

김지훈
Frontend Engineer

2001년생

Frontend Engineer

✉️ wlgns6921@gmail.com

📞 010-5664-5041

🌐 Github

💻 개발 블로그

🔗 포트폴리오 사이트

About Me

“문제를 해결하고 개선하는 것에 큰 성취감과 즐거움을 느낍니다”

기획 단계부터 개발, 배포, 운영까지 서비스를 완성해본 경험을 갖춘 프론트엔드 개발자입니다.

실제 사용자 환경을 고려하여 성능과 UI/UX를 개선하고 문제를 해결하는 과정에 가치를 두며,

협업과 지속적인 학습을 통해 더 나은 제품을 만드는 것을 목표로 성장하고 있습니다.

2025.02 Education 성결대학교 정보통신공학과 학사 졸업 2019.03 ~ 2025.02

학점 3.71 / 4.5

Awards

프로그래밍 설계 경진대회 우수

2020.06

웹서버 프로그래밍 교내 경진대회 최우수

2023.06

2025.03 Project 1 펫토리(PETORY) - 반려동물 일정 및 공유 돌봄 웹 2025.03 ~ 2025.12



펫토리(PETORY) - 반려동물 일정 및 공유 돌봄 웹

친구/가족/돌보미와 함께 반려동물의 일정을 공유 및 관리할 수 있는 웹서비스

개발 환경 | 2025.03 - 2025.12 / 2명 (BE 1, FE 1)

기술 스택 | React, JavaScript, CSS(Responsive Design)

개발 내용 | ① React Query 도입으로 중복 API 호출 및 재방문 지연 문제 해결

- 화면 전환, 재방문 시 동일 API가 반복 호출되는 문제 발생
- 캐싱/무효화로 중복 API 호출 70~80% 감소
- 재방문 로딩 시간 35~40ms 단축 및 서버 상태 관리 흐름 개선

② Axios 인터셉터 기반 전역 에러 및 인증 처리 구조 개선

- 네트워크/서버 에러가 각 페이지에 분산 처리되어 예외 로직 중복 발생
- 인터셉터로 에러 처리를 단일화하여 유지보수 통합
- 인증만료 감지 시 보호 라우팅 및 로그인 리다이렉션 적용으로 접근 제어 안정화

③ 실서비스 환경에서 발생한 UI/UX 이슈 해결 및 사용성 개선

- 모바일 브라우저 주소창 차이로 발생한 레이아웃 깨짐 문제 해결
- 브라우저/기기별 뷰포트 영역을 실시간 측정하는 로직을 추가하여 구조적으로 해결
- 반응형 레이아웃 및 크로스 브라우징 대응으로 기기·플랫폼 간 UI 일관성 확보
- 멀티 스텝 모달 품 UX 개선 및 전역 테마 관리로 사용성 및 일관성 향상

④ 배포 구조 개선 및 운영 환경 재설계

- EC2 프리티어 환경에서 리소스 한계로 성능 저하 발생
- 백엔드 서버 분리(미니 PC 구매) 및 Vercel 이전으로 배포 및 운영 구조 개선

⑤ 컴포넌트 구조 개선으로 확장 시 유지보수 비용 감소

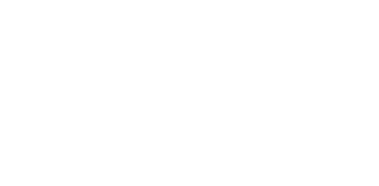
- 페이지/레이아웃/공통 컴포넌트 분리가 미흡해 변경 영향 범위 확대
- UI 컴포넌트와 서비스 로직 분리로 확장성과 유지보수성 개선

Github | <https://github.com/wlgns5041/Petroy-FrontEnd>

배포 URL | <https://www.petory.site>

PDF | [펫토리 소개](#)

2025.12 Project 2 개인 포트폴리오 웹사이트 2025.12 ~ 2026.01



개인 포트폴리오 웹사이트

나 자신과 프로젝트 경험을 소개하고 개발 역량을 보여주는 웹 기반 포트폴리오

개발 환경 | 2025.12 - 2026.01 / 1명 (개인 프로젝트)

기술 스택 | React, TypeScript, Tailwind CSS

개발 내용 | ① 초기 로딩 속도 저하 문제 개선

- 상세 모달·PDF 뷰어가 초기 로딩에 포함되어 초기 화면 로딩 지연 발생

- React.lazy + Suspense 적용으로 사용자 인터랙션 시점에만 로드하도록 개선

- TBT 30ms → 20ms (약 33% 감소)로 초기 로딩 체감 성능 개선

② 인터랙션 중심 화면 렌더링 병목 개선

- 스크롤·모달·화면 전환 로직이 컴포넌트 상태에 집중되어 불필요한 리렌더링 발생

- 애니메이션·스크롤 연동 로직을 외부 상태 기반으로 분리해 렌더 비용 최소화

- 인터랙션이 많은 화면에서도 프레임 드랍 없이 UI 응답성 유지

③ TypeScript 기반 컴포넌트 구조 및 타입 안전성 확보

- 도메인 구조를 타입으로 정의해 데이터 구조 혼선 방지

- props 및 상태를 타입으로 고정하여 기능 확장 시 사이드 이펙트 감소

- 개발 단계에서 타입 오류를 사전 차단하여 버그 발생 가능성 감소

④ 실서비스 운영을 고려한 UI 구조 및 사용자 흐름 안정화

- 모바일·브라우저 환경별 동작 차이를 고려해 레이아웃 기준 정립

- 비동기 처리·로딩·실패 상황에서 대체 UX 흐름을 함께 설계해 사용자 혼란 최소화

- Framer Motion 기반 인터랙션 일관성 유지

Github | <https://github.com/wlgns5041/portfolio>

배포 URL | <https://jihoon.cloud>

Skills

Languages



TypeScript



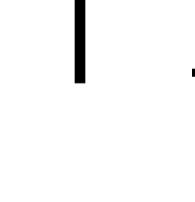
JavaScript

Frameworks



React

Styling & CSS



Tailwind CSS

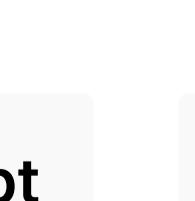


CSS(Responsive Design)

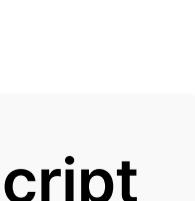
Deployment



Nginx



Vercel



Docker

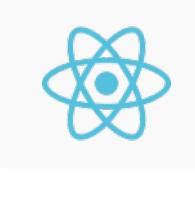


Jenkins



AWS

Tools



Git



Figma



Notion



Slack