

About Me

React, Next.js, Typescript 기반의 6년차 프론트엔드 엔지니어입니다.

초기 스타트업 서비스의 프론트엔드 소규모 팀 리딩 및 MVP 개발부터 정식 런칭, 그리고 유지보수까지 모든 과정을 성공적으로 이끌어온 경험이 있습니다.

스타트업 환경에서는 비즈니스 우선순위와 일정에 맞춰 빠르게 개발하면서도, 향후 **확장과 개선이 가능한 코드 구조를 설계**합니다.

빠르게 구현한 기능을 개선하는 것을 팀의 학습 기회이자 제 개선 과제로 활용하여 지속적인 성장과 코드 품질 향상을 동시에 이끌어 왔습니다.

이를 통해, 타이트한 일정을 스트레스로 느끼기보다는 효율적인 해결책을 찾아내는 도전이자 성장의 기회로 받아들이며 즐길 수 있습니다.

이 경험을 토대로 **프론트엔드 팀 매니징, 팀원의 원활한 적응과 성장을 주 업무로 한 경험**이 있습니다.

Work Experiences

두낫페이팩토리

2024.10 ~ 2025.04 (7개월)
프론트엔드 엔지니어

- 실험적 비즈니스 모델 검증을 위한 MVP 및 홈페이지 개발

위즈코어

2024.07 ~ 2024.09 (3개월)
AI Ground 프론트엔드 리드

- 중대재해처벌법 관련 AI-OCR기반 안전 문서 관리 플랫폼 개발

플라스크

2020.12 ~ 2023.03 (2년 3개월)
프론트엔드 리드(~2022.04) → 프론트엔드 엔지니어

- AI기반 영상 모션캡처를 통해 캐릭터에 애니메이션을 입히는 편집 툴(Plask Motion) 및 홈페이지 개발
- MVP/베타버전 개발(**30억 규모 Pre A 투자 유치, CES 2023 혁신상**), 정식 버전 런칭 및 개발, 유지보수
- 베타버전 런칭 이후 **약 평균 8만 명의 월간 활성 사용자 달성**

왓비타

2018.12 ~ 2020.10 (1년 11개월)

프론트엔드 엔지니어 → 프론트엔드 리드(2020.02~)

- 복용 중인 약, 섭취 중인 영양제, 생활 습관, 걱정되는 부분, 개인 유전체 분석 등을 기반으로하는 건강기능식품 추천, 구매, 관리 서비스 및 홈페이지 개발
- 이네이트 포헤어 서비스 개발 및 런칭(약 2주 만에 500만원 매출 달성)

코스모컨설팅

2018.03 ~ 2018.09 (7개월)

프론트엔드·백엔드 엔지니어

- 일본 군마 현청에서 경찰이 PDA 통해 업무를 수행할 수 있도록 지원하는 시스템인 GP-WAN의 교통계 방치주차위반 관리 기능 개발

Skills

- React, Next.js, Typescript
- Tailwind CSS, SCSS
- Tanstack Query, Zustand, Jotai, Redux(Redux Toolkit, Redux-Saga)
- shadcn/ui, Storybook
- Webpack, Babel
- Jest, Cypress, Sentry, Vercel, PNPM, Yarn

Competency

프론트엔드팀 매니지먼트

- 기존에 개발되어있는 레거시 코드와 치명적인 버그, 그리고 현재 개발해야하는 기능을 동시에 진행하는 환경을 경험
- '사용자 영향도 > 비즈니스 목표 > 기술부채 해소'로 업무 우선순위 결정하여 진행한 경험
- 각 스프린트에서 신규 기능 개발과 관련된 레거시 코드를 리팩토링을 일부 병행하는 전략 수립
- 팀원의 관심 도메인과 기술적 역량을 고려한 업무 분배로 팀원의 의욕 향상과 이를 토대로한 생산성 개선
- 코드 스타일 규칙 정의, 코드 리뷰 도입 및 상호 피드백으로 코드 퀄리티를 개선하고 팀원 서로의 성장을 도모
- 신규 합류하는 팀원의 원활한 적응을 위한 온보딩 프로세스 구축

DX(Developer Experience) 개선

- 개발자의 생산성과 만족도를 높이는 개발 경험을 지속적으로 고민하고 개선
- 문제 해결 사례 1 - 사내 에러 모니터링 및 대응 시스템 구축

- 문제 상황
 - 개발자가 파악하지 못한 사일런트 에러가 발생하여 잦은 CS 피드백 발생
- 해결 방법
 - Sentry를 이용한 로그 수집 및 모니터링, 알림 시스템 구축
 - 서비스 성격에 따라 알림을 받기 원하는 조건 및 중요도 레벨을 정의하여 Slack 채널 별 알림 수신
- 문제 해결 사례 2 - **Git Branch 전략 수립 및 개선**
 - 문제 상황
 - 스프린트 기반의 GitFlow 전략에서 develop에서 일부 기능의 QA와 버그로 인해 main으로 병합되지 못하고 머무르는 코드 발생
 - feature가 모두 develop에 직접 병합되어 일부 문제 기능이 전체 릴리즈를 차단
 - develop에서 문제가 발생한 특정 feature만을 제외하고 릴리즈하거나 롤백하는 어려움 발생
 - 해결 방법
 - 유사 기능 단위로 묶은 story 브랜치를 정의하고, 이를 기반으로 feature 브랜치를 분기하여 개발 후 상호 리뷰를 거쳐 병합
 - story 브랜치에서 통합 QA 진행 후, 검증 완료된 기능만 develop에 병합되어 특정 story에 문제가 발생해도 다른 검증된 story들은 정상적으로 릴리즈 진행 가능
 - 이를 통해 한 기능의 문제가 전체 릴리즈 일정을 지연시키는 현상 제거
 - 팀 회고를 통해 story 간 의존성이 높은 기능들에 대한 사전 계획 단계를 프로세스에 추가하는 등 지속적인 관찰과 개선을 진행
- 문제 해결 사례 3 - **CSS 클래스 중복 방지**
 - 문제 상황
 - 구버전 프레임워크가 지원하지 않는 CSS Modules로 인해 동일한 클래스명을 사용할 수 없어 의도하지 않은 스타일 오버라이드가 발생
 - CSS 클래스 중복을 피하기 위해 네이밍을 고민해야하는 상황으로 생산성 저하
 - 해결 방법
 - CSS 클래스의 지역을 컴포넌트 단위로 제한하여 충돌 방지를 위해 css-loader를 이용하여 빌드시 랜덤한 Base64로 인코딩한 5자리 해시를 적용
 - 이를 통해 개발자의 스타일 관리가 쉬워지고, 컴포넌트별로 스타일이 적용

UI 컴포넌트 직접 개발

- 접근성 및 합성 컴포넌트 등을 고려한 컴포넌트 직접 개발
- **컴포넌트 직접 개발 및 라이브러리 활용의 균형점 도출**
- 문제 해결 사례 1
 - 문제 상황
 - 프로젝트별/디자이너별 상이한 디자인 시스템으로 인해 일관된 UI 구현의 어려움 및 스타일 일관성 저하
 - 각 개발자마다 특수하게 필요한 상황에 맞춰서 컴포넌트를 직접 구현하여, 누락된 접근성 및 호환 문제로 재활용할 수 없는 문제
 - 라이브러리 컴포넌트 커스텀 과정에서 예상보다 많은 개발 리소스 소모

- 해결 방법
 - 필요한 핵심 UI 컴포넌트를 바닥부터 직접 설계 및 구현
 - PO/디자인팀과의 협업으로 공통 스타일을 정의하여 디자인 시스템을 기반으로한 컴포넌트 라이브러리 구축
 - Storybook을 활용한 컴포넌트 문서화 및 시각적 테스트 환경 구성으로 디자인팀과 협업 환경 구성
 - 요구사항 충족을 위한 라이브러리 커스텀과 직접 구현에 대해 리소스 및 복잡도를 판단하여 결정
 - 각 레이아웃 패널 리사이즈 기능을 React-Resizable 라이브러리를 사용하여 초기 마우스 이벤트 구현에 대한 시간을 절약
 - 요구사항 구현에 필요한 기능만을 추가 개발하여 적용

Projects

실험적 비즈니스 모델 검증을 위한 MVP 프론트엔드 개발

- 두넛페이팩토리
- 기술 스택: React 19, Next.js 15, Typescript, Tailwind CSS, Turborepo, shadcn/ui, Tanstack Query, Zustand, Supabase, Vercel, PNPM
- 문제 해결 사례 1
 - 문제 상황
 - 다양한 비즈니스 모델을 빠르게 개발하고 시장 반응에 따라 지속/중단을 결정해야 하는 환경에서 반복적인 프로젝트 세팅으로 인한 리소스 낭비
 - 해결 방법
 - 모노레포 템플릿 구축하여 신규 비즈니스 모델마다 프로젝트 초기 구성을 통일시키고 구성 시간을 단축
 - Feature-Sliced Design 아키텍처를 기반으로 기술적 관심사가 아닌 기능 중심으로 코드를 구조화함으로써 컴포넌트 위치 결정에 대한 모호함을 해결
- 문제 해결 사례 2
 - 문제 상황
 - 뉴스레터 플랫폼에서 하나의 코드베이스로 여러 서브도메인을 통해 개별 브랜드 페이지를 제공해야하는 상황
 - 해결 방법
 - Next.js 미들웨어를 활용하여 호스트 헤더를 통해 서브도메인을 파악하여 내부 지정된 경로로 redirect가 아닌 rewrite하여 SEO 친화적으로 해결
 - 하나의 Vercel 배포로 여러 서브도메인을 관리

SSE 기반 실시간 AI 챗봇 시스템 개발

- 위즈코어
- 기술 스택: React 18, Next.js 14, Typescript, Tailwind CSS, shadcn/ui, Tanstack Query, Jotai, SSE(Server-Sent Events), Sentry, Yarn
- 주요 구현 내용

- 안전 문서 관리 플랫폼 내 문서 데이터에 대한 질의응답이 가능한 AI 챗봇 개발
- SSE(Server-Sent Events) 활용한 토큰 단위 실시간 응답 스트리밍 구현
- LangChain과 연동된 백엔드 RAG 파이프라인과의 효율적 통신을 위한 실시간 스트리밍 데이터 구조 설계
- 텍스트와 JSON이 혼합된 스트리밍 응답 데이터를 실시간으로 분리 및 파싱 처리
- retry-after 기반 과도한 요청에 대한 제한 전략 수립 및 재시도 타이머 UI 개발
- 사용자 경험 향상을 위한 AI 응답의 타이핑 애니메이션, 스크롤 자동 조정 등 인터랙티브한 채팅 UI 디자인 및 개발

ML(AI)/WebGL 기반 모션캡처 애니메이션 편집 툴 개발

- 플라스크
- 기술 스택: React 18, Next.js 12, Typescript, SCSS, Redux(Redux-Saga), Storybook, Sentry, Vercel, Yarn
- 주요 구현 내용
 - 초기 CSR 기반 웹앱을 Next.js로 마이그레이션하여 유저 데이터의 서버 사이드 렌더링 구현 및 초기 로딩 성능 개선
 - 비디오 파일 업로드 및 웹캠 녹화 기능 구현과 영상 전처리(파일 시그니처 검증, 메타데이터 추출, 브라우저별 포맷 변환) 개발
 - 3D 모델, 비디오, 모션 데이터의 통합 관리를 위한 계층형 에셋 관리 시스템 개발
 - Babylon.js를 통해 렌더링된 3D 모델 데이터를 분석하고, 비디오에서 추출한 모션 데이터를 적용하거나 편집하는 모션 제어 시스템 개발
 - 정식 런칭 이후 결제 플랜에 따른 권한 제어 시스템 개발
- 문제 해결 사례 1
 - 문제 상황
 - 확장자만 변경된 비디오 파일(예: .txt를 .mp4로 변경)이 서버로 전송되어 처리 실패
 - 잘못된 파일이 서버까지 전송되고, 이를 검증하기 위한 불필요한 자원 및 네트워크 트래픽 발생
 - 해결 방법
 - 바이너리 헤더 검사 구현
 - 파일의 첫 16바이트 시그니처를 읽어 실제 파일 형식을 검증
 - 클라이언트 즉시 피드백 제공
 - 1차적으로 클라이언트에서 파일 문제를 감지하고 오류 메시지를 표시함으로써, 불필요한 업로드 트래픽을 차단하여 백엔드 로딩 및 예외 처리 작업 부하를 감소

모션캡처 애니메이션 편집 툴 리팩토링

- 플라스크
- React 18, Next.js 12, Typescript, SCSS, Recoil, React Query 3, Socket.io-client, MSW, Storybook, Husky, Lint-staged, Sentry, Vercel, Yarn
- 주요 개선 내용
 - 빠른 시장 출시 압박 속에서 개발된 MVP/베타 단계의 코드베이스를 리팩토링하여 기술 부채 해소 및 안정성 확보
 - Redux-Saga의 방대한 코드량 및 러닝커브, 유지보수 중단으로 Recoil과 React Query로 리팩토링하여 데이터 캐싱 등으로 중복 API 요청 감소 등 개선

- 애니메이션 편집 기능의 잦은 API 호출로 인해 RESTful API 기반 방식에서 WebSocket 기반 실시간 통신으로 전환
- MSW 도입으로 API 의존도 없는 독립적인 개발 환경을 구축
- Husky, Lint-staged 도입하여, 커밋 단계에서 코드 품질을 검증하여 코드 리뷰에서 스타일 문제에 피드백을 하지 않게함

이커머스 웹 서비스 홈페이지 개발

- 왓비타, 플라스크
- 기술 스택: React 16, Next.js 8~12, Typescript, SCSS, Redux(Redux-Saga, Redux Toolkit), Framer Motion, Chart.js, Turborepo, Storybook, Vercel, Yarn
- 주요 구현 내용
 - 쿠키를 통한 로그인 구현 및 서브 도메인 간 로그인 상태 공유
 - Framer Motion을 활용한 인터랙티브 UI 구현
 - 유저 제품 사용 사례를 카드 형태로 제작하고, 가로 스크롤 인터랙션 및 세로 스크롤 방향에 따른 가로 스크롤 방향 제어
 - 좌우 분할 레이아웃을 구성하여, 스크롤에 따라 Fade In/Out 적용
 - 성능 개선 및 SEO 최적화
 - CLS, LCP, FCP 등을 개선하여 기존에 개발되어있는 홈페이지의 Lighthouse 평균 점수 30점에서 96점으로 개선
 - 한글 폰트의 고용량 문제를 해결하기 위해 서브셋 폰트를 적용하고, 비슷한 폴백 폰트를 설정하여 한글 폰트 로드 전 까지 비슷한 폰트를 통해 디자인 파손을 최소화
 - Redux-Saga의 방대한 코드량 및 러닝커브로 인해 Redux Toolkit으로 리팩토링하여 불필요한 코드를 줄이고, 유지보수성과 퍼포먼스를 향상
 - Chart.js를 활용하여 유저의 플랜에 의한 사용량 등 여러 데이터를 시각화
 - Formik과 Yup을 이용하여 폼 데이터 유효성 검사 및 제출의 과정을 통일, 이후 React Hook Form과 Zod로 리팩토링
 - Webpack 번들 사이즈 분석을 통해 ESM을 지원하지 않는 Lodash, Moment.js 등 여러 문제를 발견하여 번들 사이즈를 줄이고, 네트워크 트래픽 요금 절감

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

- 왓비타
- 기술 스택: React 16, Next.js 8~9, Typescript, SCSS, Redux(Redux-Saga), Chart.js, Yarn
- 주요 구현 내용
 - Redux 기반의 Action, Reducer, Store를 분리하여 Flux 아키텍처 패턴 기반의 구조 적용
 - Redux-saga의 비동기 흐름을 관리하고, Redux-Logger를 통해 상태 변화를 추적
 - Redux의 상태 불변성으로 인한 깊은 복사 문제를 해결하기 위해 immer.js를 이용하여 불변성을 유지하되 코드 가독성을 높임
 - SCSS를 이용하여 변수, 믹스인, 상속 등을 통해 재사용가능한 스타일을 정의
 - 인터랙티브한 채팅 UI를 통해 설문을 받는 페이지 개발
 - Chart.js를 이용하여 Gauge, Doughnut, Bar, Line, Pie 등 다양한 차트 구현하여 설문 결과에 따른 부족/과다 성분, 추천 성분 등 여러 데이터를 시각화
 - JavaScript에서 TypeScript로의 마이그레이션

- 기존 PropTypes를 TypeScript 인터페이스로 마이그레이션하여 타입 안정성 확보
- API 응답 데이터에 대한 타입 정의로 데이터 핸들링 오류 감소
- 타입스크립트 namespace를 이용하여 관련 타입을 계층 구조화하고 같은 타입 이름을 다른 컨텍스트에서 사용할 수 있게 함

일본 군마 현청 경찰 업무 시스템 개발

- 코스모컨설팅
- 기술 스택: React, Java, Spring Boot 2.0, MySQL
- 주요 내용
 - React Class Component의 라이프사이클을 이용하여 컴포넌트 상태 관리 및 데이터 흐름을 통일
 - RESTful API 기반의 데이터 조회 및 수정 기능 개발
 - MVC 구조 기반의 Controller, Service, Mapper 패턴을 이용하여 백엔드 개발
 - Eclipse 환경에서 JUnit 기반의 단위 테스트를 작성하고, Coverage를 통해 핵심 로직의 안정성을 향상
 - 신입 개발자로서 기획자와 팀과 협업 및 코드리뷰 등 프로젝트의 진행 방식을 배움

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