

MOINEE

모이니 : 모임 모집 플랫폼

※ 한국어 '모임'을 영어 음차로 작명하였습니다.

팀명 : ILJO

목차 Index

Section I

프로젝트 소개

- 프로젝트 소개
- 팀원 소개와 역할
- 주 기술 스택

Section II

세부 설명

- 아키텍처 설명
- 도커 & 클라우드 배포
- 자연어처리 & 머신러닝
- 기능 시연 및 설명

Section III

정리 및 마무리

- 문제 해결 고민
- 후기 및 마무리

I. 프로젝트 소개

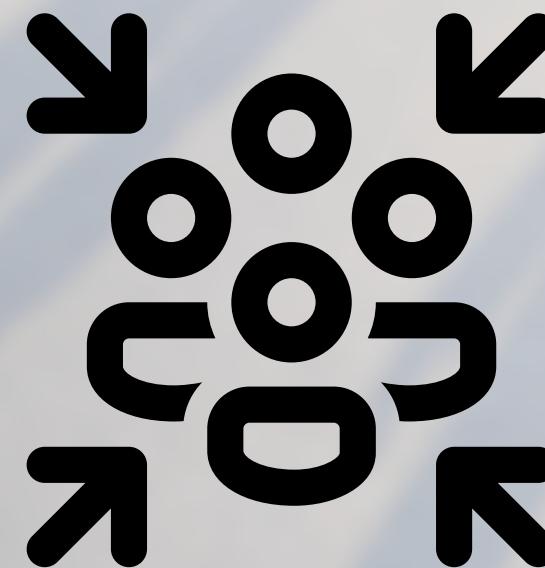
what is MOINEE??



※ 모이니는 취미 공유 웹 어플리케이션 서비스입니다.

MOINEE

모임을 영어로 풀어서 귀엽게 표기



[REDACTED]

<동기>

- Field Technology
 - Practical problem-solving
 - Useful Service

현업의 기술들을 최대한 사용하여
개발자로서 현실의 문제를 해결하여
유용한 서비스를 제공해주고 싶은 게
мот티브이다.

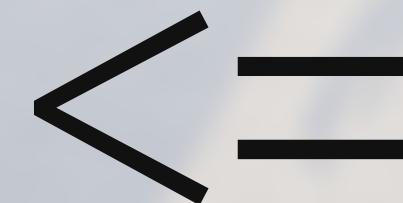
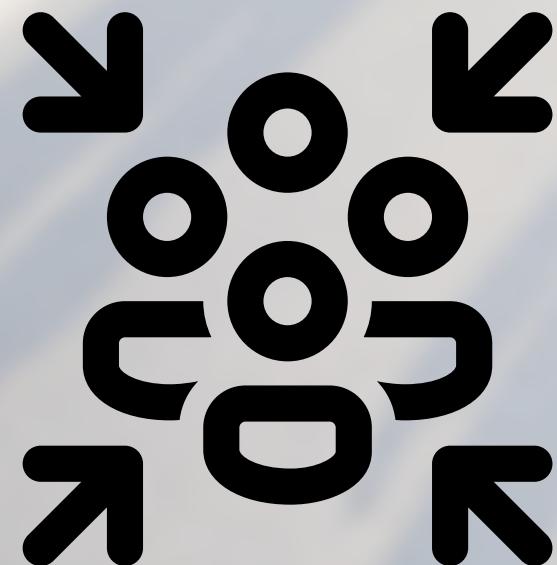
<목적>

- ▶ Networking
 - ▶ Gathering
 - ▶ Connection

POST 코로나 시대에 앞으로 대면 활동이 더욱 활발해질 텐데 사람끼리 만날 수 있는 기회가 많아졌으면 하는 바람으로 moinee라는 프로젝트를 진행하게 되었다.

who is REFERENCE?

MOINEE



<Reference>



*URL : [HTTPS://WWW.FRIENDSCUBE.COM/](https://www.friendscube.com/)

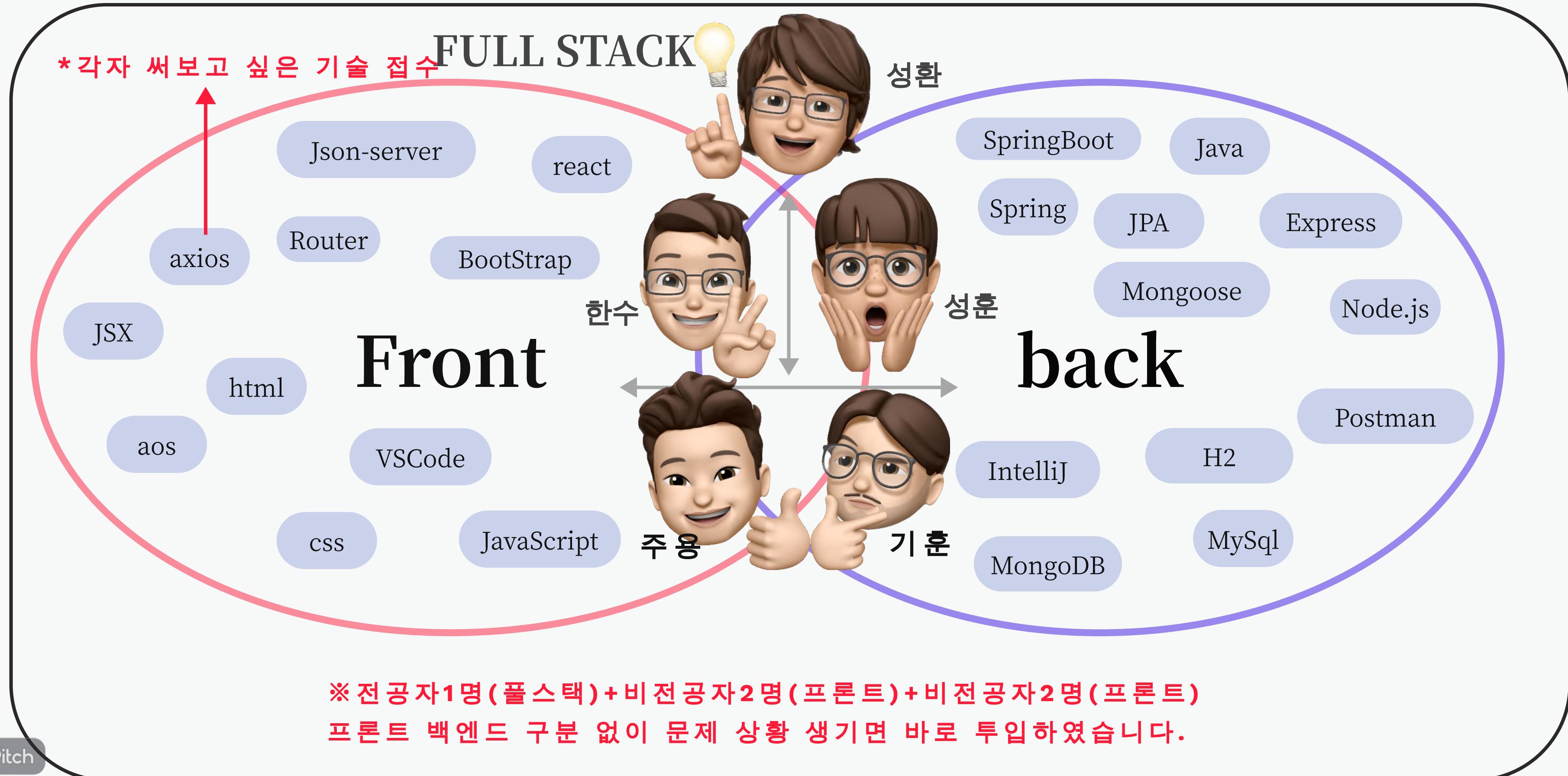
*소모임이라는 기준 모바일 앱을 참고하였습니다.

나와 같은 관심사
나와 같은 지역의 멤
버들과 함께해요.

취미 생활로 더 알차고
재미있는 여가시간을 만들어요.



팀원 소개와 역할



<Full stack>

<성환>(팀장)

[front] React, Bootstrap, json-server 컴포넌트, 라우터, 네비게이션, 차트, 애니메이션

[back] Spring, SpringBoot, Node.js, Express 스프링서버, 노드 서버, 스프링 게이트웨이, 유레카 서버, JPA, 스프링 보안, 토큰

[else] Mongoose, MySQL, MongoDB, H2, Cloud, 데이터베이스 구축, 클라우드에 서버 실행, 버킷생성, MSA 및 아키텍처 설계, 자연어 처리 및 ML Model 설계, Dockerfile 및 dockerhub를 이용한 container 이미지 생성 및 클라우드에 배포

<Front-end>

<주용>

[Front] React, Bootstrap, json-server 로그인, 회원가입 페이지

[Back] Spring, SpringBoot 해시태그 컨트롤러

[else] Mysql, H2, json-server 스프링 서버와 MySql 연동,
요청응답 테스트

<한수>

[front] React, Bootstrap, json-server 게시판, 댓글 기능

[Back] Spring, SpringBoot 유저팔로우, 방 좋아요, 참여한 방 컨트롤러

[else] H2, json-server, postman 서버와 DB 테스트

<Back-end>

<성훈>

[Front] React, Bootstrap 마이페이지, cors 오류 해결

[Back] Spring, SpringBoot 툴 컨트롤러, jpa, feign client

[else] H2, MongoDB, MySQL 더미데이터 CRUD(mockaroo)

<기훈>

[Front] React, Bootstrap 로그인 페이지 수정, 간략정보 수정

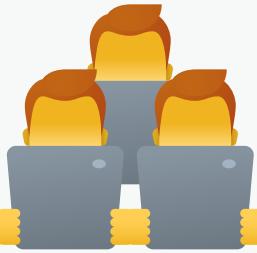
[Back] Spring, SpringBoot 유저컨트롤러

[else] Cloud, MySQL 서버와 클라우드 연동, 버킷생성, 더미데이터

<공통>

Entity-Relationship Diagram 설계, wire frame 설계, DataBase 설계, Router 설계, Use Case Diagram 설계

역할 분배 Team Role



공동 작업

- 주제 선정
- 기능 설계 및 Use-case Diagram 작성
- 화면 설계 및 Wire Frame 작성
- 백엔드 Controller Router 설계
- Database 설계 및 ER 다이어그램 작성

* 모두 함께 지속적인 팀 회의를 통해 요구사항 확인, 화면 설계, 애플리케이션 설계, 인터페이스 설계, DB 모델링을 진행하였습니다.

역할 분배 Team Role



임성환 (팀장)

- 프로젝트 총괄
- 리액트 컴포넌트 작성
(네비게이션 바 / …)
- 리액트 페이지 작성
(전체 게시물 보기 / …)
- 리액트 기능 구현
(로그인 / …)
- 백엔드 서버 구축
(스프링 / 노드JS / 스프링 게이트웨이 / 유레카)
- 스프링 보안, 토큰 시스템 구축
- 데이터베이스 구축
(MySQL / MongoDB)
- 도커를 이용해 클라우드에 애플리케이션 배포
- msa 아키텍처 설계 및 구현
- 자연어처리 & 머신러닝 모델 설계 및 구현



이한수

- 리액트 컴포넌트 작성
(게시물 보기, 댓글 창,)
- 리액트 기능 구현
(게시판, 댓글창)
- 스프링 서버 기능 구현
(팔로우 / 게시물에 좋아요 / 유저가 참여한 방 목록)

역할 분배 Team Role



강성훈

- 리액트 페이지 작성
(회원 정보 보기)
- 스프링 서버 구현
(룸 컨트롤러)
- feign client 구현
- cors 오류 해결
- 더미 데이터 생성 & 입력



심주용

- 리액트 컴포넌트 및 페이지 작성
(로그인, 회원가입)
- 리액트 기능 구현
- 스프링 서버 구현 (게시물 해시태그)
- Spring - MySql 연동
- Spring & React 테스트



이기훈

- 리액트 페이지 작성
(게시물 미리보기 창 / 회원정보 수정)
- 더미데이터 생성 & 입력
- 스프링 서버 구현 (유저 컨트롤러)
- 서버 - 클라우드 연동
- 클라우드 버킷 생성

주 기술 스택



DB

MySQL®

mongoDB®

※ MongoDB, node.js 사용 이유

비정형 data 및 빅 data를 처리 하기에 용이하기 때문입니다.

node.js를 는 MongoDB와 호환이 잘 되기 때문에 사용했습니다.



BACK-END

spring
BOOT

node
JS®



FRONT-END

React

Bootstrap

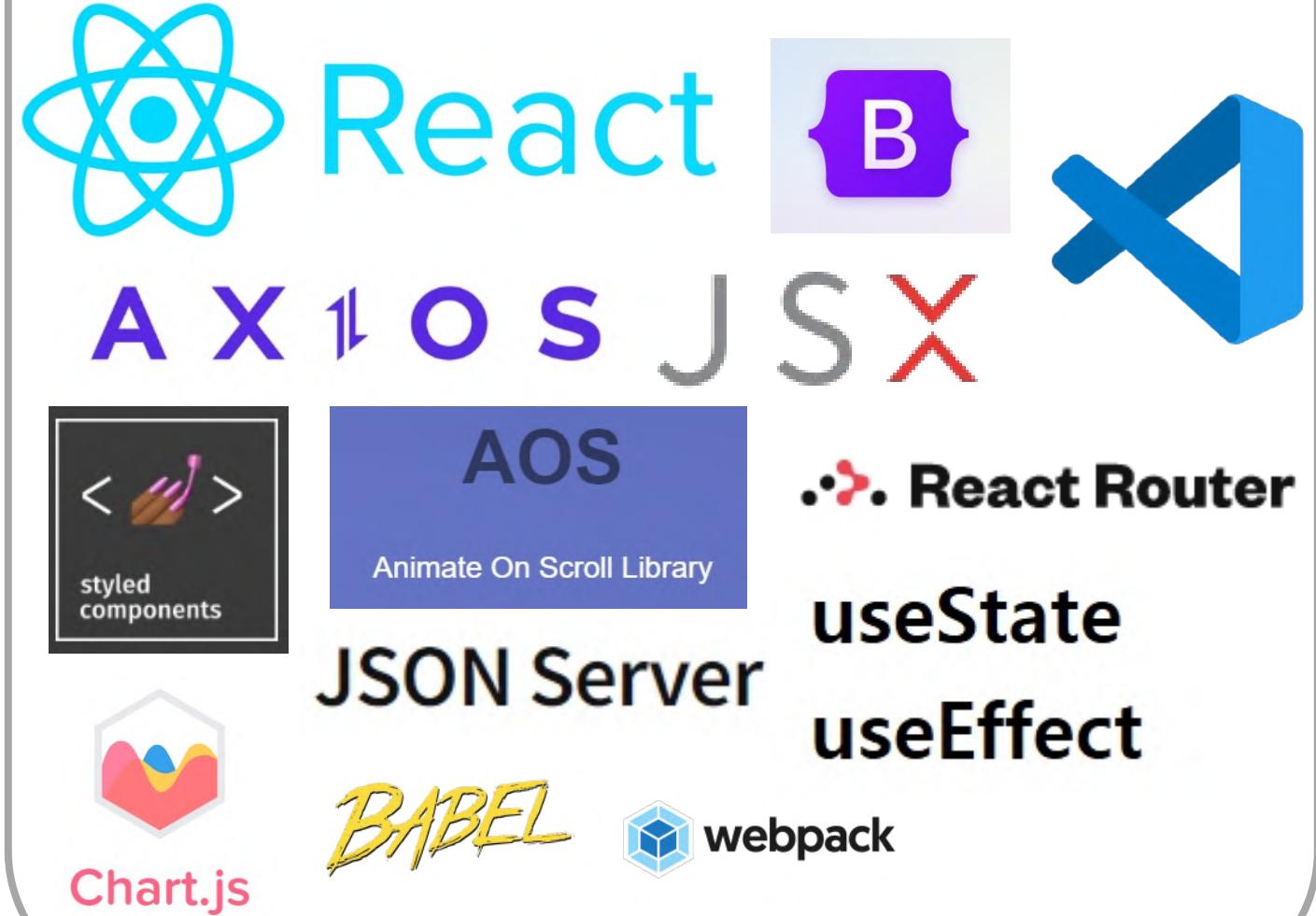


CLOUD

Google Cloud Platform

※총 기술 스택 정리입니다.

Front-End



Back-End



DB



Machine Learning



Else

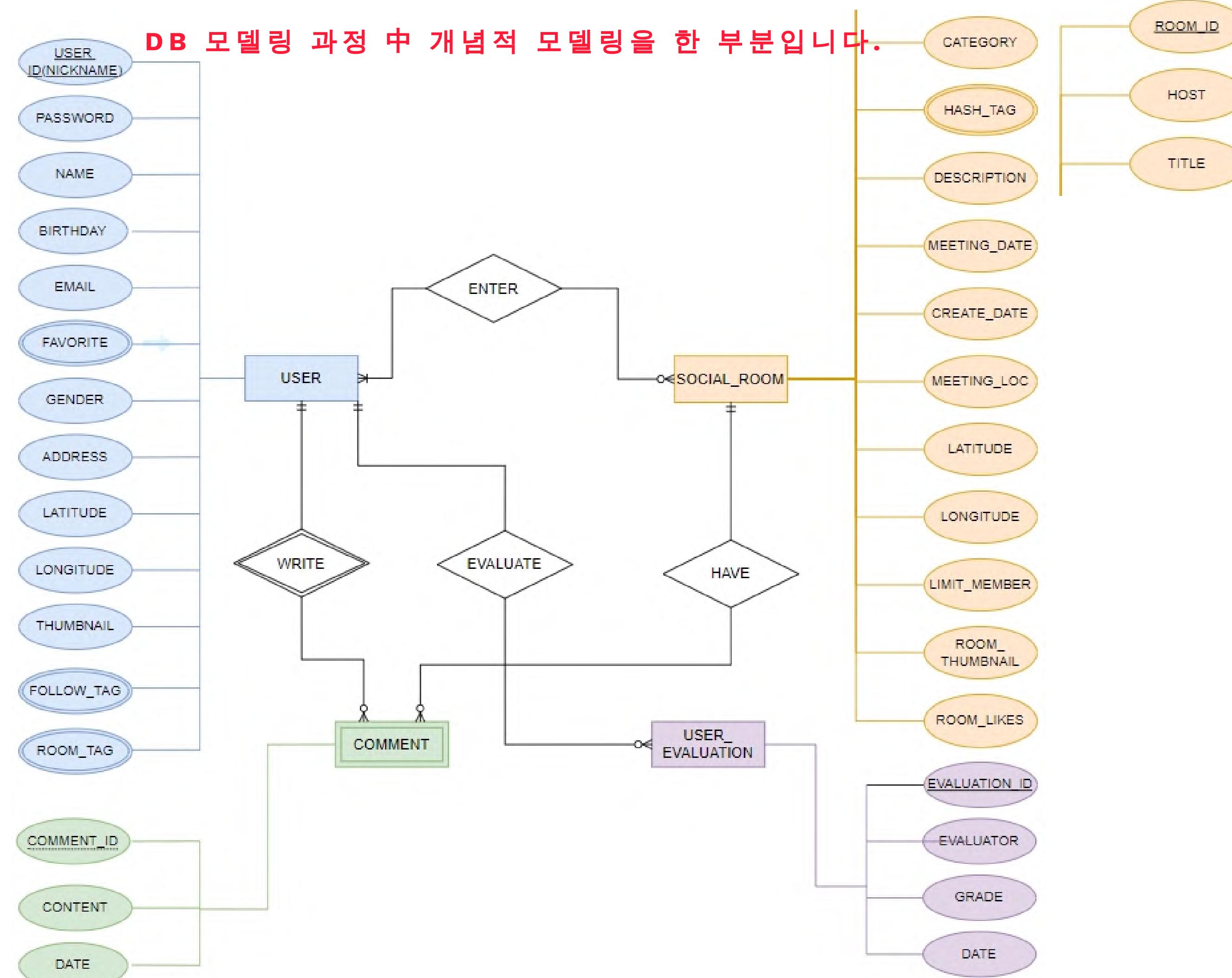


Cloud



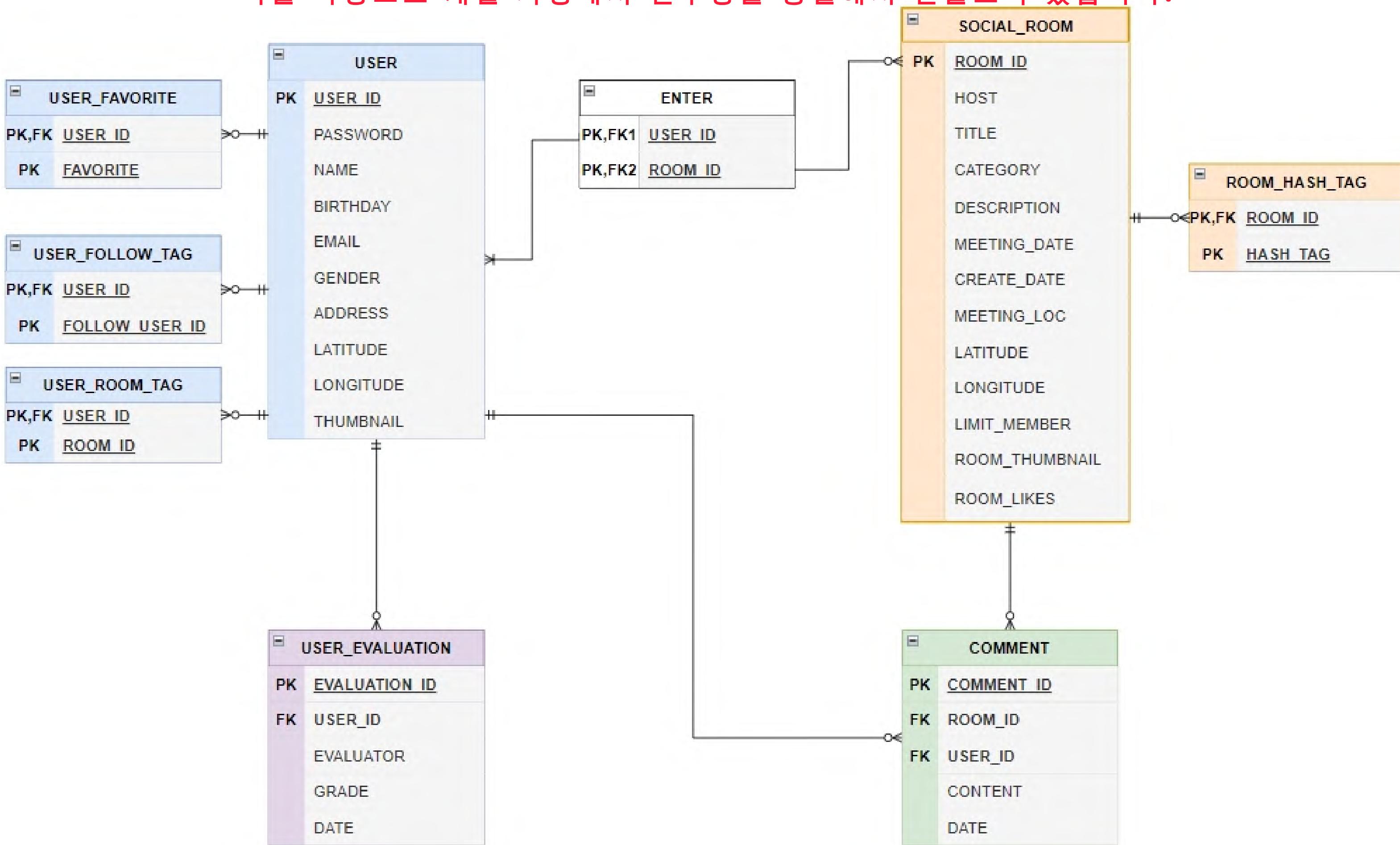
ER Diagram

DB 모델링 과정 中 개념적 모델링을 한 부분입니다.



ER Diagram

개념적 모델링을 바탕으로 논리적 모델링을 하였습니다.
이를 바탕으로 개발 과정에서 변수명을 통일해서 만들도록 했습니다.



Wire Frame

*WIREFRAME으로 웹사이트의 골격과 애플리케이션의 UI 및 핵심 기능을 나타냈습니다.

Navigation Bar



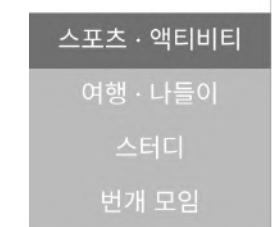
footer



Main



Category bar



Sign up

회원가입

아이디
아이디를 입력하세요

비밀번호
영문, 숫자, 특수문자 포함 8자 이상

비밀번호확인
비밀번호를 다시 입력해주세요

이름
이름을 입력하세요

생년월일
년도-월-일

관심사
운동 □ 여행 □ 독서 □ 음악 □ 게임 □ 영화 □ 운동 □
성별
남 □

이메일
이메일을 입력하세요

주소
주소 □ 우편번호 찾기 □

확인

Login

로그인

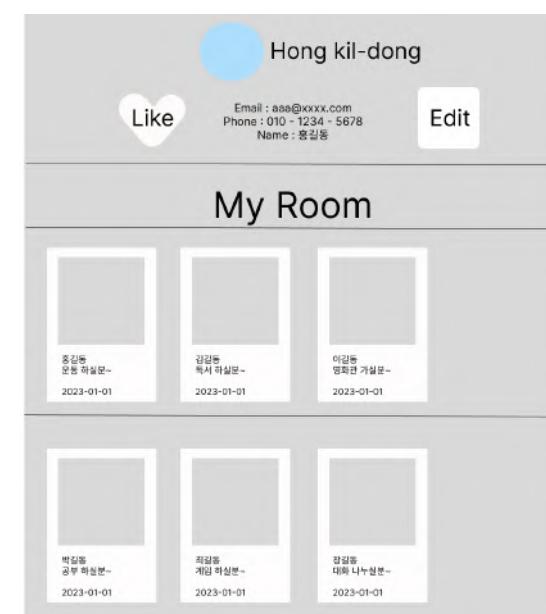
아이디
아이디를 입력하세요

비밀번호
영문, 숫자, 특수문자 포함 8자 이상

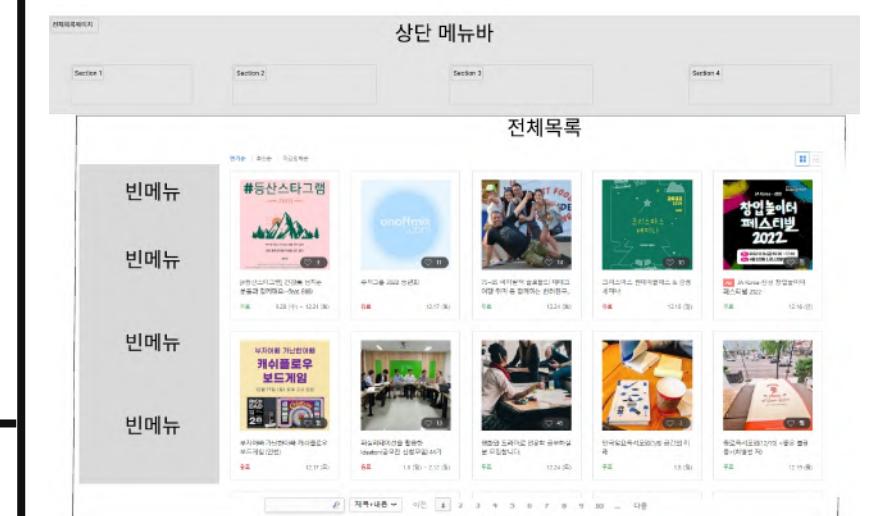
회원가입 아이디찾기 비밀번호찾기

확인

MyPage, Profile



Room List



Make Room

방 생성 NEW

제목을 입력하세요.

태그
운동 여행 음악 독서 게임 음식 스터디

모임 날짜
연-월-일

장소

제한 인원
2

내용

확인

User List



Room Detail



Room brief information



DataBase 설계, Router 설계

DataBase

*DB설계 과정을 보여주는 부분입니다.

user_id	password	encrypted_password	name	birthday	email	gender	address	latitude	longitude	thumbnail
String 20 O Identity	String 크기 x O	String O	String O	String(10) 6 date O unique	String 50 O O	String 1	String O	double O	double O	String Default (기본이미지)
gkesby0			안준호	2006-02-23	gkesby0@nytimes.co	M	울산광역시 강동구 서조중앙03로	59.94043	30.2813321	http://dummyimage.com/232x100.png/cccccc/000000
kmorgon1			이미정	2011-04-03	kmorgon1@jigsy.co	F	전라남도 여수시 석촌호수215가 (서영김)	34.728584	112.132488	http://dummyimage.com/215x100.png/cccccc/000000
rcowburn2			허정숙	2012-04-10	rcowburn2@icq.co	M	대전광역시 영동포구 반포대3가	27.304802	112.5438	http://dummyimage.com/103x100.png/ff0000/ffffff
kskett3			홍예은	2019-09-27	kskett3@wufoo.co	F	인천광역시 용산구 선릉길	26.888665	112.616612	http://dummyimage.com/186x100.png/5fa2dd/ffffff
estruis4			노건우	2011-09-23	estruis4@weebly.co	M	대구광역시 관악구 역삼길	28.0547578	81.614468	http://dummyimage.com/186x100.png/5fa2dd/ffffff
mbroomer5			주영자	2018-04-28	mbroomer5@ucoz.ru	F	서울특별시 동구 양재천길	17.1547981	42.6268969	http://dummyimage.com/236x100.png/cccccc/000000
erassmann6			성성민	2005-09-12	erassmann6@bloglo	M	부산광역시 남구 반포대길 (영수김집면)	8.3102946	-81.2188355	http://dummyimage.com/155x100.png/ff0000/ffffff
bgrote7			한채원	2017-03-22	bgrote7@upenn.edu	F	인천광역시 송파구 영등대로	58.3410191	12.2530819	http://dummyimage.com/227x100.png/ff0000/ffffff
thitschke8			안성현	2021-10-13	thitschke8@state.gov	F	경기도 충주시 압구정가	53.8298337	20.275033	http://dummyimage.com/181x100.png/ff0000/ffffff

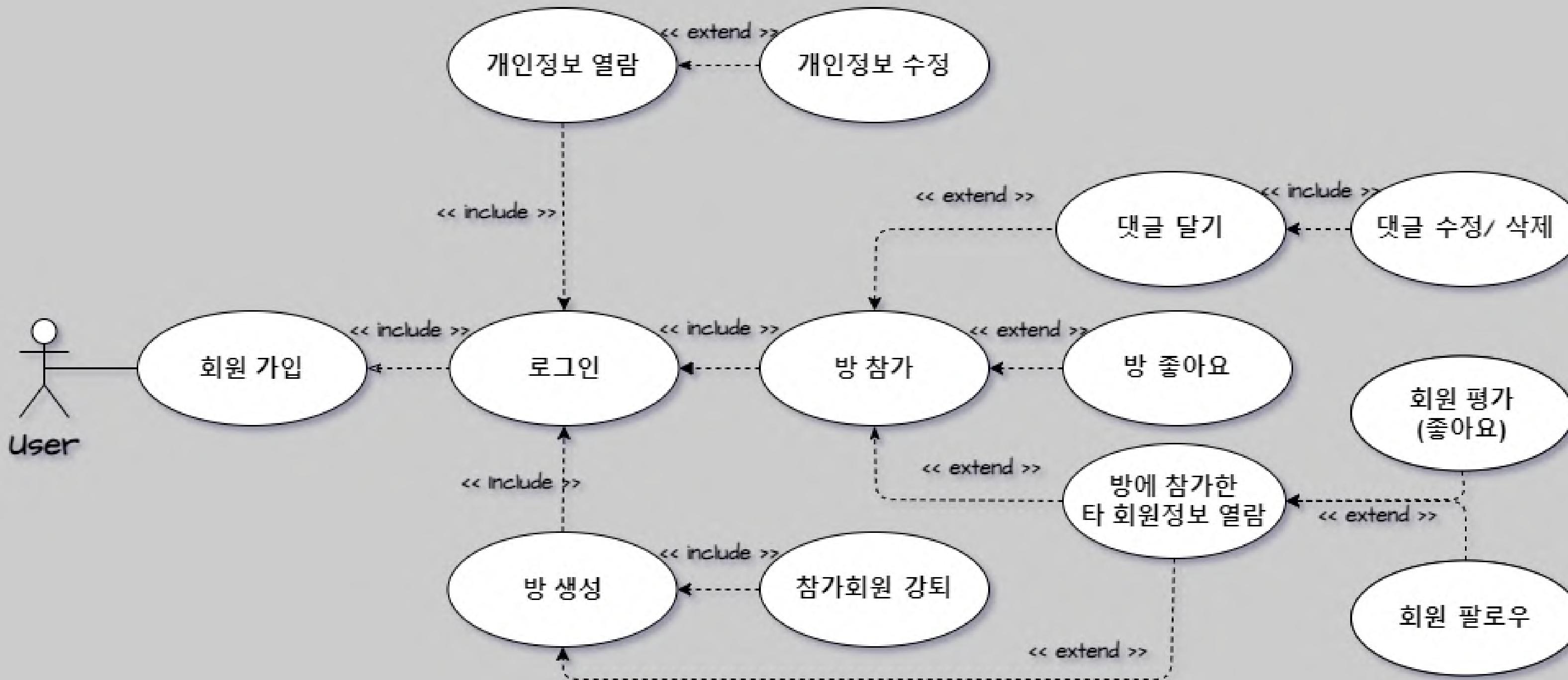
Router

*Router를 설계하여 RESTful API를 구현하였습니다.

Getmapping	CRUD	uri	parameter	return(responsebody)	action
user	post	/	user / req.params	user	회원가입
	post	/login	id,password	jwt(id,password)	로그인
	get	/:userId		user, roomList : [myroom], user	마이페이지
	get	/:userId/room		user	Room에 보내줄 User?
	put	/:userId	user	user	회원 정보 변경
	Delete	/:userId			회원 탈퇴
user/ :userId	user_follow_tag	CRUD	uri	parameter	return(responsebody)
		post	/	userId	User_Follow_Tag
		get	/		List<User_Follow_Tag>
	user_room_tag	delete	/: user_follow_tag		string(삭제 되었다는 흔적)
		post	/	roomId	User_Room_Tag
		get	/		List<User_Room_Tag>
	user_favorite	delete	/:user_room_tag		string(삭제 되었다는 흔적)
		post	/	List<String> (각각의 favorite String(입력 되었다는 흔적))	(각각의 favorite String(입력 되었다는 흔적))
		delete	/:user_favorite		string(변경 되었다는 흔적)
	post	get	/	-	List<User_Favorite>
		post	/	roomId	Enter

Use-Case Diagram

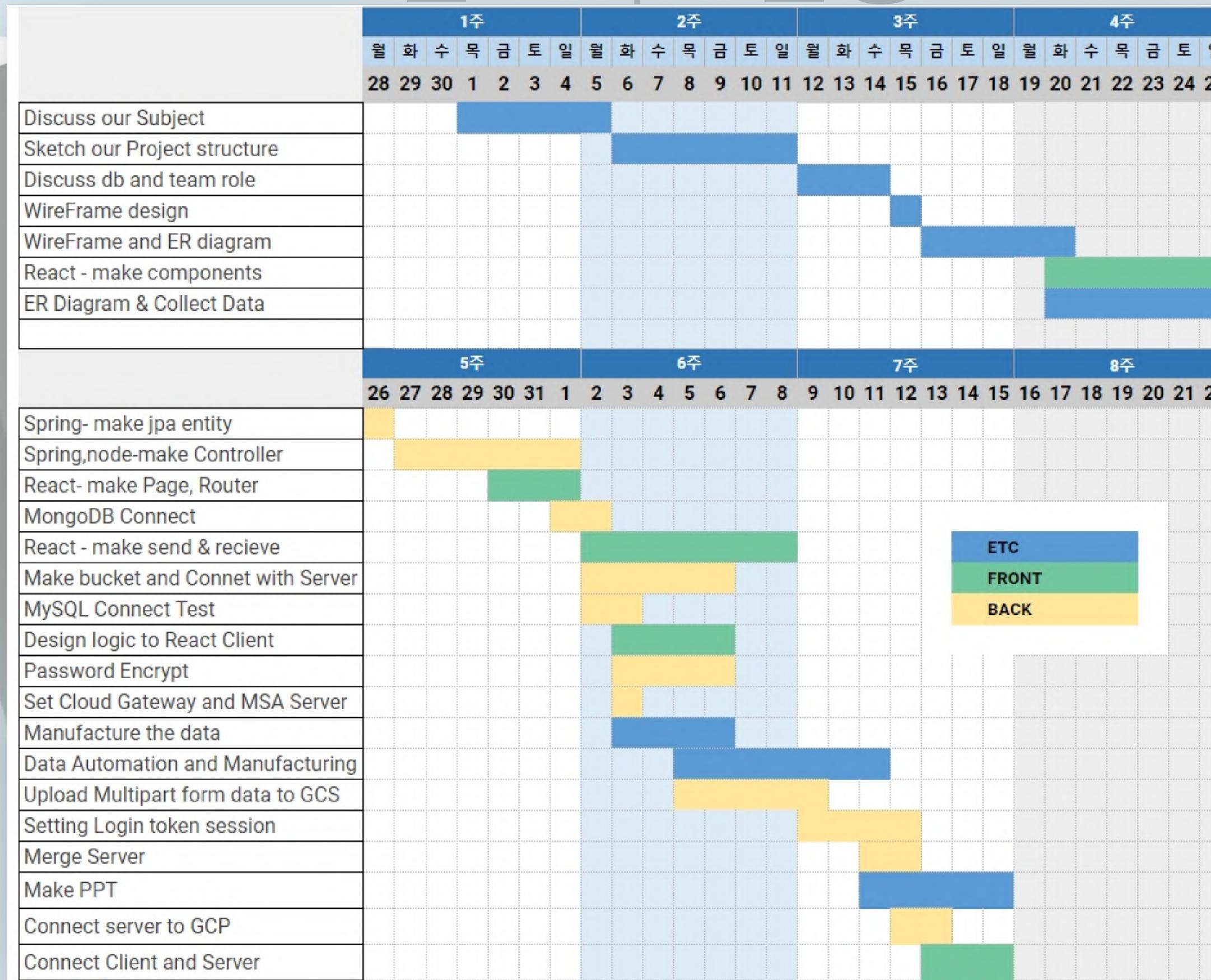
*유즈케이스 다이어그램을 통해 구현할 기능 설계 및 정리를 하였습니다.



프로젝트 관리

*간트차트를 활용하여 일정을 관리하였습니다.

간트차트 활용

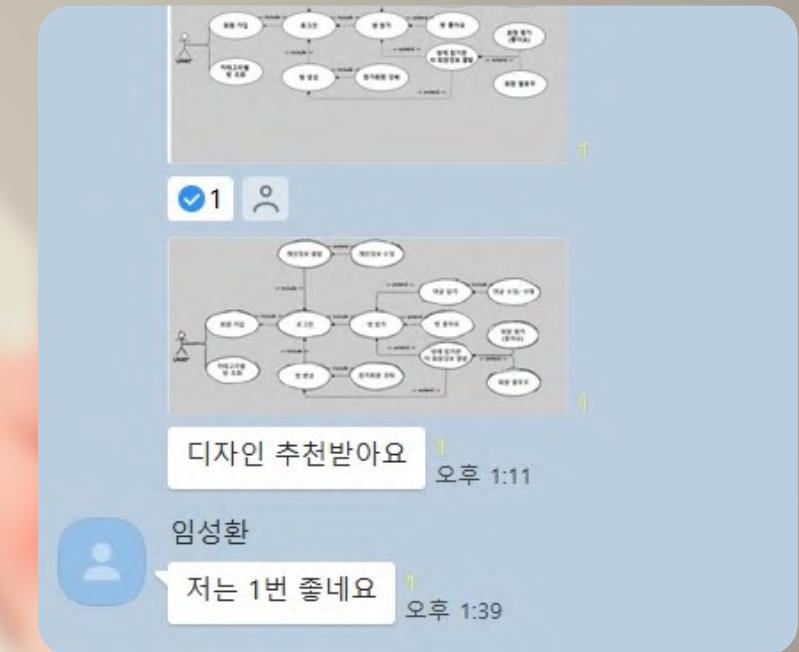


Project Management

◎ 형상관리



◎ 협업방식



This screenshot shows a GitHub repository page for "SeoulPolarBear / ILJOFrontEnd". The page includes a code editor, issue list, pull requests, actions, projects, wiki, security, and insights sections. A specific commit from "moondolph" is highlighted, showing changes made to the "public" and "src" directories. The commit message indicates fixes to a loop and some CSS changes. A README file is present at the bottom of the page.

This screenshot shows a Notion page titled "BackEnd". It lists several git repositories under a bulleted list:

- User : [user_server.git](#)
- Room : [heim.git](#)
- User_evaluation : [ILJOUUserEvaluation.git](#)
- Comment : [ILJOComment.git](#)
- Api-gateway : [iljo-api-gateway.git](#)
- Eureka : [ILJODiscoveryservice.git](#)

Below this section is another titled "FrontEnd" with a single item:

- React : [ILJOFrontEnd.git](#)

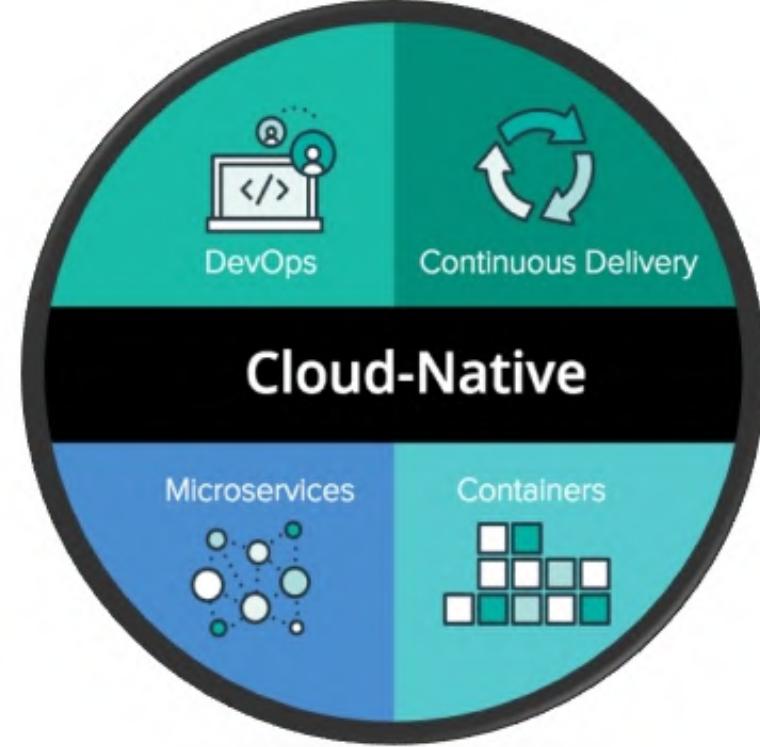
The Notion logo, which is a large black letter "N" inside a black cube-like shape, is displayed next to the word "Notion" in a large, bold, black sans-serif font.

This screenshot shows a Trello board with three main columns: "planning", "In Progress", and "Done". The "planning" column has one card: "Discuss our Subject". The "In Progress" column has several cards, each with a progress bar and a due date: "Design logic to React Client" (due 1월 6일), "Setting Login token session" (due 1월 3일), "React - make Page, Router" (due 1월 2일), "Sketch our Project structure" (due 2022년 12월 31일), "Connect Client and Server" (due 2022년 12월 30일), "Make PPT" (due 1월 13일), and "WireFrame design" (due 2022년 12월 14일). The "Done" column also has several completed cards. The interface includes various filters and search functions at the top.

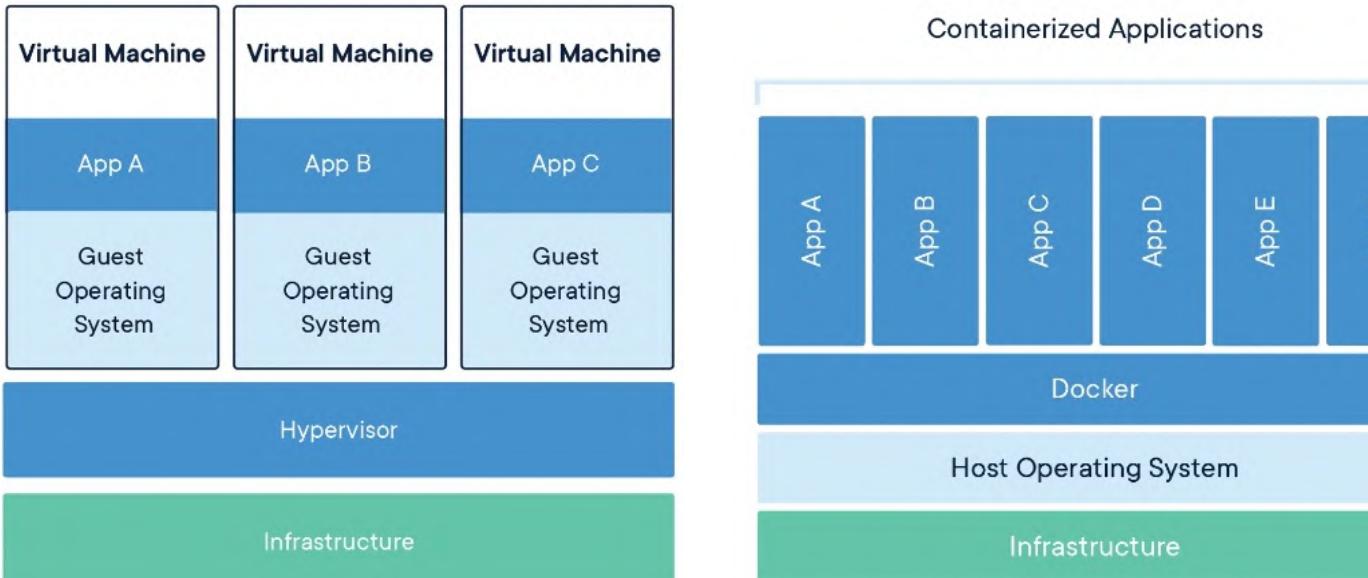


III. 세부설명

클라우드 네이티브란?

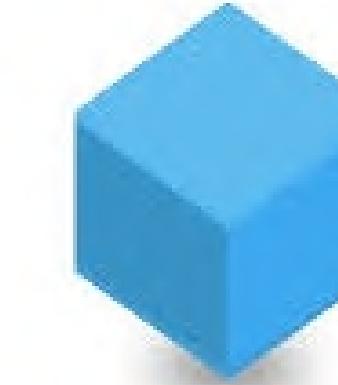


Containers



Microservices

Monolithic



Microservices



현대적인 회사는 고객의 요구를 충족하기 위해 신속하게 업데이트할 수 있는 확장성, 유연성 및 복원력이 뛰어난 애플리케이션을 구축해야 합니다. 클라우드 네이티브는 클라우드 컴퓨팅 환경에서 현대적 애플리케이션을 구축, 배포 및 관리할 때의 소프트웨어 접근 방식입니다.

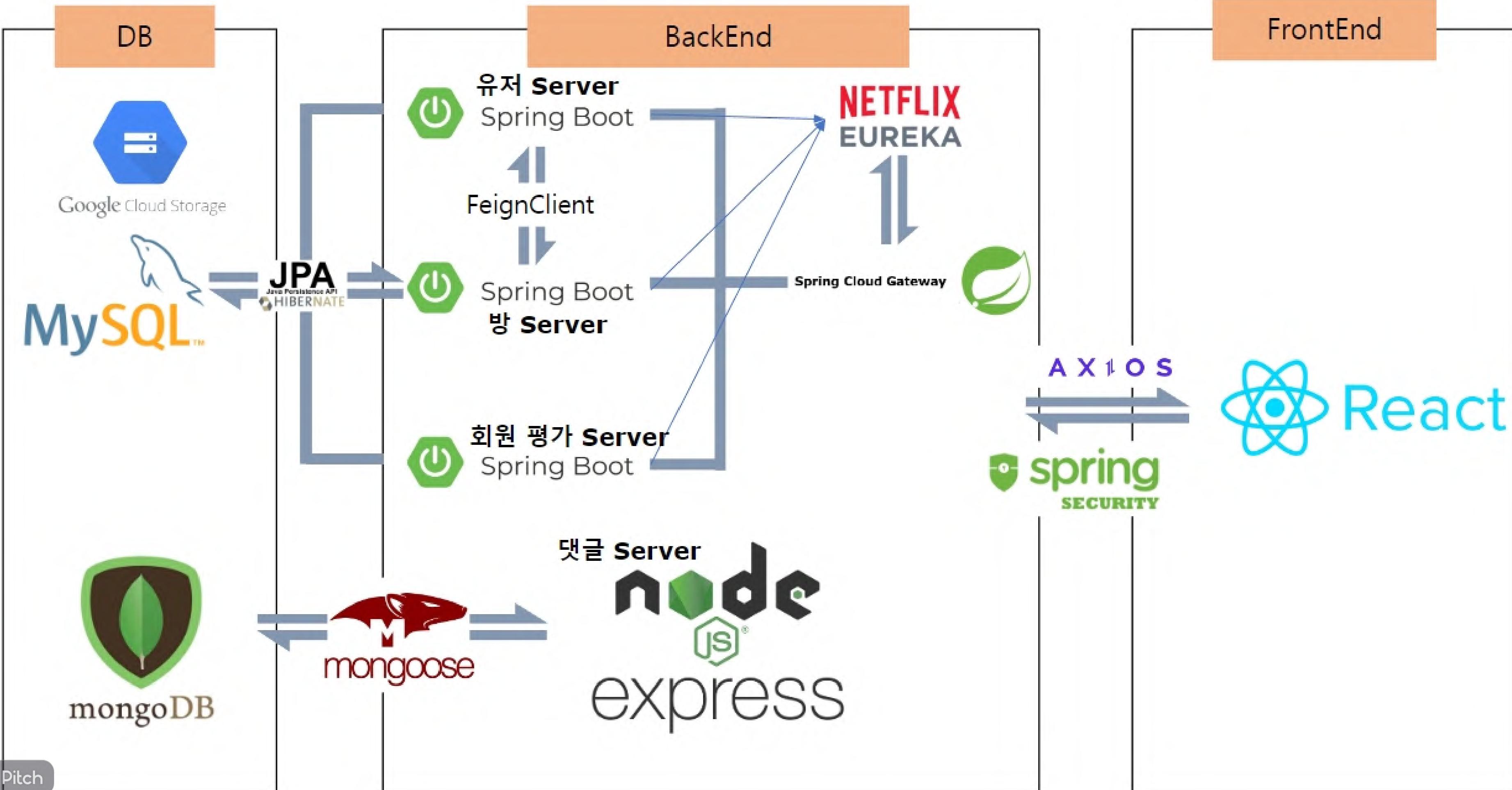
다음과 같은 장점이 있습니다.

- **독립성**: 애플리케이션을 서로 독립적으로 구축하여 개별적으로 관리하고 배치할 수 있습니다.
- **복원성**: 인프라스트럭쳐가 중단되어도 온라인 상태를 유지할 수 있습니다.
- **표준 기반**: 상호 운용성과 작업 로드 이식성을 위해 클라우드 네이티브 서비스는 오픈 소스 및 표준 기반 기술에 기반하는 경우가 많습니다. 이를 통해 벤더 종속성이 줄어들고 이동성이 향상됩니다.
- **비즈니스 민첩성**: 네트워크에서 유연한 배포 옵션을 제공하며, 기존의 앱보다 작기 때문에 보다 쉽게 개발, 배포, 반복 작업을 수행할 수 있습니다.
- **자동화**: DevOps 자동화 기능을 사용하여 정기적으로 릴리스되는 소프트웨어 변경 사항을 지속적으로 전달 및 배포할 수 있습니다.
- **작동 중지 시간 없음**: Kubernetes와 같은 컨테이너 통합 관리자 덕분에 기본적으로 다운 타임 없이 소프트웨어 업데이트를 배포할 수 있습니다.

저희 팀은 이 중 독립성과 복원성, 비즈니스 민첩성 및 이미지 배포의 기반이 되는 Microservices와 Container 기술을 사용했습니다.

* Pitch url : <https://aws.amazon.com/ko/what-is/cloud-native/> , <https://www.oracle.com/kr/cloud/cloud-native/what-is-cloud-native/>

아키텍처 설명



Google Cloud & Docker hub & MongoDB Atlas

Docker Hub interface showing repositories for user polarbear96:

- polarbear96 / iljo-front-server: Contains: Image | Last pushed: 15 minutes ago
- polarbear96 / iljo-comment-server: Contains: Image | Last pushed: 10 hours ago
- polarbear96 / iljodiscoveryservice: Contains: Image | Last pushed: 12 hours ago
- polarbear96 / iljo-room-server: Contains: Image | Last pushed: 17 hours ago
- polarbear96 / iljoapigateway: Contains: Image | Last pushed: a day ago
- polarbear96 / iljo-user-server: Contains: Image | Last pushed: a day ago
- polarbear96 / iljo-userevaluation: Contains: Image | Last pushed: 2 days ago

VM 인스턴스 목록 (Google Cloud Platform):

인스턴스	관측 가능성	인스턴스 일정
iljo-comments	✓	us-central1-c 매월 \$22 절감
iljo-eureka	✓	us-central1-c 매월 \$24 절감
iljo-front	✓	us-central1-c 매월 \$22 절감
iljo-vm-instance	✓	us-central1-a 매월 \$11 절감 매월 \$22 절감

관련 작업:

- 백업 및 DR 살펴보기
- VM 모니터링

MongoDB Atlas Collections: iljomongodb.comments

QUERY RESULTS: 1-20 OF MANY

_id: ObjectId('...')	roomId: 1	userId: "kmitie45"	content: "다같이 영화보러 갈 생각"	createdAt: 2023-01-11T07:33:00	updatedAt: 2023-01-11T07:33:00	--v: 0
_id: ObjectId('...')	roomId: 5	userId: "cwoodburnqr"	content: "저도 주말에 뜻을 같아요"	createdAt: 2023-01-11T07:40:01	updatedAt: 2023-01-11T07:40:01	--v: 0

Docker를 통해서 container 이미지를 생성하고 Docker hub에 올려 배포할 수 있게 했습니다.

DB를 포함하여 모든 서버의 이미지를 클라우드에 업로드하여 서비스할 수 있게 했습니다.

MongoDB는 Atlas가 서비스하는 클라우드를 이용하여 서비스했습니다.

NLP(3단계) → ML

1 18만의 train 댓글 data, 5만 test 댓글 data

2 학습을 위한 데이터를 형태소로 분리

[['부작'], ['마지막'], ['마치'], ['러시아'], ['인형'], ['처럼'], ['전작'], ['을'], ['풀다']]

```
[[2066, 76, 989, 2144, 2262, 242, 1014, 3, 3257, 656, 226, 4729, 1988
```

4 자연어처리 한 문장들로 LSTM 기반 model 학습

```
=====>.] - ETA: 0s - loss: 0.3882 - accuracy: 0.8208WARNING:tensorflow:Can save best model  
=====] - 29s 10ms/step - loss: 0.3880 - accuracy: 0.8209 - val_loss: 0.3610 - val_ac  
  
=====>.] - ETA: 0s - loss: 0.3289 - accuracy: 0.8557WARNING:tensorflow:Can save best model  
=====] - 16s 8ms/step - loss: 0.3289 - accuracy: 0.8557 - val_loss: 0.3494 - val_ac  
  
=====>.] - ETA: 0s - loss: 0.3036 - accuracy: 0.8703WARNING:tensorflow:Can save best model  
=====] - 15s 7ms/step - loss: 0.3036 - accuracy: 0.8703 - val_loss: 0.3384 - val_ac  
  
=====] - ETA: 0s - loss: 0.2857 - accuracy: 0.8798WARNING:tensorflow:Can save best model  
=====] - 15s 7ms/step - loss: 0.2857 - accuracy: 0.8798 - val_loss: 0.3408 - val_ac
```

 5 test data를 이용하여 머신의 정확도 확인

*긍정 리뷰에 대한 긍정 부정 판독기 입니다.

1527/1527 [=====] - 5s 3ms/step - loss: 0.2774 - accuracy: 0.8850

테스트 정확도: 0.885

6 임의 문장으로 머신을 정확도 확인

```
sentiment_predict("아 오늘 집에 가고 싶다.")
```

1/1 [=====] - 0s 21ms/step
84.91% 확률로 긍정 리뷰입니다.

('84.91', '금정 린프')

 7 실제 Moinee 댓글 476개 긍정 부정 판독

0% | 0/476 [00:00<?, ?it/s] 1/1 [=====] - 0s 326ms/step
97.30% 활용률로 금전 리본입니다.

1/1 [=====] - 0s 15ms/step
0% | 1/476 [00:00<03:16 2.42it/s] 97.30% 활용률로 글전 리붓입니다.

1/1 [=====] - 0s 17ms/step
95.27% 확률로 극적 리본입니다.

1/1 [=====] - 0s 15ms/step
100% |██████████| 475/476 [00:55<00:00, 8.22it/s] 53.23% 확률로 부정 리뷰입니다

1/1 [=====] - 0s 22ms/step
99.33% 확률로 긍정 리뷰입니다.

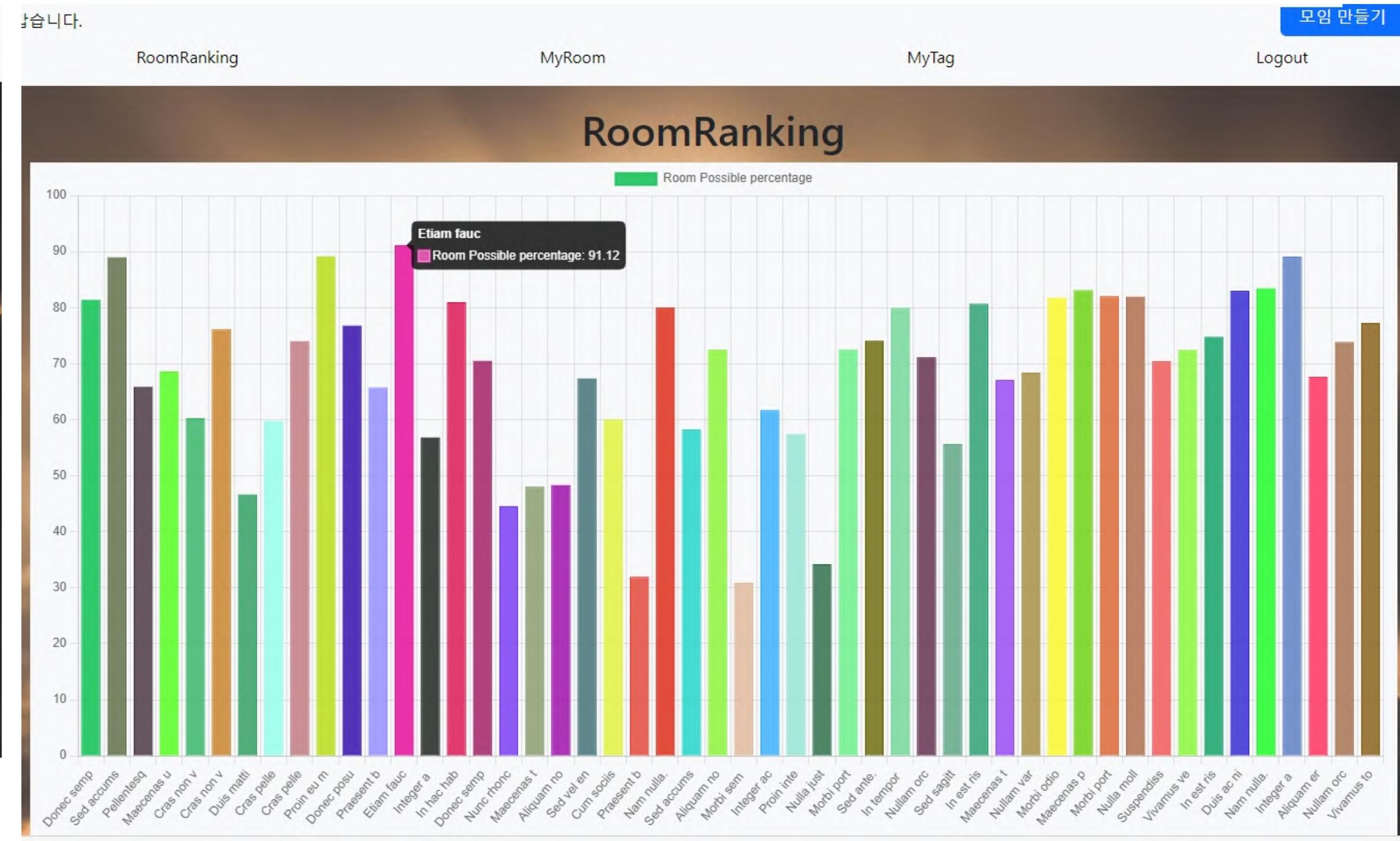
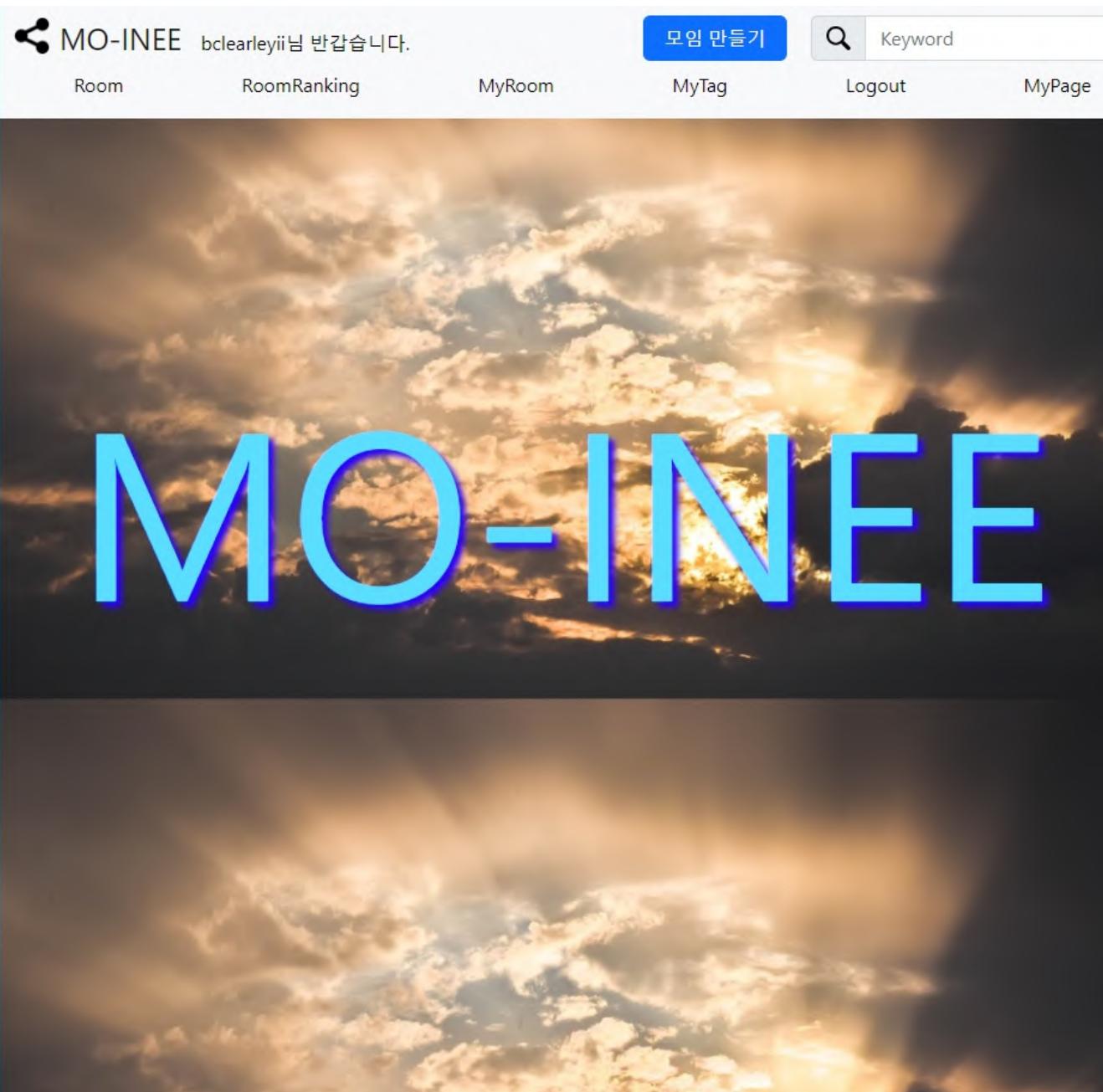
1/1 [=====] - 0s 17ms/step
100%|██████████| 476/476 [00:55<00:00, 8.63it/s] 99.33% 활용률로 글정 리붓입니다!

웹 애플리케이션 시연

*웹 어플리케이션을 시연하고 사진과 함께 기능을 설명합니다.



메인 페이지, 긍정적인 모임 랭킹



※ 위 그래프는 현재 모이니 사이트에 올라와있는 476개의 실데이터 기반 통계자료입니다.

회원가입, 로그인, 로그아웃

*회원가입 페이지입니다.

※회원가입※ **유효성검사기능**

ID 아이디 -체크박스 on/off 처리
아이디를 입력하세요

P 비밀번호 -다양한 input 속성 활용
영문, 숫자, 특수문자 포함 8자 이상

비밀번호확인

비밀번호를 다시 입력해주세요

⚠️ 이름
이름을 입력하세요

🎂 생년월일
연도-월-일

📍 관심사
운동 여행 독서 음악 게임 영화 음식

SEX 성별
남



반갑습니다, bclearleyii 님.

Room Sign-up

*로그인 페이지입니다.

로그인 -로그인에 성공하면 성공
아이디 알림과 함께 내비게이션
바 변경

bclearleyii 비밀번호

.....

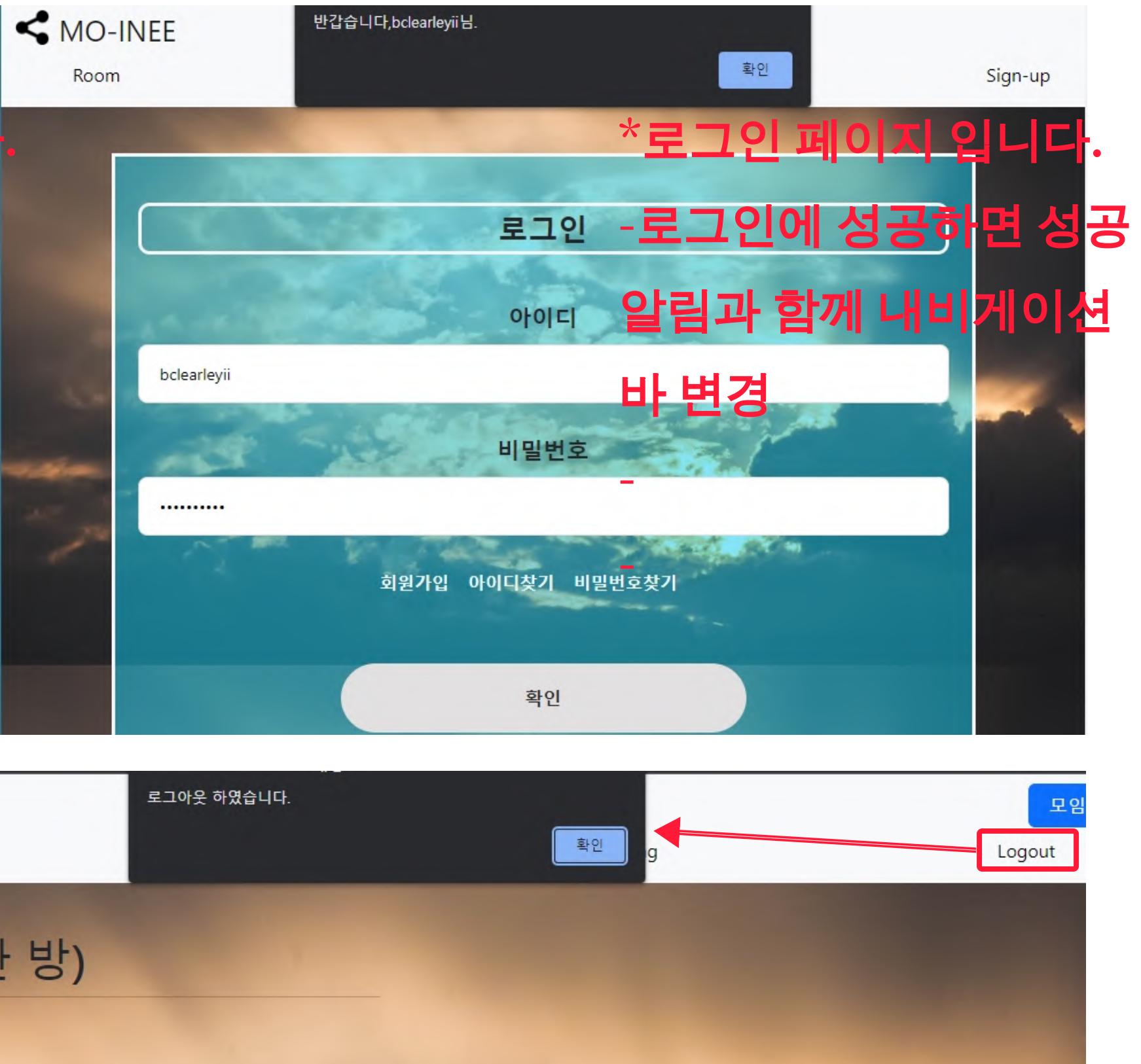
회원가입 아이디찾기 비밀번호찾기

확인

로그아웃 하였습니다.

확인 Logout

방)



프로필 수정

bclearleyii

Email: bclearleyii@google.ru
Phone: 010-1234-5678
Name: 문미정

※프로필수정※

프로필사진

파일 선택 선택된 파일 없음

이름

문미정

생년월일

2018-10-22

아이디

bclearleyii

비밀번호

*프로필 수정 페이지
-유저 ID를 통해서 DB에
서 정보 가져와 띄어놓기
-UPDATE 기능 구현

비밀번호

비밀번호를 수정하려면 입력하세요

비밀번호확인

비밀번호 확인

이메일

bclearleyii@google.ru

주소

인천광역시 도봉구 서초대90가 (민수강동)

관심사

운동□ 여행□ 독서□ 음악□ 게임□ 영화□ 음식□

OK 확인

취소

내가 만든 모임, 내가 짐한 모임

RoomRanking MyRoom MyTag

bclearleyii

Email: bclearleyii@google.ru
Phone: 010-1234-5678
Name: 문미정

3

My Room

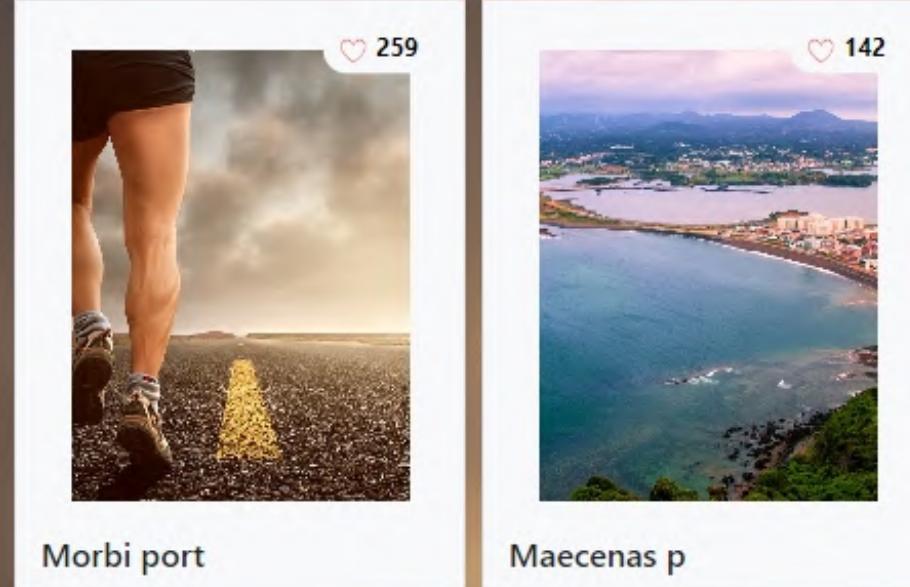


Donec semp
광주광역시 성북구
2023. 01. 12 14시 19분

MO-INEE bclearleyii님 반갑습니다.

Room RoomRanking

Room Tag(내가 짐한 방)



259 142

Morbi port Maecenas p

MO-INEE

Let's Introduce MO-INEE

This site has opened since 2023-01-17

*내가 만든 모임을 보여주는 페이지입니다.

X를 누르면 모임을 삭제할 수 있습니다.

Pitch

*내가 짐한 모임도 볼 수 있습니다.

하트를 눌러서 추가하거나 삭제할 수 있습니다.

모임방 생성

*방 만들기 페이지입니다.

다양한 Input 속성을 사용해봤고 제한인원은 20명

만갑습니다.

입니다.

RoomRanking

MyRoom

MyTag

모임 만들기

Logout



방 생성 **New**

제목

태그

운동 여행 독서 음악 게임 영화 음식

모임날짜

장소

제한인원

내용

MyRoom MyTag

연도-월-일

장소

제한인원

내용

*방에 대한 소개가 나옵니다.
썸네일도 등록할 수 있습니다.

사진을 등록해주세요
 선택된 파일 없음

OK 확인

모임방 리스트

*생성된 모든 방을 보여줍니다.

왼쪽 카테고리별로 볼 수 있는 기능도 있습니다.

커서를 방에 갖다 대면 세부정보도 볼 수 있습니다.

The screenshot shows a user interface for a social networking or meeting platform. At the top, there's a header with the logo 'MO-INEE' and a message 'bclearleyii님 반갑습니다.' Below the header are navigation links: 'Room', 'RoomRanking', 'Logout', and 'MyPage'. A search bar with the placeholder 'Keyword' is also present. On the left side, there's a sidebar with a list of categories: 'total', 'Trip/Going Out', 'Study Groups', 'Flashmob', and 'Sports/Activity'. The main area displays a grid of room thumbnails. Each thumbnail includes a small image, the room name (e.g., 'Sed accums'), the location ('대전광역시 동작구'), and the date ('2023. 01. 12 14시 19분'). Some thumbnails have additional details like a profile picture, social media sharing icons, and a 'Profile' button. A central modal window is open over the grid, showing a detailed view of a room named 'Sed accums' with a thumbnail of a globe and some text. The overall theme is a clean, modern design with a focus on social interaction and community building.

모임방 상세 정보, 댓글

MO-INEE bclearleyii님 반갑습니다.

Room RoomRanking MyRoom MyTag

대전광역시 동작구 오금가
모임 시간 2023-01-12T14:19:53

Sed accums

295 댓글 개수 나타내기

참여인원

- cogus6r
- gkesby0
- hungerer6v
- kmitie45
- mbroomer5

어떤 모임 인가요?
Nulla neque libero convallis eget eleifend luctus

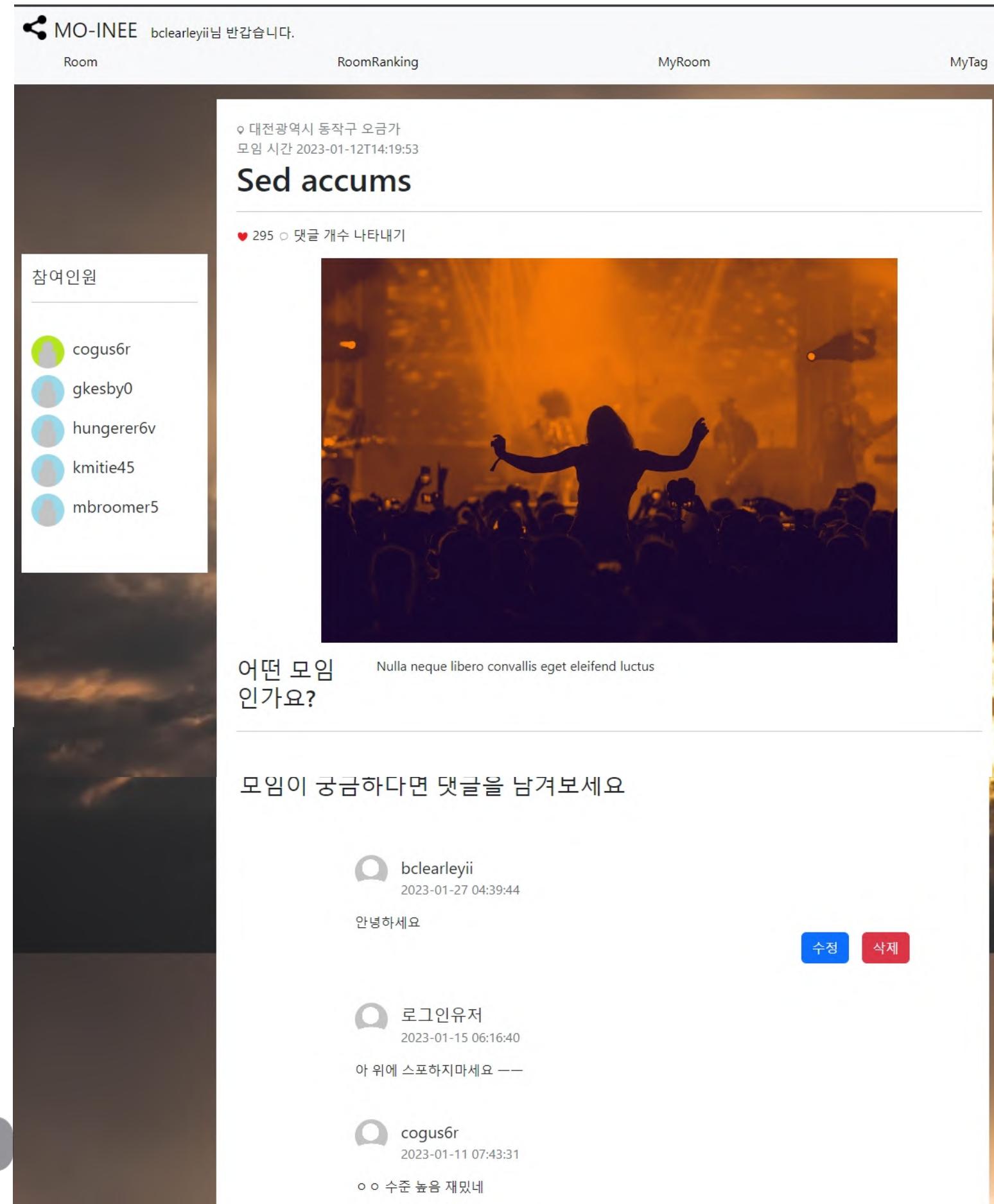
모임이 궁금하다면 댓글을 남겨보세요

bclearleyii 2023-01-27 04:39:44
안녕하세요

로그인유저 2023-01-15 06:16:40
아 위에 스포하지마세요 ——

cogus6r 2023-01-11 07:43:31
○ ○ 수준 높음 재밌네

Pitch



rleyii님 반갑습니다.

댓글이 등록되었습니다.

RoomRanking

난 보는눈이 낮은건가?어제 블랙아담 봤는데 재밋던데;; ㄹㅇ 대사 살짝 유치한거 빼곤 전투씬이나 중간중간 개그도 재밋었음;;

kmitie45 2023-01-11 07:43:31
다같이 영화보러 갈 생각에 벌써 기대되네요 ㅎㅎ

kmitie45 2023-01-11 07:33:08
다같이 영화보러 갈 생각에 벌써 기대되네요 ㅎㅎ

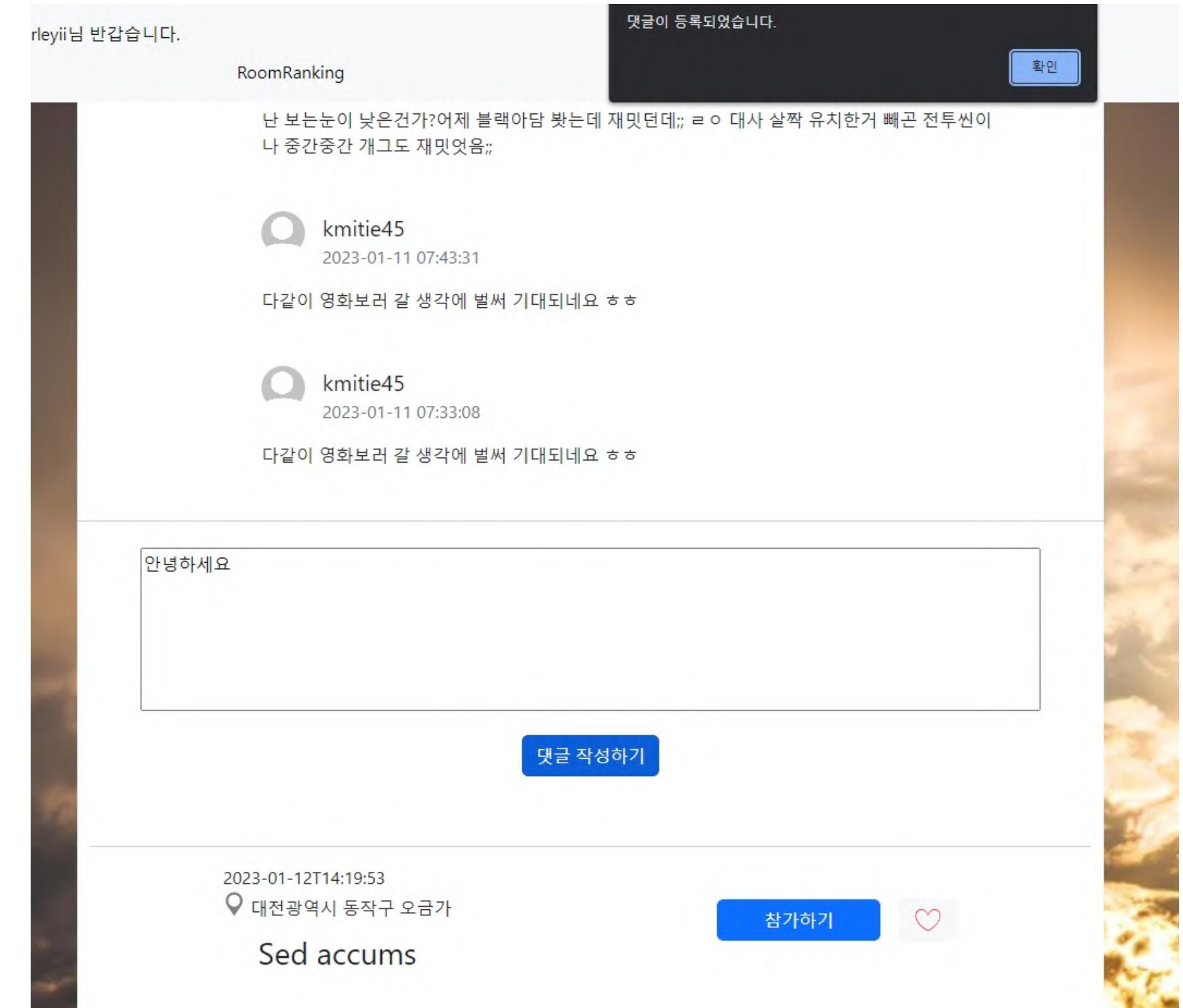
안녕하세요

댓글 작성하기

2023-01-12T14:19:53
대전광역시 동작구 오금가

참가하기 

Sed accums



*방에 입장한 상태입니다.
댓글 CRUD를 구현하였습니다.



III. 정리 및 마무리

문제 해결 고민

문제 & 교훈

처음 접한 기술 사용의 어려움

교훈 : 처음부터 잘할 수 없습니다.
끊임 없이 배우고 깨달음을 얻어야
함을 느꼈습니다.

해결방안

검색과 질문을 통해 빠른 습득

stackoverflow와 구글 및 인강(인프런, 코딩
애플, 유튜브)을 적극 활용해서 기술을 사용하
는 방법을 배움으로써 문제를 해결했습니다.
이 과정 때문에 일정이 지연되는 것이 걱정되었
지만, 프로젝트의 완성도를 높이겠다는 생각으
로 꾸준히 공부했습니다. 또한 배운 기술을 공
유하기 위해 Notion에 Reference 공간을 만들
어 쉽게 찾아볼 수 있게 하였습니다.

문제 해결 고민

문제 & 교훈

예상보다 더딘 진행

교훈 : 부족한 실력은 부족한 만큼
몸으로 메꿔서라도 해결을 해야할
때가 있음을 느꼈습니다.

해결방안

할 수 있는 조치는 다시도하여 해결

혼자 해결을 못 하는 부분이 있으면 바로 다른 팀원에게 빠른 도움 요청을 했고 바로바로 협업 했습니다. 팀장은 9:30 ~ 18:30인 코딩 시간을 7:30 ~ 18:30, 21:00 ~ 23:00로 코딩 시간을 늘렸습니다. 팀원들 또한 주말에 모이거나 늦게 혹은 일찍 나와 같이 협업 해서 해결하지 못한 부분을 해결했습니다.

문제 해결 고민

문제 & 교훈

서로 다른 생각

교훈 : 서로의 생각을 분명하게 파악 하는게 얼마나 중요한 부분인지 느꼈습니다.

해결방안

상시 회의와 소통을 통해 해결

오전 10시 30분 ~11시 30분에 회의를 2주간 진행 했습니다. 팀원들이 생각하는 방향을 확인했고 실제로 서로 다른 의견을 가지고 있음이 드러났습니다. (예를 들어 ERD 설계를 할 때 User relation과 Room relation의 관계가 방 생성 관계로 알고 있는 팀원과 방에 소속된 관계로 알고 있는 팀원으로 나뉘었습니다.) 팀원들의 의견을 적극 반영해 절충안을 찾을 수 있었습니다.

문제 해결 고민

문제 & 교훈

단순한 디버깅 = 2p X 4h

교훈 : brainstorming의 중요성

포기하지 않는 오기

해결방안

코드 수정 후 모든 작동 체크

이를 해결하기 위해 문제가 되는 파트를 만날 때마다 **2명이상**이서 계속 팀을 이뤄 **error의 원인을 도출**하고 **searching** 하여 평균 4시간을 쓸어 부어 **문제를 해결했습니다.**

문제 해결 고민

문제 & 교훈

검색해도 해결법이 안 나옴

교훈 : 검색 시 내가 원하는 요구사항은 정확해야 해야 하고 오타 없이 작성할 때마다 느꼈습니다.

해결방안

구글링과 영어 공부의 중요성

이를 해결하기 위해 영어로 검색하고 해석이 안 되면 한국어로 번역기를 돌려서 필요한 내용을 얻을 수 있도록 하였습니다. 구글링 시 필요한 keyword 입력 및 필터를 걸어서 내가 원하는 결과가 나올 수 있게 검색을 했습니다.

Project Reference, github

Docker

- Docker install ubuntu 18.04 : <https://brownbears.tistory.com/174>
- node docker images : <https://velog.io/@dhk22/Docker를-이용한-간단한-node-app-만들기>
- react docker images : <https://codechacha.com/ko/dockerizing-react-with-nginx/>

Linux

- background service : <https://brownbears.tistory.com/174>
- background service2 : <https://tyson.tistory.com/88>

Cors

- exposed Header : <https://kyung-a.tistory.com/33>
- get Header from res.data : <https://www.freecodecamp.org/news/how-to-setup-jwt-authorization-and-authentication-in-spring/>

API-Gateway

- set uri of service : <https://stackoverflow.com/questions/70974502/gateway-service-500-server-error-for-http-get>

FeignClient

- FeignClient uri : <https://stackoverflow.com/questions/43733569/how-can-i-change-the-feign-url-during-the-runtime>

Data Modeling

- Concept of data modeling and drawing ER Diagram : <https://inpa.tistory.com/>
- Relation Data Model key concept and principle : [관계형 데이터 모델링 - 생활코딩](#)

JPA

- mysql connect : <https://memostack.tistory.com/155>
- composite key entity : <https://jforj.tistory.com/84>
- MySQL(GCP) connect Springboot : <https://coding-is-fun.tistory.com/10>
- convert csv to sql : <https://products.appose.app/cells/ko/conversion/csv-to-sql>
- Caution of Operating DB(DDL) : <https://smpark1020.tistory.com/140>

API

- What is REST API? <https://www.youtube.com/watch?v=iOueE9AXDQQ&t=214s>
- The way of communication between machines - REST API <https://www.youtube.com/watch?v=PmY3dWcCxXI>
- Is it okay with that way of REST API? <https://tv.naver.com/v/2292653>
- 'RESTful' Now I get it, (REST API past test questions in the 122nd information management technician) <https://www.youtube.com/watch?v=CBXIJjJOHerU>
- Designing legit REST API for just 5 mins <https://www.youtube.com/watch?v=4DxHX95Lq2U&list=WL&index=69>
- Look into REST API | API <https://www.youtube.com/watch?v=MK7aSFA5ohI&list=WL&index=68>
- good reference site when u Learn JSON <http://www.tcp school.com/json/intro>

InteliJ

- InteliJ Github : <https://brunch.co.kr/@mystoryg/168>

Parsing

- Create Dummy Data : <https://www.mockaroo.com/>
- Create Dummy Data with Python1: <https://www.lesstif.com/python/python-faker-125305343.html>
- Create Dummy Data with Python2 : https://velog.io/@seanlee/01_커머스-더미데이터-생성
- Can get any answer from chatAI : <https://openai.com/blog/chatgpt/>

React

- React-Router-Dom install : <https://www.npmjs.com/package/react-router-dom>
- React-Router-Dom WebSite : <https://reactrouter.com/en/main>
- what is Router? : <https://goddaehee.tistory.com/305>
- Bootstrap : <https://getbootstrap.com/>
- Tailwind CSS : <https://tailwindcss.com/>
- styled-components : <https://styled-components.com/>
- AOS : <https://michalsnik.github.io/aos/>
- Bootstrap spacing : <https://velog.io/@leyuri/Bootstrap-공백-Spacing-주는-법-mt-mb-ml-mr-mx-my-pt-pb-pl-pr-px-py>
- Client Token management : <https://velog.io/@yaytomato/프론트에서-안전하게-로그인-처리하기>
- Navigate : <https://velog.io/@jin0106/React-Link로-데이터-전달하기>

CSS

- popover, transition, … : <https://www.w3schools.com/>

•

PPT

- <https://pitch.com/>

BackEnd

- TestCode : <https://github.com/SeoulPolarBear/ILJOInsertTest.git>

BackEnd

- User : https://github.com/fastwon/user_server.git
- Room : <https://github.com/Whiteheim/heim.git>
- User_evaluation : <https://github.com/SeoulPolarBear/ILJOUUserEvaluation.git>
- Comment : <https://github.com/SeoulPolarBear/ILJOComment.git>
- Api-gateway : <https://github.com/SeoulPolarBear/iljo-api-gateway.git>
- Eureka : <https://github.com/SeoulPolarBear/ILJODiscoveryService.git>

FrontEnd

- React : <https://github.com/SeoulPolarBear/ILJOFrontEnd.git>
- multipart_test : <https://github.com/fastwon/test-react-app.git>

*개발을 하면서 참고한 약 50건의 레퍼런스를 정리해놓았습니다.
다른 팀원들도 함께 공부할 수 있도록 하였고, 어떤 기술을 사용해서 어떤 방식으로 개발을 했는지 참고할 수 있습니다.

※ 클릭시 해당 사이트로 이동합니다

후기

*개발을 하며 느낀점들을 서로 정리하여 공유해보았습니다.



성환

저는 기술을 사용하고 협업을 함에 있어 유사 경험을 했는지가 매우 중요하다고 생각합니다. 따라서 저희 팀원이 backend server를 한번이라도 만들어보는 경험을 하기위해 Microservices 기술을 도입했습니다. 하지만 그렇게 되면 서버는 총 7개… '혼자서 할 수 있었을까?'라는 생각이 들었던 프로젝트였습니다. 팀원들이 모두 열정적으로 임해줘서 가능성이 보였던 프로젝트였던 것 같습니다. 새로운 도전은 팀원들이 있어서 가능했습니다. 팀원의 소중함을 느낄 수 있었고 새로 배운 기술들을 많이 사용하여 단계 성장할 수 있어서 좋았습니다.

Pitch



한수

이번 프로젝트에서는 처음 사용하는 기술 위주로 작업했기 때문에 어려움이 많았습니다. 특히 프론트를 구현할 때 사용한 리액트는 작동 방식이 기존의 기술들과 정말 다르게 느껴져서 많이 어려웠습니다. 처음에는 막막했지만 인터넷 강의와 구글링을 통해 공부하고, 지식이 많은 팀장님의 도움도 받으며 작업해나갔습니다. 이후 다른 팀원들이 만든 결과물과 취합할 때마다 눈에 띄게 완성 되어가는 프로젝트를 보면 정말 뿌듯했습니다. 이번 팀 프로젝트를 통해 팀 협업의 장점을 느낄 수 있어서 좋았습니다.

후기



성훈

개발에 무지했던 나를 팀장님이 잘 이끌어주셔서 개발 초보자 단계까지 올 수 있었습니다. 모놀리틱? MSA? 유레카서버? 아무것도 알아들을 수 없었지만 5주간 코딩에 빠져들며 많은것을 배울 수 있었습니다. 아직 완벽하게 이해했다고는 할 수 없어 아쉬움이 남지만 이 때의 기억을 가지고 꾸주히 정진하겠습니다.



주용

가장 빠르게 개발을 배우는 방법은 '일단 만들고 보는 것'이라 느꼈습니다. 생전 처음 보는 내용과 기술도 일단 시작하면 절반은 해결한 것입니다. 게다가 혼자가 아니라 동료와 함께라면 어떤 문제와 난관도 헤쳐나갈 수 있을 것입니다.



기훈

더미데이터 작업을 통해 반복적인 노동에 가까운 작업이지만 결과를 위해 꼭 필요한 일도 있다는 것을 몸소 깨닫게 되었습니다. 그리고 서버 하나를 맡아서 entity 부터 controller까지 만들어 봄으로써 rest api를 이해할 수 있었습니다. 이해가지 않았던 많은 부분들은 동료들의 도움으로 해결할 수 있었고, 덕분에 협업의 중요성을 제대로 느낄 수 있었습니다.

Our Project is Continued...

- 리팩터링을 통한 간결하고 효율적인 코드 수정
- 다른 팀원이 맡은 기능을 공부하고 분석하여 전체적인 웹 사이트 동작과정 학습
- 깔끔하고 세련된 UI로 수정
- TypeScript를 배우면서 JS 코드를 TS코드로 구현

Thank You