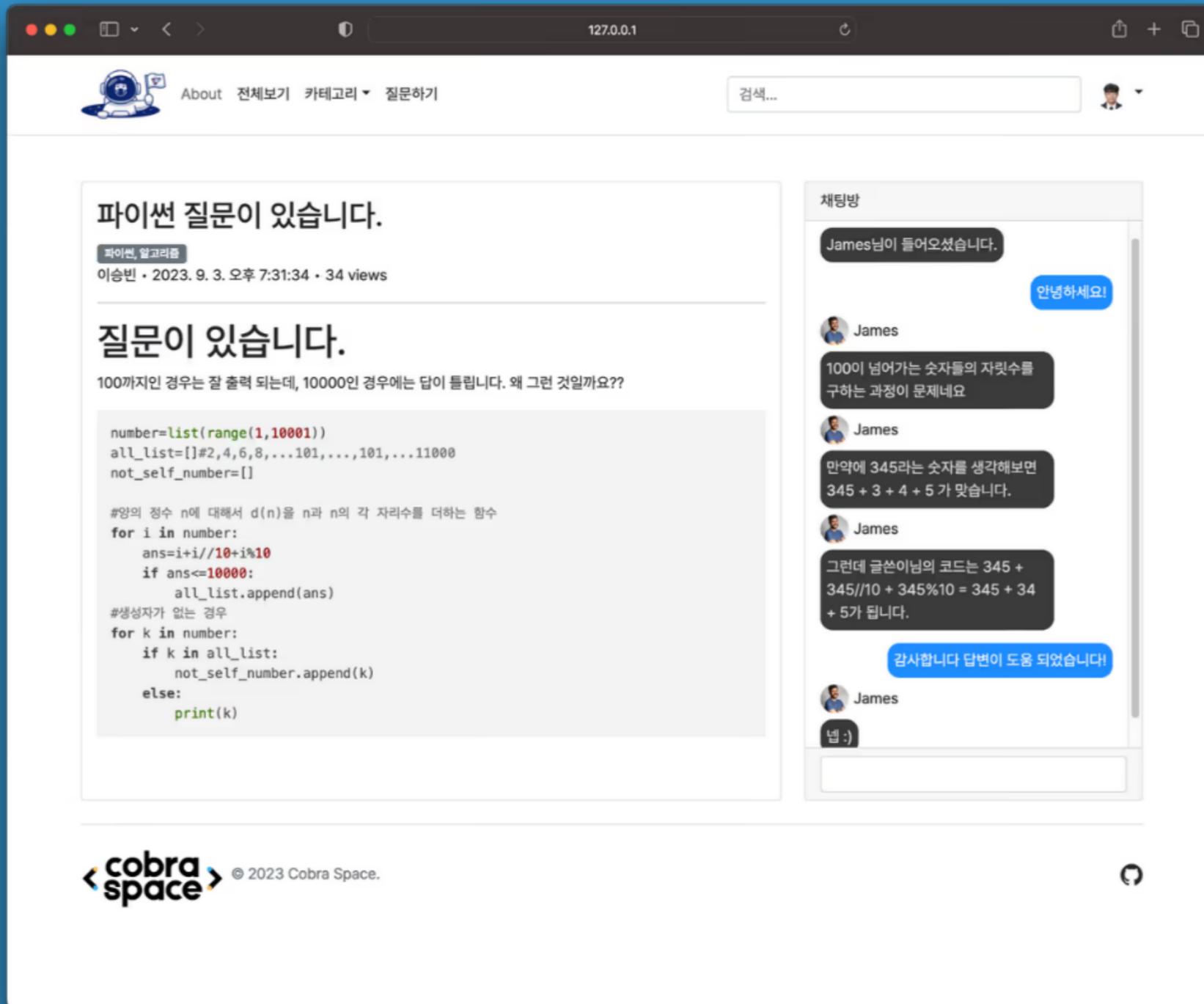


← cobra  
space →



# < cobra space >

# COding BRAinstorm



## ❌ 협력의 공간

코딩과 프로그래밍 분야에서 아이디어를 공유하고 협력하는 공간을 상징합니다.

개발자들의 지식 공유, 네트워킹 그리고 협력을 촉진합니다.

## ❌ 아이디어 탄생의 장

커뮤니티 내에서 새로운 아이디어와 프로젝트의 탄생을 돕습니다.

# Team 소개



이승빈

팀 리더 👑  
코브라 스페이스 Chat 구현



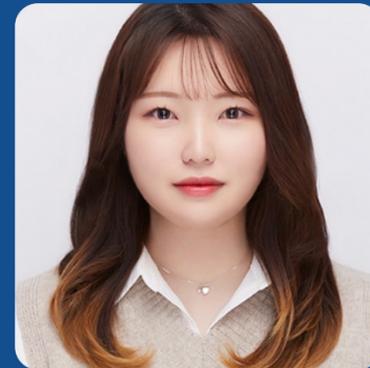
이준태

코브라 스페이스 Chat 구현



김권이

코브라 스페이스 User 구현



심민정

코브라 스페이스 User 구현



박승원

코브라 스페이스 Post 구현



이정효

코브라 스페이스 Post 구현



# COBRA SPACE

# 타임라인



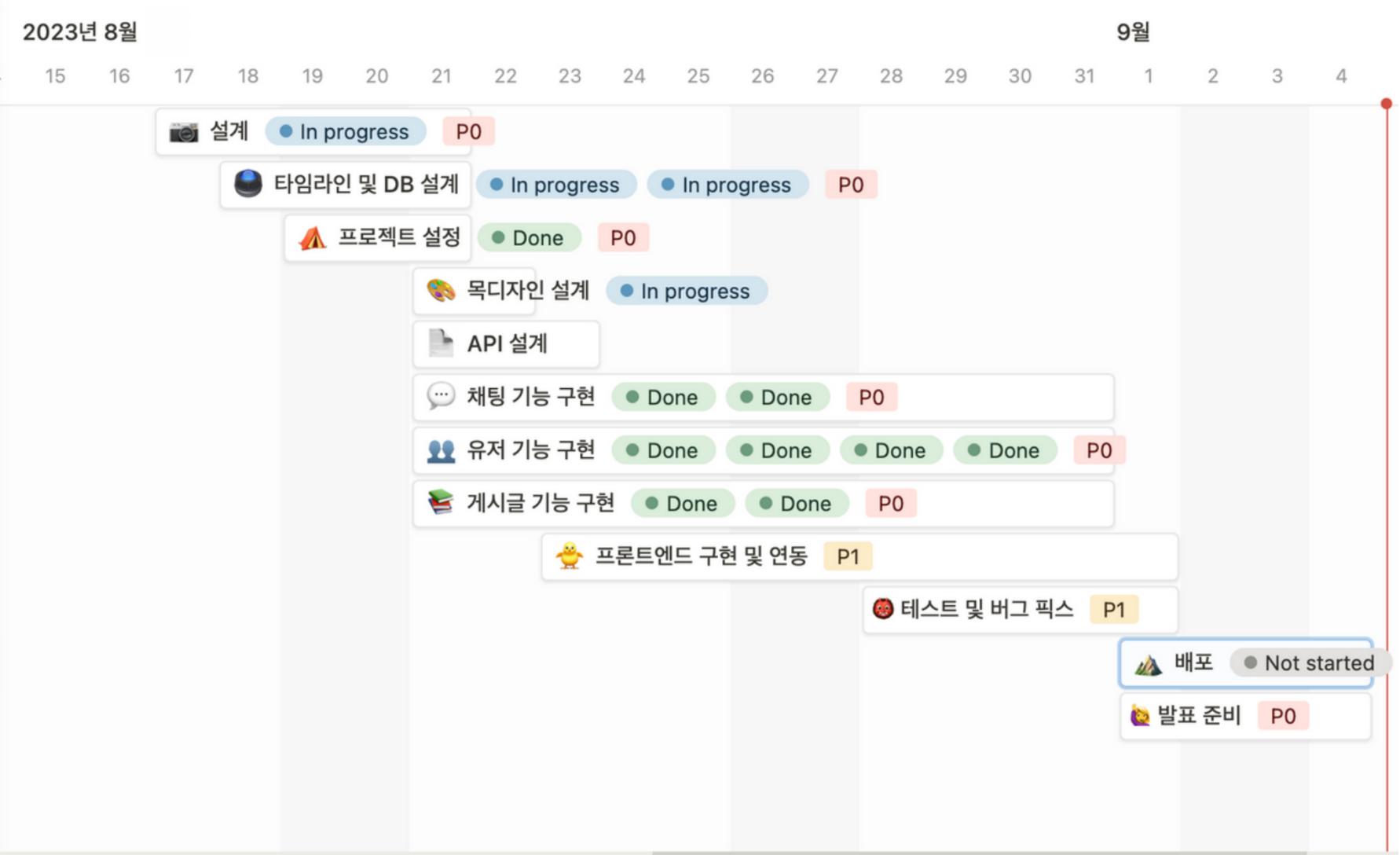
+ :: 코딩에 관한 주제를 라이브 채팅으로 소통하는 커뮤니티

타임라인 By Priority All projects

## Projects

- Project
- 설계
- 타임라인 및 DB 설계
- 프로젝트 설정
- 목디자인 설계
- API 설계
- 채팅 기능 구현
- 유저 기능 구현
- 게시글 기능 구현
- 프론트엔드 구현 및 연동
- 테스트 및 버그 픽스
- 배포
- 발표 준비
- + 새로 만들기

개수 12





개요

### Cobra Space

#### 1. 소개

- 1.1. 목적
- 1.2. 프로젝트 범위
- 1.3. 용어집
- 1.4. 문서 개요

#### 2. 전체 설명

- 2.1. 시스템 환경
- 2.2. 사용사례
  - 2.2.1. 방문자의 사용 사례
    - 2.2.1.1. 사용 사례: 회원...
  - 2.2.2. 유저의 사용 사례
    - 2.2.2.1. 사용 사례: 로그인
    - 2.2.2.2. 사용 사례: 로그아웃
    - 2.2.2.3. 사용 사례: 프로필...
    - 2.2.2.4. 사용 사례: 프로필...
    - 2.2.2.5. 사용 사례: 팔로우...
    - 2.2.2.6. 사용 사례: 팔로우...
    - 2.2.2.7. 사용 사례: 팔로우...
    - 2.2.2.8. 사용 사례: 팔로우...
  - 2.2.3. 게시물 사용 사례
    - 2.2.3.1. 사용 사례: 게시물 ...
    - 2.2.3.2. 사용 사례: 게시물 ...
    - 2.2.3.3. 사용 사례: 게시물 ...
    - 2.2.3.4. 사용 사례: 게시물 ...
    - 2.2.3.5. 사용 사례: 게시물 ...

# Software Requirement Specification

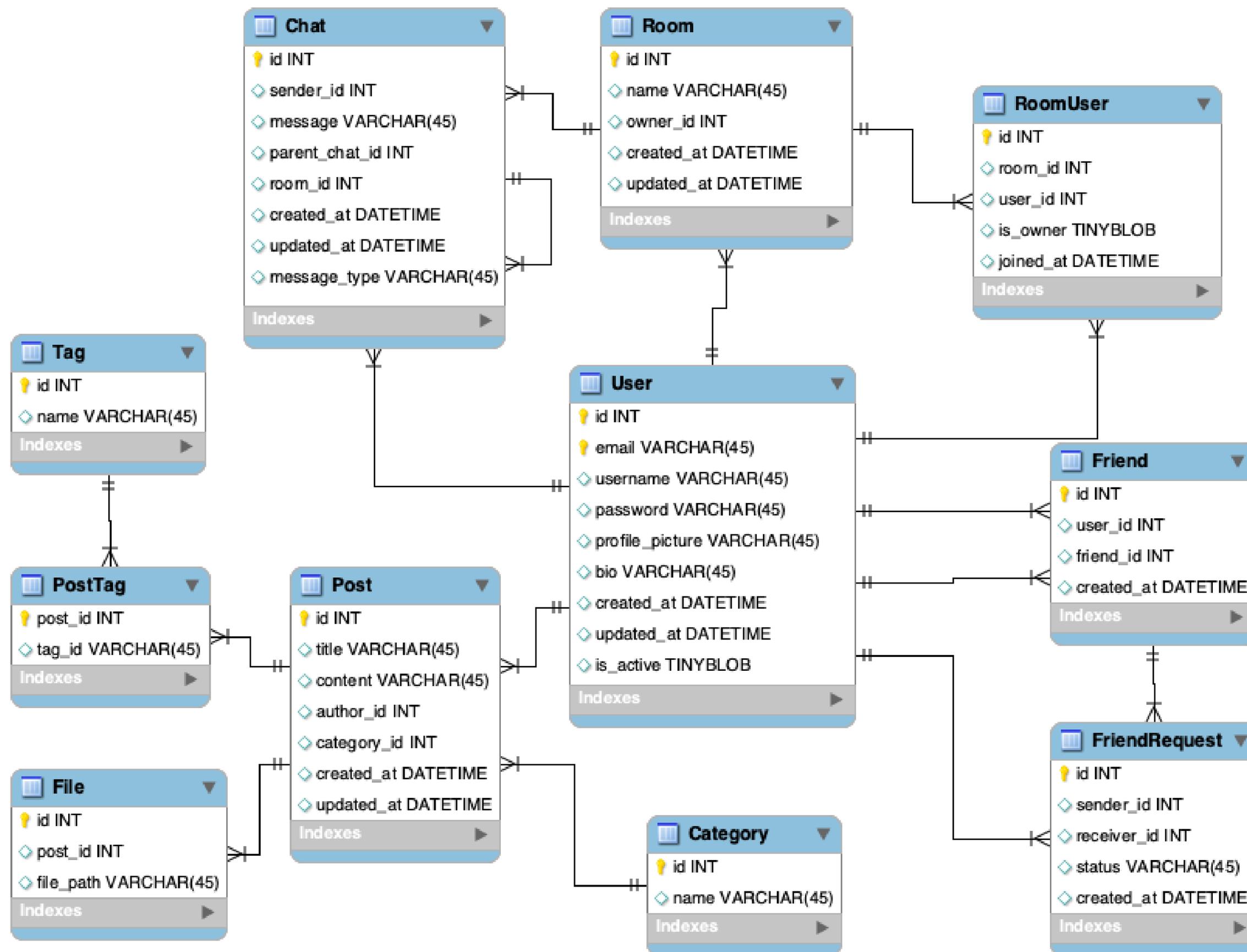


## Cobra Space

# SRS



# DB





# 주요 기능 소개

1. User
2. Post
3. Chat

# User

사용자는 자신의 계정으로  
어플리케이션의 기능을 사용가능



## 로그인

사용자의 입력 정보를 검증하고, 올바른 정보  
라면 인증된 세션을 생성 합니다.



## 로그아웃

로그아웃 버튼 클릭시 시스템은 현재 세션을  
종료 후 초기 화면으로 복귀합니다.



## 회원 가입

사용자의 이메일, 비밀번호를 입력받고, 받은  
정보에 따라 회원가입, 프로필을 생성합니다.

# Friend



## 친구 추가

사용자가 다른 사용자의 이메일을 입력하여 친구를 추가합니다.



## 친구 목록 조회

친구 추가된 사용자들의 이메일을 조회합니다.



## 친구 삭제

사용자가 친구 목록에 있는 친구(사용자)를 삭제합니다.



## 친구 프로필 조회

특정 친구(사용자)의 프로필 정보를 조회합니다.

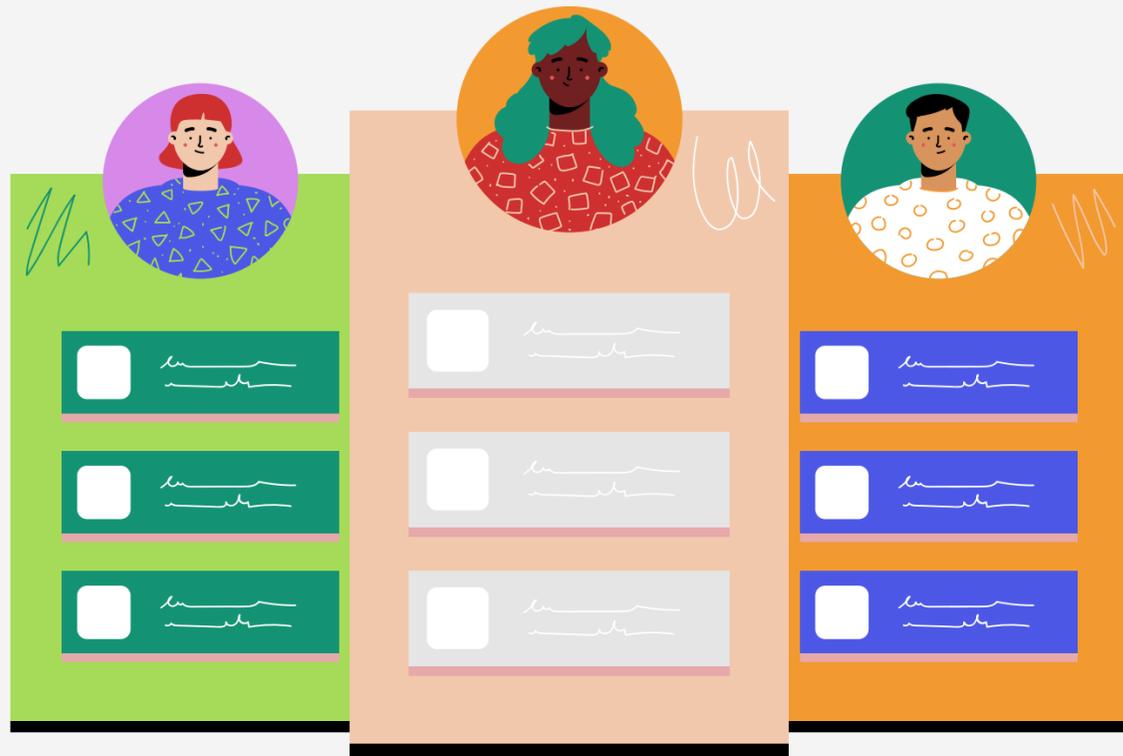


## 친구 검색

키워드를 기반으로 친구(이메일)를 검색합니다.

# Profile

사용자는 회원가입시 생성된  
프로필을 조회, 수정할 수 있습니다.



## 프로필 조회

회원가입시 생성된 프로필을 조회 합니다.

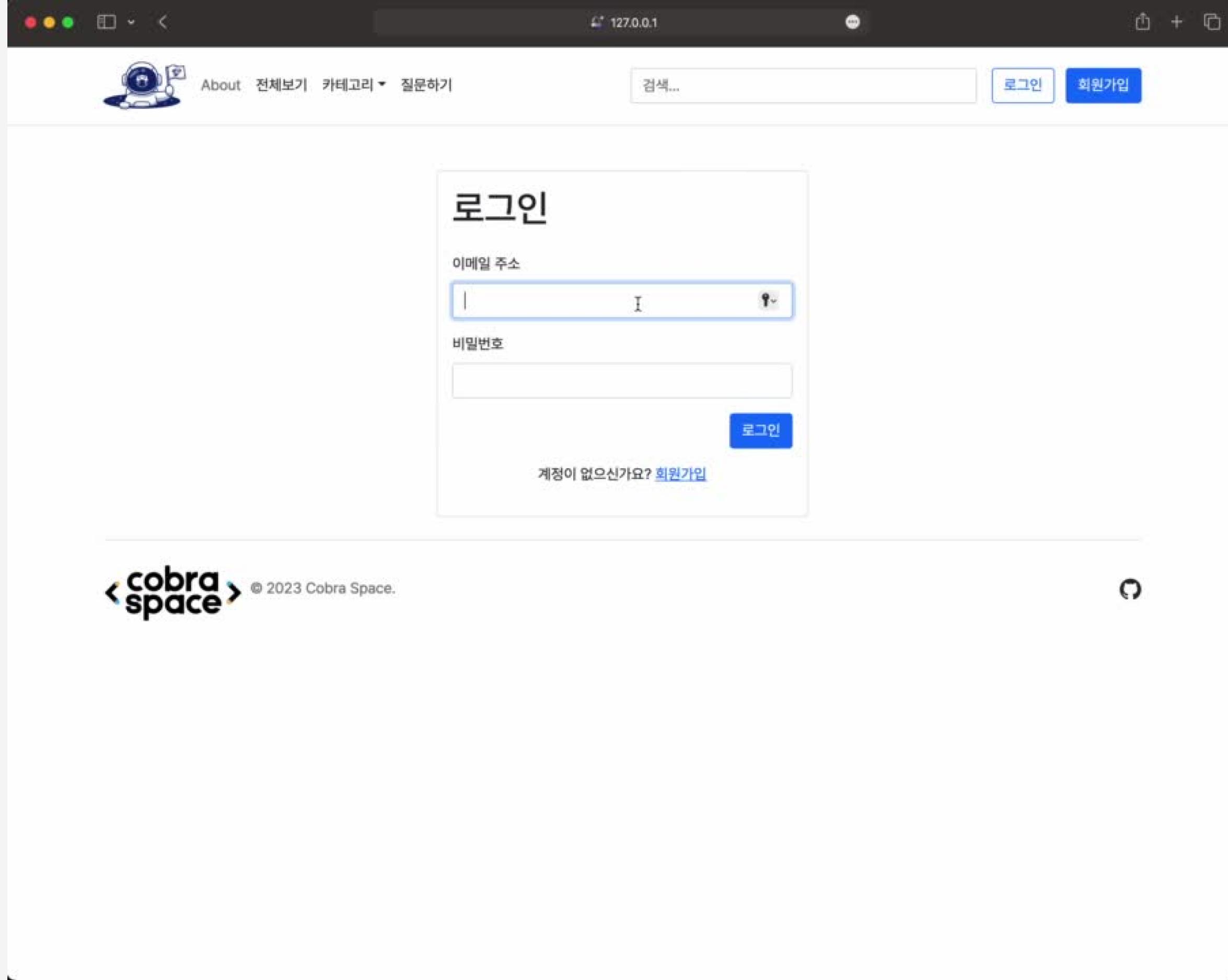


## 프로필 수정

사용자가 프로필 수정 요청을 보내고 수정을 할  
수 있습니다.

# User

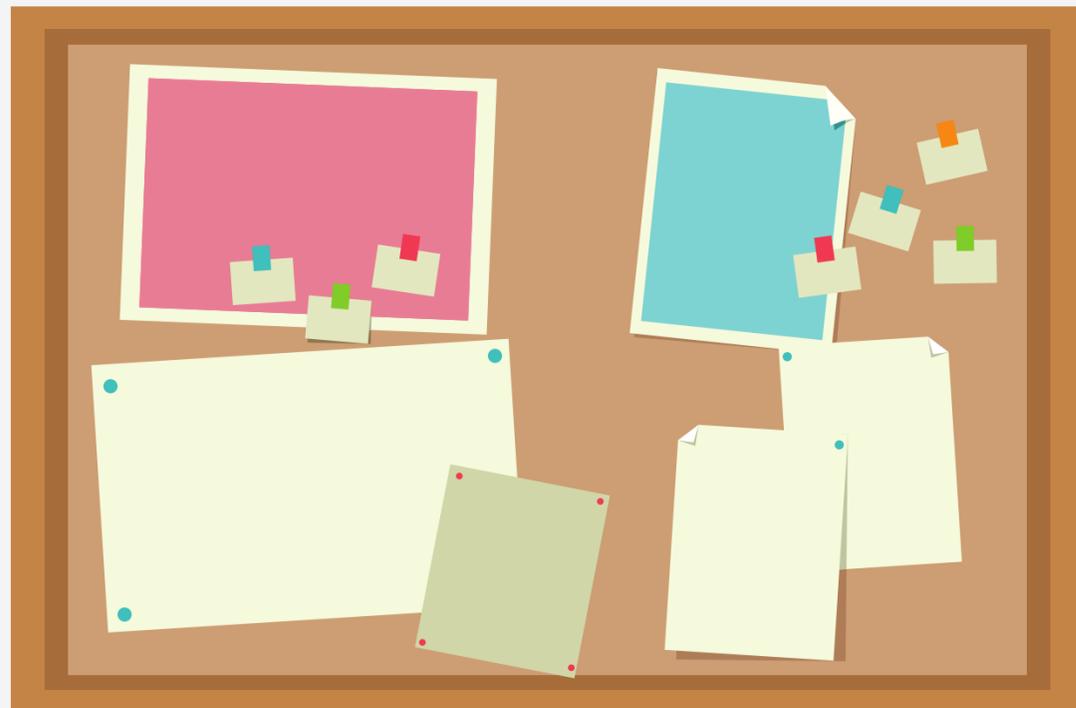
로그인 & 로그아웃





# Post

사용자가 다양한 주제에 대한 소통이 가능한 어플리케이션 기능



## 게시글 작성

로그인된 사용자가 홈페이지에서 “게시글 작성” 을 선택하여 제목,내용,해시태그 입력된 내용을 저장합니다.



## 게시글 수정 및 삭제

사용자가 자신이 작성한 게시글의 제목,내용,해시태그 등에 대해 수정하고 자신의 게시글에 대한 삭제가 가능합니다.



## 게시글 조회

특정 게시물에 대한 내용, 작성자 및 기타 세부 정보를 확인할 수 있습니다.

# Post



## 게시물 검색

작성한 키워드에 해당하는 게시물의 목록을 볼 수 있습니다.



## 게시글 목록

게시된 모든 게시물의 목록을 볼 수 있습니다.

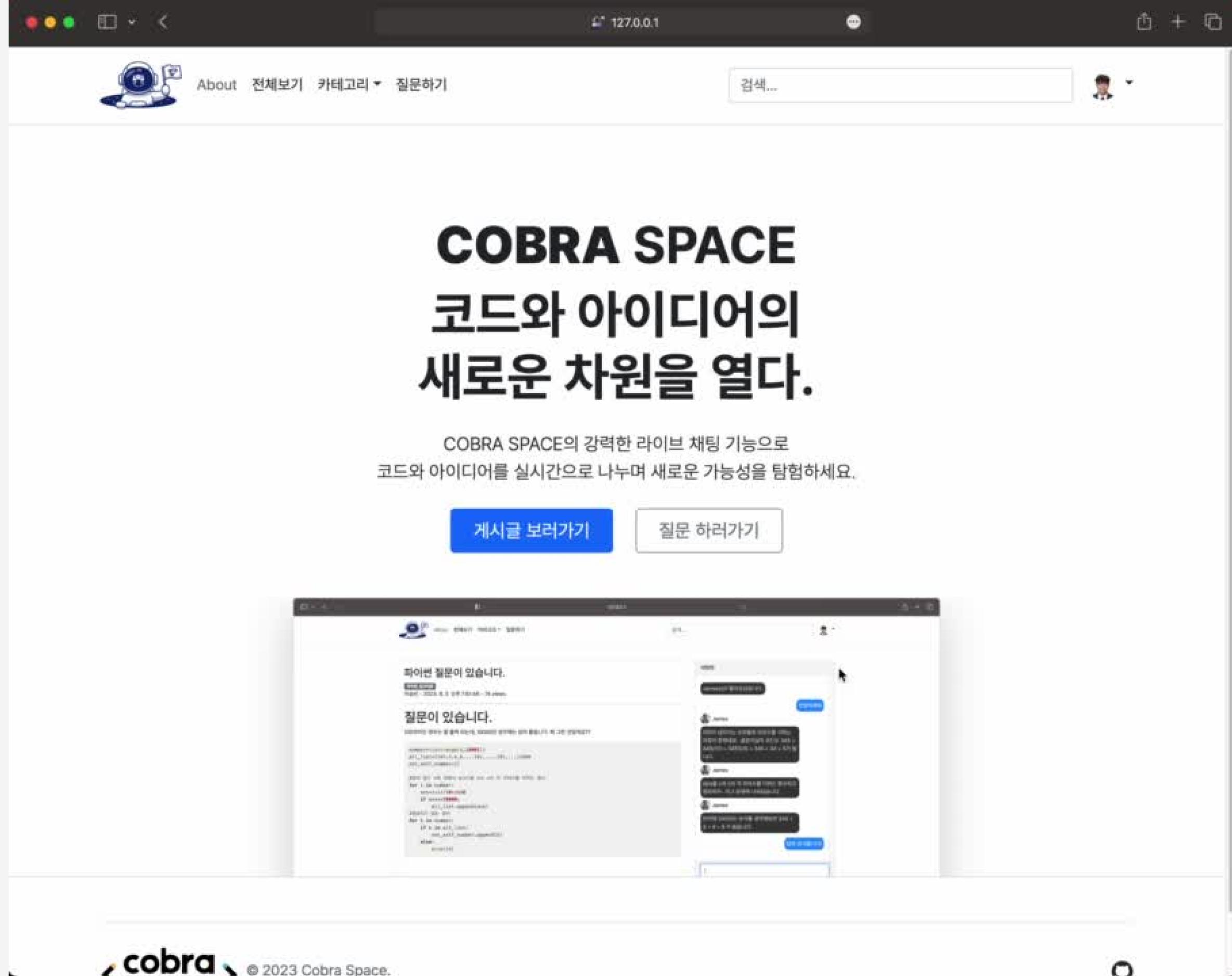


## 카테고리

존재하는 모든 카테고리의 목록을 볼 수 있습니다.

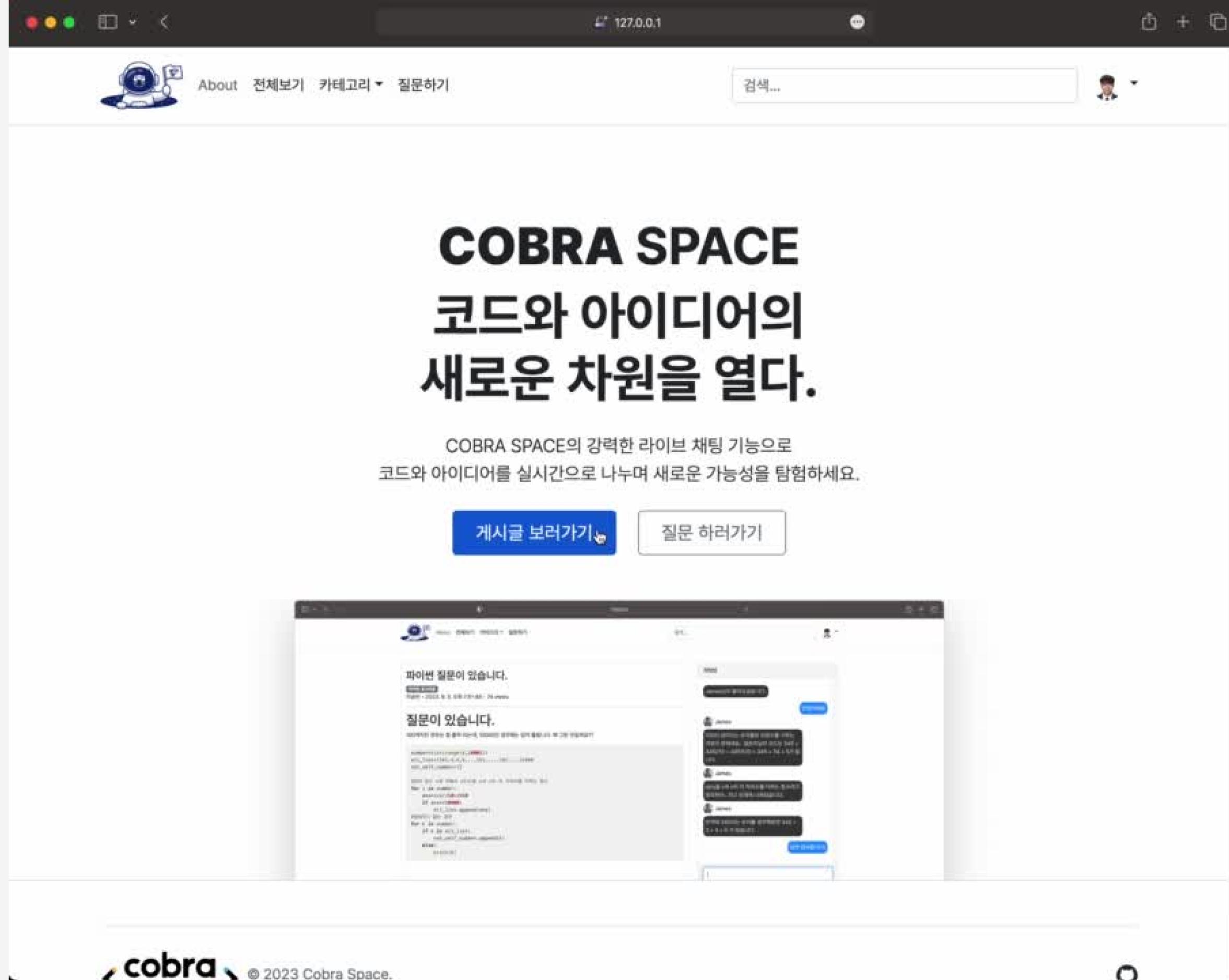
# Post

글 작성



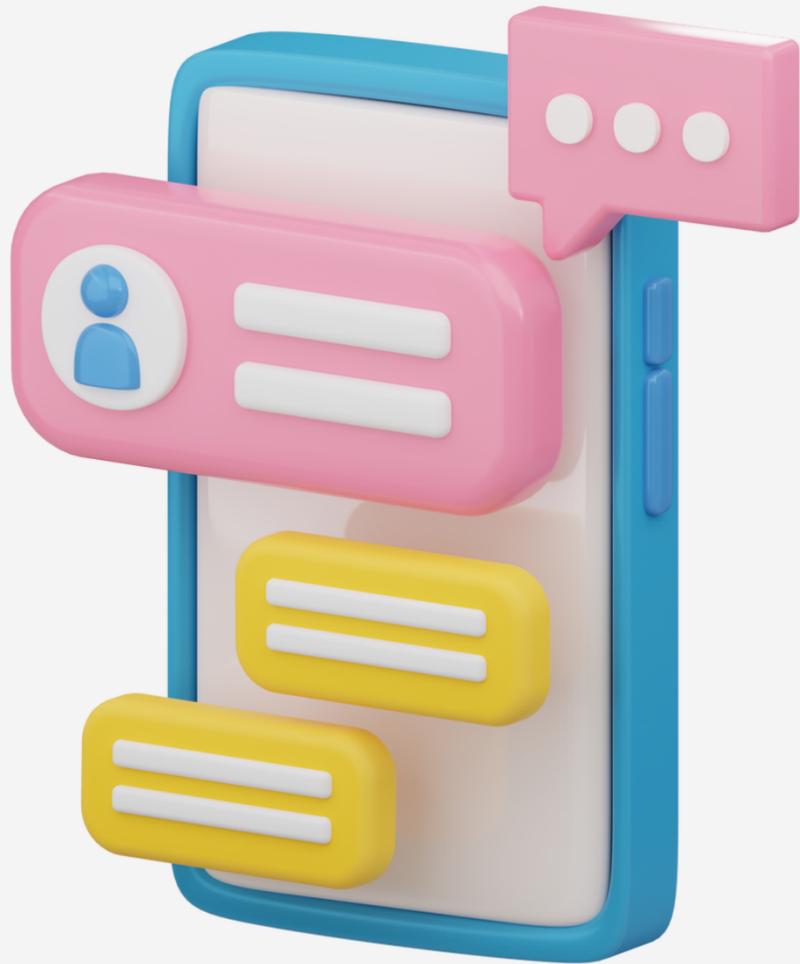
# Post

게시글 조회



# Chat

대화방을 개설하여 상대방과 대화를 주고받을수 있습니다.



## 채팅 작성

사용자가 특정 채팅방에서 새로운 대화를 시작하거나 이미 진행중인 대화에 메시지를 추가하여 상대방과 소통합니다.



## 채팅 수정 및 삭제

채팅 작성자가 작성한 채팅에 대한 수정 및 삭제가 가능합니다.



## 추가 기능

멘션 기능과 답글 기능을 추가하여 의견, 답변, 논의를 보다 구체적으로 발전시키는 기능을 합니다.



### 파이썬 질문있습니다

#### 장고 - 장고 기본에서는 HTTP 요청을 처리하는 주체는 View, 함수와 클래스 형태 View 에서는 HttpRequest 객체를 통해서 유저/세션/쿠키/헤더...

장고 파이썬



이승빈

2023-09-02 10:24

### 파이썬 질문이 있습니다.

# 질문이 있습니다. 100까지인 경우는 잘 출력 되는데, 10000인 경우에는 답이 틀립니다. 왜 그런 것일까요?? ```python number=list(range(...

파이썬 알고리즘



이승빈

2023-09-02 10:55

### SyntaxError 구문 오류

```javascript const handleClick = (e: React.MouseEvent) => { if (e.current...

자바스크립트



James

2023-09-03 21:12



# Chat

채팅방 접속 & 퇴장

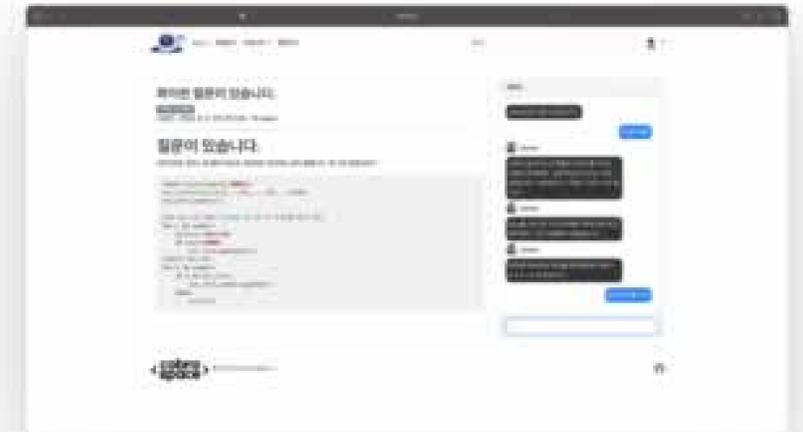


## COBRA SPACE 코드와 아이디어의 새로운 차원을 열다.

COBRA SPACE의 강력한 라이브 채팅 기능으로  
코드와 아이디어를 실시간으로 나누며 새로운 가능성을 탐험하세요.

게시글 보러가기

질문 하러가기



# 추가 API

친구추가,  
친구찾기,  
게시글 검색,  
게시글 수정,  
태그 검색,  
등등

...

## 1.3.2 메시지 교환

메시지 유형:

WebSocket 연결을 통해 주고받는 메시지는 다음과 같은 유형을 가질 수 있습니다:

- `chat.message` : 일반 채팅 메시지
- `chat.user.join` : 사용자가 채팅방에 입장한 이벤트
- `chat.user.leave` : 사용자가 채팅방에서 퇴장한 이벤트

메시지 전송:

클라이언트는 JSON 형식의 메시지를 WebSocket 서버로 전송할 수 있습니다. 메시지의 형식은 다음과 같습니다:

```
{  
  "type": "chat.message",  
  "message": "Hello, World!"  
}
```

메시지 수신:

WebSocket 서버는 다른 클라이언트로부터 수신한 메시지를 현재 연결된 클라이언트에게 전달합니다.

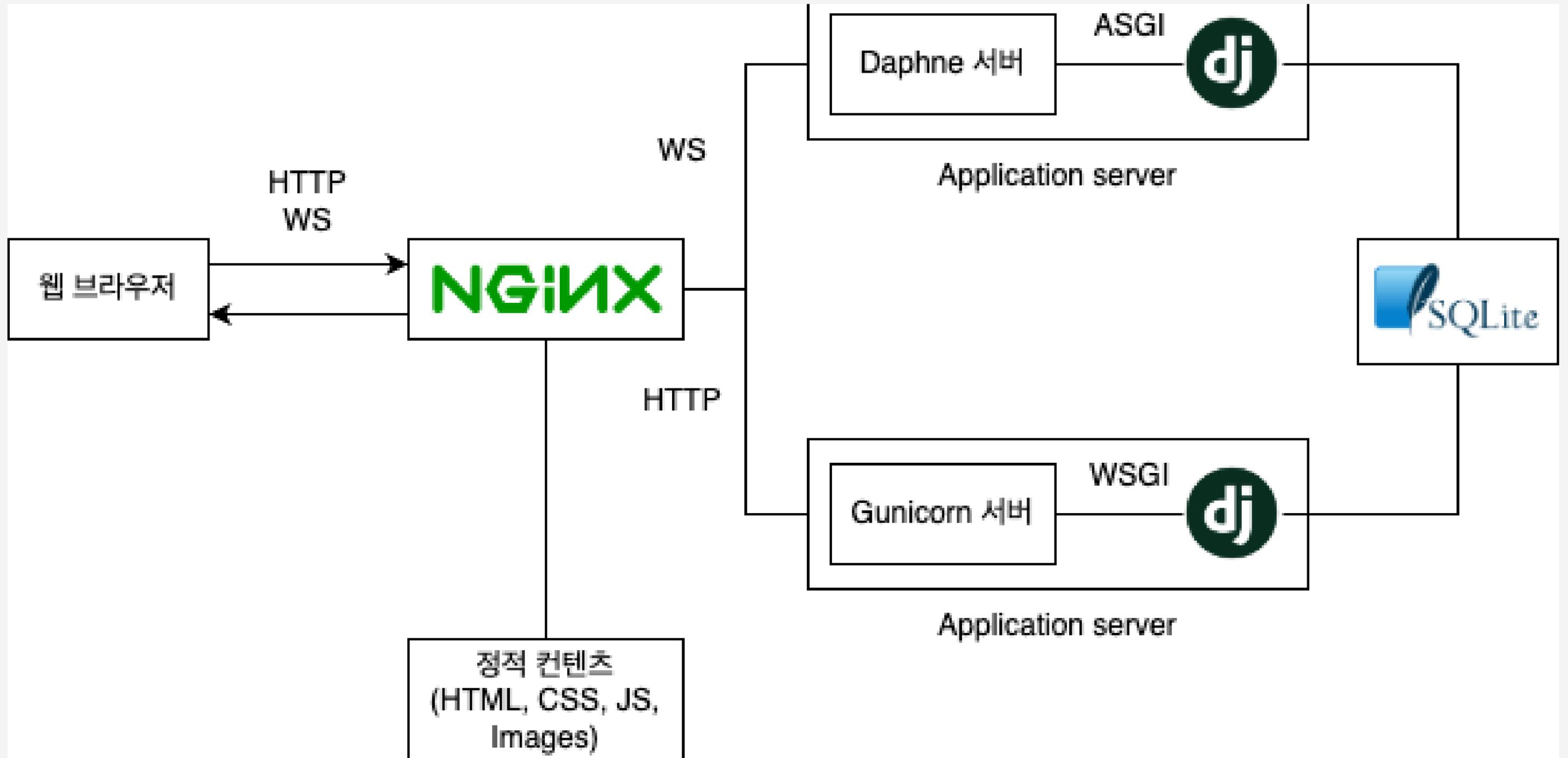
```
{  
  "type": "chat.message",  
  "message": "Hello, World!",  
  "sender": "user@example.com",  
  "nickname": "John",  
  "profile_picture_url": "https://example.com/profile.jpg"  
}
```

이벤트 처리:

클라이언트는 `chat.user.join` 및 `chat.user.leave` 이벤트를 수신하여 사용자의 입장 및 퇴장을 처리할 수 있습니다.

```
{  
  "type": "chat.user.join",  
  "username": "John"  
}
```

# 배포





# 라이브 데모

<http://api.cobraspace.xyz/api/post/>

<http://cobraspace.xyz>

Q & A



감사합니다