

# 이중표

Frontend Developer

leejpsd@gmail.com | 010-4615-7794 | <https://velog.io/@leejpsd>

## ABOUT

---

사용자가 "빠르다"고 느끼는 순간을 설계하는 프론트엔드 개발자입니다.

대규모 트래픽 환경에서 SSR·ISR 렌더링 전략 설계, React Query + Suspense 기반 선언적 비동기 UI, 이벤트 기반 캐시 무효화로 실시간 데이터 동기화를 구현해왔습니다. WebView 환경에서 네이티브 앱 연동 경험을 갖고 있으며, Core Web Vitals 73→99점, SEO 76→100점, CTR +32% 등 기술 개선을 비즈니스 성과로 검증해왔습니다.

## CORE COMPETENCIES

---

### 선언적 비동기 UI

React Query + Suspense + ErrorBoundary로 로딩/에러/성공 상태를 선언적 관리. 스켈레톤 UI 타이밍 설계로 체감 속도 개선.

### 캐시 전략 설계

ISR 3가지 패턴(시간/Webhook/사용자 이벤트) 설계.  
revalidateTag 기반 정밀 캐시 무효화로 실시간 UX 보장.

### WebView 최적화

iOS/Android WebView 성능 최적화, 네이티브 브리지 설계,  
앱-웹 인증 동기화, 생체 인증 연동 경험.

### 마이그레이션 & 아키텍처 설계

Django→Next.js 전면 마이그레이션 리드, SSR·SSG·ISR 혼합  
렌더링 전략 설계, 서버 비용 54% 절감.

## EXPERIENCE

---

### 플래튼 (FLATTEN) Frontend Developer (정규직)

2022.11 — 현재

- Django → Next.js 14 마이그레이션 리드, CWV 73→99·SEO 76→100·CTR +32%·신규 유입 +28% (6개월 검증)
- SSR·SSG·ISR 렌더링 전략 설계, revalidateTag 기반 캐시 무효화로 실시간 UX와 성능 양립
- React Query + Suspense 기반 선언적 비동기 UI 구현, Skeleton UI 타이밍 설계로 체감 속도 개선
- 신한 SOL·솔드아웃 제휴를 위한 UI/UX 개편 주도, 신규 유저 30만 명 유입·MAU 33% 성장 기간

- WebView 네이티브 브리지 설계, 앱-웹 인증 동기화·생체 인증 연동으로 하이브리드 앱 UX 향상
- Sentry 에러 모니터링·Jenkins CI/CD 도입, 장애 대응 1시간→20분 단축
- SEO TF(PM·AOS·BE·FE) 리딩, Search Console·GA4 기반 분석 체계로 클릭수 105%↑·순위 12.5→6.8
- 프론트엔드 3명 리드, 점심 스터디로 Next.js·React Query 학습 주도, 가이드 문서화로 팀 자산 정착

## PROJECT DETAILS

---

**Shoeprize 슈프라이즈** 2022.11 — 2025.08 | 프론트엔드 리드 (3명)

MAU 90만, 동시접속 3만 / 한정판 스니커즈 발매정보 플랫폼

### 문제 상황

한정판 발매 알림 시 동시접속 3만 명으로 서버 과부하 및 페이지 지연 발생. Django SSR의 성능 한계로 Core Web Vitals 73점, TTFB 2.8초. SEO 점수 76점으로 검색 유입 확대에 제약. 프론트엔드 독립 아키텍처가 필요한 상황.

### 해결 과정

- 렌더링 전략 설계:** 페이지 특성별 SSR·SSG·ISR·CSR 조합 설계
  - ISR(시간 기반): 랭킹/피드 등 주기 간접 영역에 revalidate 적용
  - ISR(이벤트 기반): Webhook → revalidatePath로 어드민 수정 시 즉시 무효화
  - ISR(사용자 이벤트): revalidateTag로 투표/좋아요 등 사용자 액션 시 캐시 갱신
  - SSG: 변동이 적은 정보는 정적 프리렌더
- React Query + Suspense 선언적 비동기 UI:** Suspense를 above-the-fold 기준으로 분리, 지연 구간은 Skeleton/Fallback으로 체감 공백 최소화
- 상태 관리 아키텍처:** Zustand를 도메인 스토어 단위로 분리, selector + subscribe 패턴으로 필요한 컴포넌트만 리렌더
- WebView 브리지 설계:** 웹뷰 컨트롤 및 브리지 함수 구현, 앱 개발자와의 협업을 통한 네이티브 연동

### 성과

- Core Web Vitals:** 73 → 99 (LCP 8.6s→0.9s, FCP 55% 단축, CLS 0.25→0.009)
- 인프라 최적화:** 서버 인스턴스 13 → 6대, 비용 54% 절감, TTFB 2.8s → 1.2s
- 사용자 지표:** 이탈률 -20%, 세션 지속시간 +18%, 재방문율 상승

### 트러블슈팅: 라우터 캐시와 실시간 UX

**문제:** 투표 후 페이지 이동 → 복귀 시 최신 투표 수 미반영. 백엔드는 정상이나 라우터 캐시가 이전 데이터 노출. 사용자는 "내 투표가 반영 안 됐다"는 불신.

**분석:** Next.js 14.1 라우터 캐시는 opt-out 불가. prefetch 기본값(null)은 동적 경로 30초, prefetch={true}는 5분 캐시. prefetch={false}도 방문 이력은 30초간 캐시 유지.

**해결:** API 응답에 태그 부여 + Server Action에서 revalidateTag 호출로 해당 캐시만 정밀 무효화. 투표 직후 캐시 정리 → 재방문 시 항상 최신 데이터.

**인사이트:** 캐시를 끄는 것이 아닌, 사용자 이벤트에 맞춘 정밀 무효화가 실시간 UX와 캐시 성능을 양립시키는 핵심. 이 경험을 페이지별 SSR/SSG/ISR 경계 문서와 팀 가이드로 정리해 이후 관련 이슈 0건 달성.

**Kasina 카시나** 2023.08 — 현재 | 프론트엔드 팀원

프리미엄 스니커즈 커머스 플랫폼 / 안정적 운영과 사용자 경험 개선

### 문제 상황

Shopby + 사내 API 인증 복잡, API 실패/UX 불일치 빈발. WebView 환경에서 앱-웹 인증 동기화 필요. 안정적 커머스 운영/확장을 위한 프론트엔드 독립 아키텍처 필요.

### 해결 과정

- API 통합 레이어 설계:** Axios 래퍼로 단일 엔드포인트 인터페이스 설계(타임아웃, 공통 쿼리, 로깅)
  - 요청 인터셉터에서 액세스 토큰 삽입, 응답 인터셉터에서 상태코드별 에러 정규화
  - 토큰 갱신 큐를 두어 동시 401 발생 시 단일 refresh만 수행, 실패 시 일괄 로그아웃
  - AbortController 기반 요청 취소 + idempotent 요청만 재시도
- 인증/보안 플로우 구현:** 자체 OAuth + 리프레시 토큰 자동 갱신, 앱 토큰·생체 인증 연동
  - 로그인/주문 등 민감 경로에 가드 미들웨어 적용
- WebView 브리지 설계:** 앱 내 WebView 브리지를 통한 네이티브 ↔ 웹 인증 컨텍스트 동기화
- 클라이언트 데이터 관리:** React Query 캐시 키 설계(사용자/장바구니/주문), 의존 쿼리·낙관적 업데이트·에러 복구
  - UI 상태는 Zustand로 경량 관리(모달, 토스트 등)
- 운영 신뢰성:** Sentry 릴리즈 트래킹/소스맵/Slack 알림, 장애 시나리오별 화면 처리·복구 플로우 사전 정의

### 성과

- API 실패율:** -40%, 로그인 성공률 +25%
- TTR:** 1시간 → 20분 (에러 추적 체계화)
- 비즈니스:** 장바구니 이탈 -12%, 결제 전환 +7%, 세션 지속 +15%
- 사용자 편의:** 자체 인증·생체 인증으로 고객 편의성↑, 월간 수십만 명 이용 환경에서 UX 안정성 확보

**트러블슈팅: 동시 API 요청의 토큰 갱신 Race Condition**

**문제:** 장바구니·주문·사용자 정보 등 여러 API가 병렬 호출 시 토큰 만료되면 모두 401 → 각각 refresh 요청 시도 → race condition 발생. 불필요한 네트워크 비용, 일부 요청 실패.

**해결:** isTokenRefreshing flag + 재시도 폴링 패턴 구현. 첫 401은 refresh 진행, 이후 401은 500ms 대기 후 재시도(최대 2회). refresh 실패 시 일괄 로그아웃. Shopby·Kasina 이중 API 환경에서 각각 독립적 토큰 관리 구조 설계.

**인사이트:** 동시 요청의 토큰 갱신 race condition 완전 해소. 분산된 비동기 요청의 동시성 제어가 커머스 안정성에 직결됨을 확인. 단순 재시도보다 중앙화된 상태 관리가 인증 플로우의 핵심.

## Flatten Corporate Site 2025.04 — 2025.05 | 단독 담당

requestAnimationFrame + Lenis로 60fps 패럴렉스 구현, GPU 합성 레이어 최적화. 한 달 내 기획→배포 완료, 신규 파트너사 영입 기여.

## EDUCATION

---

### 스파르타 코딩클럽 | 항해99 프론트엔드 집중 과정 수료

2022.07 — 2022.10

### 신구대학 | 색채디자인과 졸업 (컬러리스트 산업기사)

2012.03 — 2017.02