



## 서비스

### 개발자 협업 웹 서비스

GitPulse는 팀 개발자의 GitHub 활동 데이터를 수집하고 시각화 하여  
개발 습관과 협업 패턴을 한눈에 파악할 수 있도록 돕는 웹 애플리케이션입니다.

## 개발 기간

2025. 05. 07 - 2025. 05. 20 (2주)

서비스 링크

상세정보(노션)

디자인 링크(피그마)

<https://github.com/GitPulse-04>

## 사용 기술 스택

JS

JavaScript

B

Bootstrap

⚛️

React

⚡

Vite

🎨

Figma

📄

HTML5

## 구성원 (4명)

김현우 / 이예은 / 박용규 / 임재찬

## 역할 팀장

### Design

프로토타입 제작 (figma)  
UI/UX 디자인 [반응형\_Mobile, PC] (figma)

### FrontEnd

로그인 & 회원가입  
IT NEW 페이지  
백엔드 연동 DB 설계

## 프로젝트 회고

- 협업을 위한 플랫폼을 만들게 되면서 팀원과 소통을 많이 생각해보는 시간이 되었고 선과 여백 하나에도 모두가 모여 좀 더 나은 결과물을 창작하는 과정에서 협업의 힘을 실감 할 수 있었습니다. 좋은 팀원들을 만나 즐겁게 프로젝트를 진행해서 재미있었습니다.
- 직관적인 UI가 사용하는데 있어서 많은 영향을 끼친다는 것을 알게되었습니다. 이를 통해 피드백을 받으며 UI/UX를 개선하는 경험이 매우 값진 시간으로 이어질수 있었습니다.

## 주요 시스템 구성 요소 및 기술 스택

### Frontend (React + TailwindCSS)

- UI/UX
  - 로그인 화면: GitHub OAuth 로그인
  - 대시보드: 팀/개인 활동 시각화
  - 커밋/이슈 이벤트 발생 시 알림 & 자동 업데이트
  - 데이터 시각화
    - Chart.js / Recharts 등으로 활동 데이터 시각화
    - “허수 커밋” 필터링 결과 표시 (LOC, 메시지 분석)

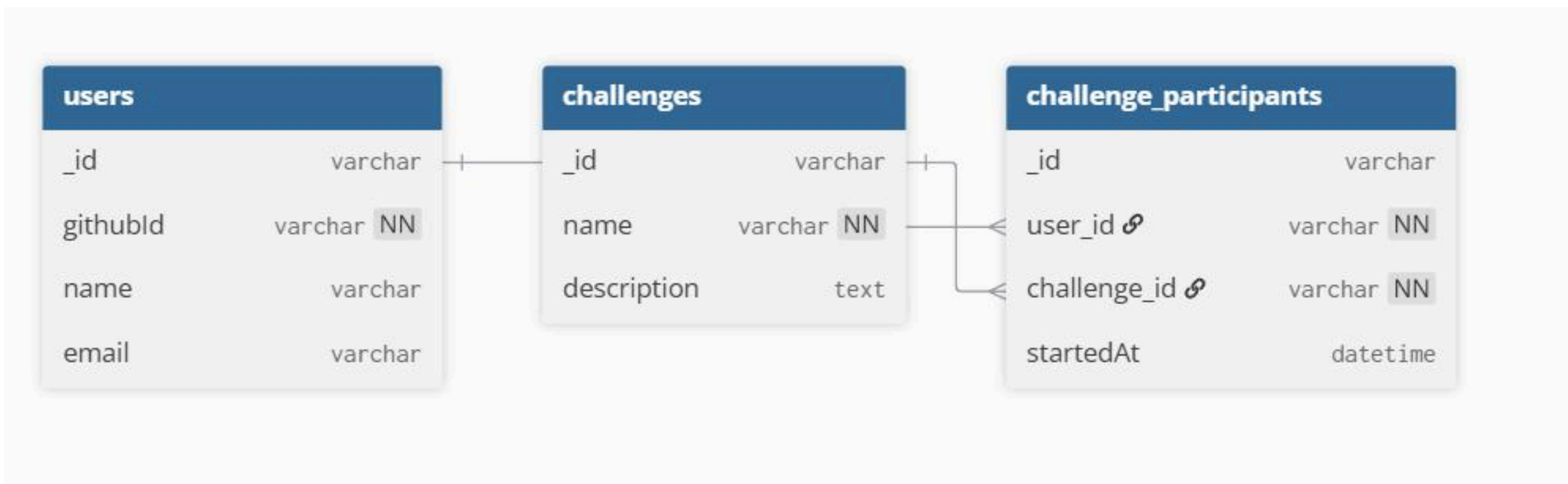
### Backend (Node.js + Express )

- 인증
  - GitHub OAuth 로그인 (Access Token 발급 및 저장)
- API 연동
  - GitHub REST / GraphQL API 호출
    - Repo 생성/삭제
    - Branch 생성
    - Issue 생성/닫기/수정
    - PR 목록 조회 & 머지 상태 확인
  - DB 연동 (MongoDB)
    - 사용자 프로필 저장 (GitHub 계정)
    - 대화 기록 저장 (GPT 질의응답 로그)
    - 프로젝트/팀 활동 데이터 캐싱

### 차별화 기능 (GitHub 대비)

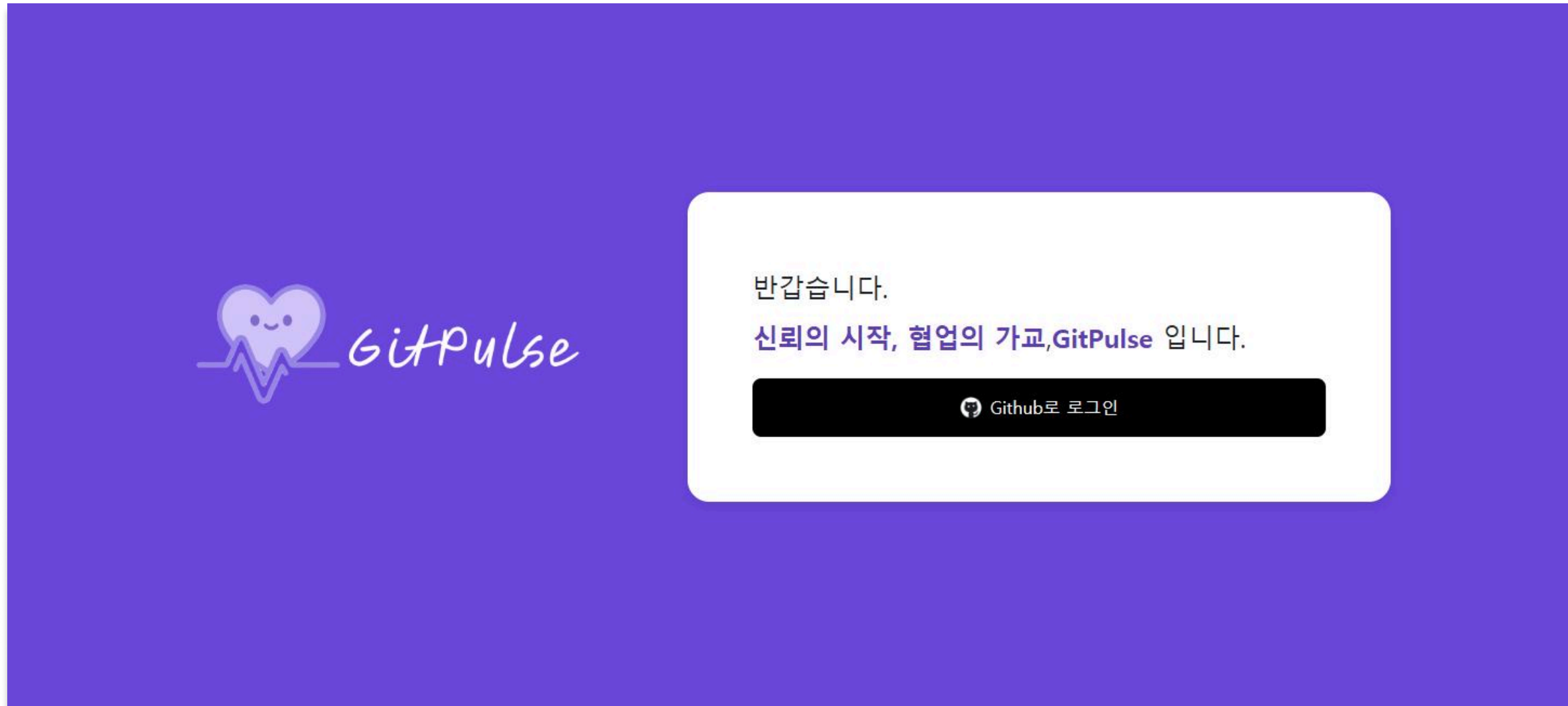
1. 허수 커밋 감지
  - 커밋 메시지 필터링: "fix typo", "update", "." 등
  - LOC(Line of Code) 변화량 기반 → 무의미 커밋 제거
2. 팀 협업 인사이트
  - 주간 커밋 분석 → 배지/멘트 제공
  - 팀원별 최근 활동 타임라인 → 바로 리뷰 가능
  - 좋아요/즐거찾기 → 인기 커밋 상단 노출
3. 품질 중심 분석
  - 리뷰 개수 + 리뷰 내용 반영 여부 분석
  - 컨벤션 준수 여부 체크
4. AI 지원
  - “누가 허수 활동을 많이 하는가?” 자동 인사이트 제공
  - 팀 협업 개선 팁 제안

## ERD



## Git OAuth

Git OAuth 로그인을 통한 회원가입 처리 및 토큰 발급



### 주요 기능 및 구현 기술

- OAuth 2.0 기반 인증:  
GitHub 계정을 이용해 사용자 인증을 수행, 별도의 회원가입 절차 없이 빠르게 로그인 가능.
- Access Token 발급:  
GitHub이 발급한 Authorization Code를 이용해 Access Token을 받아 사용자의 기본 프로필 정보, 이메일, 저장소 권한 등을 확인.
- 소셜 로그인 연동:  
passport-github2 와 같은 라이브러리 또는 직접 OAuth 플로우 구현을 통해 Node.js/Express, React 기반 서비스에 통합.
- 보안 강화:  
사용자 비밀번호를 직접 관리하지 않고 GitHub 인증을 위임하여 보안 리스크 감소.

### 기술적 도전 및 해결 방안

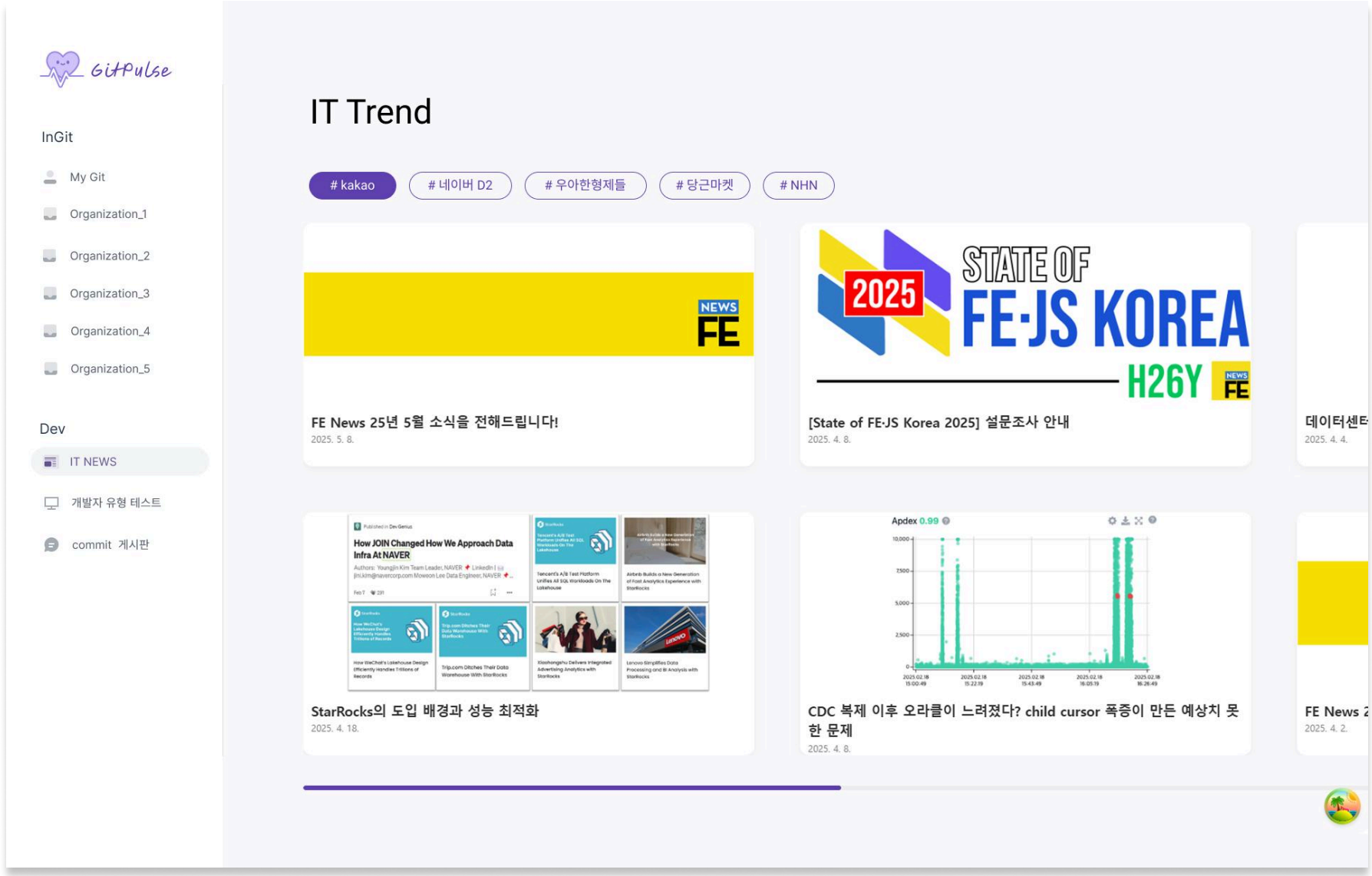
- 문제: OAuth 플로우 이해 부족  
→ OAuth는 Authorization Code Flow, Implicit Flow 등 여러 방식이 있으며, GitHub에서는 Authorization Code Flow를 권장. 이를 명확히 이해하지 못하면 잘못된 토큰 처리나 보안 이슈 발생 가능.  
→ 해결: GitHub 공식 문서와 OAuth RFC 문서를 참고해 흐름(인증 요청 → 코드 발급 → 토큰 교환 → API 접근)을 단계별로 구현.
- 문제: 리다이렉션 및 콜백 처리  
→ GitHub 인증 후 리다이렉트될 URL을 서버에 명확히 등록하지 않으면 redirect\_uri\_mismatch 오류 발생.  
→ 해결: .env에 GITHUB\_CLIENT\_ID, GITHUB\_CLIENT\_SECRET, REDIRECT\_URI를 설정하고 GitHub OAuth App 설정에 동일하게 등록.

### 주요 성과

- 사용자 경험 개선:  
별도의 회원가입 없이 GitHub 계정으로 원클릭 로그인 가능, 진입 장벽을 낮춤.
- 보안성 강화:  
비밀번호를 직접 저장하지 않아 유저 데이터 유출 위험 감소.
- 개발 효율성 향상:  
GitHub API를 통해 사용자 프로필, 저장소, 활동 데이터까지 쉽게 가져올 수 있어 다양한 부가 기능(예: 커밋 기반 챌린지 서비스)으로 확장 가능.

## IT NEWS

RSS 기반 최신 IT 트렌드 뉴스 제공



### 주요 기능 및 구현 기술

- RSS 피드 연동
  - Kakao Tech, 네이버 D2, 우아한형제들 기술 블로그 등 주요 IT 기업의 RSS 피드를 수집.
  - 서버에서 주기적으로 데이터를 가져오고, 클라이언트(React 기반)에서는 카드 형태로 시각화.
- 태그 기반 필터링
  - #kakao, #네이버 D2, #우아한형제들 같은 태그 버튼으로 특정 기업 소식만 빠르게 조회 가능.
- 카드 UI/UX
  - 각 뉴스 항목을 이미지, 제목, 날짜와 함께 카드 레이아웃으로 표시하여 가독성 강화.

### 기술적 도전 및 해결 방안

- 문제: 최신 뉴스 동기화 지연
  - RSS는 실시간 푸시가 아니어서 새로운 글 반영이 늦어질 수 있음.
  - 해결: 일정 주기(예: 10분/1시간)로 Cron Job을 실행해 피드를 갱신.
- 문제: 이미지/메타데이터 누락
  - 일부 피드에는 썸네일 이미지가 없거나 설명이 HTML 태그로만 제공됨.
  - 해결: Open Graph(OG) 메타태그 크롤링으로 보강하거나 기본 썸네일 제공.

### 주요 성과

- IT 최신 동향 집약 제공
  - 여러 기업 블로그를 직접 방문하지 않아도, 한눈에 최신 소식을 확인 가능.
- 사용자 편의성 증대
  - 태그별 필터와 카드형 시각화로 빠른 탐색과 직관적 접근 제공.
- 확장 가능성 확보
  - 새로운 RSS 피드 추가가 용이해 지속적인 확장성 보장.