

Frontend 개발 명세서

1. 기술 스택

- Framework: React 19
- 언어: JavaScript
- 상태 관리: 없음 (Context API, Redux 등 미사용)
- 스타일링: 기본 CSS
- 빌드 도구: Vite

2. 프로젝트 구조

```
frontend/
├── .dockerignore
├── .env.local
├── .env.production
├── .gitignore
├── abc.txt
├── Dockerfile
├── eslint.config.js
├── index.html
├── package-lock.json
├── package.json
├── vite.config.js
├──
├── docker/
│   └── nginx.conf
├──
├── public/
│   └── vite.svg
├──
└── src/
    ├── App.jsx
    ├── App.css
    ├── main.jsx
    ├──
    ├── api/
    │   ├── auth.js
    │   └── axios.js
```

Docker 빌드 설정
ESLint 설정
앱의 진입 HTML

프로젝트 의존성
Vite 설정

Docker 관련 설정 디렉터리
Nginx 프록시 설정

정적 자산
Vite 기본 로고

소스 코드 루트
루트 컴포넌트

진입점 (ReactDOM.render)

API 요청 모듈
로그인/회원가입 관련 API
Axios 인스턴스 설정

charts.js	# 차트 데이터 API
diet.js	# 식단 관련 API
record.js	# 기록 관련 API
todo.js	# 할 일 관련 API
user.js	# 사용자 정보 API
user info.js	# 추가 사용자 정보 API
assets/	
favicon-32x32.png	# 이미지, 아이콘 등 정적 파일
components/	
charts/	# 재사용 가능한 컴포넌트 모음
ChangeTrendChart.jsx	# 차트 컴포넌트
UserCompareChart.jsx	
cheering/	# 응원 메시지 팝업
CheeringPopup.css	
CheeringPopup.jsx	
header/	# 헤더
Header.css	
Header.jsx	
loading/	# 로딩 화면
Loading.css	
Loading.jsx	
main/	# 메인 페이지 구성 요소
Calendar.css	
Calendar.jsx	
MainModal.css	
MainModal.jsx	
Todolist.css	
Todolist.jsx	
Meal.jsx	
Meal.css	
Video.jsx	
Video.css	
toast/	# 토스트 메시지
ErrToast.jsx	



3. 환경 설정

.env.local

VITE_API_URL=

.env.production

VITE_API_URL=

4. 주요 페이지

페이지 경로	주요 컴포넌트	기능 설명
/ 또는 /login	LoginPage	사용자 로그인, JWT 저장 및 인증 처리
/signup	SignupPage	회원가입, 입력 유효성 검사 및 사용자 등록
/main	MainPage, Calendar, Todolist, MainModal, CheeringPopup	메인 대시보드, 일정 확인 및 응원 메시지 출력

/profile	ProfilePage, Header 등	사용자 정보 조회 및 수정, 로그아웃 기능
/graph	GraphPage, UserCompareChart, ChangeTrendChart	건강 데이터 통계 시각화 (체중, BMI 등)
—	PrivateRoute	인증되지 않은 사용자의 보호 페이지 접근 제한

5. 컴포넌트

5.1 공통 컴포넌트

컴포넌트명	설명
Header	상단 네비게이션, 로고 및 사용자 메뉴
Toast / ErrToast	알림 메시지 토스트 (성공/에러)
Loading	비동기 처리 중 스피너 표시

5.2 메인 컴포넌트 (달력/기록)

컴포넌트명	설명
Calendar	날짜별 기록 표시용 달력
Todolist	할 일 목록 (운동/식단 등)
MainModal	날짜 선택 시 팝업 상세 보기
CheeringPopup	응원 메시지 표시
Meal	식단 표시
Video	유튜브 영상 표시

5.3 차트 컴포넌트

컴포넌트명	설명
UserCompareChart	내 기록 vs 평균 비교 그래프
ChangeTrendChart	최근 기록 변화 추이 그래프

6. 상태 관리 (localStorage 사용)

→ useState, useEffect, useContext 기반 로컬 상태 관리

추후 상태 관리가 필요할 경우 Zustand 또는 Redux Toolkit 도입 가능

7. API 연동

- HTTP 클라이언트: axios + 공통 인스턴스 구성
- API 모듈 위치: src/api/

API 모듈	역할
auth.js	로그인, 회원가입, 인증 처리
record.js	건강 기록 조회 및 저장
diet.js	식단 조회/등록 API
todo.js	할 일(Todo) CRUD
charts.js	통계용 차트 데이터 조회
userinfo.js	사용자 프로필 조회 및 수정

8. 주요 기능

- 반응형 디자인 (CSS 및 MUI 일부 사용)
 - 건강 기록/식단/운동 데이터 시각화 (Recharts)
 - 일정 기반 기록 등록 (달력 UI 기반)
 - JWT 기반 로그인 / 보호 라우팅 처리
 - 로딩 상태 및 토스트 메시지 처리 (Loading, Toast 컴포넌트)
-

9. 빌드 및 배포

```
"scripts": {
```

```
"dev": "vite",  
"build": "vite build",  
"lint": "eslint .",  
"preview": "vite preview"  
}
```

- 개발 서버 실행: npm run dev
- 빌드: npm run build
- 미리보기: npm run preview
- 코드 검사: npm run lint

10. Docker 설정

- 멀티스테이지 빌드 사용
 - node:lts → 앱 빌드
 - nginx:alpine → 정적 파일 서빙
- 관련 파일
 - Dockerfile (최상단)
 - docker/nginx.conf (프록시 설정 포함)