

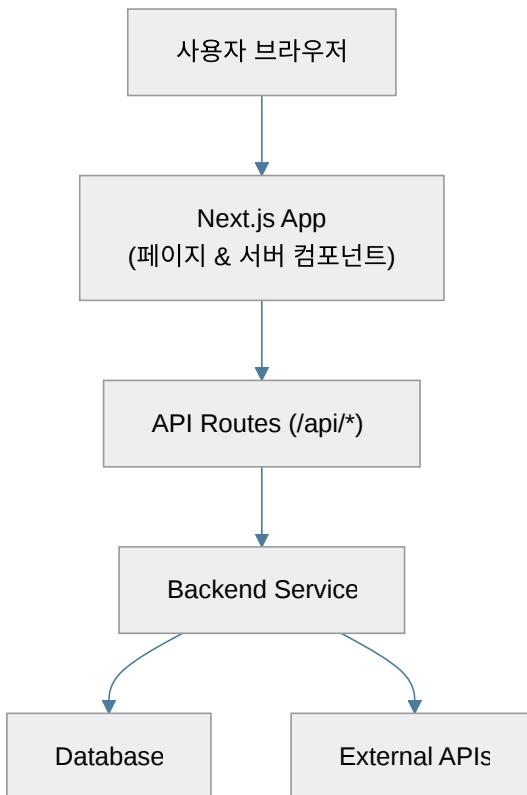
구조 가이드 예시

1. 프로젝트 개요

이 프로젝트는 사용자의 활동 데이터를 기반으로 한 개인화된 웹 서비스입니다. 프론트엔드는 Next.js 기반으로 페이지/데이터 렌더링을 최적화하며, 핵심 기능은 인증, 데이터 시각화, 사용자 설정 관리입니다.

기술 스택 요약: **Next.js** · **TypeScript** · **Zustand** · **shadcn/ui** · **REST API** · **Vercel**

2. 전체 구조



전체 구조는 클라이언트 → Next.js → 서버 API → DB의 단방향 흐름을 유지합니다.

3. 폴더 구조 요약

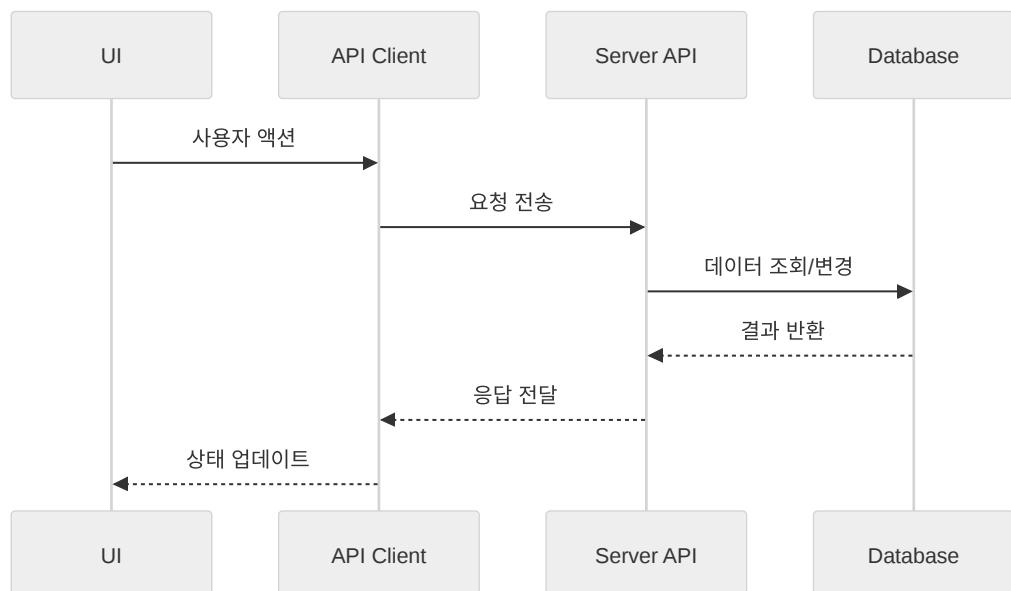
```
src/
  ├── app/      페이지 라우팅 및 서버 컴포넌트
  ├── components/  공통 UI 컴포넌트
  ├── features/   기능 단위 도메인 모듈
  ├── lib/        API 클라이언트, 유ти
  ├── styles/    전역 스타일
  └── types/    전역 타입 정의
```

폴더는 역할 기반 + 기능 단위(feature-first) 원칙을 따릅니다.

4. 기술 스택 및 선택 이유

영역	선택	이유
Framework	Next.js	CSR/SSR 혼합 + SEO 최적화
상태관리	Zustand	최소 전역 상태 + 간결한 API
API Layer	fetch wrapper	에러/응답 형식 일원화
UI Library	shadcn/ui	커스텀 가능한 Headless 기반
배포	Vercel	프리뷰/프로덕션 관리 편리
테스트	Playwright	핵심 플로우 자동화

5. 데이터 흐름 요약



6. 핵심 아키텍처 원칙

- 단방향 데이터 흐름 유지: UI → API → Server → DB
 - 기능 단위 분리(Feature-first): 도메인 중심 구조
 - 컴포넌트 계층화: Primitive → Core → Feature
 - API 에러 처리 일원화: wrapper에서 통합 처리
 - 재사용 우선 구조: 공통 컴포넌트는 components/에 한정
-

