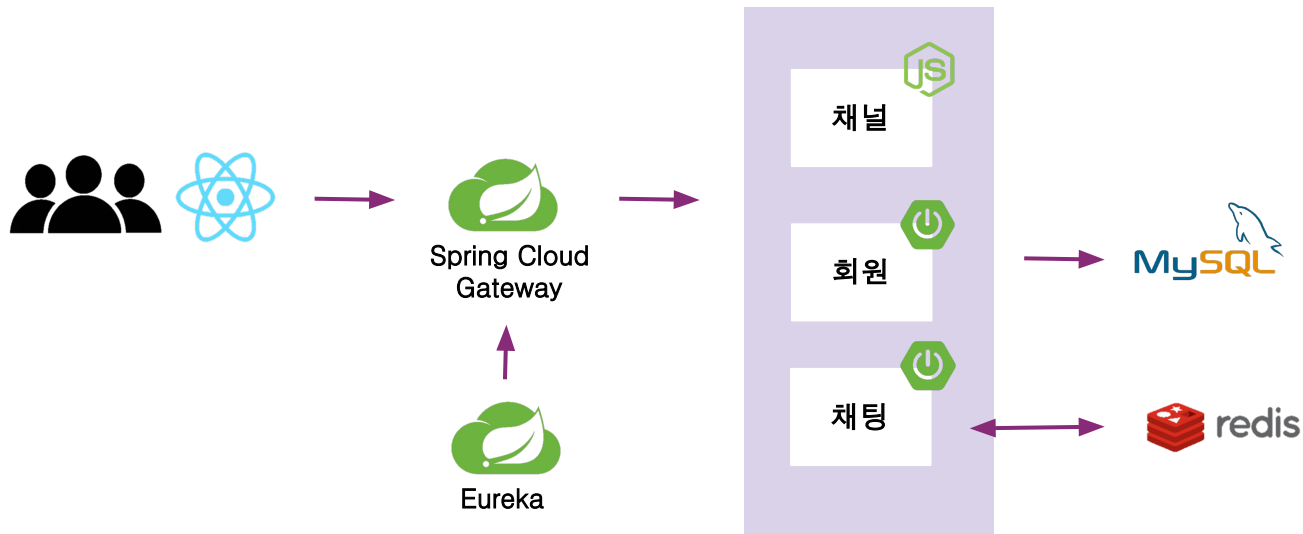


Slack Clone Project – 아키텍처

프로젝트 소개



결과



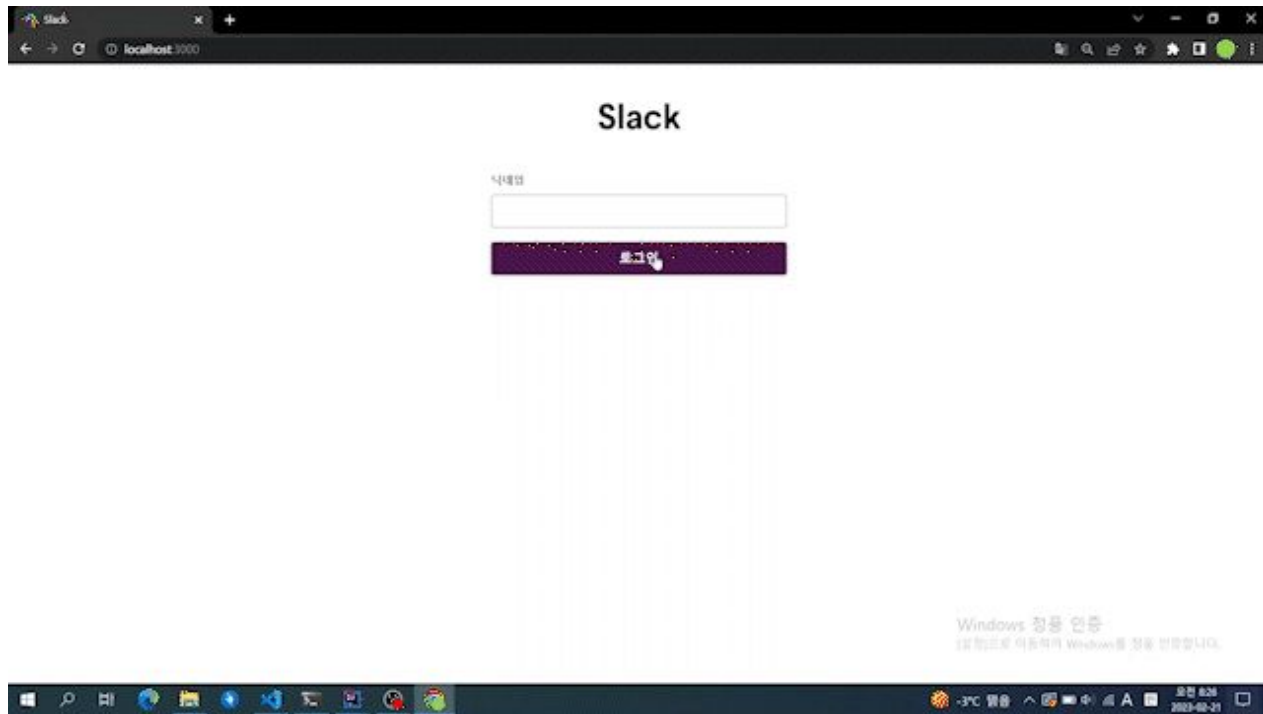


Slack Clone Project – 결과물

프로젝트 소개



로그인 페이지



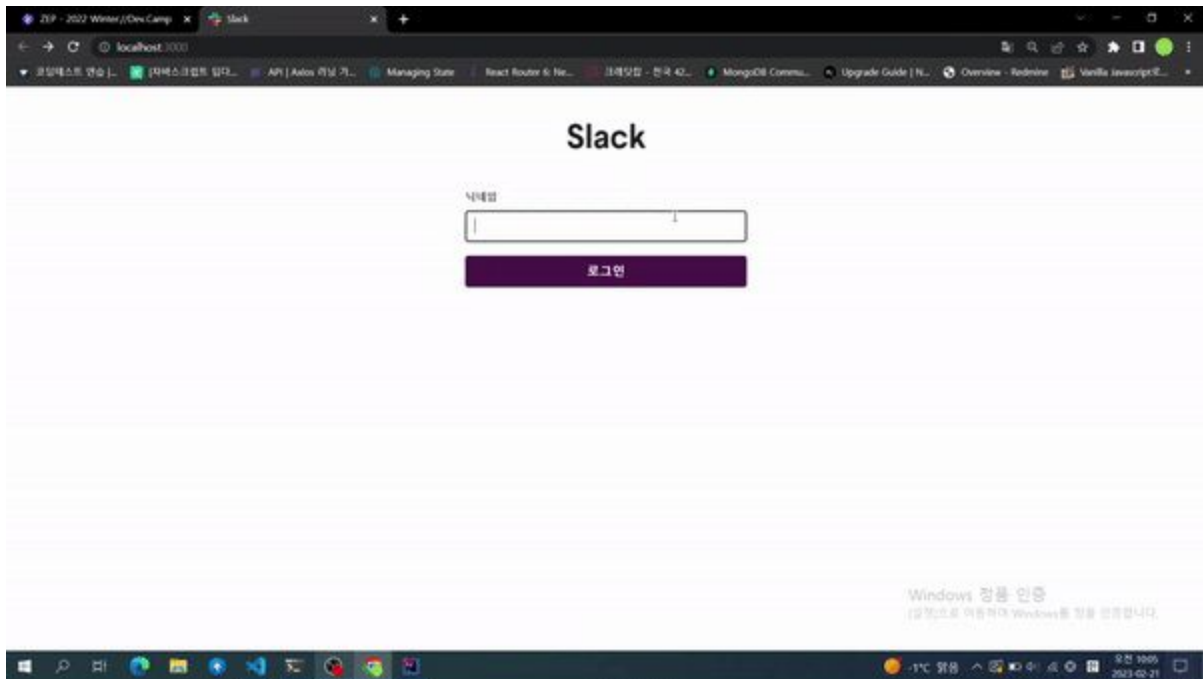


Slack Clone Project – 결과물

프로젝트 소개



채널 페이지



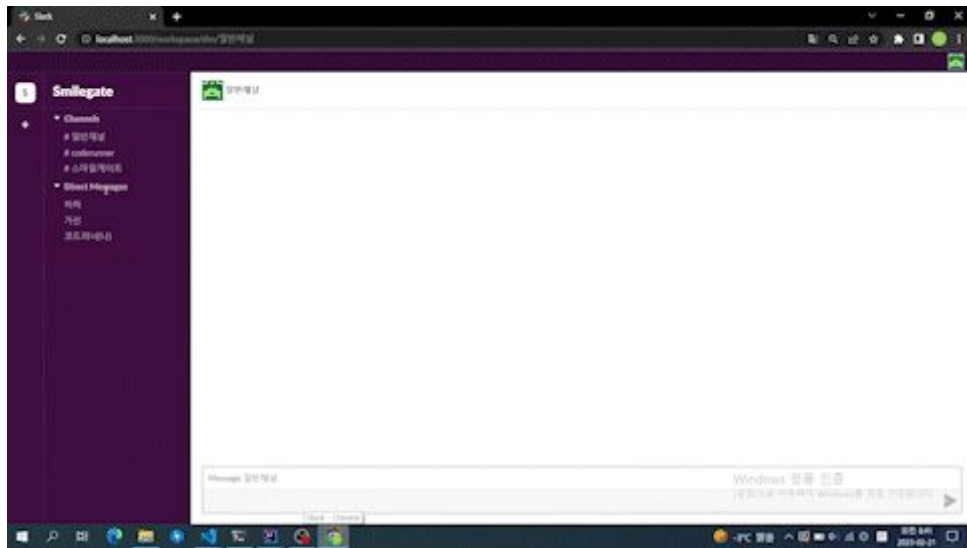


Slack Clone Project – 결과물

프로젝트 소개



채팅 페이지





2. 역할 소개 및 이슈 해결 과정



장소명 (BE) - 역할 소개

역할 소개 및 이슈 해결



기존 업무

1. 게이트웨이, 서비스디스커버리 구축을 통한 인프라 구성
2. 채팅 서버 구축



변경된 업무

1. 게이트웨이, 서비스디스커버리 구축을 통한 인프라 구성
2. 채팅 서버 구축
3. 회원 서버 구축



장소명 (BE) - 이슈 해결 과정 1

역할 소개 및 이슈 해결



기존 목표

JWT를 이용한 회원 인증 구현

시행착오

- **원인:** 팀원 간 진행상황 공유 부족
- **해결방법:** 일주일도 안남은 시점에서 간단히 할 수 있는 방법 선택하기로 함

결과

인증 절차가 필요없는 **익명 로그인 방식**으로 바꾸기로 함



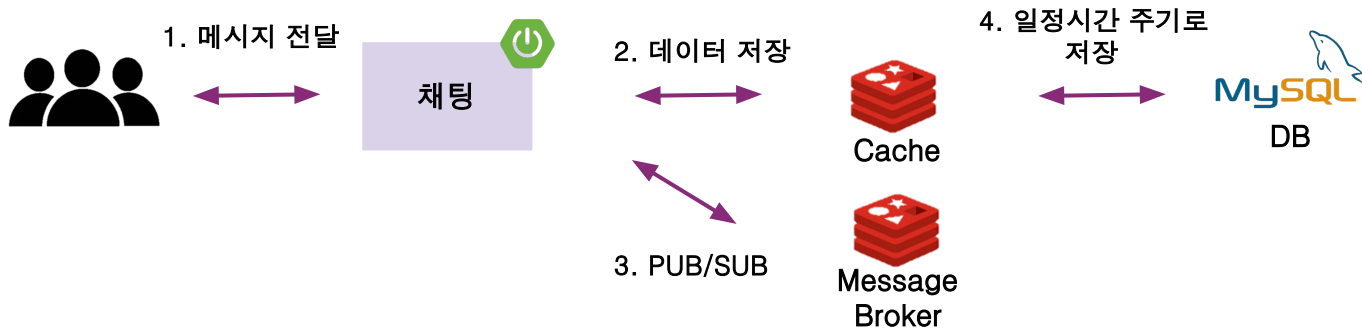
장소명 (BE) - 이슈 해결 과정 2

역할 소개 및 이슈 해결



기존 목표

일정 주기가 되면 Redis 캐시에 저장했던 Data를 DB에 저장





장소명 (BE) - 이슈 해결 과정 2

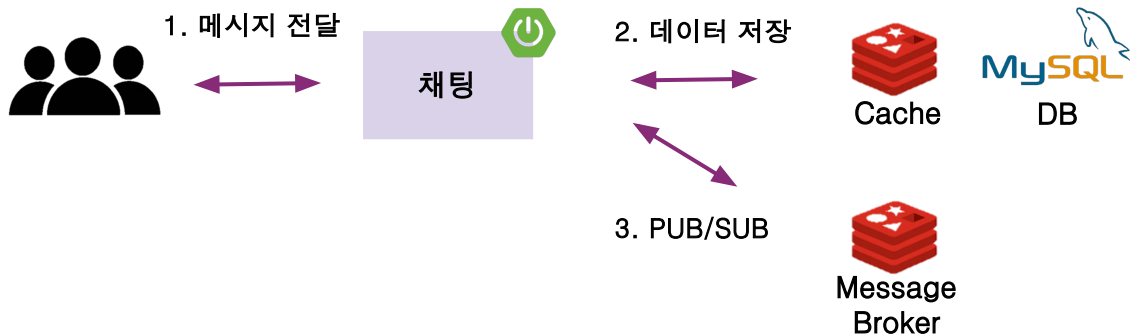
역할 소개 및 이슈 해결



시행착오

- **원인:** Redis에 저장되는 것 까지 확인했지만, Redis에서 조회하는 부분에서 막힘 + 오류해결 시간 부족
- **해결 방법:** Cache와 DB에 동시에 저장하는 방식으로 변경

결과





박가인 (FE, BE) - 역할 소개

역할 소개 및 이슈 해결



기존 업무

1. 워크스페이스, 채널 프론트
2. 채팅 클라이언트
3. 채널 생성,조회, 글작성,조회 api



변경된 업무

1. 전체 프론트부분
2. 채널 생성,조회, 글작성,조회 api



박가인 (FE, BE) - 이슈 해결 과정

기존 목표

채팅, 게시글을 post 요청으로 보내면 바로 렌더링되어 보낸 글이 화면에 보이게 함

시행착오

채팅, 게시글을 작성해도 바로 화면에 렌더링이 되지 않음!

😭 get 요청을 두번 보내도 안됨!



컴포넌트 안에 있는 채팅내용의 **상태값을 변경** 해주면 렌더링이 되기 때문에

```
const [chatData, setChatData] = useState([]);  
setChatData([...chatData, chat]);
```

spread 연산자로 깊은 복사 한 뒤 input 입력값을 넣어줌



박가인 (FE, BE) - 이슈 해결 과정

역할 소개 및 이슈 해결



결과



상태관리 하던 값의 상태값이 변경되어 화면이 렌더링됨



박가인 (FE, BE) - 이슈 해결 과정

역할 소개 및 이슈 해결



기존 목표

Recoil이라는 라이브러리의 selector 함수를 사용해서 api 호출을 컴포넌트와 분리하여 관리해보자!



시행착오

selector 함수로 get 요청을 처리하고자 아래와 같이 작성하였으나 api 호출을 통해 받은 데이터를 따로 저장소에 담기 힘들



결과

selector로 api 호출을 관리하는 것이 따로 api 호출하는 hook을 만들어서 하는 것보다 효율적이지 않다고 깨달음

```
export const channelDataSelector = selectorFamily({
  key: "channelDataSelector",
  get: (id) => async () => {
    try {
      const { data } = await axios.get(
        `http://localhost:8082/channel/${id}/get`
      );
      return data;
    } catch (err) {
      console.err(err);
    }
  },
  set: ({ set }, newValue) => set(channelDataAtom, newValue),
});
```



박가인 (FE, BE) – 이슈 해결 과정

역할 소개 및 이슈 해결



기존 목표

채팅 클라이언트 구현할 때 웹 소켓을 사용하여 구현하자!





박가인 (FE, BE) – 이슈 해결 과정

역할 소개 및 이슈 해결



시행착오

[@stomp/stompjs](https://stomp.github.io/) 라이브러리를 사용하여 간단하게 구현하기로 결정함

클라이언트에 아래와 같이 두 가지 유형이 있다는 것을 모르고 send 대신 publish를 사용했더니 당연하게도 연결이 되지 않음!

- 생산자: CONNECT, SEND frame
- 소비자: CONNECT, SUBSCRIBE frames

채팅방 생성





박가인 (FE, BE) – 이슈 해결 과정

역할 소개 및 이슈 해결



결과

여러 과정을 거쳐 **connect()**, **subscribe()**, **send()**
메소드로 채팅 클라이언트 개발

```
52 };
53 const connectSocket = () => {
54   StompClient.connect(
55     { "client-id": "my-client-id" },
56     "",
57     () => {
58       StompClient.subscribe(
59         `sub/chat/room/${roomId}`,
60         (message) => {
61           if (message.body) {
62             alert("got message with body " + message.body);
63           } else {
64             alert("got empty message");
65           }
66         },
67         { id: roomId }
68       );
69     },
70     function (frame) {
71       console.log(frame + "Web socket disconnected");
72     }
73   );
74 };
75
```



3. 프로젝트 리뷰



장소명



배운점

- 기술을 선정하고 고민할 수 있었음
- MSA 구조를 직접 구성하고 서버 간 통신 경험할 수 있었음
- Redis를 이용한 메시지브로커와 캐시를 직접 구현할 수 있었음
- 팀원간 진행상황을 말뿐이 아니라 직접 확인하면서 공유해야 한다는 것을 깨달음

아쉬운점

- 성능 테스트 하지 못함
- Redis Cache를 적용해보지 못함
- Gateway 내 인증 절차 적용해보지 못함

향후 포부

- 향후 개인적으로 앞서 말한 아쉬웠던 점들을 보완하고 싶음



배운점

- 전역상태관리 라이브러리 recoil의 atom과 selector를 언제 어떻게 사용하는지 배움
- 팀빌딩은 단순 역할분담이 아니라는 것과 각자 개발한 코드를 주기적으로 커밋 푸시 머지하는 것의 중요성을 깨달음

아쉬운점

- jest 테스트 도구로 채널부분에서 테스트코드를 작성하고자 했으나 결국 못함
- 핵심이 되는 화면부터 개발하지 않아 백엔드팀원이 프론트코드가 필요할 때 아직 개발을 끝내지 못한 상황인 경우가 있었음

향후 포부

- 비동기 호출은 따로 훅으로 관리하고 recoil로 전역상태관리를 하여 리액트에서 api 호출부분과 같은 코드와 UI코드를 분리해 더 깔끔하게 코드를 작성할 것입니다!
- 팀에서 각자 개발한 코드를 적어도 3일에 한번은 merge하여 서로의 진행상황을 꼼꼼하게 확인하겠습니다!



감사합니다 :)