하예지

Product Engineer

010-5021-9862
1126yeji@naver.com
Portfolio ↗

불편함을 그냥 두지 못하는 만 5년 4개월차 프로덕트 엔지니어입니다.
웹을 중심으로, 필요하면 앱·서버·데이터까지 넘나듭니다.

01 손으로 하던 일을 코드로
- 손으로 관리하던 화면 목록 → 코드에서 화면 520여 개의 지도를 자동 생성하는 '크플 프로덕트 맵'
- 손으로 쓰던 API 코드 → OpenAPI 한 장에서 생성하는 Orval, 웹 레포 앱 6개의 표준으로 정착

02 경계를 넘어 끝까지
- 레거시 모노레포 이관 첫 주자 → 3개 도메인 217개 화면 이관, 사이트 첫 화면까지 전환
- 필요하면 Android·iOS 레포에도 직접 기여, 웹 신분증 촬영 첫 시도 성공률 67% → 96%

03 먼저 만들어서 보여 주기
- Orval은 먼저 적용해 보인 사례로 웹 전면 도입을 끌어냄
- 프로덕트 맵은 직접 제안하고 설계부터 구현까지 맡음(커밋 94%)
- 6명 스타트업 첫 프론트엔드로 서비스 하나를 출시부터 2.0까지 책임

WORK EXPERIENCE

피에프씨테크놀로지스 Client Team · Product Engineer

2024.12 — 현재

P2P 온투금융(투자·대출) 플랫폼 '크플' 운영 · **2025 하반기, 2026 상반기 연속 Exceeds Expectations**

0→1 · ARCHITECTURE

크플 프로덕트 맵 — 코드만 바뀌면 다시 그려지는 서비스 지도

기술 블로그 연재 · #1 코드만 바뀌면 다시 그려지는 서비스 지도 ↗ · #2 웹·Android·iOS·React Native 코드에서 한 장의 지도를 뽑기까지 ↗

520+	1,300+	6,000+	94%
화면·다이얼로그	화면 간 연결	자동 촬영 스크린샷	본인 커밋 비중

PROBLEM
- 웹·Android·iOS·React Native에 화면이 흩어져, 한 사용자 흐름이 어느 플랫폼의 어느 화면을 거치는지 레포를 하나씩 열어봐야 알 수 있음. 손으로 만든 화면 목록·플로우 문서는 코드가 바뀌는 순간 낡아 아무도 믿고 쓰지 못함.

DECISION & TRADE-OFF
- "사람이 관리하는 값은 관리되지 않는 값"을 원칙으로, 화면 목록·연결·스크린샷을 각 플랫폼의 코드·API 명세·테스트 인프라에서 파생하는 구조를 선택.
- 런타임 서버 없는 정적 파이프라인으로 운영 부담을 없애는 대신, 각 단계의 정확성을 테스트·가드로 지키는 비용을 수용. 의사결정은 ADR 290여 건으로 기록.

ACTION
- **플랫폼별 추출기:** 웹 라우트·이벤트·API 호출, Android·iOS 네이티브 화면과 화면 전환, RN 화면 연결, 네이티브↔웹뷰 브리지를 코드에서 읽어 하나의 그래프로 합침.
- **크로스플랫폼 화면 페어링:** 같은 화면의 Android·iOS 버전을 이벤트·화면 문구 등 5단계 신호로 자동 매칭해, 사람이 직접 적던 화면 쌍 11개를 0개로 만듦.
- **스크린샷 · 데이터:** 웹은 Playwright, 네이티브는 앱 레포의 스냅샷 테스트로 상태별 자동 촬영. Amplitude 퍼널로 연결 350여 개에 실제 전환율을 붙임.
- **AI 네이티브 개발:** 설계·ADR·검증 기준을 직접 정하고, 구현은 티켓 단위로 AI 에이전트(Claude Code)에 병렬 위임. 테스트 204개 파일·자동 검사 14종·AI 리뷰를 가드로 PR 380여 건 운영.

IMPACT

- 화면 520여 개·연결 1,300여 개·스크린샷 6,000여 장이 잡 한 번이면 손 수정 없이 다시 그려짐. 전체 파이프라인 62.6분 → 49.6분(같은 범위 기준 약 21% 단축).
- 지도로 만들고서야 보인 결함 발견(4xx·5xx를 화면에서 구분하지 않는 문제, 같은 계좌 인증이 iOS 3화면·Android 1화면). 사내 AI 어시스턴트가 이 지도를 데이터 소스로 사용.

ARCHITECTURE

거대 레거시 → 모노레포 마이그레이션

217개 화면
3개 도메인 이관

15 → 3분
투자 배포 시간

1,500+
미사용 파일 제거

PROBLEM

- 웹 화면 대부분이 한 레거시 레포에 혼재해 작은 수정에도 전체 빌드에 약 15분. 신규 기능이 계속 레거시 위에 쌓여 부채가 커지는 구조.

DECISION & TRADE-OFF

- 이관 첫 주자로, 조용한 도메인 대신 **신규 개발이 가장 활발한 투자 도메인**을 먼저 선택. 부채가 늘어나는 곳을 먼저 막고 이관 비용을 신규 개발 안에 흡수.
- 상품별 동적 SEO 때문에 SSR이 필요해, 모노레포 라우팅과의 일관성을 우선해 TanStack Start 선택(성숙도는 트레이드오프로 수용).

ACTION

- **데이터로 고른 정리:** AWS 접근 기록과 Amplitude를 대조해 안 쓰는 페이지를 가려 이관 전 약 640개 파일 정리(로컬 빌드 58% 단축).
- **되돌릴 수 있는 컷오버:** 새 앱을 약 한 달 병행 운영하고 재배포 없이 경로만 바꿔 전환. 운영 전환 중 문제를 발견하자 즉시 되돌리고 원인 해결 뒤 재전환.
- **확장과 마무리:** 회원·대출(가심사·계약·마이페이지)까지 같은 방식으로 이관하고, 사이트 첫 화면(/)까지 새 앱으로 전환. 옛 주소는 서버 리다이렉트로 넘기고 이관 뒤 미사용 코드(knip) 약 1,500개 파일 제거.

IMPACT

- 전환 뒤 신규 서비스 '세컨드 마켓'도 새 앱에서 런칭, 레거시 투자 영역 추가 개발 0건. 투자 배포가 앱 단독으로 분리(약 15분 → 약 3분).

DX & QUALITY

Orval 기반 API 연동 자동화와 백엔드 협업 표준

6개 앱
웹 레포 표준 API 레이어

1,800+
생성 operation

- API마다 fetch·훅·타입을 손으로 쓰고, 서버 변경이 공유되지 않아 타입이 실제 응답과 어긋나던 문제를 OpenAPI 하나로 생성하는 Orval로 해결. 직접 제안하고 먼저 적용해 적용기를 공유.
- 적용 중 명세에 없던 API를 찾아 문서화를 요청하고, 백엔드용 API 작성 가이드(Do & Don't) 중 투자 서버 몫을 작성해 전사 API 문서 규약에 반영.
- CTO 요청으로 웹 전면 도입. 팀장 평가: "백엔드-프론트엔드 간 API 협업 표준 수립".

WEBVIEW

앱 내 웹뷰 경험과 신분증 자동 촬영 공용화

67 → 96%
신분증 첫 시도 성공률

47 → 93%
OCR 성공률

59 → 21%
재시도율

- **신분증 자동 촬영:** 표준 API가 없는 웹에서 카드 검출과 네이티브 정책 기반 판정을 직접 구현하고, 공용 패키지로 묶어 투자·핀크·스탁론에 적용, 카카오페이에도 이식(배포 전후 비교, 위 수치).
- **네이티브 같은 화면 전환:** 네 방식을 직접 만들어 비교한 뒤 View Transitions API로 구현하고, iOS 스와이프 백은 터치 좌표로 추정해 전환이 겹치지 않게 처리. 팀장 평가: "새로운 기술들을 실무에 도입하여 성공 사례를 만듦".
- **책임 분리 · 안정화:** 웹뷰 세션 문제를 계기로 「웹뷰 페이지는 웹뷰 앱이 맡는다」를 ADR로 명문화. 웹뷰 Sentry 이슈의 약 절반(2,587건) 노이즈 정리, AI 보안 점검으로 찾은 민감 정보 로깅 경로 차단.

OTHERS

기타 기여

- **개발망 간편 로그인:** QA 때마다 계정 주인을 찾아 로그인을 부탁하던 문제를 제기하고, 계정 목록에서 누르면 바로 로그인되는 방식을 고안해 웹·앱 화면을 구현. 운영 번들에서는 코드 자체를 빼고 산출물로 검증. 웹·앱 개발의 기본 도구로 정착.
- **리브랜딩 홈탭(React Native):** 한 명에게 몰린 작업을 가져와 RN 화면부터 iOS·Android 작업까지 담당.
- **AI가 읽기 좋은 디자인 시스템:** AI가 모르면 조용히 틀리는 함정을 규칙·ADR로 남기고, Tailwind 기본값을 지워 Figma 토큰만 쓰게 고정. 규칙은 hook·CI 검사로 강제(동료와 공동 구축).

피치그로브 프론트엔드 개발자 · 웹 프론트엔드 프로젝트 **100% 단독 담당** 2021.07 — 2024.12

공동구독 중개 플랫폼 '링키드' 운영

0 TO 1 & OWNERSHIP

공동구독 플랫폼 '링키드' 구축부터 2.0 개편까지

2개월 + 1개월
베타 → 정식 출시

54 → 91
Lighthouse Performance

75 → 100
Lighthouse SEO

−87%
이미지 용량

- 6명 스타트업의 첫 프론트엔드로 파티 찾기·만들기·관리·출금 등 핵심 기능과 백오피스를 단독 구축, 베타 2개월·정식 1개월 만에 출시하고 1.0 → 2.0까지 책임.
- 서버가 입력 단계와 다음 화면을 정의하는 시나리오 시스템(Server-Driven Flow) 기획에 참여해, 배포 없이 새 상품을 운영할 수 있게 함.
- 2.0 개편에서 Next.js로 SEO가 필요한 페이지에 SSR을 점진 도입하고 이미지·번들을 최적화(이미지 158MB → 19MB). 고객 응대(CX)를 직접 맡아 피드백을 바로 코드에 반영.

PROJECTS

SIDE PROJECT · DOMAIN DESIGN

커뮤니티 운영·회비 정산 플랫폼 '97oxrunners'

2026.07 — 현재 · 초기 구조와 핵심 도메인 설계 주도 · 운영 중

80명
회원이 사용

1번
거래내역서 업로드로 회비 정산

- 카톡 공지와 엑셀로 흩어진 80명 커뮤니티 운영을 한곳으로. 초기 앱 구조와 AI 작업 기반(지침·용어·ADR)을 만들고 회원제·일정·회비·회계 도메인을 설계(Next.js·tRPC·Supabase).
- 은행 거래내역서를 올리면 입금을 회비·참가비·굿즈 대금에 자동으로 맞추고, 애매한 입금은 응답·주문 시각으로 추천해 운영진이 확정. 결산·회원 공시까지 한 흐름으로 연결. 일정 공지는 웹 푸시로 대체.

SKILLS

프론트엔드	TypeScript, React (v19), Next.js (App Router), TanStack Router · Start · Query
모바일	React Native, Android · iOS 네이티브 레포 기여 (Roborazzi · swift-snapshot-testing 캡처 인프라, 개발망 로그인)
아키텍처 / API	Turborepo · pnpm 모노레포, Orval · OpenAPI 코드 생성, tRPC, Supabase (Postgres · RLS · Auth)
테스트 / 자동화	Playwright, Vitest, knip
데이터 / 관측	Amplitude, Sentry
AI 협업	Claude Code (멀티 에이전트 병렬 작업), Cursor, Codex

EDUCATION & CERTIFICATIONS

강릉원주대학교 컴퓨터공학과 · 우등 졸업 — 2016.03 — 2021.02

정보처리기사 한국산업인력공단 — 2019.05