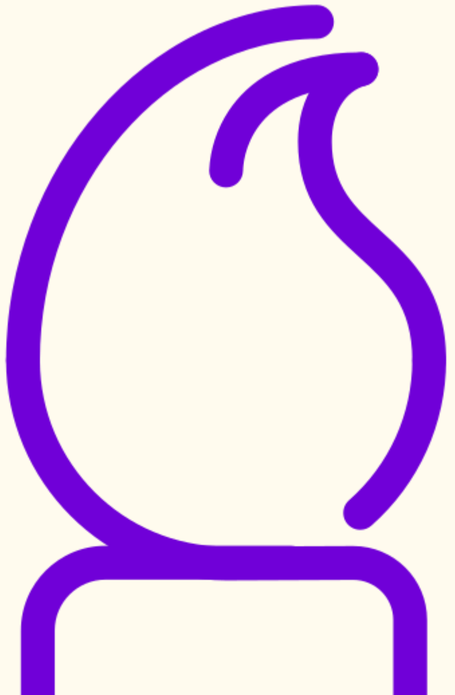
딩동! 여행 플랫폼

Web 3.0 기반 분산형 커뮤니티 서비스

2023 - summer - 딩동 - 05

공소연
김민정
윤석규
이지민



기업요구사항

- **과제 :** Web 3.0 기반 분산형 커뮤니티 서비스

  블록체인 분산 네트워크를 기반으로 데이터를 공유하는 새로운 웹 형태

  → 백엔드 : Solidity, Rust

But

- **기업 측 :** 블록체인 기술을 사용하지 않고, 어떤 데이터가 중앙화된 객체에 종속되지 않는 블록체인의 '탈중앙성'이라는 특징에 초점을 맞추는 방향성 제시

  → 백엔드 : Node.js + Express

→ **블록체인의 탈중앙화적 특성을 지향하는 커뮤니티 서비스 제작!**

추진배경

기존 질의응답 커뮤니티

답변이 더 유용한 경우가 많음

but

질문 글이 사라지면 답변 글도 함께 사라짐

→ 중앙화된 구조

해결

딩동 커뮤니티

질문 글이 사라지더라도,
그에 딸린 답변과 댓글들은 남아있어
유용한 정보 유지 가능

→ 탈중앙화된 구조

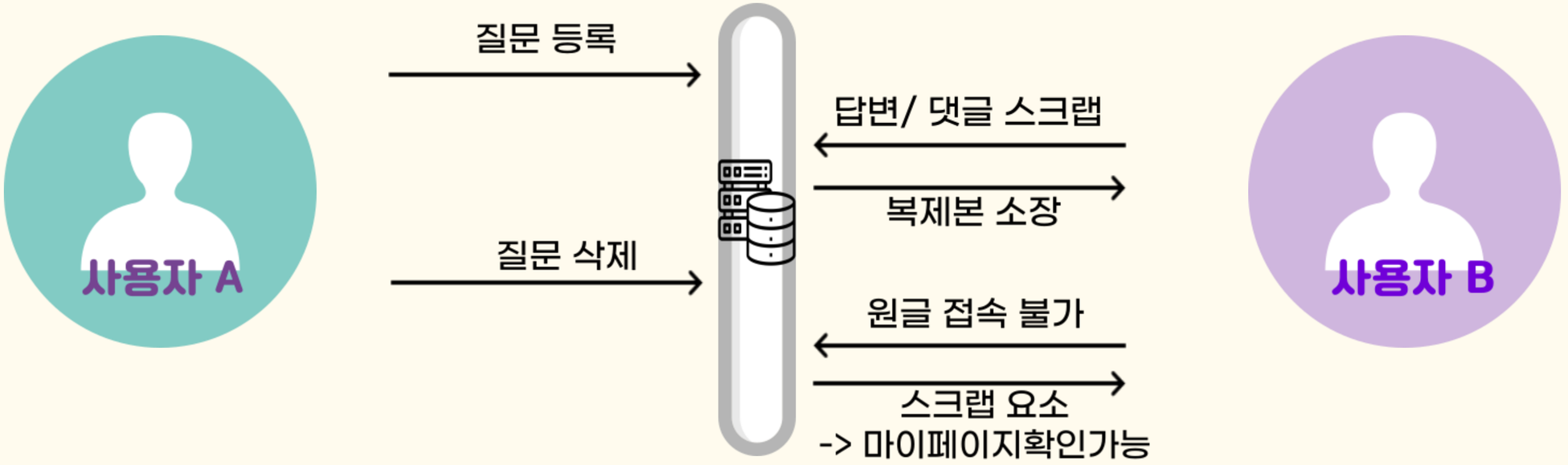
과제 목표

> **여행에 관한 유용한 정보의 원활한 제공**

1. 탈중앙적 구조의 Q&A 게시판 (답변과 댓글 객체화)
2. 투표 기능을 통한 유용한 정보 판별
3. 해시태그 기능을 통한 정보 카테고리화
4. 검색 기능을 통한 정보 찾기

접근 방법

Web 3.0 《 불변성
탈중앙화



접근 방법 FE

react-Quill
rich text 편집기

real Carousel
해시태그 애니메이션

recoil
(+ local storage)
해시태그, 로그인, 글 item 변수관리

dompurify
XSS 방지

접근 방법 BE

soft / hard deletion

사용자 삭제 option에 따른 명령어 구분

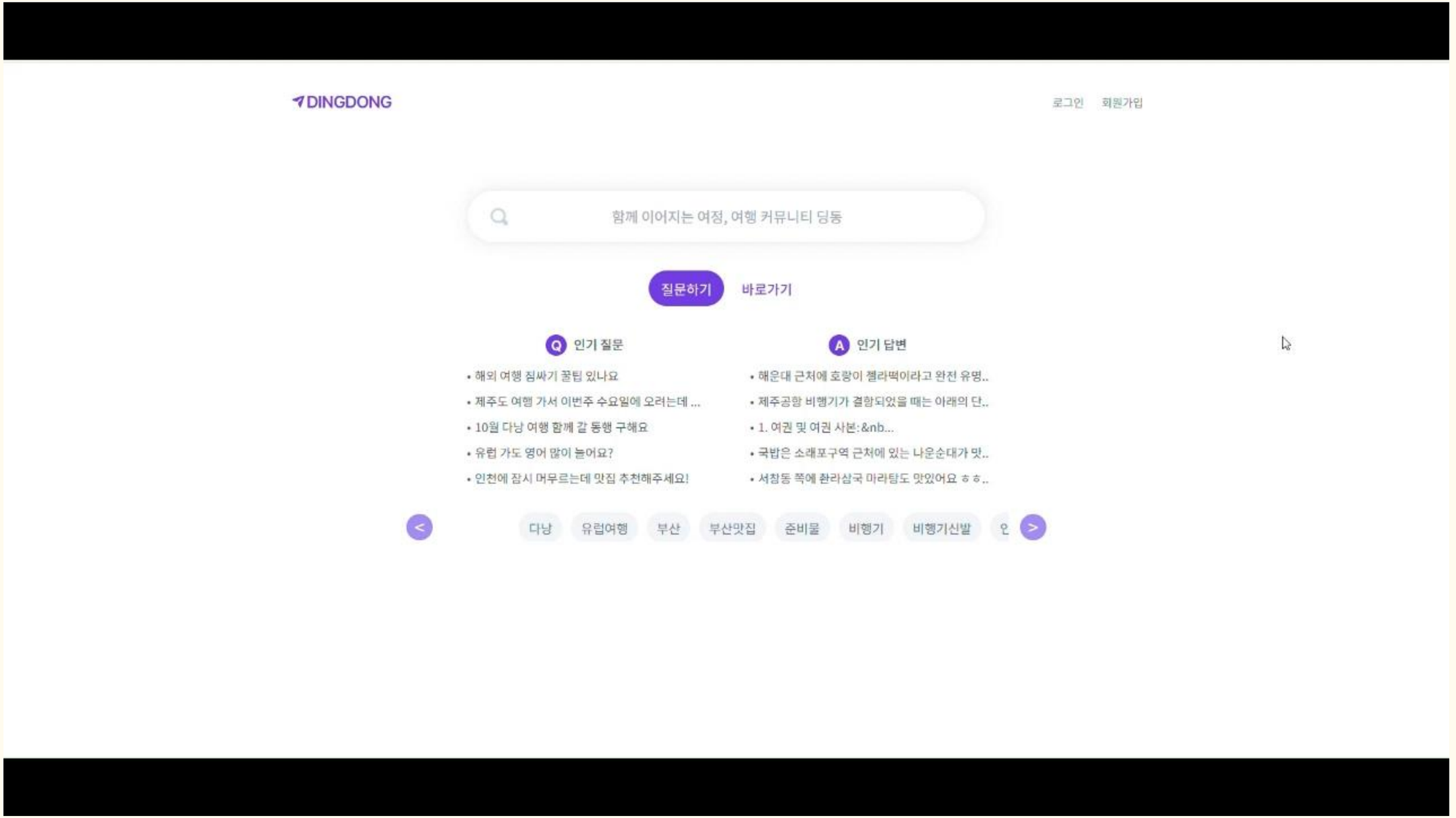
CopyQuestion

복제본을 위한 질문 copy 모델

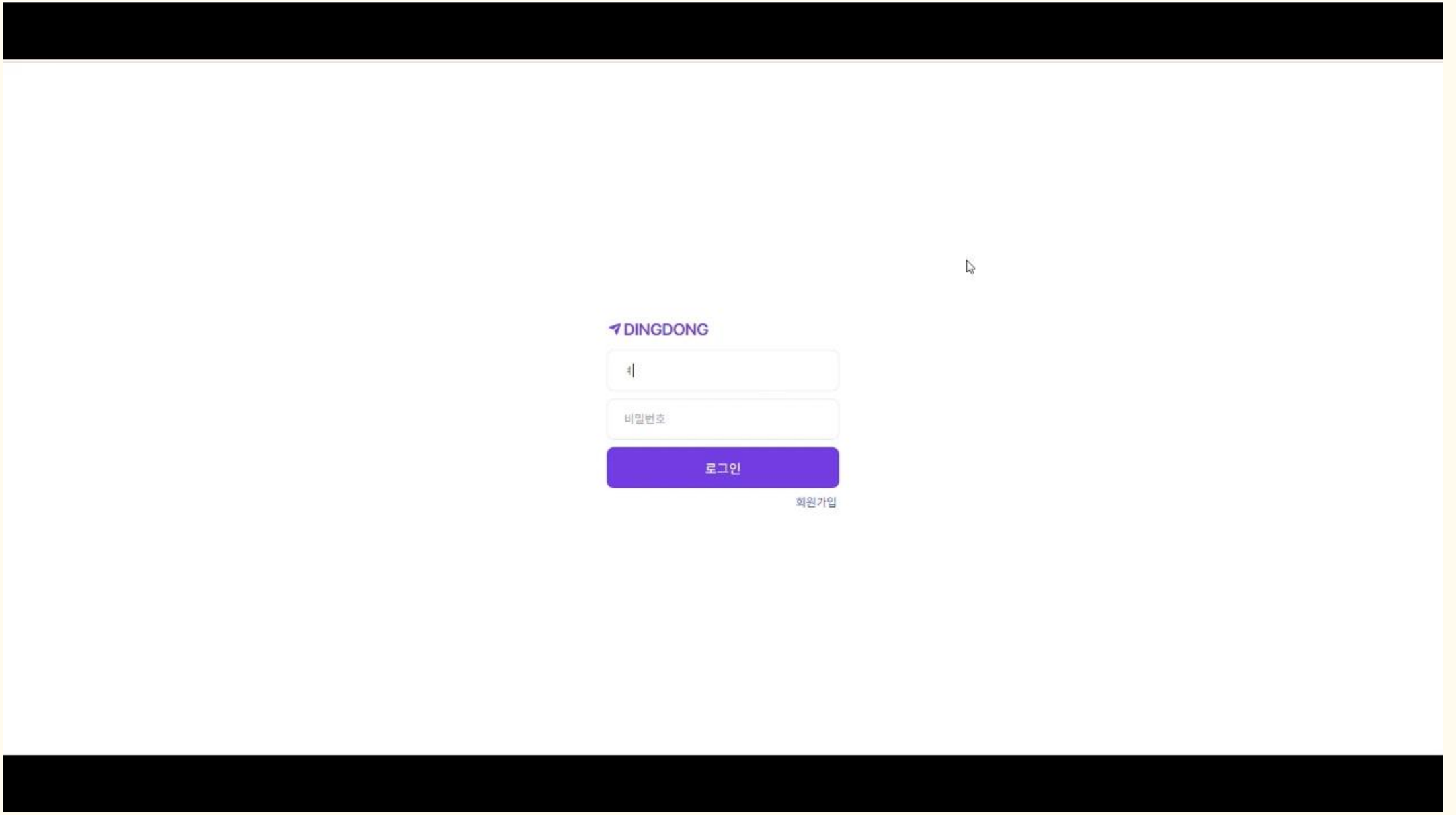
CopyAnswer

복제본을 위한 답변 copy 모델

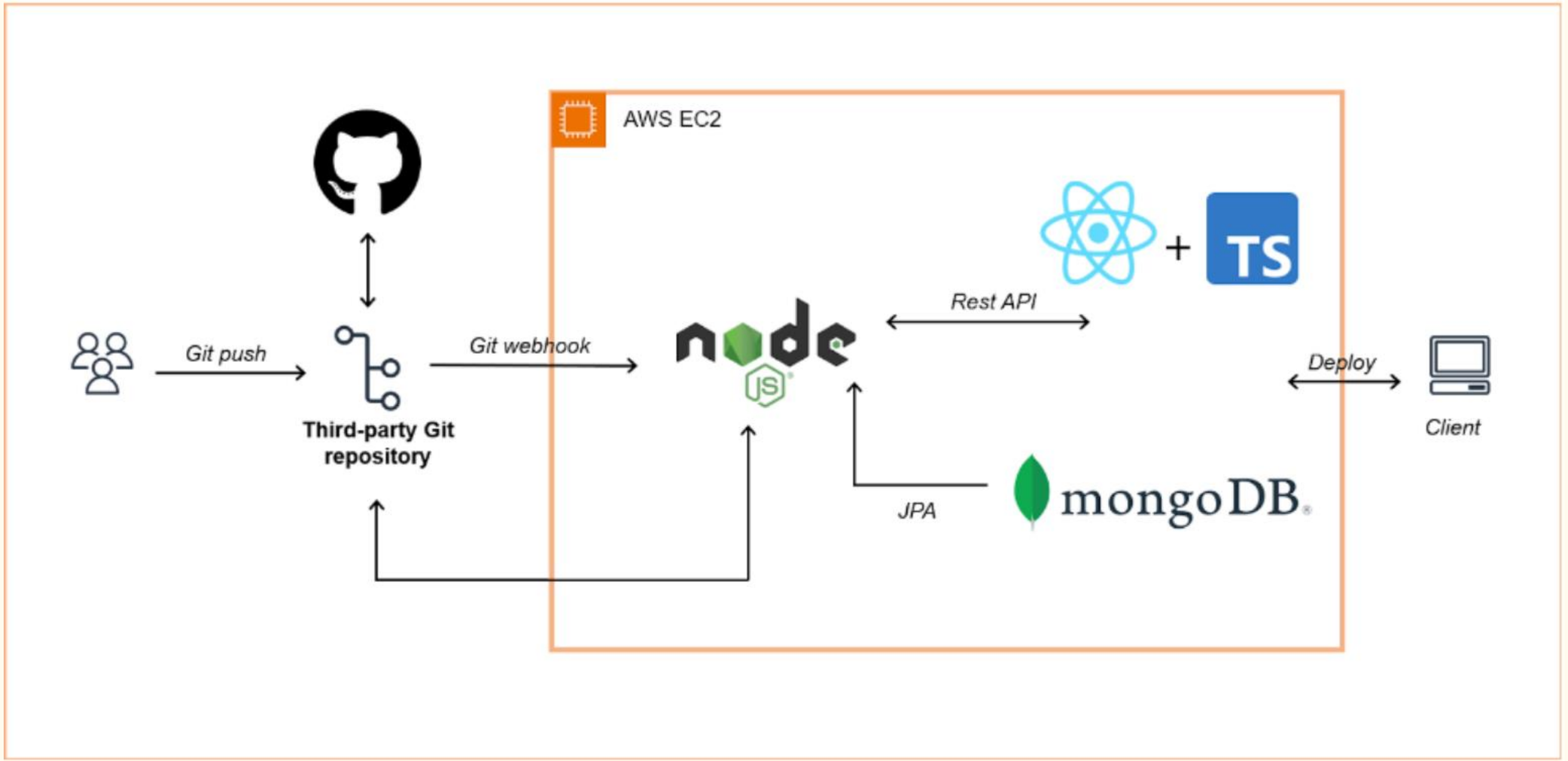
시연 영상(비회원)


DINGDONG
로그인
회원가입
함께 이어지는 여정, 여행 커뮤니티 딩동
질문하기
바로가기
Q 인기 질문
• 해외 여행 짐싸기 꿀팁 있나요
• 제주도 여행 가서 이번주 수요일에 오려는데 ...
• 10월 다낭 여행 함께 갈 동행 구해요
• 유럽 가도 영어 많이 늘어요?
• 인천에 잠시 머무르는데 맛집 추천해주세요!
A 인기 답변
• 해운대 근처에 호랑이 젤라떡이라고 완전 유명..
• 제주공항 비행기가 결항되었을 때는 아래의 단..
• 1. 여권 및 여권 사본:&nb...
• 국밥은 소래포구역 근처에 있는 나운순대가 맛..
• 서창동 쪽에 촨라삼국 마라탕도 맛있어요 ㅎㅎ..
다낭
유럽여행
부산
부산맛집
준비물
비행기
비행기신발

시연 영상(회원)



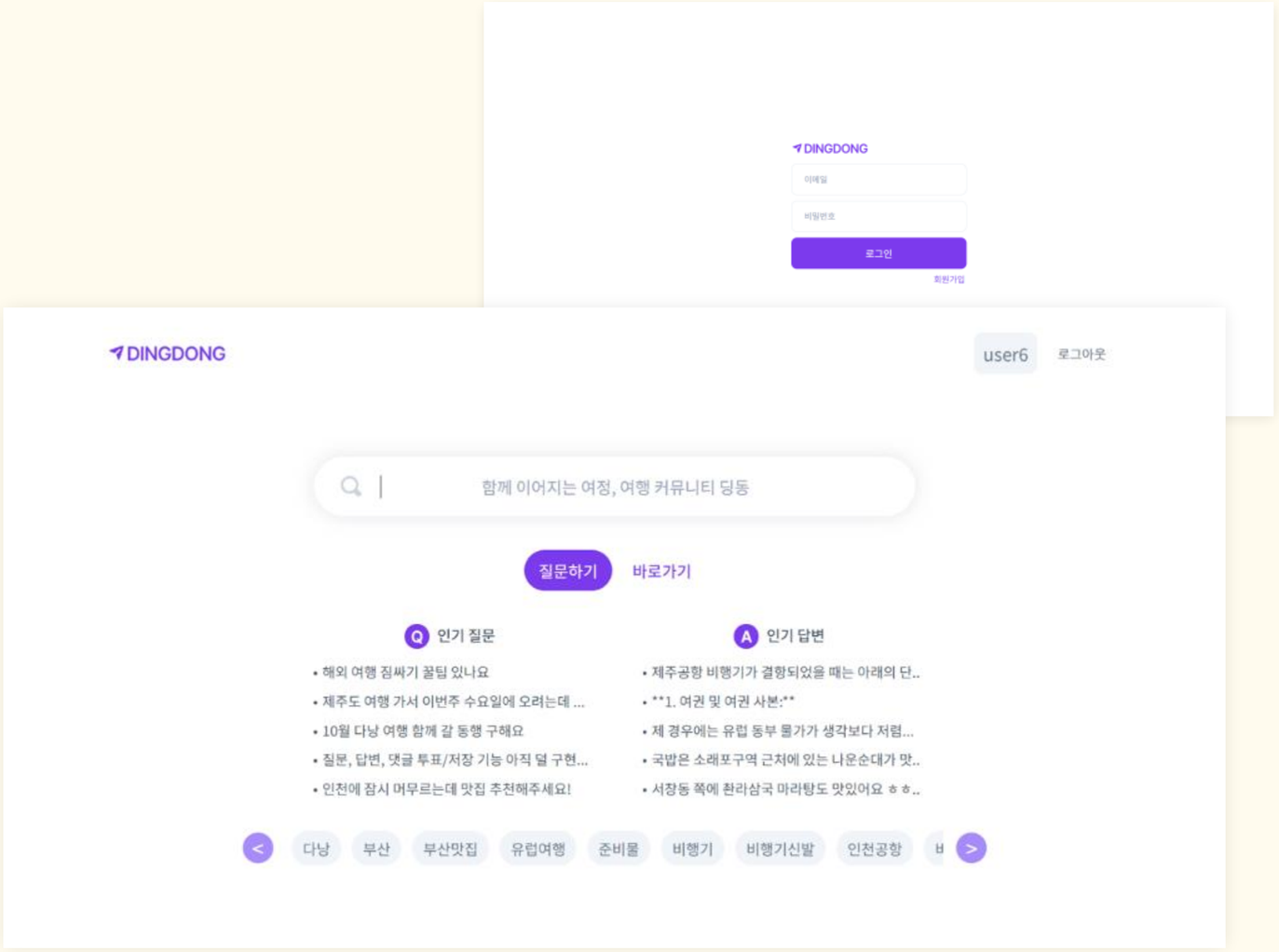
2. 과제 해결방안



시스템 구조

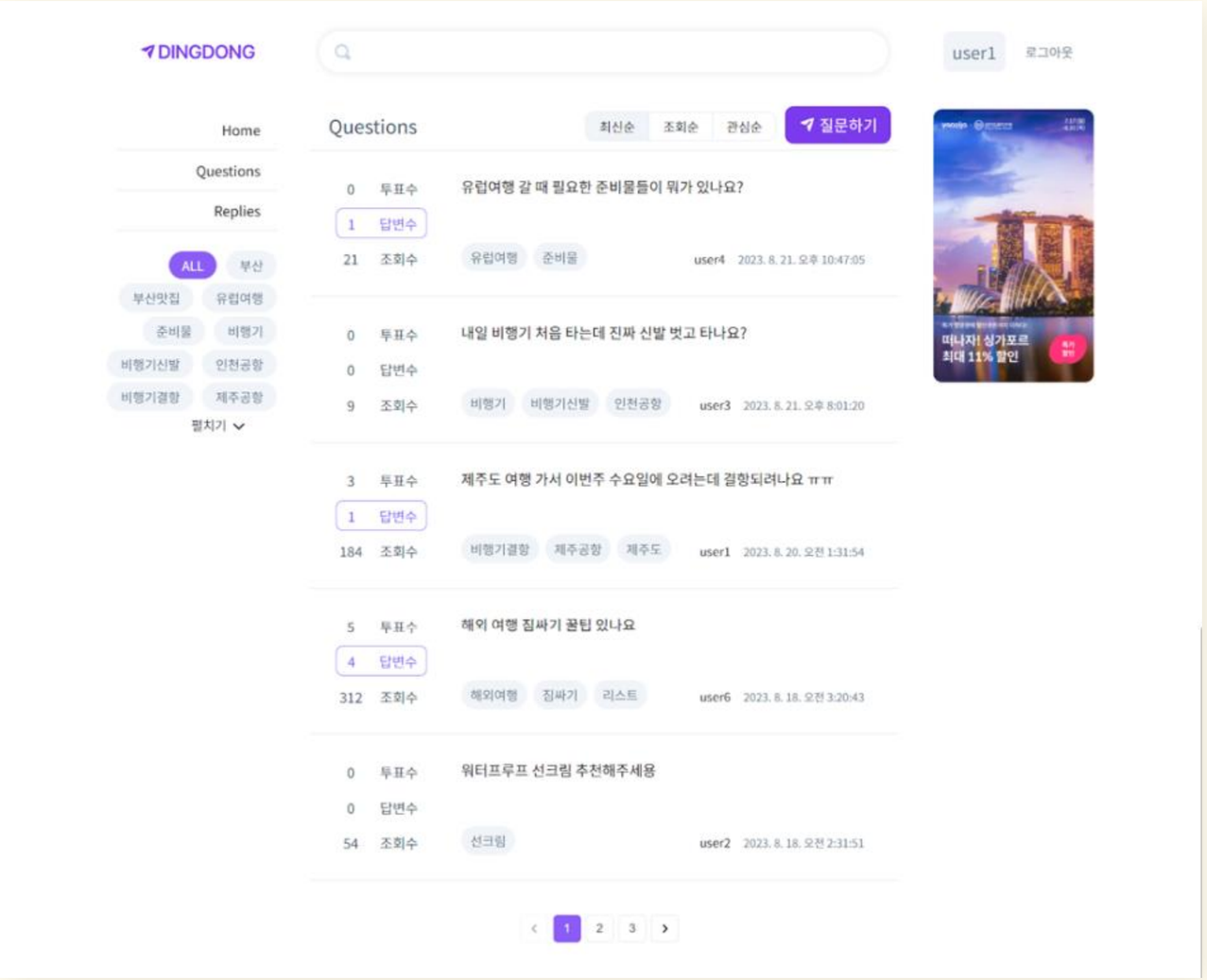
3. 주요 개발내용(구성요소 별 개발사항)

(1) 메인 페이지, 로그인 페이지



- **로그인**
  - passport 이용하여 구현
- **검색**
  - 질문과 답변 테이블에서 검색 단어를 포함한 글 조회
- **인기질문 5개, 인기답변 5개**
  - 투표순으로 정렬
  - 인기답변의 경우, dangerouslySetInnerHTML과 DOMPurify 이용
- **해시태그(키워드)**
  - 클릭 시 해당 해시태그에 대한 검색결과 페이지로 이동
  - 정렬 API 설계하여 구현
    - ■해시태그 언급 빈도 순 정렬
    - ■빈도가 같을 시 최신 순 정렬
  - Carousel 구현
    - ■라이브러리를 디자인에 맞춰 커스터마이징

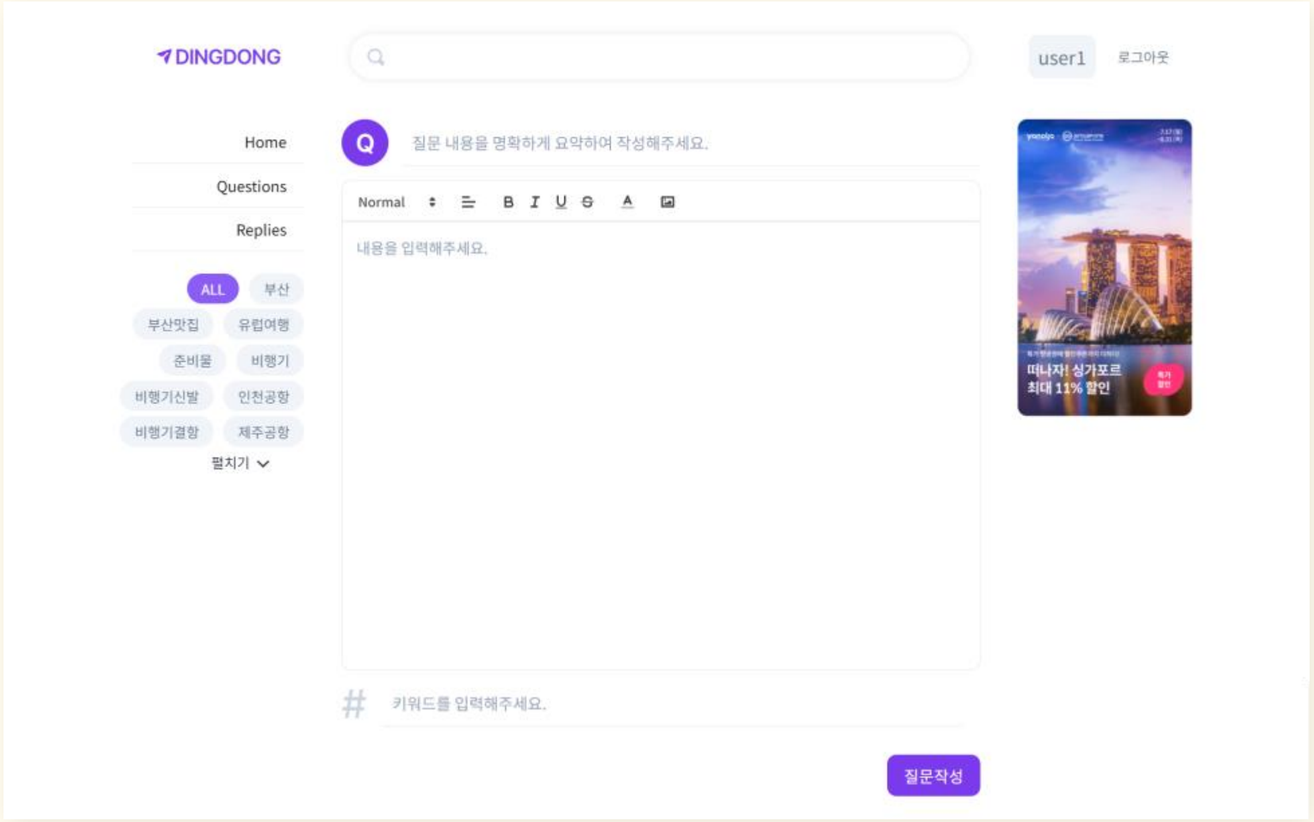
(2) 질문 리스트 페이지 (Questions)



- **검색**
  - 검색 시 검색결과 페이지로 이동
- **해시태그 네비게이션**
  - 클릭 시 해당 해시태그를 포함한 게시글 조회 가능
  - 해시태그 언급 빈도 순 정렬
  - 빈도가 같을 시 최신 순 정렬
- **정렬**
  - 최신순, 조회순, 관심순(투표순)으로 질문글 정렬 가능
- **질문리스트**
  - 각 질문의 제목, 투표수, 답변수, 조회수, 키워드, 작성자 등 표시
  - 페이지네이션 - 질문글을 5개 단위로 받아옴

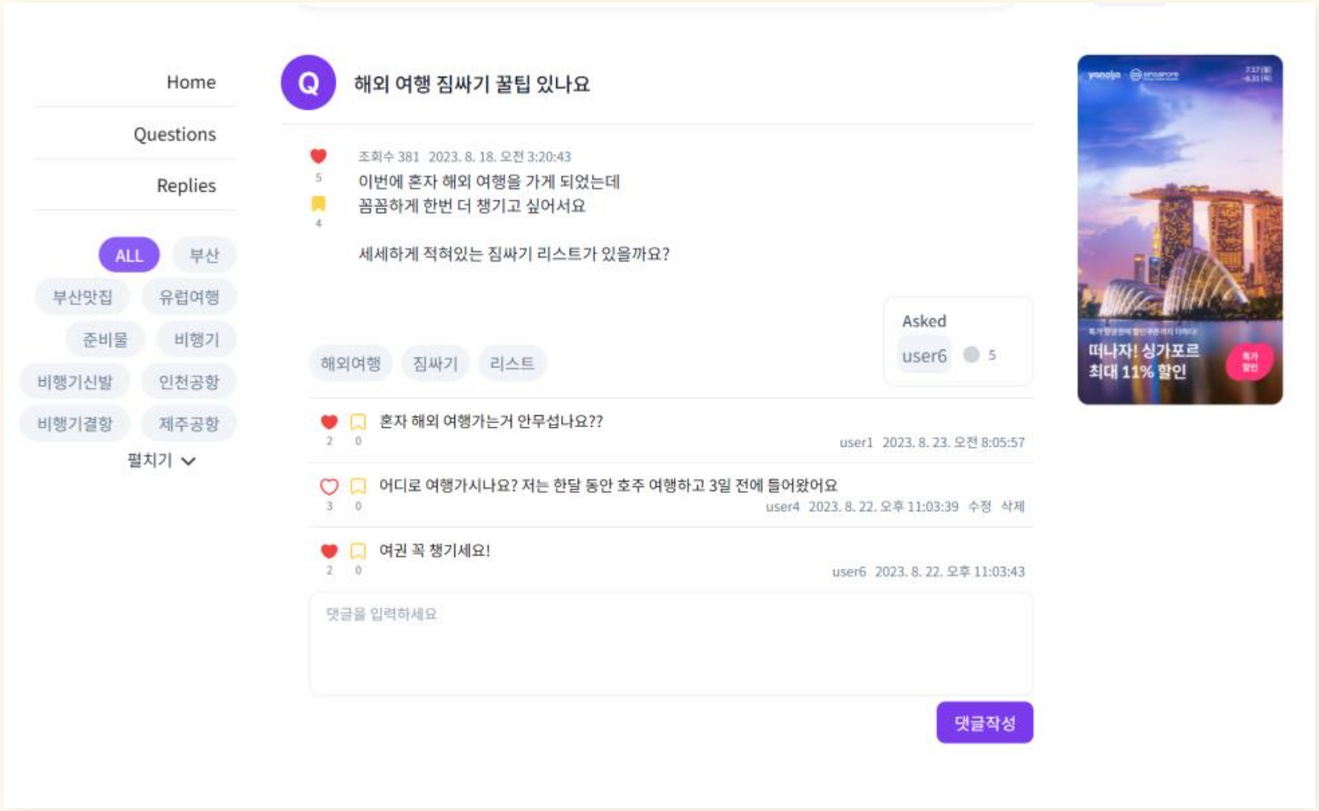
3. 주요 개발내용(구성요소 별 개발사항)

(3) 질문 작성 페이지



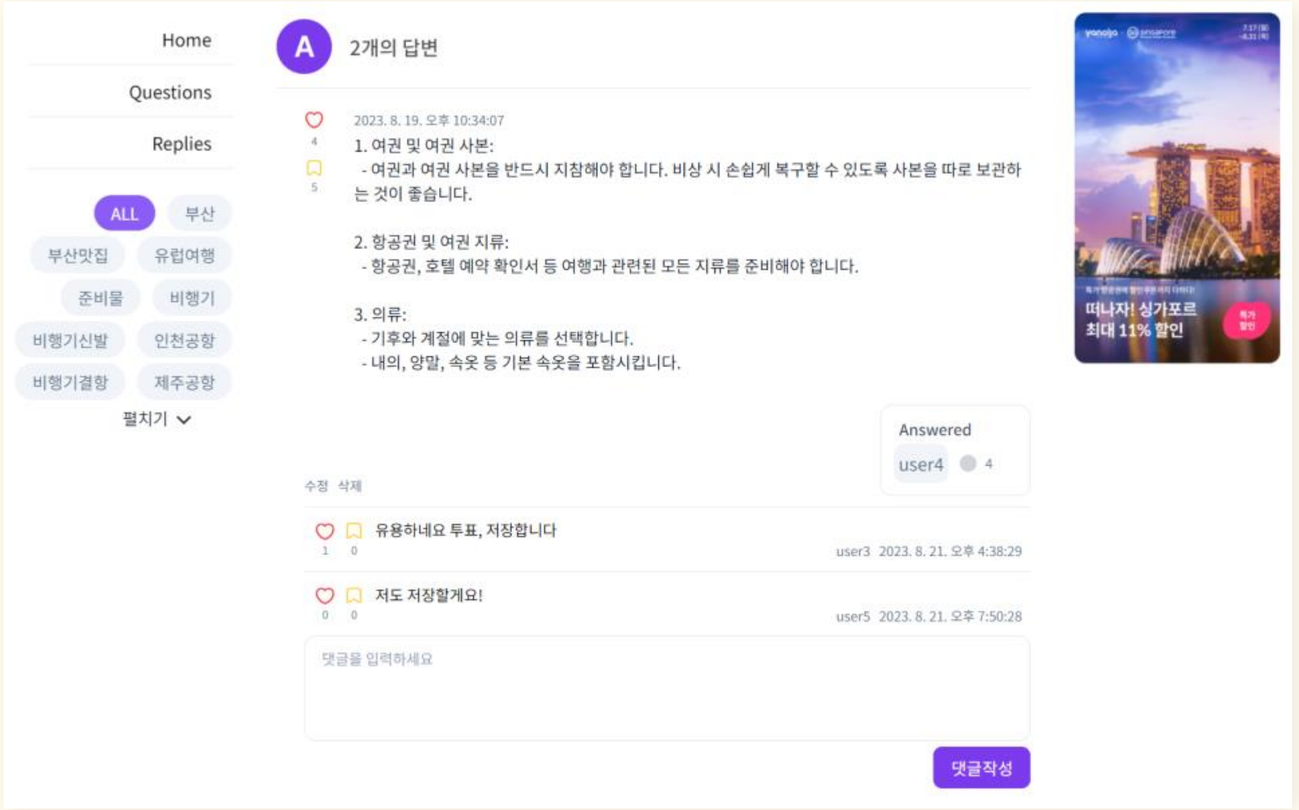
- **로그인 시에만 접속 가능**
- **게시글 작성 보드**
  - 글씨크기, 굵기, 색상 등 스타일 조정 가능
  - 이미지 첨부 가능
  - HTML 형태로 DB에 저장
- **해시태그 입력**
  - 하나의 키워드는 최대 6글자로 제한
  - 최대 3개의 키워드 등록 허용
- **질문작성**
  - 질문리스트에 질문이 등록됨
  - axios 이용해 Question 테이블에 Post

(4) 질문글 상세 페이지 - 질문



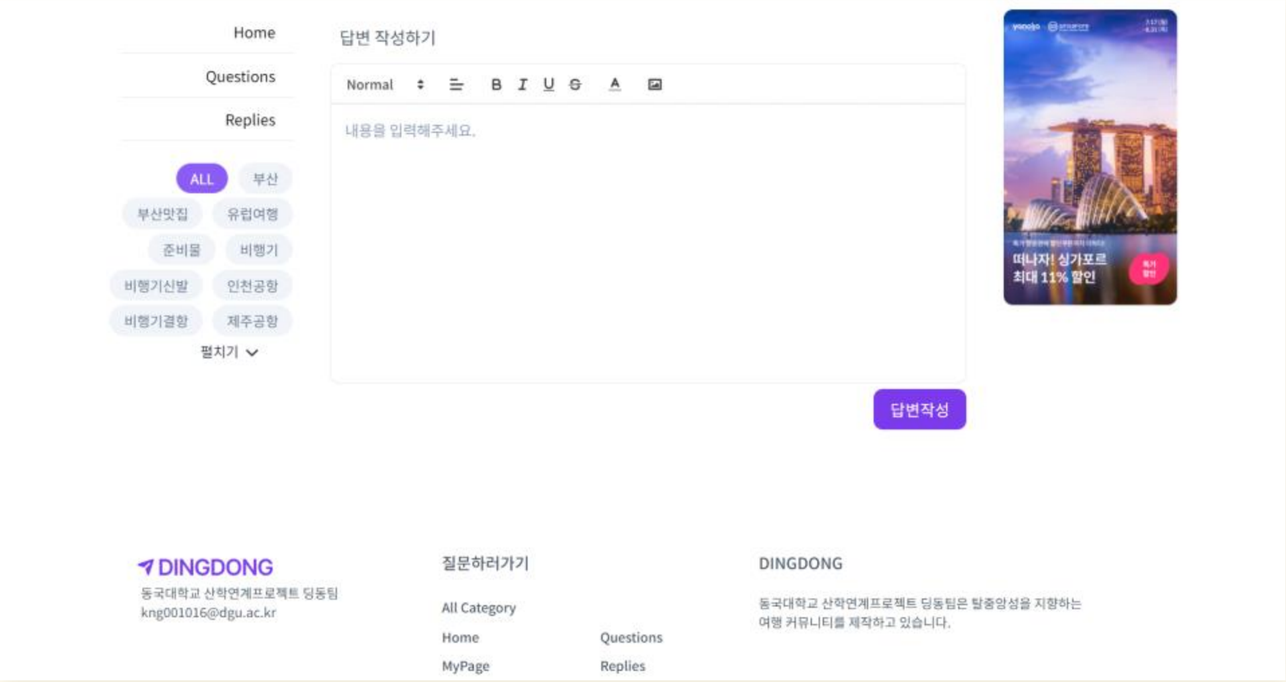
- **질문 내용**
  - HTML 형태로 전달된 content를 렌더링하기 위해 dangerouslySetInnerHTML과 DOMPurify 이용
- **질문 투표/저장**
  - 자신이 작성한 글은 투표/저장 불가
  - (로그인 한) 사용자들이 유용한 질문에 투표/저장 가능
  - 투표 횟수는 사용자 당 1회로 제한
- **질문 키워드(해시태그)**
- **질문 수정/삭제**
  - 질문 작성자만 수정/삭제 가능
- **작성자 정보**
  - 작성자의 username과 받은 투표 수 조회
- **질문에 대한 댓글**
  - 댓글 저장/투표 및 댓글 수정/삭제 가능

(4) 질문글 상세 페이지 - 답변



- **답변 내용 조회**
  - HTML 형태로 전달된 content를 렌더링하기 위해 dangerouslySetInnerHTML과 DOMPurify 이용
- **답변 투표/저장**
  - 자신이 작성한 글은 투표/저장 불가
  - (로그인 한) 사용자들이 유용한 질문에 투표/저장 가능
  - 투표 횟수는 사용자 당 1회로 제한
- **답변 수정/삭제**
  - 답변 작성자만 수정/삭제 가능
- **작성자 정보**
  - 작성자의 username과 받은 투표 수 조회 가능
- **답변에 대한 댓글**
  - (로그인 시) 댓글 저장/투표 및 수정/삭제 가능

(4) 질문글 상세 페이지 - 답변

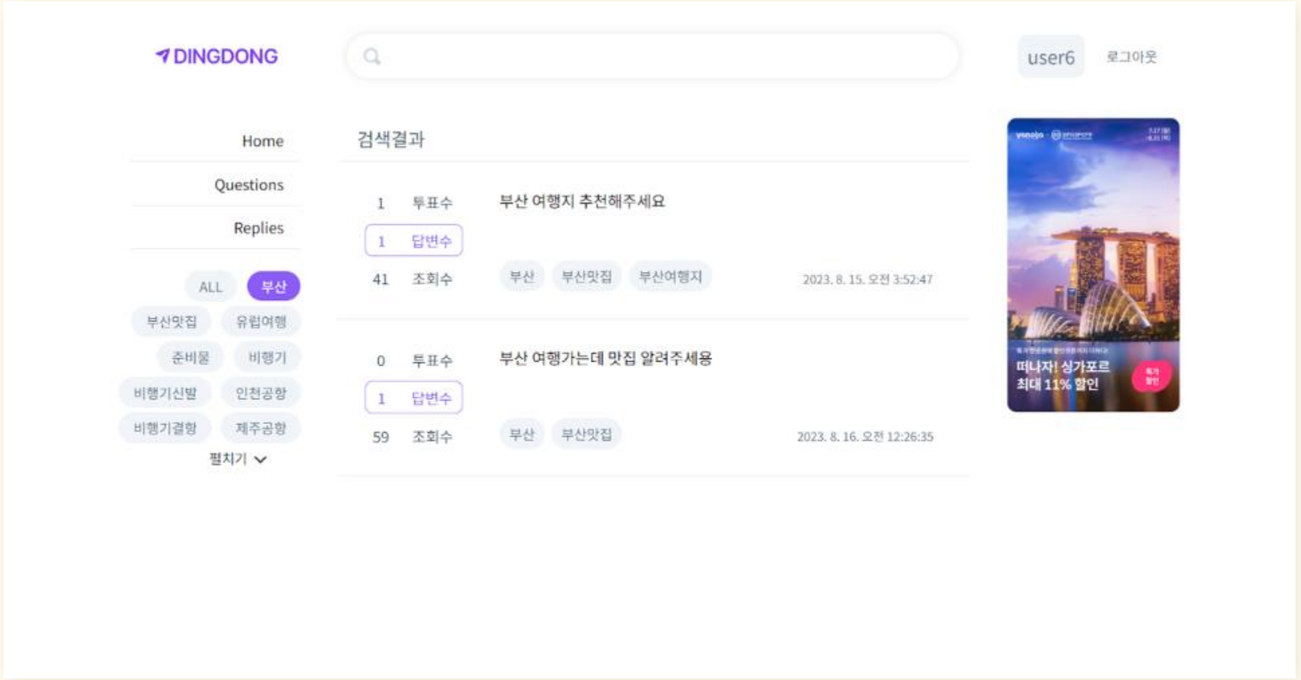


- **답변 작성/수정 가능**
- 답변글에서 수정 버튼 클릭
  → 답변 작성 폼으로 스크롤 자동 이동
- **자문자답 방지**
- **답변 횟수 사용자 당 1회 제한**
- 글씨크기, 굵기, 색상 등 스타일 조정 가능
- 이미지 첨부 가능

(5) 답변/댓글 페이지 (Replies)

- **투표순으로 정렬**
  - 답변, 댓글 각각에 대해 정렬 API 설계하여 구현
- **무한스크롤 구현**
  - react-query와 react-infinite-scroller 이용
- 각각의 답변/댓글에 질문의 해시태그 포함 렌더링
- 긴 답변/댓글의 경우 **더보기 버튼**을 추가하여
  사용자가 해당 답변/댓글을 펼치고 접을 수 있도록 함
- **삭제된 질문에 대한 답변/댓글도 조회 가능**
  - 질문이 *soft deleted* 되도록 구현함

(6) 검색결과 페이지



- 검색창에서 단어 검색 시 해당 단어를 포함하는 질문/답글 렌더링
- 해시태그 클릭 시 해당 해시태그를 포함하는 질문 렌더링

(7) 마이페이지

- 작성한 질문
- 작성한 답변
  - **질문이 지워지더라도 조회 가능**
- 작성한 댓글
  - **질문이 지워지더라도 조회 가능**
- 저장한 질문
- 저장한 답변
  - **삭제된 질문에 대한 답변도 조회 가능**
- 저장한 댓글
  - **삭제된 질문에 대한 댓글도 조회 가능**

기대효과 측면

- 지속적인 정보 활용성 증진
- 댓글 작성자 신뢰성 확보 및 커뮤니티 내 신뢰 구축
- 정보 교류와 협업 활성화
- 다양한 문화 이해와 교류 강화

- 효율적인 여행 계획 구성으로 비용절감
- 지역 경제 활성화 및 서비스 다각화
- 지역 경제 지속적 기여 및 관광 활성화

과제를 통해 배운내용 

- 답변이 질문에 종속적이지 않은 **객체화** 특성 적용
- **협업**에 알맞은 메서드 • 필드 명, commit message
- 개발 중 API **endpoint 변경** -> FE, BE 간의 소통 중요
- .prettierrc 설정

  -> single • double quote 등 코드 스타일 통일

과제를 통해 배운내용 

- NoSQL (MongoDB) 사용 - 요구사항 변경에 유연
- 게시물 삭제 될 경우 soft • hard deletion 으로 분리
  - 사용자 삭제 철회
- 어떤 에러가 발생했는지 alert 등으로 사용자에게 알림
- 복잡한 정렬 기준은 FE 보다 BE 에서 처리하는 것이 효율적

과제를 통해 배운내용 

- 'real carousel' library 를 사용한 해시태그 기능
- 검색 시 자동 완성 기능으로 검색어가 포함된 글 매핑
- react query 를 사용하여 refetch 없는 데이터 캐싱
- 절대경로로 import 하여 경로 설정 시간 단축

주요문제점 극복사항

문제점

질문과 답변을 독립적으로 객체화

-> 각각을 어떻게 저장하고,
어떻게 가져와야 할까?

-> 질문이 삭제될 경우,
어디까지 보여줘야 할까?

해결

- 답변 북마크 할 경우 기존 스키마와
복사본 스키마에 각각 저장
- 질문이 삭제될 경우
복사본 스키마에서 데이터 가져오기
- soft • hard deletion 으로
질문 삭제 철회 가능

주요문제점 극복사항

문제점

로그인 할 때 부여된 jwt token 을 localStorage 에 저장하면

계속 로그인 상태로 머무른다.

어떻게 로그인 상태를 관리할까?

해결

- token 발급 시 만료 시간 1시간으로 설정 -> 로그인 유효 시간 = 1시간
- token이 localStorage에 저장될 때 token **만료 시간**도 같이 저장
- **recoil library** 와 localStorage 활용해서 token 상태 저장 및 관리

주요문제점 극복사항

문제점

게시물 **페이지네이션** 할 때,

BE 로부터 전체 데이터를 받아서

FE 에서 페이지 별로 데이터 분할

-> 데이터 양이 많을 경우 서버 부하,

렌더링 속도 및 사용자 경험 저하 발생

해결

- **서버 사이드 페이지네이션** 사용
- 불필요한 데이터가 넘어가지 않도록
  BE 로 페이지 번호를 보내서
  해당하는 데이터만 받아옴
- 과도한 메모리 사용, 렌더링 지연,
  FE 측 코드 복잡성 해결

향후계획

로그인

- 가입한 이메일로 비밀번호 찾기
(임시 비밀번호 메일로 전송)
- 페이지 권한 설정
(회원가입, 로그인 페이지에 접근 X)
- 회원 정보 수정하기

soft deletion

- soft deletion 기간에
저장한 답변, 댓글을 통해 질문 보기 가능
- hard deletion 이후
저장한 답변, 댓글 보기 가능
해당 질문 보기 불가능

향후계획

최적화

- webpack 대신 vite 로 번들링 속도 향상 예정
- 중복되거나 불필요한 코드 제거
- 코드 스플리팅 : 논리적 단위로 분할하여 필요한 부분만 로딩
- 정적 파일 캐싱
- 서버 사이드 렌더링 (SSR) or 클라이언트 사이드 렌더링 (CSR)

Thanks!

QnA