

# 김리안

프론트엔드 개발자

Email

5ffcut@gmail.com

Mobile

+82 10 9001 1250

Website

liankim.kr

GitHub

github.com/lianKim

## 만들고 팔아본 프론트엔드 개발자

React, TypeScript로 커머스와 B2B2C 플랫폼을 만들어왔습니다.

빈티지 조명 브랜드를 직접 운영하며, 고객이 상품을 보고 결제하는 화면과 운영자가 주문·상품을 관리하는 관리자 페이지를 만들고, 배포까지 맡았습니다. 전 과정을 경험해본 덕에 함께 일하는 다른 직무의 관점을 이해하며 소통하게 되었습니다.

이후 패션 산업 플랫폼에서는 성격이 다른 서비스들이 한 코드베이스에 섞이며 생기는 문제를 줄이기 위해, 프론트엔드 구조를 재편하고 팀이 같은 기준으로 개발할 수 있도록 문서화했습니다. 새로운 것을 도입할 때는 그대로 따르기보다, 지금 상황에 맞는지 따져보고 조정해 적용합니다.

## Work Experience

### 유희주식회사 2024.07 - 2025.08 · Frontend Engineer · FE 1인 → 3인

패션 제품 라이프사이클 추적 및 디지털 제품 여권(DPP) 발행 플랫폼

#### 프론트엔드 아키텍처 재구성 ↗

문제

권한·사용자 유형이 다른 독립 도메인들이 단일 CRA 구조에 혼재해, 한 도메인의 변경이 다른 도메인으로 번질 수 있는 상태

해결

- 표준 단일 스코프 FSD(Feature-Sliced Design)로는 도메인 격리가 부족하다고 판단해, 영역·권한 단위로 상위를 나눈 **멀티 스코프 구조 구성**
- 레이어 단방향 의존 원칙을 **ESLint 규칙으로 강제**하고, 아키텍처 위반을 커밋 단계에서 자동 차단
- 반복적인 파일 이동·import 수정은 AI 도구로 자동화하고, 도메인 경계와 의존 방향은 직접 검증·교정
- Atomic Design 기반 공통 컴포넌트를 정리하고 Storybook으로 50+ 컴포넌트 문서화
- 팀이 같은 기준으로 구조를 판단할 수 있도록 FSD 개요·구조 가이드·체크리스트 문서화

성과

- 아키텍처 위반을 리뷰어 판단이 아닌 커밋 단계에서 자동 검출
- CRA → Vite 전환으로 빌드 시간 **58%** 단축 (81s → 34s)

React

TypeScript

Vite

ESLint

Storybook

Husky

lint-staged

#### Server-Driven UI(SDUI) 시스템 설계 ↗

문제

브랜드마다 레이아웃·정보 구조·인터랙션 요구가 달라, 브랜드별 분기 컴포넌트가 늘며 번들과 유지보수 부담이 커지는 구조

해결

- 테마 시스템을 검토했으나 브랜드마다 다른 레이아웃 구조를 감당할 수 없어, **UI 구조 자체를 JSON으로 표현**하는 SDUI를 설계

- JSON의 role을 React 컴포넌트로 매핑하는 재귀 렌더러를 설계, 오류는 요소 단위로 격리해 일부 데이터가 잘못돼도 나머지 화면은 그러지게 처리
- element를 HTML 태그 수준의 최소 단위로 설계하고, 변형은 새 컴포넌트가 아닌 prop으로 흡수해 요구사항이 늘어도 컴포넌트 수가 늘지 않도록 유지
- 서버 JSON의 검증 규칙을 런타임에 **Zod 스키마로 변환**해 폼 검증을 동적으로 구성
- 상태·의사 선택자(hover 등)를 포함한 확장형 스타일 시스템을 설계해, 브랜드별 인터랙션·조건부 스타일을 JSON으로 제어

성과

- 신규 브랜드 화면 추가 시 기존 element를 JSON으로 조합해 대응
- **배포 없이** 서버 JSON 수정만으로 UI 변경을 즉시 반영

React Query   Jotai   Emotion   Zod   React Hook Form

### 어드민 백오피스 구축 및 고도화 ↗

문제

라우트·사이드바·접근 권한을 따로 관리해, 페이지 하나를 추가할 때 세 곳을 모두 수정해야 해 누락 위험이 있는 구조

해결

- 라우트 정의에 메뉴 메타데이터를 함께 선언하는 구조를 만들어, 라우트·사이드바·접근 권한을 **단일 정의로 동기화(SSOT)**
- 로그인 시점에 **권한별 라우트만 동적으로 등록**해, 허용되지 않은 경로를 라우터에서 제외
- 라우트 단위 코드 스플리팅으로 초기 로드 JS 축소
- 70+ 필드 폼 검증, 4개 언어(ko/en/ja/zh) 입력, OCR 기반 사업자등록증 자동 입력 등 운영 기능 구축

성과

- 페이지·권한 추가를 단일 정의 수정으로 처리
- 초기 로드 JS **48%** 경량화 (gzip 713KB → 373KB)

React Router   React Hook Form   Zod   react-i18next   CLOVA OCR API

### 의류 케어라벨 WYSIWYG 에디터 개발 ↗

문제

디자인 파일과 소통 기록이 흩어져 있어, 케어라벨 제작 진행 상황을 파악하기 어려운 상태

해결

- 그리드 1칸을 실물 1mm에 대응시켜, **편집한 치수 그대로 인쇄**되는 에디터 개발 (px 변환은 인쇄 표준 300 DPI 기준)
- react-grid-layout이 지원하지 않는 다중 선택·바운딩 박스 그룹 변형과 배치 전 영역 검증을 직접 구현
- 디자인을 JSON으로 직렬화해 재편집 가능하게 저장하고, 폰트 깨짐 없는 출력을 위해 사용 폰트를 임베딩한 고해상도 이미지로 캡처

상태

- 사내 검증 단계까지 개발

react-grid-layout   html-to-image   styled-components   FontFace API

## 커머스 웹사이트 구축 ↗

문제

SNS 판매의 수기 주문 처리에 한계가 있어, 브랜드 아이덴티티를 담은 자체 커머스가 필요한 상태

해결

- 기획·디자인·프론트엔드·배포·운영 전반을 맡아, **고객 스토어와 주문·배송·환불을 관리하는 관리자 페이지**를 갖춘 커머스 서비스 구축
- 주문·결제를 포함한 API 에러 케이스를 백엔드와 함께 정의하고, API 영역과 전역의 2단 Error Boundary로 상태별 분기·복구 처리
- PortOne 연동으로 카카오페이·토스페이·KG이니시스 **다중 PG 결제**를 구현해 실주문 결제 처리
- 피그마로 아토믹 디자인 기반 디자인 시스템을 만들어, 화면과 컴포넌트의 일관성을 유지
- AWS EC2 + GitHub Actions 배포 자동화

React

TypeScript

React Query

Jotai

styled-components

PortOne

AWS EC2

GitHub Actions

## Side Project

### 포트폴리오 & 기술 블로그 ↗

- Next.js App Router 기반 SSG 사이트 설계·운영
- MDX 콘텐츠 파이프라인을 구성해 마크다운 파일 추가만으로 글 발행
- react-three-fiber로 3D 명함 인터랙션 구현

Next.js

TypeScript

Tailwind CSS

shadcn/ui

MDX

react-three-fiber

## Skills

Frontend

TypeScript

React

Next.js

Vite

React Query

Jotai

React Hook Form

Zod

Styling & UI

Emotion

styled-components

Tailwind CSS

Storybook

DevOps & Quality

GitHub Actions

Vercel

AWS EC2

ESLint

Husky

lint-staged

Tools

Git

Figma

Cursor

Claude Code

## Education

### 홍익대학교

2013.03 - 2019.02 | 디자인영상학부