최우석



화면 앞의 사용자를 위한 경험을, 옆자리의 동료와 함께 만드는 개발자입니다.

전화번호 010-2426-9068 **GitHub** github.com/chldntjr1321

이메일 chldntjr6113@gmail.com **포트폴리오** choiwooseok.vercel.app

자기소개

화면을 마주할 사람의 입장에서 생각하며, 성능과 접근성처럼 사용자 경험에 영향을 주는 부분에 중점을 두고 개발합니다.
장애인과 보호자를 위한 서비스를 개발하며 같은 화면이라도 이용하는 방식은 환경마다, 사람마다 다르다는 것을 경험했습니다.
누가 어떤 환경에서든 자연스럽게 이용할 수 있는 화면을 만들고자 합니다.

팀 프로젝트에서는 개인의 역량만큼이나 팀이 같은 방향과 속도로 나아가는 것이 중요하다고 생각합니다.
팀이 정한 목표를 가장 우선시하며, 작업에 앞서 충분한 소통으로 방향을 명확히 하고, 작업 후에는 직접 점검한 뒤 팀원과 공유해 제가 놓친 부분을 보완합니다. 이러한 과정을 통해 팀원의 관점과 팀의 작업 방식을 배우며 성장합니다.

AI를 활용하면서 작업 속도가 빨라진 만큼, 코드를 책임지는 역할은 더 중요해졌다고 생각합니다.
AI가 생성한 코드라도 최종적으로 책임지는 것은 사람이기에, 사전에 정한 규칙과 사용 맥락에 맞는지 직접 읽고 검증합니다.
코드 작성 외에 반복되는 작업은 AI로 자동화하고, 그렇게 확보한 시간을 검증에 더 투자하고 있습니다.

프로젝트

2026.03 - 진행 중

URL
GitHub
배포 URL

머먹지 (Mumukji) | 디자인 시스템

팀 구성: 팀 프로젝트 | 디자이너 2인, 프론트엔드 3인

사용 기술: React, TypeScript, Rollup, Turborepo, Storybook, GitHub Actions

주요 활동

1. **Tokens 패키지 설계 및 구현**
   - semantic 토큰을 하드코딩하지 않고 primitive 토큰을 참조하도록 primitive/semantic 2계층 토큰 구조로 설계
   - semantic 토큰의 primitive 참조를 CSS 변수로 변환하는 로직 구현
   - Figma에서 export한 JSON을 파싱해 CSS 커스텀 프로퍼티와 TS 상수를 동시 생성하는 빌드 스크립트 작성
   - Storybook 문서 작성 및 README를 통해 tokens 패키지 사용 가이드 정리
2. **모노레포 구조 도입**
   - 역할에 따라 icons, tokens, ui로 분리한 패키지들을 일관되게 관리하기 위해 pnpm workspace와 Turborepo를 활용
   - 패키지들 간 코드 스타일과 커밋 컨벤션을 통일하기 위해 ESLint, Stylelint, commitlint 등 코드 품질 도구를 루트 단일 설정으로 공통 적용 및 관리
3. **버전 관리 및 배포**
   - Changesets로 tokens 패키지에 변경 사항이 생길 때마다 버전 업데이트 및 CHANGELOG 이력을 기록하고, GitHub Actions 파이프라인으로 현재 v1.0.0까지 총 4회 npm 배포 진행

2026.06 - 2026.07
(3주)

URL
GitHub
배포 URL

DigTheCrate | LP 아카이브 서비스

팀 구성: 개인 프로젝트

사용 기술: Vite, React, TypeScript, Tailwind CSS, TanStack Query, Supabase

주요활동

1. **메인 스레드 병목 개선**
   - 앨범 커버 이미지에서 대표색을 추출하여 앨범 상세 모달 배경에 그라데이션으로 적용하는 과정에서 메인 스레드를 점유하고 있던 색상 계산 연산을 **Web Worker**로 위임
   - Web Worker 위임으로 메인 스레드 스크립트 점유율 **85.8% → 5.7%**로 개선
2. **애니메이션 렌더링 최적화**
   - LP 플레이어의 회전 모션을 JS setInterval로 회전각 state를 갱신하는 방식과 CSS animation 두 가지로 각각 구현해 성능 비교 측정
   - CPU 4x 스로틀링 환경에서 **14%**(JS) **→ 3.2%**(CSS animation) 로 메인 스레드 점유율 감소
   - JS 방식은 setInterval이 실행될 때마다 state 변경에 의한 리렌더링이 일어나는 반면, CSS animation의 transform 속성은 GPU에서 처리되어 메인 스레드와는 독립적으로 동작한다는 것을 확인
3. **모달 키보드 접근성**
   - 마우스 없이 키보드만으로 이용하는 경우를 고려해, 모달에서 Esc 키로 닫기, Tab/Shift+Tab 포커스 순환, 닫힐 때 이전 포커스 복원을 처리하는 커스텀 훅을 구현
4. **게스트 계정 구현**
   - 회원가입 없이 체험할 수 있는 게스트 로그인을 서버리스 함수로 구현
   - 로그아웃 시 게스트 컬렉션을 정적 데모 컬렉션으로 초기화

2026.04 - 2026.05
(3주)

URL
GitHub
배포 URL

마주봄(matchzoom) | 장애인·보호자 대상 AI 직종 탐색 서비스

팀 구성: 팀 프로젝트 | 프론트엔드 4인

사용 기술: Next.js, TypeScript, Tailwind CSS, TanStack Query, Zod, React Hook Form, Vitest

주요활동

1. **대량 리스트 렌더링 최적화**
   - 많은 양의 공고를 한 번에 불러와 렌더링할 때 발생하는 성능 저하 문제를 화면에 보이는 만큼만 DOM에 유지하는 **react-virtual 가상화**로 최적화
   - 공고 리스트 영역 DOM 노드 **34,000개 → 453개**로 감소 (공고 1,000개 기준)
2. **채용공고 필터 성능 개선**
   - 클라이언트에서 전체 공고를 받아 필터링하던 방식을 서버에서 처리하도록 전환하고, 공고 목록을 무한 스크롤로 구현
   - 사용자별 랭킹 결과를 서버에 5분간 캐싱해 필터 요청마다 반복되던 조회·계산을 줄이고, 필터 상태를 클릭 즉시 화면에 반영
3. **SSR 데이터 동기화 구조 구축**
   - 대시보드 진입 시 서버에서 이미 채용공고 목록을 조회했음에도, 화면이 그려진 뒤 클라이언트가 같은 목록을 다시 요청하던 문제 확인
   - 서버에서 조회한 첫 페이지 목록을 클라이언트 캐시에 미리 채워 전달해 중복 요청 없이 해결
   - 이후 스크롤을 통해 추가되는 공고는 같은 캐시에 이어서 쌓이도록 구조 설계
4. **접근성 개선**
   - 장애인 사용자를 위한 서비스로서 **WCAG 2.1 AA** 기준을 목표로 접근성을 고려해 개발
   - 스크롤 잠금과 inert 처리, 포커스 트랩을 적용해 모달 활성화 시 포커스가 배경으로 이탈하지 않도록 제어하여 일관된 키보드 탐색 경험을 제공

5. Claude Code 기반 자동화 워크플로우

- FSD 구조와 CLAUDE.md의 레이어 경계·코드 컨벤션을 기준으로 AI와 협업하고, 생성된 코드가 규칙에 맞는지 직접 검증하며 개발
- ESLint가 레이어 단방향 의존성 위반을 빌드 시점에 자동 차단하는 환경에서 작업
- **로컬 CI 검증 → 이슈 생성 → 커밋 → PR 생성 → 위키 동기화 → CI 모니터링**이 하나의 명령으로 자동 실행되는 워크플로우를 도입

2025.12 - 2026.01
(5주)

URL
GitHub
배포 URL

글로벌 노마드 (Global Nomad) | 체험 예약 서비스

팀 구성: 팀 프로젝트 | 프론트엔드 4인

사용 기술: Next.js, TypeScript, Tailwind CSS, Zustand, TanStack Query

주요 활동

1. BFF 기반 인증 구조 설계

- 백엔드 API가 토큰을 response body로 반환하는 구조로 클라이언트에서 직접 관리 시 **XSS** 위험이 있어, BFF 패턴을 적용하여 Next.js API Route에서 소셜 및 일반 로그인을 처리하고 토큰을 **HttpOnly 쿠키**에 저장
- 서버 컴포넌트에서 인증 API를 직접 처리하고 클라이언트 컴포넌트는 BFF를 통해 인증 요청을 위임하는 구조 설계

2. 마이페이지 예약 내역 구현

- 예약 상태별 필터와 배지, 예약 카드를 구현하고 예약 내역 목록 API를 연동
- 예약 취소와 리뷰 작성 기능 구현, 체험 상세 페이지로 이동 연결
- 리뷰 작성 후 메인 페이지의 리뷰 수·평점이 새로고침 전까지 반영되지 않던 문제를 서버 재조회 대신 **TanStack Query 캐시 직접 갱신**으로 해결

3. 재사용 가능한 컴포넌트 설계

- 페이지마다 동일한 스타일의 제목을 각각 다른 태그로 써야 하는 상황에서, as prop으로 렌더링할 HTML 태그(h2~h6)를 주입받는 **polymorphic 패턴**을 적용하여 시맨틱 마크업을 지키면서 컴포넌트 재사용성 확보

학력

2019.03 - 2025.02
(졸업)

우송대학교 | IT융합학부 스마트IT・보안전공

학점 | 3.97 / 4.5

자격증

2025.09

정보처리기사

한국산업인력공단

교육

2025.07 - 2026.01
(6개월)

코드잇 | 프론트엔드 엔지니어 부트캠프

수료

- TypeScript, Next.js, Zustand, TanStack Query 등을 활용한 실무형 프로젝트 진행
- 코드 품질과 협업 효율 개선을 위한 lint, pre-commit 등의 개발 환경을 주도적으로 도입
- GitHub Projects와 마일스톤을 활용한 프로젝트 전체 일정 및 진행 상황 관리 경험
- 코드 리뷰를 통한 팀 단위 개발 문화 경험