

칙칙퐁퐁

정찬웅

INDEX

01

About

목적

02

Contents

채팅
게임
대시보드

03

Skills

사용한 기술 스택

About

칙칙퐁퐁은 실시간 게임과 채팅을 결합한 서비스입니다.

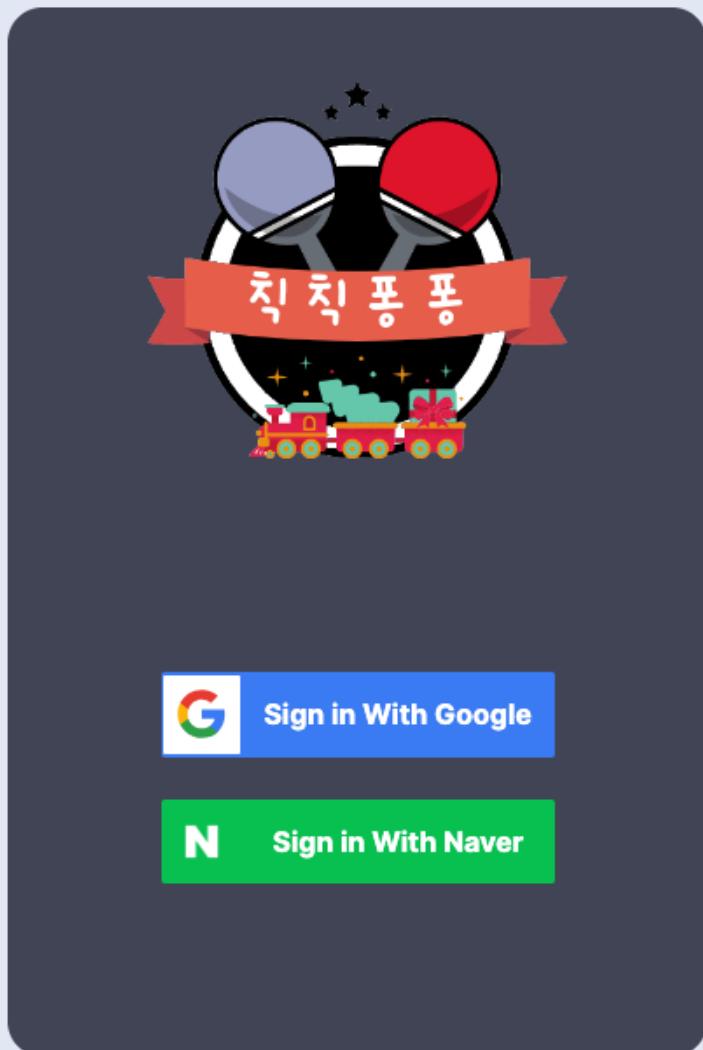
클라이언트와 서버간의 연결을 소켓과 Rest API로 구현하였습니다.

게임은 실시간으로 진행되며,
백엔드 서버에서 게임 엔진을 구현하였습니다.

게임 종료 후 결과는 대시보드를 제공하였습니다.

AWS EC2를 통해 배포하였습니다.

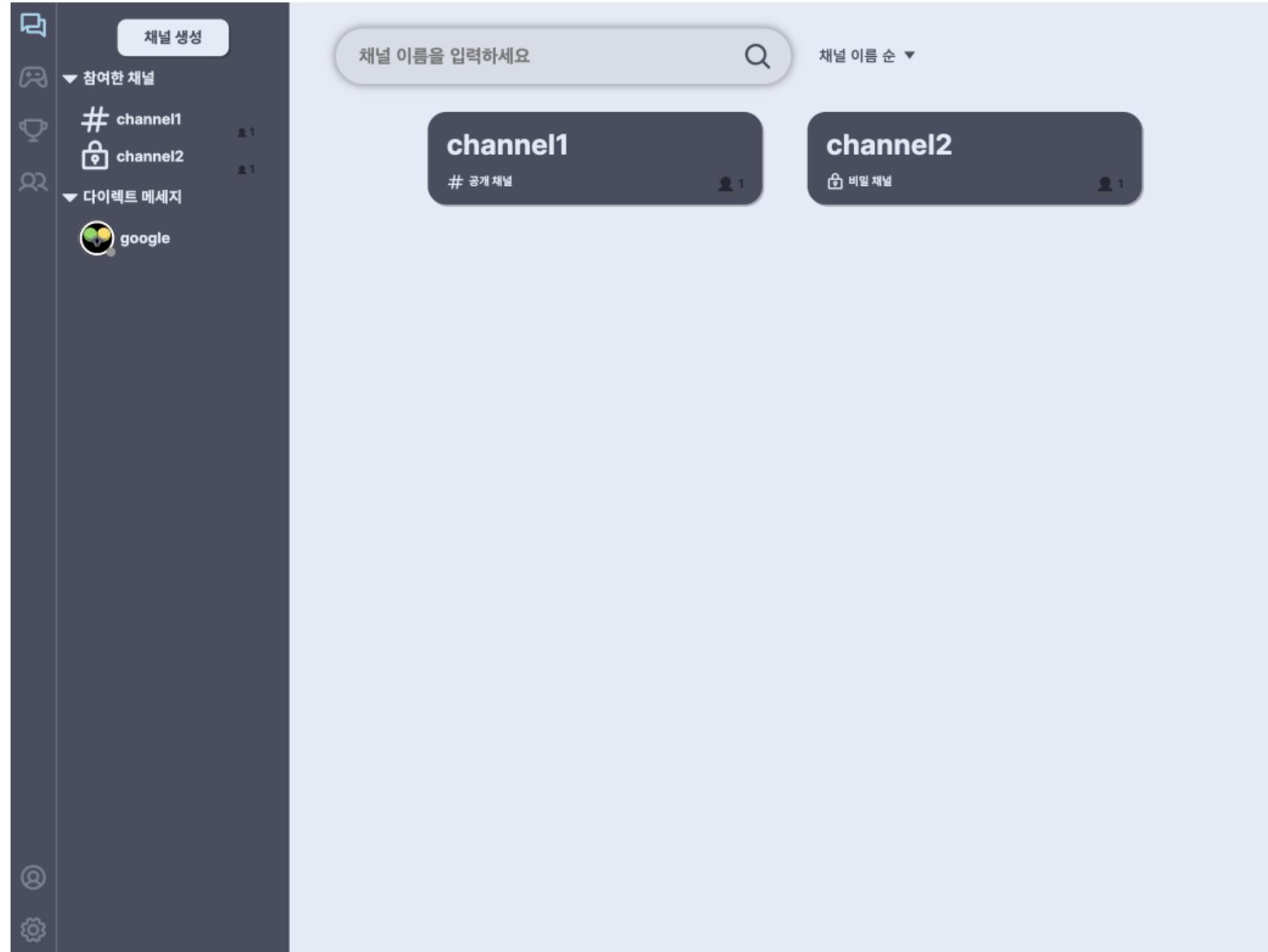
Contents



로그인을 간편하게 하기위해 소셜 로그인을 추가하였습니다.

추가적인 개인정보를 필요로 하지 않아
인증의 용도로만 사용했습니다.

Contents



로그인하면 보여지는 화면입니다.

채팅의 채널을 구현하였고,
채널은 [공개, 비공개, 비밀]로 나뉩니다.

Contents

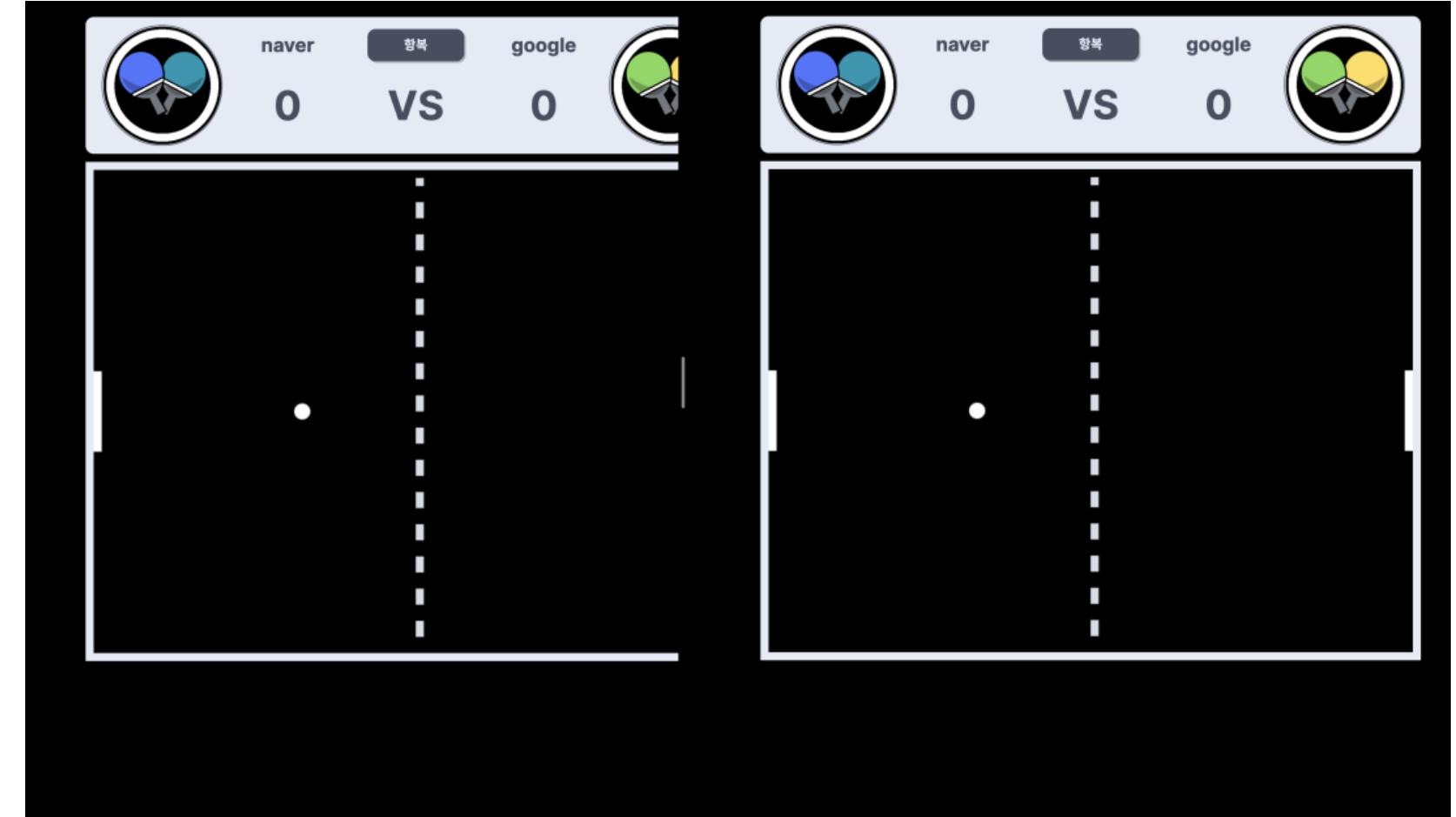


DM 가능입니다.

상대방에게 DM을 하게 되면, 알림이 표시됩니다.

채팅은 새로고침 없이 실시간으로 전송 및 수신 가능합니다.

Contents



게임 대기방 및 게임 화면입니다.

게임 대기방의 채팅창은 터미널처럼 구성했습니다.

게임은 실시간으로 진행되고, 두 유저는 같은 정보를 수신할 수 있도록 서버에서 정보를 전송합니다.

Contents

#	닉네임	점수	승률
1	dummy3 (admin)	2042 LP	95 194 32.87%
2	dummy15	1969 LP	109 226 32.54%
3	dummy8	1948 LP	146 63 69.86%
4	dummy17	1943 LP	73 223 24.66%
5	dummy39	1940 LP	64 120 34.78%
6	dummy33	1924 LP	177 116 60.41%
7	dummy36	1920 LP	125 164 43.25%
8	dummy19	1919 LP	239 70 77.35%
9	dummy4	1915 LP	166 81 67.21%

Top 3
dummy3 2042 LP
dummy15 1969 LP
dummy8 1948 LP

내 전적
승률 48.7%



게임은 랜덤 매칭이 적용된 랭킹전이 가능하며, 랭킹전에서 승리 혹은 패배 시 점수가 변동됩니다.

점수 시스템은 ELO 레이팅을 구현하였습니다.

각 유저의 프로필 페이지에는 본인의 전적과 통계를 볼 수 있는 대시보드를 제작하였습니다.

Skills



프론트엔드, 백엔드 모두 타입스크립트로 작성하였습니다.

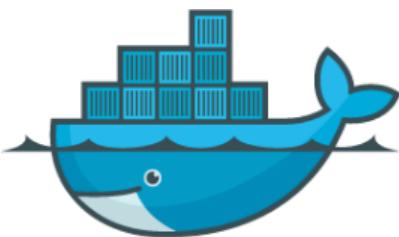
프론트엔드는 리액트를 사용하였고, 백엔드는 NestJS를 사용하였습니다.

그 이유로 하나의 언어로 프론트, 백 둘 다 구현을 할 수 있다는 편리함이 있었고, NestJS의 DTO를 통해 입력받은 데이터의 유효성을 쉽게 검사할 수 있다는 장점이 있기 때문입니다.



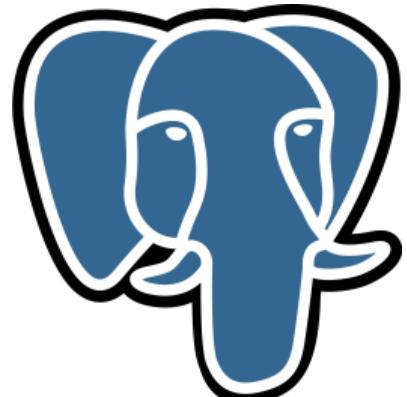
웹 서버는 NGINX를 사용했습니다.

NGINX를 통해 API와 소켓을 더욱 간편하게 활용할 수 있었습니다.



각 서버들을 독립적으로 사용하기 위해 도커 컨테이너로 관리해주었습니다.

도커 컴포즈를 통해 여러개의 도커 컨테이너들을 결합해서 사용하였습니다.



데이터베이스는 PostgreSQL을 사용하였습니다.

관계형 데이터베이스 사용을 통해 데이터 추가, 삭제, 변경이 용이하기 때문입니다.

저희 서비스에서는 지속적으로 데이터를 변경하고, 채팅 내역을 추가해주어야 하기 때문에, 이 데이터베이스를 사용하였습니다.