

# 박종광

## 프론트엔드 엔지니어

Phone 010-3502-7281  
Email [jgpark7281@gmail.com](mailto:jgpark7281@gmail.com)  
Github <https://github.com/ykhtdt>

### About Me

변화에 빠르게 대응하는 팀 환경에서 **사용자 경험을 중심으로 작업의 비즈니스 가치를 판단**하고, 일정 내에 완성도 높은 서비스를 제공해온 경험이 많습니다.  
서비스의 주인의식을 바탕으로 **팀 모두가 같은 비전과 목표를 공유할 때까지 소통**하는 것을 중요하게 생각합니다.  
따라서, 문제 해결 과정에서 함께 해결책을 찾는 과정을 주도하고 큰 보람을 느끼며 팀이 함께한다면 어떠한 어려움도 극복할 수 있다고 믿습니다.  
최근에는 **불필요한 프로세스를 개선**하고 **개발자 경험(DX)을 향상**시키는 데 관심이 많으며, 이를 통해 서비스 품질과 팀 생산성에 긍정적인 영향을 줄 수 있는 방법을 탐구하고 있습니다.

### Tech Skills

코어	React, Next.js (App Router), Typescript
스타일	TailwindCSS, SCSS
상태관리	Tanstack/React-Query, Zustand, Redux
테스트	Jest, Cypress, React Testing Library
빌드	Webpack, Babel
패키지매니저	PNPM, Yarn
개발 도구	Vercel, Github Actions, Turborepo, Storybook

### Work Experience

#### 두넛페이팩토리

프론트엔드 엔지니어  
2024.10 - 2025.04 (7개월 / 정규직)

- 다양한 도메인에서 실험적인 비즈니스 모델을 빠르게 검증하는 2인으로 구성된 조직
- 프론트엔드 1인 개발자로서 설계부터 개발까지 전담

#### 위즈코어

대리 · 기술연구소 프론트엔드 리드  
2024.07 - 2024.09 (3개월 / 정규직)

- AI-OCR 기반 중대재해처벌법 대응을 위한 솔루션 및 LLM 기반 챗봇의 프론트엔드를 설계 및 개발
- 스프린트 간 발생한 이슈의 해결 과정을 문서화·공유하고, 개발 아티클 공유 및 코드 리뷰 정착을 통해 **팀의 기술 지식 확산과 협업 문화를 적극적으로 개선**, 사내 내부 기술 블로그 도입을 주도
- 런칭 예정인 서비스에서 발생하던 보고된 **주요 이슈의 90% 이상 해결**하여 서비스 품질을 개선하고, **Sentry 기반의 모니터링/알림 체계를 구축**하여 이슈를 빠르게 파악하고 대응하는 프로세스 정착
- Jest와 React Testing Library를 이용해 테스트 코드를 점진적으로 도입하여, **버그 재발 방지와 안정적인 서비스 운영을 도모**했습니다.

플라스크

프론트엔드 리드(~2022.04) → 프론트엔드 엔지니어  
2020.12 - 2023.03 (2년 4개월 / 정규직)

- AI 기반 모션캡처 기술을 활용한 애니메이션 편집 툴(Plask Motion) 및 웹사이트 개발
- 초기 프론트엔드 팀을 리딩하여 MVP를 개발하고, **30억 규모의 Pre-A 투자 유치 및 CES 2023 혁신상 수상**에 기여
- 기술 스택 선정 및 아키텍처 설계를 주도하고, 코드 리뷰와 페어 프로그래밍 문화를 도입하여 팀 협업 효율성과 코드 품질 향상
- 웹사이트 성능 최적화를 통해 Lighthouse 점수를 30점에서 96점으로 개선하여 사용자 경험 개선
- 신규 개발자 온보딩 시스템을 구축하여 조직에 성공적으로 정착할 수 있게함
- 정식 런칭 후 평균 월 8만 명이 사용하는 글로벌 서비스로 성장한 제품의 프론트엔드를 지속 개발

왓비타

프론트엔드 엔지니어 → 프론트엔드 리드(2020.02~)  
2018.12 - 2020.10 (1년 11개월 / 정규직)

- 복용 중인 약, 섭취 중인 영양제, 생활 습관, 걱정되는 부분, 개인 유전체 분석 등을 기반으로하는 건강기능식품 추천, 구매, 관리 서비스의 프론트엔드 개발
- 유전자 검사로 탈모 유전인자를 알고 맞춤형 탈모 관리 세트를 제공하는 이네이트 포헤어 서비스 개발 (**약 2주 만에 500만원 매출 달성**)
- 영양제를 구매하고 1달 중 90% 이상 섭취 체크 시 다음 달 영양제를 무료로 제공하는 왓비타 챌린지 서비스 개발

코스모컨설팅

사원 • 풀스택 개발자  
2018.03 - 2018.09 (7개월 / 정규직)

- 일본 군마 현청에서 경찰이 PDA 통해 업무를 수행할 수 있도록 지원하는 시스템(GP-WAN)의 교통계 방치주차위반 관리 시스템의 프론트엔드 및 백엔드 개발

Work Skills

프론트엔드팀 매니지먼트

- 초기 스타트업 서비스의 프론트엔드 소규모 팀 리딩으로 **MVP 개발부터 정식 런칭, 유지보수까지의 과정**을 경험했습니다.
- **팀원 참여형 업무 분배 및 지식 확산 문화 형성**  
팀원에 대한 관심과 잦은 대화를 통해 **개인의 관심 도메인과 기술적 역량 및 성향을 파악**하고, 팀 회의를 통해 공동으로 업무를 분배하는 방식을 결정하였습니다.  
스프린트간 발생한 이슈 해결 과정과 개발 아티클을 공유하고, **페어 프로그래밍과 코드 리뷰를 통해 개인의 부담을 줄이고 코드 품질을 지속적으로 개선**하며 팀 전체의 성장을 도모했습니다. 이를 통해, 팀원들의 서비스 주인의식과 업무 만족도를 향상시켰습니다.
- **전략적 우선순위화 체계 구축**  
기존에 개발되어있는 레거시 코드 및 버그와 지속 개발해야하는 서비스가 있는 상황에서, **사용자 영향도 > 비즈니스 목표 > 기술부채 해소**로 업무를 분배하는 결정하였습니다.  
비즈니스가 지속되고 성장해야 개발팀과 팀원이 존재할 수 있다고 생각합니다.  
이를 통해, 개발팀과 팀원이 성취감과 안정감을 느끼고 이는 다시 비즈니스 성과로 이어지는 선순환을 이룬다고 생각합니다.
- **비즈니스 임팩트를 고려한 개발 소통**  
비즈니스 이해관계자들은 개발팀의 업무에 대해서 관심이 많습니다. 개발팀의 업무가 사용자 수, 전환율 등 비즈니스에 어떻게 기여하고 있는지, 개발 리소스가 효율적으로 사용되고 있는지 알고싶어합니다.  
이를 위해, 스프린트에서 팀 회의를 통해 단거리 목표와 업무 그리고 담당자를 정하고, 스프린트 회고를 통해 단순히 '코드를 개선하였다'가 아니라, 이 개선과 기능을 통해서 **'이탈률이 n% 줄었고 클릭률이 개선되고, 그로 인해 매출이 n% 증가했다'**와 같이 **비즈니스 기여도를 확인할 수 있도록** 하였습니다.  
이에 대한 수집은 Google Analytics, Hotjar를 사용한 경험이 있습니다.
- **새로 합류하는 팀원의 원활한 적응을 위한 온보딩 프로세스 구축**  
팀원과의 식사와 티타임을 진행하고, 첫 주는 스프린트 참관을 통해 의사결정 과정과 업무 흐름을 관찰할 수 있게 하여 기술 스택과 코드베이스를 이해할 수 있게 하였습니다.  
그리고 페어 프로그래밍과 소규모 기능 개발 등으로 단계적인 책임을 부여하고 지속적인 팀원들과의 미팅을 통해 모니터링했습니다. 이후 온보딩 종료 과정에서 신규 팀원이 온보딩 경험을 팀에 공유하고 프로세스 개선점을 제안하는 시간을 가져 온보딩 프로세스를 지속적으로 보완하였습니다.  
신규 인재의 합류는 기술·문화적 적합성을 모두 고려하여 긴 채용 과정을 거쳐 이루어집니다. 합류 이후의 초기 경험이 조직에 성공적으로 정착할 수 있는 결정적인 영향을 준다고 생각합니다. 체계적인 온보딩 프로세스는 단순히 기술 습득을 넘어 조직 문화 적응과 소속감을 형성하여 신규 팀원의 이탈을 방지하고 빠르게 퍼포먼스로 이어질 수 있도록 합니다.

## DX(Developer Experience) 개선

- 개발자의 생산성과 만족도를 높이는 개발 경험을 지속적으로 고민하고, 개선하기 위해 의견을 제시합니다.
- 사내 에러 모니터링 및 대응 시스템 구축**  
개발자가 파악하지 못한 사일런트 에러가 발생하여 잦은 CS 피드백이 발생했습니다.  
이는 서비스 품질에 영향을 끼치는 것 뿐아니라, 개발자 스스로가 문제를 찾지 못했다는 상황에서 스스로에 대한 불신/불안감이 증가하고  
잦은 긴급 대응으로 업무 흐름을 중단시켜 능률이 저하시킵니다.  
이를 위해, **Sentry를 이용하여 사용자 경험과 관련된 것을 중심으로 임계값과 중요도 레벨을 정의**하였습니다. 또한, 서로 다른 Slack 채널별로 알림을 수신하였습니다.  
서비스를 지속적으로 모니터링할 수 있게 하고 다음 업무 계획으로 이어질 수 있게 하였습니다.  
이를 통해, 개발자가 견고한 서비스를 만들 수 있다는 자신감을 주었으며, 서비스의 품질 개선에도 영향을 주었습니다.
- Git Branch 전략 수립 및 개선**  
애자일 스프린트를 운영하는 과정에서 Git Flow 전략의 한계점을 체감했습니다.  
복잡한 브랜치 규칙과 대규모 일괄 배포 방식으로 인해 코드 충돌 해결이 어렵고, 대량의 변경 사항이 한 번에 리뷰되면서 리뷰 부담과 피로도가 증가했습니다.  
이는 결과적으로 개발 지연과 협업 효율 저하로 이어졌습니다. 빠르고 빈번한 배포 주기를 지향하는 스프린트 방식과 Git Flow는 궁합이 맞지 않는다고 판단하여,  
**더 간결하고 통합 주기가 짧은 Trunk Based Development로 전환**하였으며, 이후에도 팀의 개발 문화와의 적합성을 지속적으로 점검했습니다.

## UI 컴포넌트 직접 개발과 라이브러리 활용 간 균형점 도출

- 프로젝트별 상이한 디자인과 개발자별 개별 구현으로 인해 호환성과 접근성 문제, 재사용성 저하가 발생하여 디자인 시스템 도입을 제안했습니다.  
**PO 및 디자인팀과 협업**해 공통 스타일과 접근성 기준을 수립하고, 이를 기반으로 사내 UI 컴포넌트 라이브러리를 구축했습니다.  
컴포넌트는 필요한 경우 직접 개발하고, 상황에 따라 외부 라이브러리를 활용하여 커스터밍하는 방식을 병행하며 **개발 효율성과 유지보수 용이성 사이에서 균형을 유지**했습니다.  
또한 Storybook을 도입해 **문서화와 시각적 테스트 환경을 구성하여 협업 효율과 유지보수성을 개선**했습니다.

## Work Projects

### 실험적 비즈니스 모델 검증을 위한 MVP 프론트엔드 개발

#### 두넛페이백토리

- 다양한 비즈니스 모델을 빠르게 개발하고 시장 반응에 따라 지속/중단을 결정해야 하는 환경에서 반복적인 프로젝트 세팅으로 인한 리소스가 낭비되었습니다.  
이를 해결하기 위해, **PNPM Workspace와 Turborepo로 템플릿을 구축**하여 신규 프로젝트의 초기 구성을 통일하고 세팅해야하는 반복 작업을 제거하였습니다.  
또한, **Feature-Sliced Design** 아키텍처를 적용하여 기술적 관심사가 아닌 기능 중심으로 코드를 구조화함으로써, 매번 컴포넌트 위치를 고민해야 했던 문제를 해결하고  
코드의 명확성과 유지보수성을 향상시켰습니다.
- 뉴스레터 플랫폼에서 하나의 코드베이스로 여러 서브도메인을 통해 개별 브랜드 페이지를 제공해야 하는 요구사항이 있었습니다.  
Next.js 미들웨어를 활용하여 호스트 헤더를 통해 서브도메인을 파악하고,  
내부 지정된 경로로 **Redirect가 아닌 Rewrite를 적용하여 SEO 친화적인 서브도메인 라우팅 구현**으로 해결했습니다.  
이를 통해 하나의 Vercel 배포로 여러 서브도메인을 관리할 수 있게 되어, 유지보수성과 배포 프로세스를 단순화했습니다.
- 기술 스택** : React 19, Next.js 15, Typescript, TailwindCSS, Turborepo, shadcn/ui, Tanstack/React Query, Zustand, Supabase, Vercel, PNPM

### LLM 기반 실시간 AI 챗봇 시스템 개발

#### 위즈코어

- 안전 문서 관리 플랫폼 내 문서 데이터에 대한 질의응답이 가능한 AI 챗봇을 개발했습니다.  
POST 요청과 응답 스트리밍을 활용하여 **토큰 단위 실시간 응답 스트리밍**을 구현하고, LangChain과 연동된 백엔드 RAG 파이프라인과의  
**효율적 통신을 위한 실시간 스트리밍 데이터 구조를 설계**했습니다.
- 텍스트와 JSON이 혼합된 스트리밍 응답 데이터를 실시간으로 분리 및 파싱** 처리하는 로직을 개발했습니다.  
또한 retry-after 기반 과도한 요청에 대한 제한 전략을 수립하고 재시도 타이머 UI를 개발하여 서버 부하를 관리했습니다.
- 사용자 경험 향상을 위해 AI 응답의 타이핑 애니메이션, 스크롤 자동 조정 등 인터랙티브한 채팅 UI를 디자인하고 개발했습니다.
- 기술 스택** : React 18, Next.js 14, Typescript, TailwindCSS, shadcn/ui, Tanstack/React Query, Jotai, Streams API, Sentry, Yarn, Jest, React Testing Library

## AI·WebGL 기반 모션캡처 애니메이션 편집 툴 개발

### 플라스크

- 초기 **CSR 기반 웹앱을 Next.js로 마이그레이션**하고, 사용자 인증 정보를 포함한 초기 데이터를 **서버 사이드 사전 패칭**함으로써 클라이언트 측 데이터 요청 지연으로 발생하는 일시적인 잘못된 UI 상태 노출 문제를 해결했습니다.
- 복잡한 3D 모델, 비디오, 모션 데이터를 **재귀적 트리 구조의 계층형 에셋 시스템**을 설계하여 사용자가 직관적으로 연관 에셋을 관리할 수 있게 했습니다.  
**react-window 기반 트리 뷰 가상화 기법**을 적용하여 수천 개의 Bone, Mesh, TransformNodes 등으로 구성된 3D 에셋의 렌더링 성능을 최적화했습니다.  
또한 Babylon.js를 통해 렌더링된 3D 모델 데이터를 분석하고, 비디오에서 추출한 모션 데이터를 적용하거나 편집하는 모션 제어 시스템을 개발했습니다.
- 구독 모델 도입 시 **RBAC와 ABAC** 기반의 기능 권한 관리를 구현했습니다. 결제 플랜에 따른 기능 접근은 역할 기반(RBAC) 권한 관리로 제어하고, 사용자별 에셋 개수나 크레딧 같은 사용량 제한은 속성 기반(ABAC) 권한 관리를 적용하여 세분화된 접근 제어를 구현했습니다.
- 비디오 타임라인 조작을 위한 **단계적 스크러버 컴포넌트 개발**을 진행했습니다. 초기에는 빠른 출시를 위해 HTML input 태그의 type="range"를 활용하여 **개발 공수와 성능 사이의 균형점**을 찾았습니다. 이후 UX 개선을 위해 썸네일, 롤러, 선택 영역, 커서 등을 포함한 복합적 UI 요소로 구성된 커스텀 컴포넌트로 고도화했습니다.  
마우스 이벤트 핸들링을 통해 드래그 기반 타임라인 조작과 양쪽 경계점 조절이 가능한 크롭 기능을 구현했습니다.  
각 단계에서 성능 저하 없이 사용자 경험을 개선하는 점진적 접근 방식을 통해 안정적인 기능 개발을 달성했습니다.
- 확장자만 변경된 비디오 파일이 서버로 전송되어 처리 실패하는 문제가 발생했습니다.  
이를 해결하기 위해 바이너리 헤더 검사를 구현하여 파일의 첫 16바이트 Magic Number를 읽어 실제 파일 형식을 검증하고, 클라이언트에서 즉시 피드백을 제공하여 문제를 조기에 감지하도록 했습니다.  
이를 통해 **불필요한 업로드 트래픽을 차단**하고 **백엔드 로깅 및 예외 처리 부담을 크게 감소**시켰습니다.
- 기술 스택** : React 18, Next.js 12, Typescript, Babylon.js, SCSS, Redux, Redux-Saga, Storybook, Sentry, Vercel, Yarn, Webpack, Babel, react-window

## 모션캡처 애니메이션 편집 툴 리팩토링

### 플라스크

- 빠른 시장 출시 압박 속에서 개발된 MVP/베타 단계의 코드베이스를 **리팩토링하여 기술 부채를 해소**하고 **안정성을 확보**했습니다.  
Redux-Saga의 방대한 코드량과 러닝커브, 유지보수 중단으로 인한 문제를 해결하기 위해 Recoil 기반의 상태 관리 시스템으로 전환했습니다.
- 애니메이션 편집 기능에서 발생한 잦은 API 호출 문제를 해결하기 위해 RESTful API에서 **WebSocket 기반 실시간 통신으로 전환**하여 **사용자 액션에 대한 UI 반영 지연 문제를 개선**했습니다. 이 과정에서 **React Query**의 데이터 캐싱 기능을 활용해 **중복 API 요청을 감소**시켰습니다.
- MSW를 도입하여 **API 의존도 없는 독립적인 개발 환경**을 구축했습니다. 또한 **Husky와 Lint-staged**를 도입하여 커밋 단계에서 코드 스타일을 사전 검증함으로써, 코드 리뷰에서 스타일 문제에 대한 피드백을 최소화하고 핵심 로직에 집중할 수 있는 협업 환경을 조성했습니다.
- 기술 스택** : React 18, Next.js 12, Typescript, SCSS, Recoil, React Query 3, Socket.IO-client, MSW, Cypress, Storybook, Husky, Lint-staged, Sentry, Vercel, Yarn

## 이커머스 웹 서비스 홈페이지 개발

### 왓비타, 플라스크

- 쿠키 기반 인증에 HttpOnly, Secure 플래그와 Domain 속성을 적용하여 서브도메인 간 인증 상태를 공유했습니다.  
이로 인한 CSRF 위험은 추가 토큰 검증으로 방지했습니다.
- Framer Motion의 기본 애니메이션 기능만을 활용하여 제품 사용 사례와 같은 핵심 콘텐츠를 강조했습니다.  
복잡한 애니메이션보다는 스크롤 연동 효과 등으로 개발 비용 대비 주요 정보에 대한 시각적 주목도에 중점을 두었습니다.
- 성능 개선 및 SEO 최적화 작업을 주도**하여 Core Web Vitals를 개선하여, 기존에 개발된 웹사이트의 **Lighthouse 평균 점수를 30점에서 96점으로 향상**시켰습니다.  
Speed Index는 11.3초에서 2.8초로, TTI는 21.1초에서 4.6초로 대폭 개선하였습니다.  
한글 폰트의 고용량 문제를 해결하기 위해 서브셋 폰트를 적용하고 유사한 폴백 폰트를 적용하여, 한글 폰트 로딩 이전까지의 디자인 파손을 최소화했습니다.  
이를 통해 **이탈률을 30% 감소**와 **전환율 증가에 기여**했습니다.
- Redux-Saga의 방대한 코드량과 러닝커브로 인한 문제를 해결하기 위해 Redux Toolkit 기반의 상태 관리 시스템으로 전환하고, Chart.js를 활용하여 유저의 플랜에 의한 사용량 등 여러 데이터를 시각화했습니다.
- Formik과 Yup 또는 React Hook Form과 Zod를 사용하여 폼 데이터 유효성 검사 및 제출 과정을 통일하여 코드 유지보수성을 개선했습니다.
- Webpack 번들 사이즈 분석을 통해 ESM을 지원하지 않는 Lodash를 Webpack과 Babel 플러그인을 활용해 트리쉐이킹 문제를 해결하고, Moment.js는 Day.js로 대체하는 등 여러 문제에 대응하여 **번들 사이즈를 30% 이상 감소**시켰습니다.
- 기술 스택** : React 16, Next.js 8~12, Typescript, SCSS, Redux, Redux-Saga, Redux Toolkit, Framer Motion, Chart.js, Turborepo, Storybook, Vercel, Yarn

섭취 영양제/복용 중인 약 등을 기반으로 한 건강기능식품 추천 서비스 개발

왓비타

- Redux 기반의 Flux 아키텍처를 적용하여 상태 관리 시스템을 구축했습니다.  
Redux-saga로 비동기 흐름을 관리하고, **immer.js**를 도입하여 불변성을 유지하면서도 코드 가독성을 개선했습니다.
- 구버전 프레임워크가 CSS Modules를 지원하지 않아 동일한 클래스명을 사용할 수 없어서 의도하지 않은 스타일 오버라이드가 발생했고, CSS 클래스 중복을 피하기 위해 네이밍을 고민해야하는 생산성이 저하되는 문제가 있었습니다.  
이를 위해, CSS 클래스의 지역을 컴포넌트 단위로 제한하여 충돌을 방지하기 위해 Webpack에 css-loader를 이용하여 빌드시 랜덤한 Base64로 인코딩한 5자리 해시를 적용했습니다.
- 인터랙티브한 채팅 UI를 통해 질문을 받는 페이지를 개발했으며, Chart.js를 활용하여 다양한 차트를 구현하여 여러 데이터를 시각화했습니다.  
이를 통해, **사용자별 영양 분석 결과 및 추천 성분 등을 직관적으로 전달**했습니다.
- JavaScript에서 TypeScript로의 점진적인 마이그레이션을 수행했습니다.  
**타입 안정성을 확보하고 런타임 에러를 줄이고, API 응답 데이터에 대한 타입 정의와 네임스페이스를 활용한 계층적 타입 구조화**로 개발 생산성과 코드 품질을 향상시켰습니다.
- 기술 스택 : React 16, Next.js 8~9, Typescript, SCSS, Redux, Redux-Saga, Chart.js, Yarn, Jest, Enzyme

일본 군마 현청 경찰 업무 시스템 개발

코스모컨설팅

- 클래스 컴포넌트의 라이프사이클을 이용하여 컴포넌트 상태 관리 및 데이터 흐름을 통일했으며, RESTful API 기반의 데이터 조회 및 수정 기능을 개발했습니다.
- 백엔드는 MVC 구조로 구현했습니다. Eclipse 환경에서 **JUnit** 기반의 단위 테스트를 작성하여 핵심 로직의 안정성을 확보하고, **코드 커버리지를 측정하여 테스트 품질을 관리**했습니다.
- 신입 개발자로서 기획자와 팀 간의 협업 방식과 코드 리뷰 등 프로젝트 진행 방식에 대한 경험을 쌓았습니다.  
이를 통해 **실무 프로젝트의 전체 생명주기와 협업 프로세스에 대한 이해**를 높였습니다.
- 기술 스택 : React, Java, Spring Boot 2.0, MySQL

Skills

Javascript/React

- React, Next.js 기반의 프론트엔드 개발 경험을 해왔습니다.
- 함수 컴포넌트와 React Hooks에 익숙하고, 과거에는 클래스 컴포넌트를 사용했던 경험이 있습니다.
- 상태 관리는 기존에 Redux를 사용했으나, 최근에는 **코드량을 줄이기 위해** Zustand를 사용하고 있습니다.
- **데이터 캐싱, Revalidate**를 위해 Tanstack/React Query를 도입하여 사용하고 있습니다.
- **React Suspense와 Error Boundary**를 활용해 사용자 경험과 안정성을 높이고 있습니다.
- Next.js를 이용해 **SSR, SSG, ISR, CSR 등 다양한 렌더링 방식**을 프로젝트 요구에 맞게 적용했습니다.
- 선언형 코드를 작성해 가독성이 뛰어나고, **응집도는 높으며 결합도는 낮은 모듈화**를 항상 고민합니다.

Typescript

- Javascript에서 Typescript로의 점진적인 마이그레이션을 한 경험이 있습니다.
- Type-Safe한 개발에 관심이 많아 **제네릭, 유틸리티 타입 등 Typescript의 고급 타입 기능**을 활용해 타입 추론과 타입 좁히기를 적극적으로 수행합니다.

HTML/CSS

- 의미 있는 마크업 작성을 위해 **시맨틱 태그**를 적극적으로 활용합니다.
- 스타일링은 SCSS와 Tailwind CSS를 주로 사용합니다.
- **SEO 최적화 경험**이 많으며, Sitemap.xml과 Robots.txt 작성, Google Search Console 및 Google Analytics 연동을 통해 마케터 협업한 경험이 있습니다.

성능 최적화

- 프론트엔드 성능 개선 경험이 많습니다. **렌더링 속도 향상, 불필요한 리렌더링 최소화, 번들 사이즈 경량화, 네트워크 비용 최적화**의 경험이 있습니다.

도구 및 테스트

- 팬텀 디펜던시 문제, Yarn Berry 대비 초기 설정 간편함을 고려하여 패키지 매니저로 **PNPM**을 선호합니다.
- **Jest**와 **React Testing Library**를 이용해 테스트 코드를 작성한 경험이 있고, Babel과 Typescript 초기 환경 설정의 복잡성을 이유로 최근에는 Vitest 학습 및 도입을 진행하고 있습니다.
- **Cypress**로 E2E 테스트 경험이 있으며, 테스트 코드의 속도와 Webkit 브라우저 지원 필요성으로 Playwright를 학습하고 있습니다.
- 반복적이고 불필요한 작업을 줄이기 위해 **Github Actions**를 활용한 **지속적 통합(CI) 자동화**를 적극적으로 도입하고 있습니다.
- **Storybook**을 활용해 컴포넌트 개발과 시각화를 진행한 경험이 있습니다.

협업 및 개발 문화

- 같은 비전을 공유하는 의사소통과 협업을 중요시하며, 직위나 부서 구분 없이 하나의 팀으로서 함께 일하는 문화를 지향합니다.
- **애자일과 스퀘드 조직**에서 PO, 디자이너, 백엔드, 마케터 등 다양한 직군과 원활하게 협업한 경험이 있습니다.
- 최신 기술이 항상 최선은 아니라고 생각하며, 기술이 나오게 된 배경과 적절한 적용 상황에 깊은 관심을 가지고 학습합니다.
- 조직 및 팀의 업무 프로세스와 개발 문화를 **적극적으로 개선**하고, 팀원들의 성장과 협업 효율성을 높이기 위해 협력한 경험이 많습니다.

Educations

신구대학교

- 2012.03 - 2018.02  
모바일IT 전공 전문학사  
학점평균 4.135/4.5 졸업

백산정보

- 2017.07 - 2018.02  
일본 IT 기업 코스모컨설팅 협력 교육 회사에서 프로그래밍 기초 및 일본어 교육 이수  
실무 프로젝트 진행 방식과 기획부터 개발의 협업 과정을 학습  
4인 구성 팀으로 3회의 팀 프로젝트를 경험

기타 정보

- 생년월일 : 1994년 02월 25일
- 병역사항 : 대한민국 육군 제21보병사단 병장 만기 전역(2014.02.04 - 2015.11.03)