

00 Check!



최대한 자세히 작성해주세요.
추후 포트폴리오 및 이력서를 작성할 때 큰 도움이 됩니다.

일상을 찰방으로 공유하는

Newsfeed

주특기 숙련 (React)

TEAM 12조 십이걸지마

송민수, 장소정, 장수인, 최유나
튜터 최원장, 이정민

목차

- 01 프로젝트 개요
- 02 프로젝트 팀 구성 및 역할
- 03 프로젝트 수행 절차 및 방법
- 04 프로젝트 수행 경과
- 05 자체 평가 의견

01 프로젝트 개요

1 프로젝트 주제 및 선정 배경, 기획의도

- 프로젝트 주제: 짤방과 함께 오늘 하루를 공유하는 뉴스피드
- 구현기존 유사 서비스와 차별화된 내용: 많은 소셜 미디어가 복잡한 기능을 제공하는 반면, 짤방 피드는 간단하게 짤방과 짧은 설명만으로 하루를 기록할 수 있어, 사용자의 시간과 노력을 최소화하면서 일상을 쉽게 공유할 수 있습니다.

2 프로젝트 내용

짤방과 함께 하루를 간단히 기록하고 짤방에 대한 설명을 덧붙여 하루를 공유합니다.
사용자는 짤방에 해시 태그를 추가하여 특정 주제나 상황을 태그할 수 있습니다.

3 활용 장비 및 재료

- 프레임워크 및 도구: Vite, React
- 백엔드 및 데이터베이스: Supabase
- 고유 식별자 생성: UUID
- 코드 품질 관리: ESLint 기타: 최신 빌드 및 개발 도구 설정 (Vite, @vitejs/plugin-react-swc)

4 프로젝트 구조

5 활용 방안 및 기대 효과



02 프로젝트 팀 구성 및 역할

이름	역할	담당 업무
송민수	팀원	포스트 정보를 메인 페이지에서 동적으로 표시, 비로그인 접근 허용, 메인 페이지 레이아웃과 스타일링 작업 수행
장소정	팀장	Supabase를 활용한 회원가입 기능, 사용자 프로필 정보 로드 및 사용자 이름 대체 처리 기능 구현
장수인	팀원	포스트 상세 정보, 로그인 상태에 따른 글 수정 및 삭제 처리, 비로그인 사용자에게 대한 수정 및 삭제 버튼 숨김, 포스트 열람 기능, 댓글 CRUD 기능, Supabase를 통한 찜방 상세 정보 조회
최유나	팀원	개발을 총괄하며 디자인 시스템 설정, 기능 구현 조율, Supabase 설정 및 인증을 활용한 라우팅과 사용자 인증 상태 관리, 찜방 업로드 기능 구현 및 데이터베이스 연동, 사용자 프로필 관리 기능 및 작성한 게시글 목록 표시
최원장, 이정민	튜터	SA 피드백, 주제선정 피드백, 프로젝트 기술 질의 응답

03 프로젝트 수행 절차 및 방법

▶ 프로젝트의 (1)사전 기획과 (2)프로젝트 수행 및 완료 과정으로 나누어서 작성해주세요.

구분	기간	활동	비고
기획/프로젝트 준비	8/28(수)~8/28(수)	프로젝트 기획 및 주제 선정, 기획안 작성, 역할 배분, git 관리 방안 수립	SA 문서 작성
협업 준비	8/29(목)~8/29(목)	프로젝트 세팅, 데이터베이스 테이블 설계, 개발환경 구축	주요 패키지 / prettierrc / 기본 라우팅 설정
기능 개발	8/30(금)~9/3(화)	기능 구현 1~3차	
디자인 개발	9/2(화)~9/3(수)	디자인 작업 1~2차	
최종 검토 및 배포	9/3(수)~9/3(수)	트러블 슈팅, 배포, 시연 영상 제작, 결과보고서 작성	최적화, 오류수정
총 개발 기간	8/28(수)~9/3(수) (총 1주)		

04 프로젝트 수행 경과

▶ 프로젝트 결과물이 도출된 과정을 세부적으로 작성해 주세요.

- 1 결과를 서술하는 과정에서는 활용된 기술 및 구현 방법, 핵심 기능, 구현 결과 들을 상세히 작성
- 2 프로젝트 수행 과정이 잘 드러날 수 있도록 가공과정부터 활용까지 전체적인 프로세스를 단계별로 작성
프로젝트 구현 내용, 컨셉, 훈련 내용과의 연관성 등을 포함
- 3 프로젝트 수행 과정에서의 피드백 내용과 그것을 적용(보완 등)한 내용이 포함되도록 작성
트러블 슈팅, 유저테스트, 개발환경 등
- 4 프로젝트 수행 결과물을 잘 드러낼 수 있는 자료를 첨부하여 작성
결과물 사진, 시연 영상, 구동 화면 등 프로젝트의 우수성이 드러날 수 있는 자료

04 프로젝트 수행 경과 - (1) 기술적 의사결정

1 결과를 서술하는 과정에서는 활용된 기술 및 구현 방법, 핵심 기능, 구현 결과 들을 상세히 작성

요구사항	선택지	핵심 기술을 선택한 이유 및 근거
Supabase를 활용한 CRUD 구현	Supabase, React useEffect	Supabase는 CRUD 작업과 실시간 데이터베이스 업데이트를 지원하여 빠른 구현을 가능하게 하고, React의 useEffect는 컴포넌트의 데이터 로딩과 상태 관리를 효율적으로 처리하기 때문에 선택
Supabase를 활용한 로그인 및 회원 가입	Supabase Authentication API, onAuthStateChanged	Supabase Authentication의 통합 인증 솔루션과 간편한 인증 관리할 수 있고, 이메일과 비밀번호 인증 기능을 통한 보안 강화. onAuthStateChanged를 이용한 사용자 인증 상태의 실시간 감지와 로그인/로그아웃 상태 관리 효율성을 활용하고자 선택
Context API를 활용한 전역 상태 관리	createContext(), ContextProvider, useContext를 사용하여 상태를 공유	Context API는 React 애플리케이션 내에서 전역 상태를 손쉽게 공유하고 관리할 수 있게 해주며, prop drilling을 피할 수 있고, 전역 상태 관리를 위한 간단하고 편리한 도구이기에 선택
React Router DOM (RRD) 활용	React Router DOM	React Router DOM은 SPA(Single Page Application)에서의 내비게이션을 쉽게 구현할 수 있으며, 인증 상태에 따라 접근 제어를 손쉽게 설정할 수 있는 기능을 제공하기 때문에 선택
마이 페이지 구현	Supabase Storage API	Supabase Storage는 파일 저장 및 관리 기능을 제공하고, 이미지 업로드 및 다운로드 URL을 처리하는 데 적합하다. 이를 통해 사용자 프로필 이미지 관리 및 게시물 관련 이미지 처리를 효과적으로 수행할 수 있기 때문에 선택
배포하기	Vercel	Vercel은 CI/CD 파이프라인을 지원하며, 자동 배포 및 스케일링이 용이하여 React 애플리케이션의 배포에 적합하기 때문에 선택
Git을 활용한 협업	Git	Git은 코드 버전 관리와 팀 협업을 효율적으로 지원하며, Pull Request와 브랜치 관리를 통해 협업 프로세스를 체계적으로 관리있기 때문에 선택



04 프로젝트 수행 경과 - (2) 개발 프로세스

② 프로젝트 수행 과정이 잘 드러날 수 있도록 가공과정부터 활용까지 전체적인 프로세스를 단계별로 작성

1. 프로젝트 개요

- 1) 프로젝트 이름: 짬방 피드
- 2) 목표: 짬방과 함께 하루를 기록하고 공유하는 뉴스피드 애플리케이션 개발

4) UI 구성

- 컴포넌트 개발: React 컴포넌트 설계 및 구현
- 스타일링: CSS를 통한 디자인 적용 및 반응형 디자인 구현
- 테스트 및 검토: UI/UX 테스트 및 디자인 일관성 검토

2. 기능 개발 단계

1) 짬방 게시 기능

- 기능 구현: 이미지 업로드 및 게시물 저장 기능 개발
- UI/UX 디자인: 와이어 프레임 디자인 시안 개발
- 테스트 및 검토: 기능 테스트

2) 사용자 인증

- Supabase 설정: 인증 관련 API 설정 및 연동
- 회원 가입 및 로그인: 회원 가입 페이지 및 로그인 페이지 개발
- 테스트 및 검토: 인증 기능 검증 및 보안성 테스트

3) 데이터 저장 및 관리

- 데이터베이스 설계: 데이터 모델 정의 및 실시간 데이터베이스 설정
- 데이터 저장 및 업데이트: 데이터 저장 기능 및 실시간 업데이트 구현
- 테스트 및 검토: 데이터 정확성 검증 및 성능 테스트

3. 프로젝트 완료 및 배포

- 1) 최종 테스트 및 검토: 종합 테스트 및 버그 수정
- 2) 배포 준비: 배포 환경 설정 및 애플리케이션 서버에 배포
- 3) 유지보수 및 업데이트: 애플리케이션 모니터링 및 기능 개선, 버그 수정

04 프로젝트 수행 경과 - (2) 개발 프로세스

② 프로젝트 수행 과정이 잘 드러날 수 있도록 가공과정부터 활용까지 전체적인 프로세스를 단계별로 작성

• 프로젝트 개요

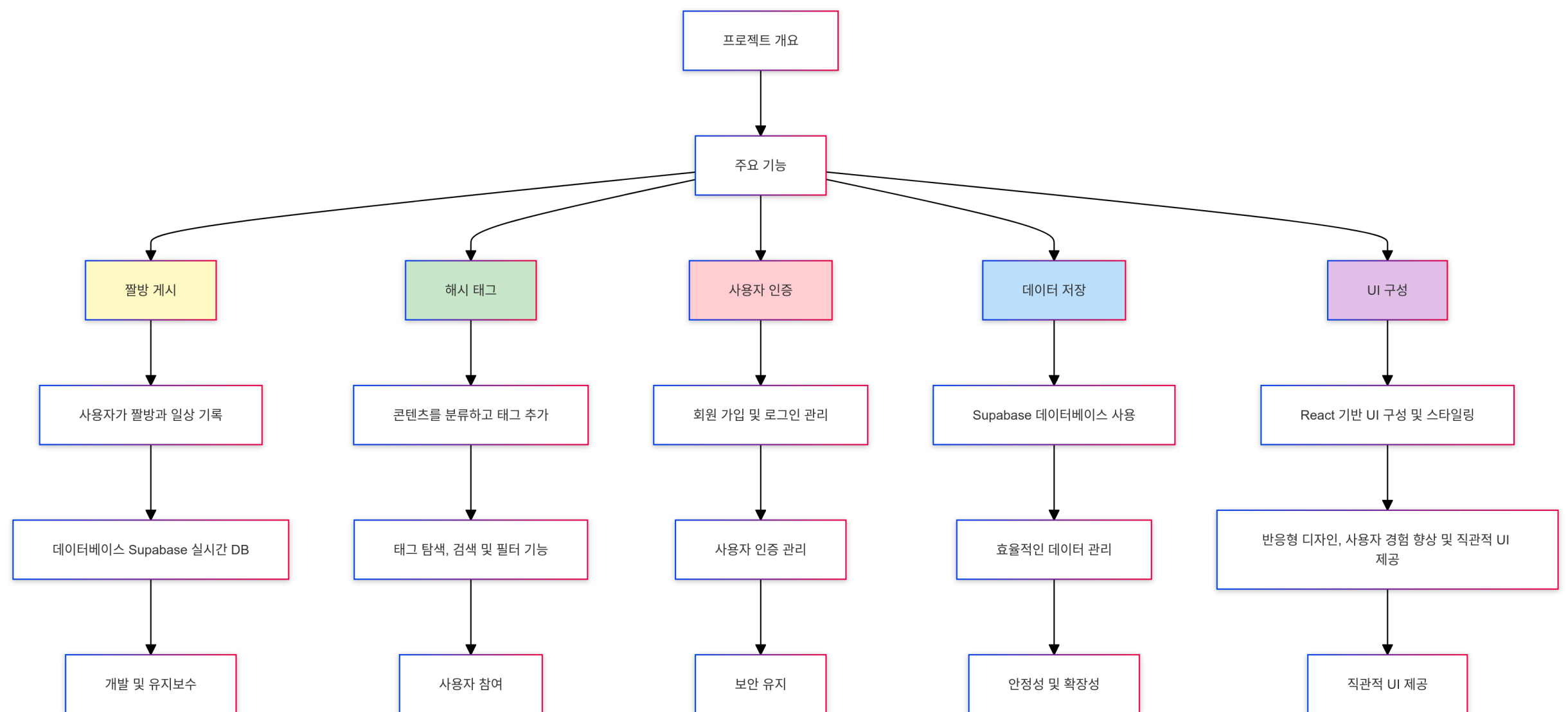
- 찰방 피드: 찰방과 함께 하루를 기록하고 공유하는 뉴스피드 기능을 가진 애플리케이션 개발

• 주요 기능

- 찰방 게시: 사용자가 찰방과 함께 일상적인 순간을 기록하고 게시
- 해시 태그: 게시물에 태그를 추가하여 분류하고 검색 가능하게 함
- 사용자 인증: Supabase를 활용하여 회원 가입 및 로그인 기능 제공
- 데이터 저장: Supabase를 통해 실시간 데이터베이스 및 데이터 관리
- UI 구성: React를 기반으로 한 UI 컴포넌트 및 디자인 구성

• 훈련 내용과의 연관성

- 찰방 게시: 사용자 경험을 고려한 UI/UX 디자인 및 기능 구현 훈련
- 해시 태그: 데이터 분류 및 검색 기능을 활용한 효율적인 데이터 관리
- 사용자 인증: 인증 시스템 구축 및 보안 관리 훈련
- 데이터 저장: 실시간 데이터베이스 관리 및 데이터 처리 기술
- UI 구성: React를 이용한 컴포넌트 기반 UI 개발 및 스타일링



04 프로젝트 수행 경과 - (3) 트러블슈팅, 유저테스트

3 프로젝트 수행 과정에서의 피드백 내용과 그것을 적용(보완 등)한 내용이 포함되도록 작성 트러블 슈팅, 개발환경 등 / 아래는 예시를 위한 이미지로, 내용에 대한 상세 설명을 작성해주셔야 합니다.

• 트러블 슈팅 1)

문제:

- 인증 정책 필요: users Auth 테이블에서 profiles 테이블로 user.uid를 받아오며 새로운 데이터를 추가할 때 인증 정책이 필요하였습니다.
- 정보 추가 어려움: profiles 테이블에서 가입 후 마이 페이지에서 정보를 받아와서 추가할 때 어려움이 발생하였으며, 특히 post 연동에서 문제가 있었습니다.
- 전체 페이지 정보 뿌리기: 이미 만들어진 session에서 getUserProfile 함수로 profile 함수의 이메일, 프로필, 닉네임을 설정한 후 이를 user 정보에 넣어 전체 메인 페이지에 뿌리는 과정에서 예외처리 문제가 있었습니다.

해결:

- 인증 정책: 인증 정책을 정의하고 profiles 테이블에 접근할 때 인증된 사용자인지 확인하는 로직을 추가하였습니다. 이를 위해 백엔드에서 인증된 사용자만이 profiles 테이블에 데이터를 추가할 수 있도록 미들웨어를 사용하거나 인증 토큰을 검증하는 방법을 적용하였습니다.
- 정보 추가 문제: 데이터 전송 과정에서 post 요청이 올바르게 구성되었는지 확인하였으며, 필요한 정보가 누락되지 않았는지 점검하였습니다. 서버에서 클라이언트로부터 받은 데이터가 올바르게 처리되도록 검토하였습니다. 데이터베이스 스키마를 점검하여 profiles 테이블에 추가할 수 있는 필드가 제대로 설정되어 있는지 확인하였습니다.
- 전체 페이지 정보 뿌리기: getUserProfile 함수가 profile 함수의 이메일, 프로필, 닉네임을 올바르게 받아와서 user 정보에 설정하도록 하였습니다. 페이지가 로드될 때 데이터를 받아오지 못한 경우를 대비하여 예외 처리 로직을 추가하였습니다. 데이터가 없을 때 기본값을 설정하거나 오류 메시지를 사용자에게 보여주는 방식으로 문제를 해결하였습니다.

• 트러블 슈팅 2)

문제:

- 댓글 작성자 정보 누락: 댓글 작성자의 username을 불러와야 하는데 null 값이 뜨며, profiles 테이블과 comment 테이블을 JOIN 하여도 여전히 null 값이 나타났습니다.
- 정보 전달 문제: 댓글 추가 시 username 정보가 누락되어 addComment 함수에서 username: user.username을 추가하여 보냈으나, 문제가 해결되지 않았습니다.

해결:

- 댓글 작성자 정보 확인: profiles 테이블과 comment 테이블의 JOIN 쿼리를 확인하여 username 필드가 제대로 조인되고 있는지 검토하였습니다. JOIN 조건이 올바르게 설정되었는지 확인하고, 두 테이블 간의 관계를 정확하게 설정하였습니다. 예를 들어, 댓글 작성자의 user_id와 profiles 테이블의 user_id가 정확히 매핑되었는지 점검하였습니다.
- 정보 전달: addComment 함수에서 username이 포함되도록 하는 것은 좋았지만, 서버에서 이 정보를 처리하는 방식도 점검하였습니다. 서버가 클라이언트에서 받은 username 정보를 올바르게 데이터베이스에 저장하고 있는지 확인하였습니다. 데이터가 저장되기 전에 username이 제대로 설정되었는지 확인하기 위해 로그를 추가하여 디버깅하였습니다.

04 프로젝트 수행 경과 - (3) 트러블슈팅, 유저테스트

3 프로젝트 수행 과정에서의 피드백 내용과 그것을 적용(보완 등)한 내용이 포함되도록 작성

트러블 슈팅, 개발환경 등 / 아래는 예시를 위한 이미지로, 내용에 대한 상세 설명을 작성해주셔야 합니다.

• 트러블 슈팅 3)

문제:

- 이미지 업로드 후 수정이 완료되었을 때 새로운 이미지가 반영되지 않는 문제

해결:

- 이미지 URL 상태 업데이트: 이미지 업로드 후 새로운 이미지의 URL이 상태에 정확히 설정되도록 확인했습니다. `uploadImage` 함수에서 이미지를 업로드한 후 `setImage(publicUrl)`을 호출하여 상태를 업데이트했습니다. 이를 통해 업로드된 이미지의 URL이 올바르게 `image` 상태에 반영되었는지 로그를 추가하여 검토했습니다. 문제가 발생한 경우, `publicUrl` 값이 예상대로 반환되는지 확인했습니다.
- Supabase 저장소에서 올바른 공개 URL을 가져오는지 검토: 이미지가 Supabase 저장소에 업로드된 후, 해당 이미지의 공개 URL을 정확히 가져오는지 확인했습니다. `uploadImage` 함수에서 `getPublicUrl`을 호출하여 이미지의 공개 URL을 가져왔습니다. 이 과정에서 오류가 발생하는지 점검하고, URL이 올바르게 반환되는지를 검토했습니다. 만약 오류가 발생한다면, Supabase의 저장소 설정이나 권한을 점검하여 해결했습니다.
- 폼 제출 시 최신 상태가 사용되도록 확인: `handleUpdate` 함수에서 `title`, `description`, `image` 상태를 사용하여 게시글을 업데이트할 때, `image` 상태가 최신 이미지 URL을 포함하고 있는지 점검했습니다. 폼 제출 시 `image` 상태가 최신 URL을 포함하여 게시글이 업데이트되도록 했습니다. 이를 위해, 상태가 업데이트된 후 `handleUpdate` 함수가 호출되기 전에 모든 상태 값이 올바르게 설정되었는지 확인했습니다.

• 트러블 슈팅 4)

문제:

- 메인페이지에서 포스트 정보를 가져올 때, 사용자 정보를 포함한 프로필 정보를 가져오지 못함

해결:

- 테이블 조인 및 사용자 정보 추가: `posts` 테이블에서 포스트 정보를 가져오는 코드가 있지만, 사용자 정보가 포함된 프로필 정보는 가져오지 못했습니다. 이를 해결하기 위해 `profiles` 테이블과 `posts` 테이블을 조인하여 포스트와 사용자 정보를 함께 가져오는 쿼리를 수정했습니다.
- 상태 관리 및 데이터 업데이트: 가져온 사용자 정보를 `data`와 함께 관리할 수 있도록 상태를 업데이트했습니다. `profiles` 테이블에서 추가적인 사용자 정보를 불러오는 로직을 포함하여, `user_profile`과 `username` 필드를 포스트와 함께 사용하는 방식으로 데이터를 처리했습니다.
- 미래의 개선 사항으로 테이블 구조를 처음부터 명확히 정의하고, 조인 쿼리의 필요성을 미리 고려하여 데이터베이스 설계를 신중하게 진행할 필요가 있음을 인식했습니다. 이를 통해 복잡한 데이터 처리 문제를 사전에 방지하고, 데이터를 효율적으로 가져올 수 있도록 개선할 예정입니다.

04 프로젝트 수행 경과 - (4) 발표영상, 이미지

④ 프로젝트 수행 결과물을 잘 드러낼 수 있는 자료를 첨부하여 작성

결과물 사진, 시연 영상, 구동 화면 등 프로젝트의 우수성이 드러날 수 있는 자료

- ▶ 발표영상이 아닌 세부 기능 소개, 화면 구동 및 기능 동작 여부를 확인할 수 있는 **시연영상**으로 제작해야 합니다.
- ▶ 용량제한 : 팀별 **5-10분 내** (100MB 이하), **기능별 소개 음성**을 포함해주세요!
- ▶ 자세한 사항은 시연영상 촬영 가이드라인을 참고해주세요 :)

05 자체 평가 의견

- ▶ 프로젝트 결과물에 대한 프로젝트 기획 의도와의 부합 정도 및 실무 활용 가능 정도, 달성도, 완성도 등
개개인 별로 자체적인 평가 의견과 느낀 점을 작성해주세요.

1

사전 기획의 관점에서
프로젝트 결과물에 대한 완성도 평가 (10점 만점)

2

개인 또는 우리 팀이 **잘한 부분과 아쉬운 점**
ex) 렌더링 속도 측정 결과, 00%로
결과도 향상을 위해 추후 개선 필요

3

프로젝트를 수행하면서
느낀 점이나 경험한 성과 (경력 계획 등과 연관)

4

프로젝트 결과물의
추후 개선점이나 보완할 점 등 내용정리

05 자체 평가 의견 - (1) 프로젝트 결과물에 대한 완성도 평가

▶ 개개인 별로 자체적인 평가 의견과 느낀 점을 작성해주세요.

이름	평가 점수 (10점 만점)	평가 점수 사유
송민수	0점	프로젝트 초반에 제가 팀원분들이 공유해주신 사항이 이해가 가지 않을 때마다 적극적으로 소통하고 이해를 하려고 했었어야 되는데, 혼자서 이해하려하다 보니까 프로젝트에 대한 전반적인 이해가 많이 늦어져서 완성도는 처음에 계획했던 것 만큼 안 나왔지만, 그래도 기능들은 다 구현을 했기 때문에, 다음번에는 더 좋은 결과물이 나올거라고 생각합니다.
장소정	8점	짤방 게시와 사용자 인증 기능이 제대로 작동하고, 데이터 저장과 관리도 안정적으로 수행되고 있고, UI/UX 디자인도 직관적이라 전반적으로 목표를 잘 달성한 것 같습니다.
장수인	5점	계획 했을 때와 다르게 개인 실력 문제로 완벽하게 하지 못한 부분이 아쉽습니다.
최유나	5점	supabase를 활용해 다양한 로그인 연동을 계획했지만 개인 능력치의 한계로 인해 구현하지 못한 것이 너무 아쉽습니다.

05 자체 평가 의견 - (2) 개인 또는 우리 팀이 잘 한 부분과 아쉬운 점, (3) 느낀점, 성과

▶ 개개인 별로 자체적인 평가 의견과 느낀 점을 작성해주세요.

송민수

‘자체 평가 의견(1)’과 마찬가지로 제 이해가 늦어져서 팀 프로젝트가 지연이 되서 개인적으로 너무 부족했습니다. 팀원분들이 깃헙도 세세하게 알려주셔서 너무 감사했고, 다들 기능구현을 잘해주셔서 팀 프로젝트가 완성이 된 것 같습니다.

장소정

기능이 잘 작동하고 UI/UX 디자인이 사용자 친화적으로 개발된 부분에 만족합니다. 아쉬운 점은 개발 기간이 길어졌고 실시간 데이터 업데이트에 성능 이슈가 있었지만, 협력과 문제 해결 과정에서 많은 것을 배웠습니다.

팀원 분들 모두 의사소통도 열심히 잘 하여 진행이 잘 된 것 같습니다
다만 제 개인 실력이 부족하다는 것을 많이 느꼈고 그로 인해 시간 소비가 너무 컸던 부분들이 아쉽습니다

장수인

팀원 분들이 서로 노력하며 잘 진행 한 것 같습니다.
팀원간의 소통이 제일 중요하고 바로바로 피드백하는 것이 중요한데 초반에 팀원분들께 잘 공유 드리지 못한 것 같아서 너무 아쉽습니다.
이해가 안 가는 부분이 있으면 빨리빨리 캐치하고 튜터님을 찾아뵈어야 했는데 너무 혼자 끙끙거리느라 시간 분배를 잘 하지 못한 점이 아쉽습니다

최유나

05 자체 평가 의견 - (4) 프로젝트 개선점 / 보완할 점

• 개발 프로세스

- 개발 기간 관리: 프로젝트 초기 계획보다 개발 기간이 길어졌습니다. 이는 일정 관리와 타임라인 설정이 부족했기 때문입니다. 향후 프로젝트에서는 더 세밀한 일정 계획과 주기적인 진행 상황 점검을 통해 개발 기간을 단축하고 목표 달성 시간을 줄일 필요가 있다고 판단하였습니다..
- 역할 분배: 프로젝트 초반에 팀 내 역할 분배가 명확하지 않아 작업의 중복과 비효율이 발생했습니다. 앞으로는 프로젝트 시작 단계에서 각 팀원의 역할과 책임을 명확히 정의하고, 이를 기반으로 작업을 분담하여 협업의 효율성을 높이겠습니다.

• 기능 개선

- 게시글 작성 시 사용자 경험을 향상시키기 위해 토스트 에디터를 적용하지 못한 점이 아쉽습니다. 이 기능을 추가하면 사용자들이 더 직관적으로 게시글을 작성할 수 있을 것입니다. 향후 업데이트에서 이 기능을 도입하여 사용자 편의성을 높이고, 콘텐츠 작성의 효율성을 개선할 계획입니다.