



# 포팅 매뉴얼

태그 문서

1. Version

2. EC2 Server

1) Docker

2) Nginx

3) Jenkins

4) MySQL

3. 외부 API

1) 카카오

4. Settings

1) application.yml

2) .env

5. 시연 시나리오

## 1. Version

- React 18.1.0
- Node.js 20.10.0
- Java 17
- Springboot 3.2.1
- MySQL 8.0

## 2. EC2 Server

### 1) Docker

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" |  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

## 2) Nginx

```
sudo apt install nginx -y  
sudo systemctl status nginx  
  
sudo apt-get install letsencrypt  
sudo apt-get install certbot python3-certbot-nginx -y  
sudo certbot --nginx
```

## 3) Jenkins

```
sudo docker exec -it jenkins bin/bash  
docker  
  
apt-get update  
apt-get install ca-certificates curl gnupg lsb-release -y  
mkdir -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg  
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/ke  
$(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list  
apt-get update  
apt-get install docker-ce docker-ce-cli containerd.io docker-  
containerd
```

## 4) MySQL

```
sudo apt-get update  
sudo apt-get install mysql-server -y  
sudo mysql -u root  
  
CREATE DATABASE stella;  
SHOW DATABASES;  
  
CREATE USER 'stella'@'%' IDENTIFIED BY 'stella2401@@';  
FLUSH PRIVILEGES;
```

### 3. 외부 API

#### 1) 카카오

- 카카오 디벨로퍼 로그인(<https://developers.kakao.com/>) 후 내 애플리케이션 이동



- 카카오 애플리케이션 추가

The screenshot shows a dashboard titled "전체 애플리케이션 (2)". It features a search bar at the top right labeled "애플리케이션 이름". Below the search bar is a blue header bar with a plus sign icon and the text "애플리케이션 추가하기". The main content area displays two application cards. The first card is for "ssafy10" (ID 965745, OWNER, Web). The second card is for "TravelShooter" (ID 965727, OWNER). Both cards include an "APP" icon.

- REST API 키 확인

The screenshot shows the "ssafy10" application details page. At the top, there is a "설정" (Settings) button. Below it is a table titled "액 키" (Key) with four rows: "네이티브 앱 키", "REST API 키", "JavaScript 키", and "Admin 키". Each row has a redacted value.

액 키	
네이티브 앱 키	[REDACTED]
REST API 키	[REDACTED]
JavaScript 키	[REDACTED]
Admin 키	[REDACTED]

- 카카오 로그인 활성화

내 애플리케이션 > 제품 설정 > 카카오 로그인

APP ssafy10

카카오 로그인 ON

활성화 설정

카카오 로그인 API ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.  
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.  
상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

- Redirect URI 등록(테스트용 주소, 실제 서버 주소)

Redirect URI

Redirect URI	http://localhost:3000/login/kakao https://i10b209.p.ssafy.io/login/kakao http://localhost:3000/regist/kakao https://i10b209.p.ssafy.io/regist/kakao
--------------	--

삭제 수정

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

## 4. Settings

### 1) application.yml

- 위치: BE > src > main > resources > application.yml
- spring.datasource.url에 server주소 입력
  - 예) i10b209.p.ssafy.io

```
server:
  port: 8080
```

```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://{{server주소}}/stella?useSSL=false&serverTimezone=UTC
    username: stella
    password: stella2401@@
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
        show_sql: false
        defer-datasource-initialization: true
  sql:
    init:
      mode: always
    output:
      ansi.enabled: always
  servlet:
    multipart:
      max-file-size: 10MB
      max-request-size: 10MB
  jwt:
    secret: {{jwt_secret}}
  social:
    url: https://i10b209.p.ssafy.io
  kakao:
    key: {{kakao_key}}
  google:
    client-id: {{google_client-id}}
    client-secret: {{google_client-secret}}
  naver:
    client-id: {{naver_client-id}}
    client-secret: {{naver_client-secret}}
  email:
    address: {{email_address}}
    password: {{email_password}}
```

```
cloud:  
aws:  
  s3:  
    bucket: ssafy-stella-bucket  
  stack.auto: false  
  region.static: ap-northeast-2  
  credentials:  
    accessKey: {{credentials_accessKey}}  
    secretKey: {{credentials_secretKey}}
```

## 2) .env

- 위치: FE

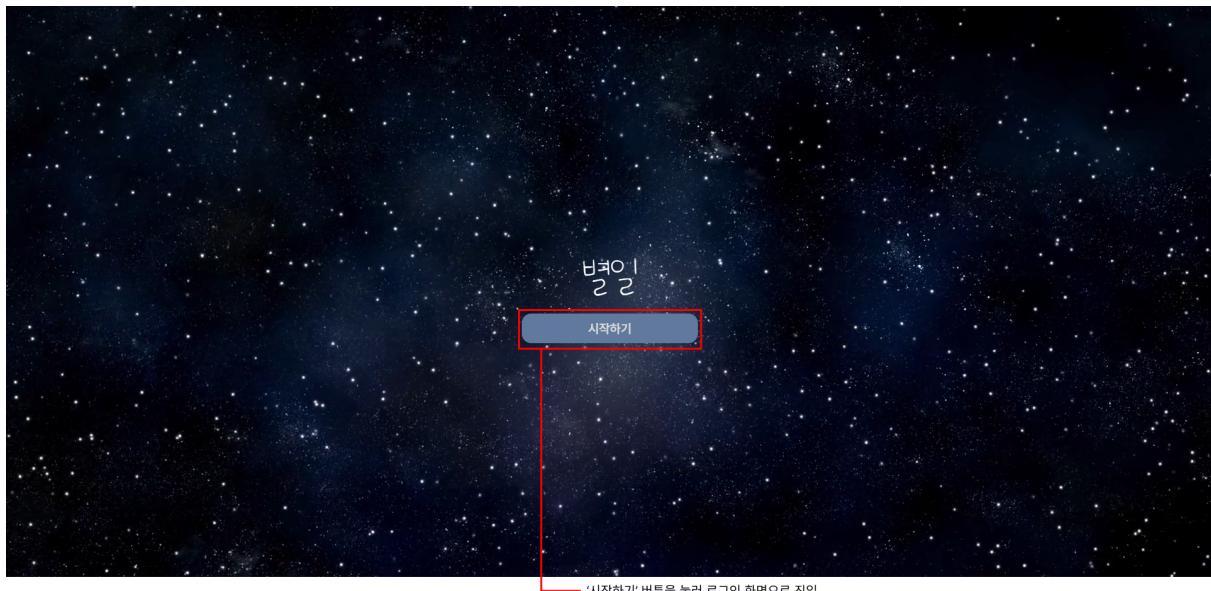
```
REACT_APP_API_URL=https://i10b209.p.ssafy.io/api  
REACT_APP_TTS_URL=https://i10b209.p.ssafy.io/tts-server  
REACT_APP_KAKAO_API_KEY=b1189e38a050b511cf3ae169bea072fe  
REACT_APP_KAKAO_REDIRECT_URI=https://i10b209.p.ssafy.io/api/m  
REACT_APP_KAKAO_JOIN_URI=https://i10b209.p.ssafy.io/api/membe  
RECOIL_DUPLICATE_ATOM_KEY_CHECKING_ENABLED=false  
REACT_APP_KAKAO_TEMP_URI=https://i10b209.p.ssafy.io  
REACT_APP_KAKAO_LOGIN_REDIRECT_URI=https://i10b209.p.ssafy.io  
REACT_APP_KAKAO_JOIN_REDIRECT_URI=https://i10b209.p.ssafy.io/
```

---

# 5. 시연 시나리오

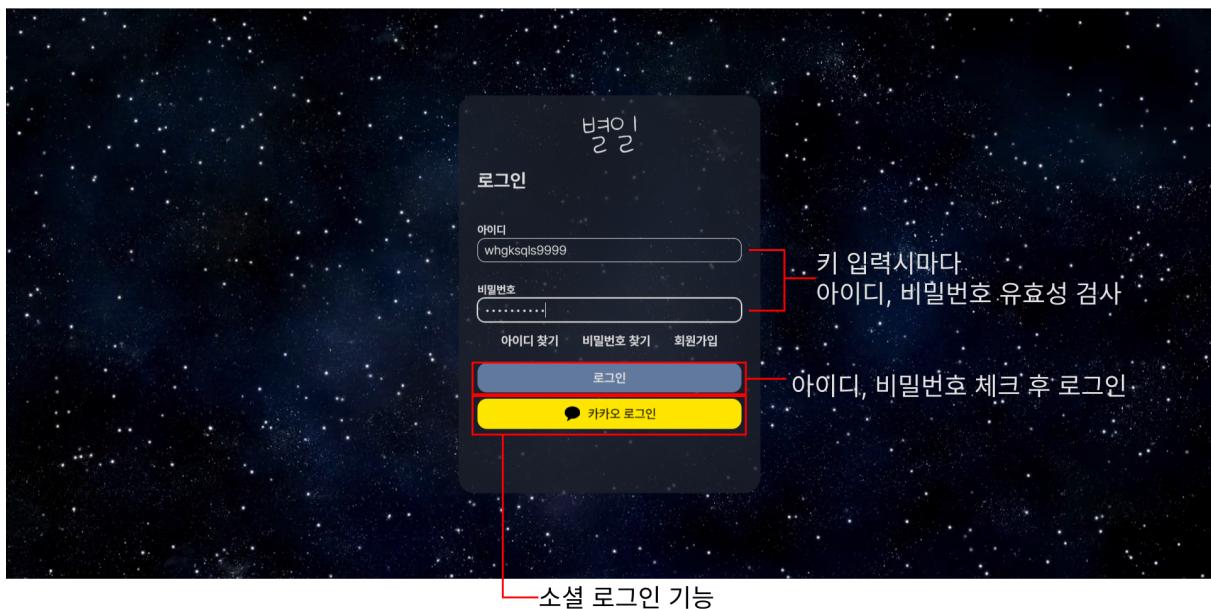
---

## 1. 시작



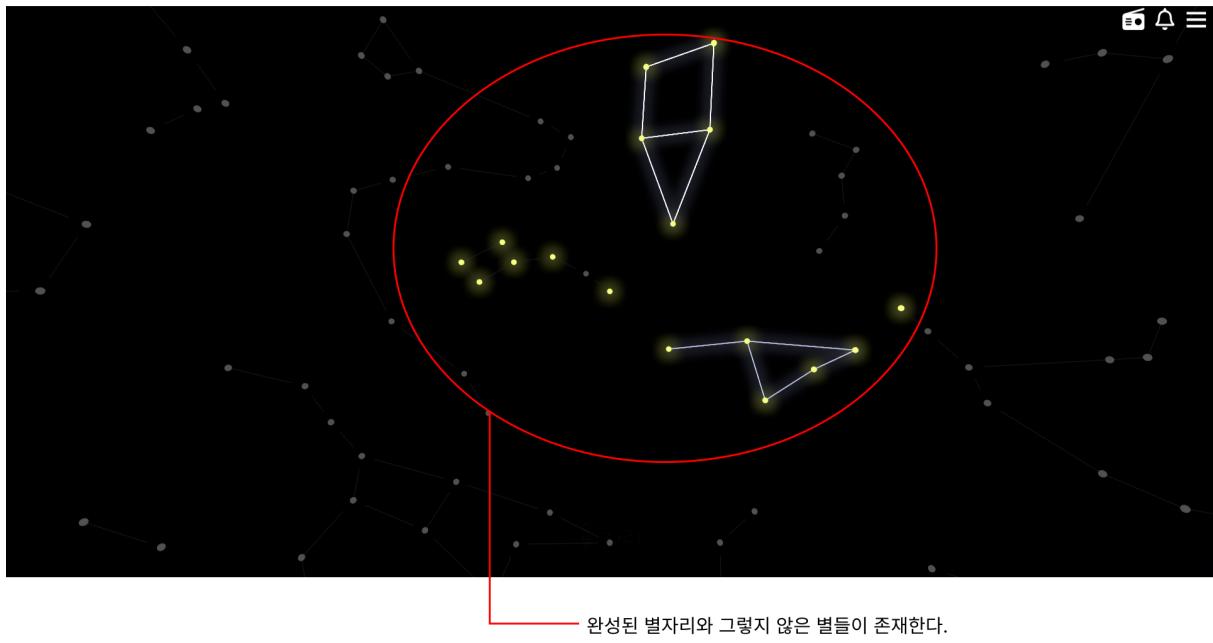
- 랜딩 페이지
- '시작하기' 버튼을 눌러 로그인 페이지로 진입

## 2. 로그인



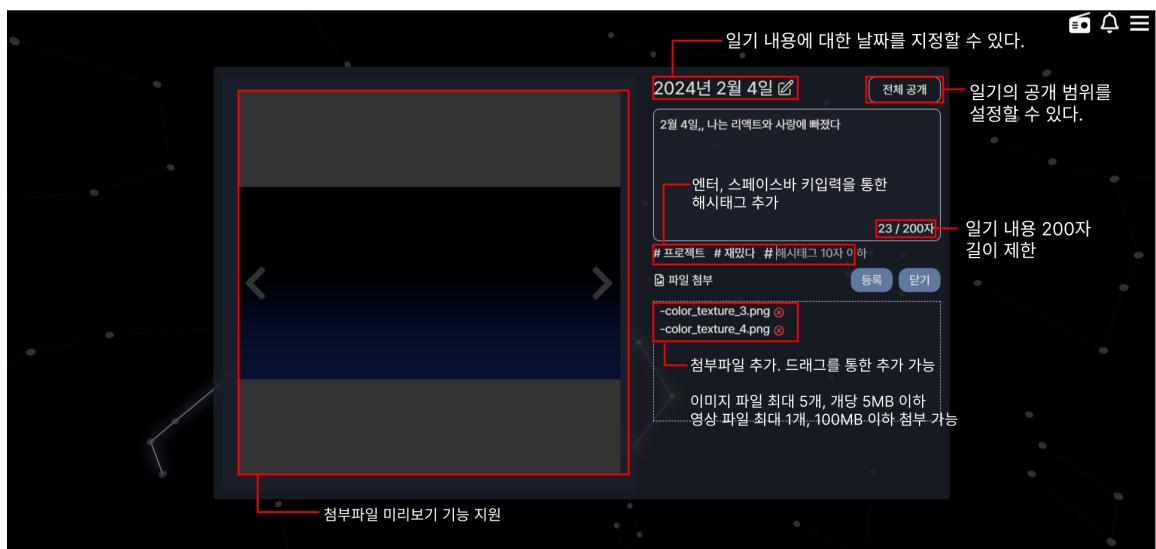
- 로그인 페이지
- 키 입력시마다 아이디, 비밀번호 유효성 검사
- 정해진 아이디, 비밀번호 (whgksqls9999 / !gksqls123) 입력 후 로그인 진행

## 3. 우주 공간



- 개인이 가지는 3D 공간
- 화면을 드래그해 주변 공간을 살펴본다

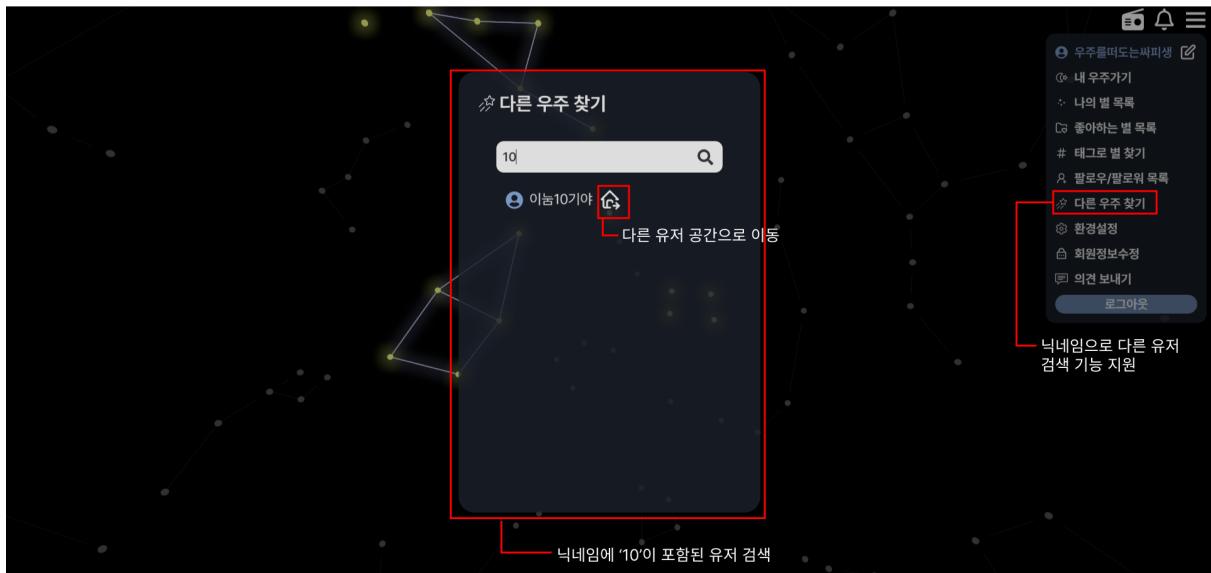
#### 4. 게시글 작성



- 일기 날짜를 **2월 4일**로 지정
- 일기 내용 입력 : “**2월 4일,, 나는 리액트와 사랑에 빠졌다**” - 미리 복사한 내용을 붙여 넣기
- 해시태그 입력 : “**프로젝트**”, “**재밌다**”
- 드래그로 파일 첨부 : 이미지 파일 1개, 동영상 파일 1개 각각 드래그로 첨부

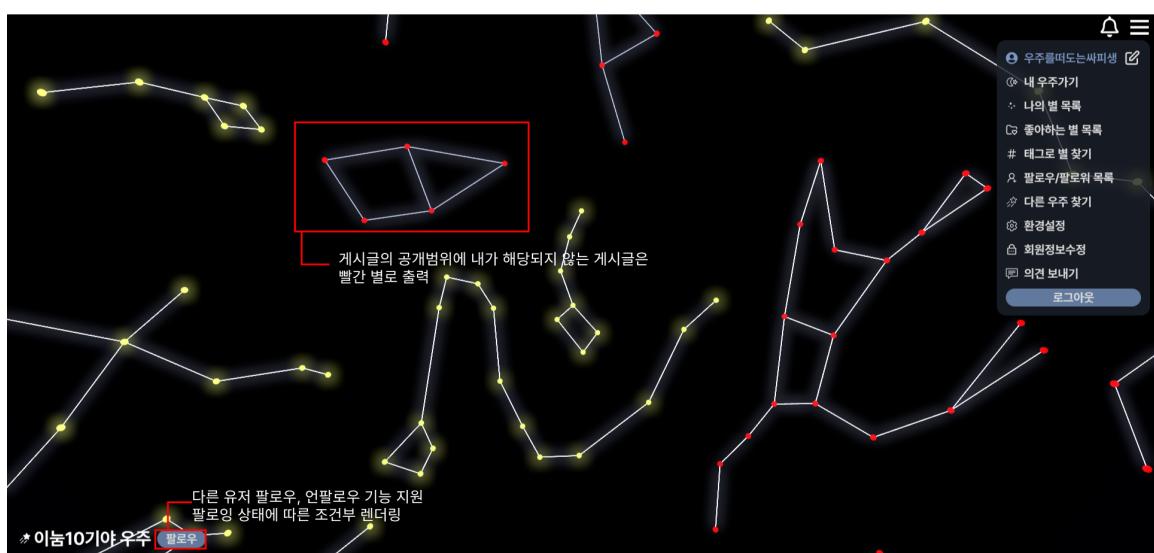
- 첨부파일 미리보기 둘러보기
- 파일 등록

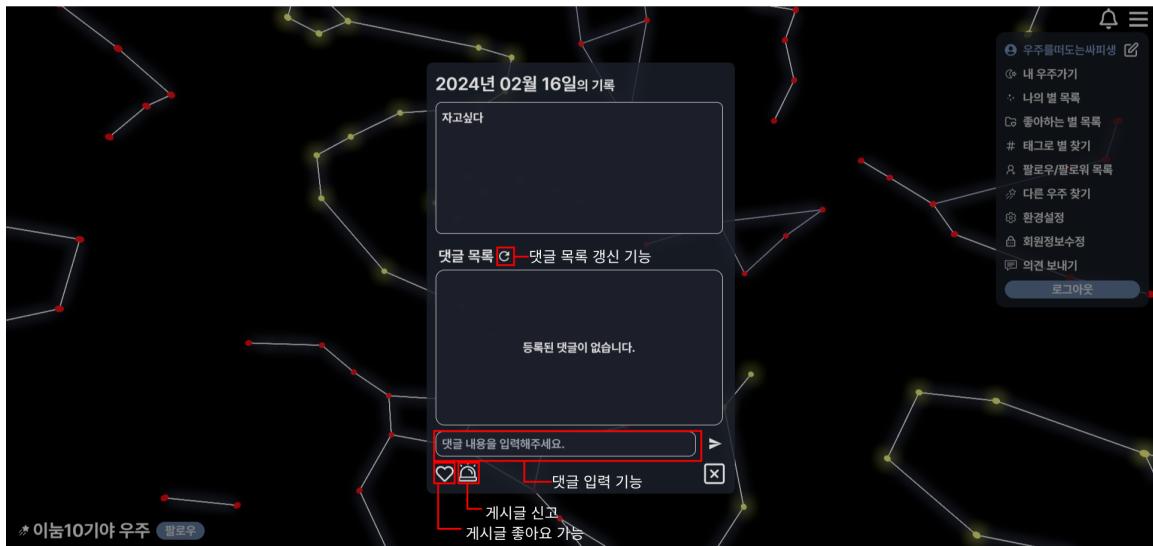
## 5. 다른 유저 닉네임 검색 및 공간 이동



- 사이드바 → 다른 우주 찾기 → '10' 검색
  - 엔터키를 입력하지 않아도 자동 검색 - 디바운스
- 이동 버튼 눌러서 다른 유저 공간으로 이동

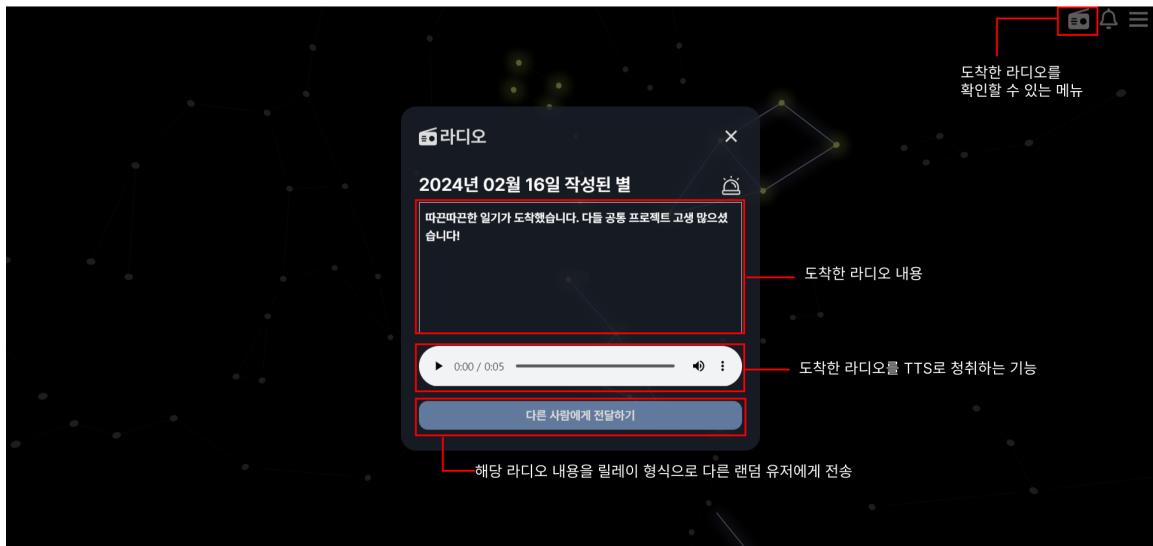
## 6. 다른 유저 공간 탐색





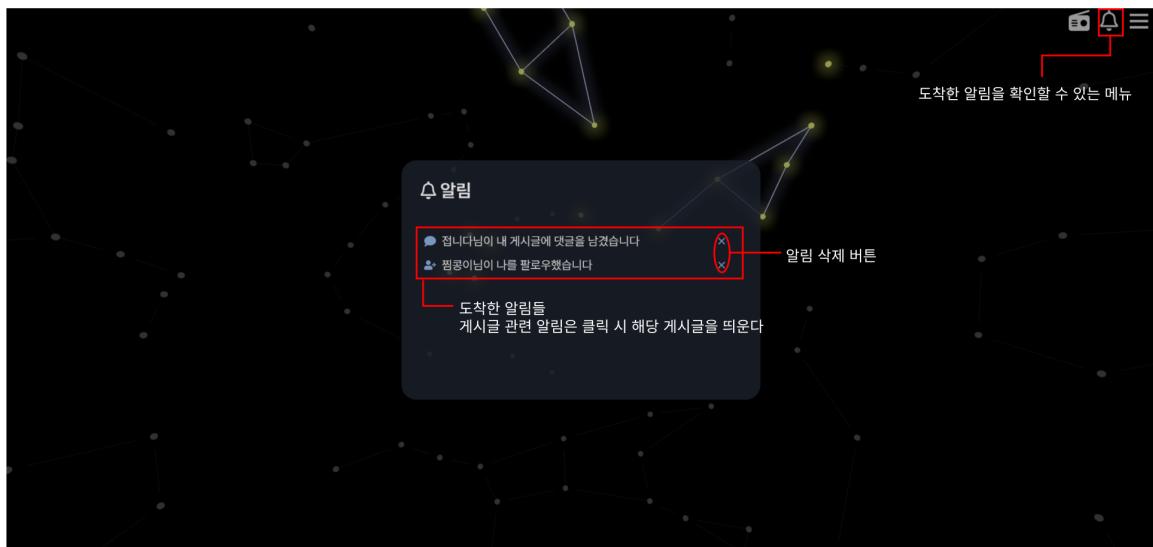
- 다른 유저 공간 한바퀴 탐색
- 노란 게시글 클릭해서 게시글 구경하기
- 팔로우 버튼 누르기
  - '팔로워 공개' 상태였던 빨간 별이 노랗게 변하는 장면 구경

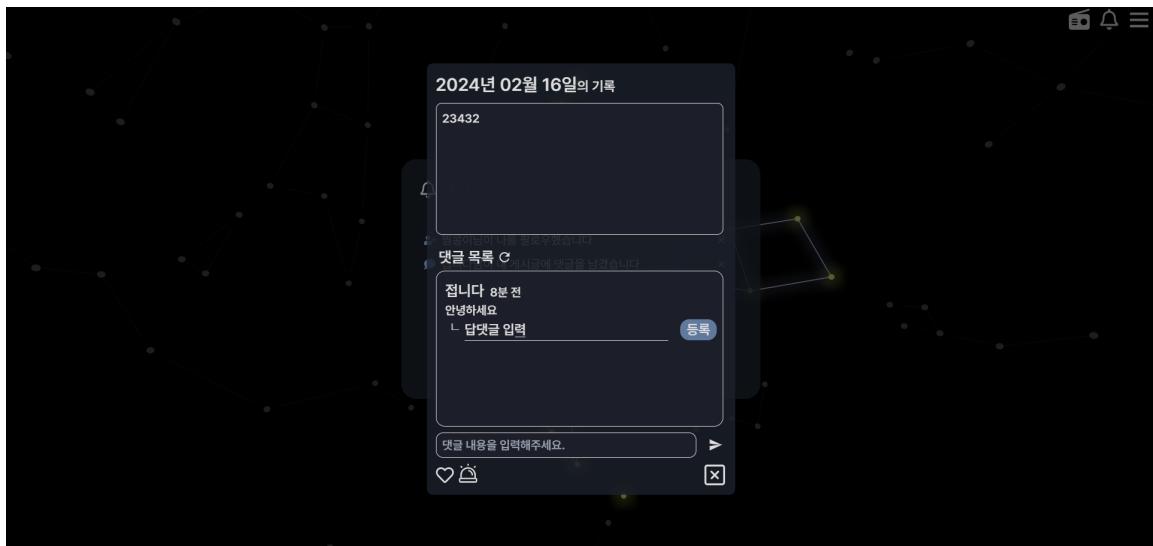
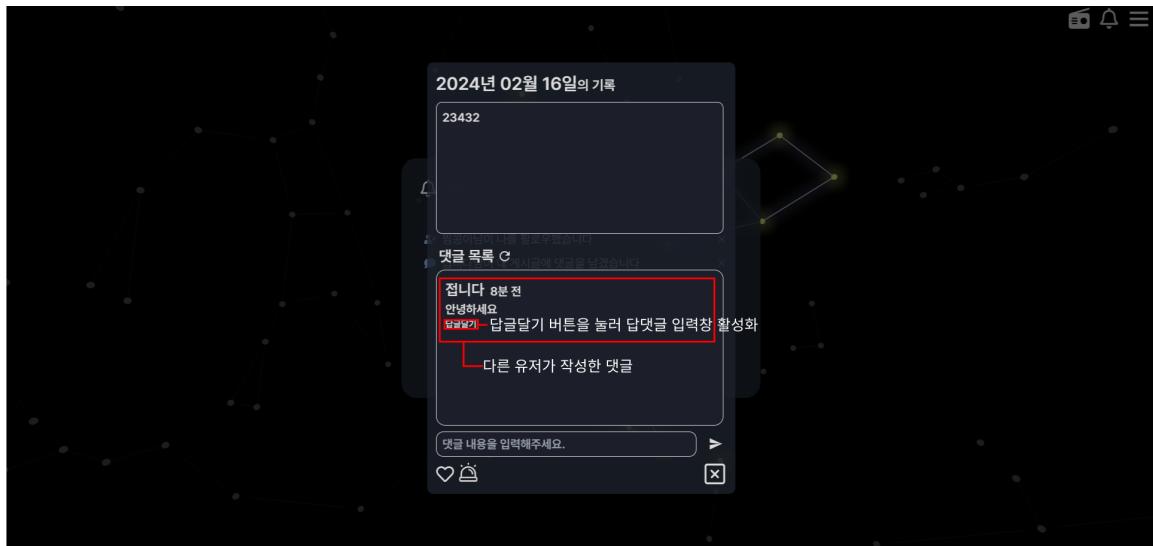
## 7. 라디오 확인



- 라디오 알림창 클릭
- 내용 확인 후 TTS 청취
- ~~다른 사람에게 전달하고 나가기~~

## 8. 알림 확인 및 답댓글 달기





- 알림 메뉴 클릭
- 알림 메뉴 내 댓글 알림 클릭
- 댓글 확인 후 답댓글 달기
- 라디오 송신

## 9. 별자리 만들기

- 글 하나 작성해서 별자리 생성해보기

## 10. 별자리 해제하기

- 방금 작성했던 글 하나 삭제해서 별자리 없애기

## 11. 의견 보내기



- 의견 보내기란 들어가서 “재미있는 서비스였어요, 만드느라 고생하셨어요!” 입력 후 의견 보내기