

편의점 종합 솔루션 : 실시간 통합 관리 플랫폼

고객, 점주, 본사를 잇는 완결형 디지털 생태계 구축



목차

01 | 프로젝트 개요

- 프로젝트 주제 및 선정 배경
- 프로젝트 기획 의도
- 프로젝트 내용

02 | 프로젝트 팀 구성 및 역할

03 | 프로젝트 수행 절차 및 방법

04 | 프로젝트 수행 경과

- 데이터베이스 구축
- 핵심 기능 구현: 실시간 주문 처리
- 피드백과 개선

05 | 자체 평가 의견

- 잘한 점: 최신 기술 도입, BaaS 활용
- 아쉬운 점: 테스트 자동화 부족

06 | 향후 개선점 및 개인 성장

- 코드 품질 및 안정성 강화
- 성능 최적화
- 기능 확장: PWA, 고급 분석 기능

프로젝트 주제 및 선정 배경

실시간 데이터 동기화 기반의 편의점 통합 관리 플랫폼 개발

현재 편의점 운영 환경은 주문, 재고, 매출 관리가 파편화되어 있어 비효율이 발생하고 있습니다. 실시간 데이터의 부재로 재고 부족이나 주문 폭주와 같은 현장 상황에 즉각적인 대응이 어렵고, 점주, 본사, 고객 간의 소통 채널이 부족한 실정입니다. 이러한 문제점을 해결하기 위해 하나의 통합 플랫폼을 통해 모든 관계자가 실시간으로 정보를 공유하고 효율적으로 업무를 처리할 수 있는 솔루션 개발이 필요합니다.



프로젝트 기획 의도

고객, 점주, 본사를 잇는 완결형 디지털 생태계 구축

본 프로젝트는 편의점 운영의 모든 측면을 하나의 플랫폼에서 처리할 수 있는 통합 솔루션을 제공하고자 합니다. 실시간 데이터 동기화를 통해 비즈니스 민첩성과 효율성을 극대화하는 것이 핵심 목표입니다.

고객에게는 편리한 주문 경험을 제공하고, 점주에게는 매장 운영을 위한 강력한 관리 도구를, 본사에게는 전체 프랜차이즈 네트워크를 효과적으로 모니터링할 수 있는 시스템을 구축합니다.

이를 통해 모든 이해관계자가 상생할 수 있는 디지털 생태계를 조성하여 편의점 비즈니스의 새로운 패러다임을 제시하고자 합니다.



프로젝트 내용

고객용 앱

- GPS 기반 지점 검색
- 상품 주문 및 장바구니
 - 실시간 주문 추적
- 토스페이먼츠 간편 결제
 - 주문 이력 조회
 - 즐겨찾기 기능
 - 푸시 알림 수신

점주용 대시보드

- 실시간 주문 접수 알림
 - 주문 상태 관리
- 스마트 재고 관리
- 일/주/월 매출 분석
 - 본사 물류 요청
 - 인기 상품 분석
 - 프로모션 설정

본사용 관리 시스템

- 전국 지점 통합 모니터링
 - 상품 마스터 관리
- 물류 승인 및 배송 관리
 - 전사 데이터 분석
 - 매출 트렌드 분석
 - 지점별 성과 비교
- 신규 상품 출시 관리
- 프로모션 일괄 적용

핵심 특징

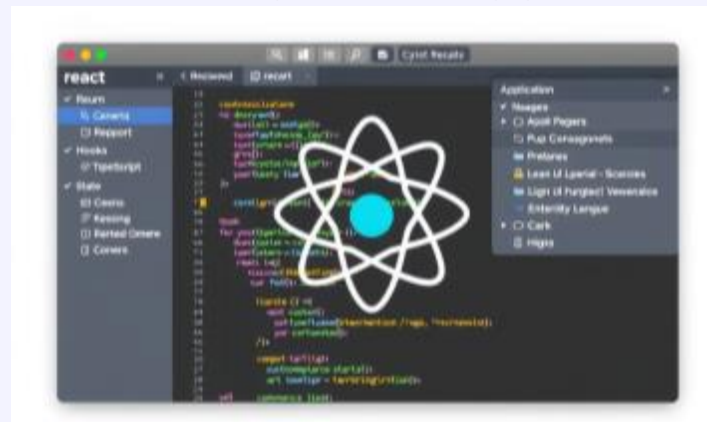
- Supabase Realtime을 통한 실시간 데이터 동기화
- 역할 기반 접근 제어(RLS)
 - 자동화된 재고 관리
- 클라우드 기반 확장성
- 모바일 최적화 UI/UX
- 다중 결제 수단 지원
- 데이터 백업 및 복구

활용 장비 및 재료 (기술 스택)



Frontend

- React 19
- TypeScript
 - Vite
- Zustand
- TanStack Query
- Tailwind CSS



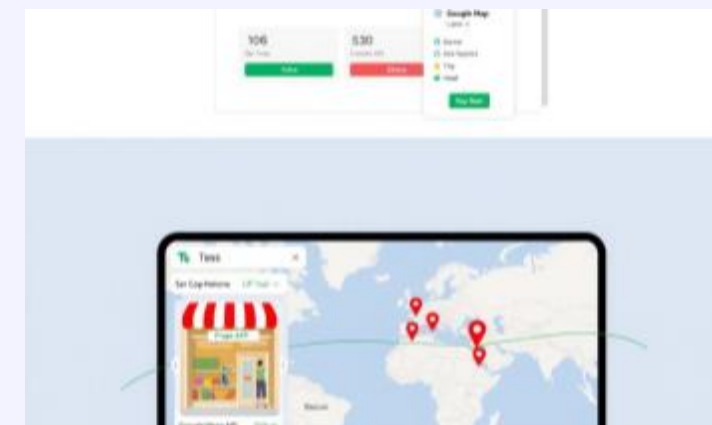
Backend & Database

- Supabase
- PostgreSQL
 - Auth
- Realtime API
- Storage
- RLS 정책



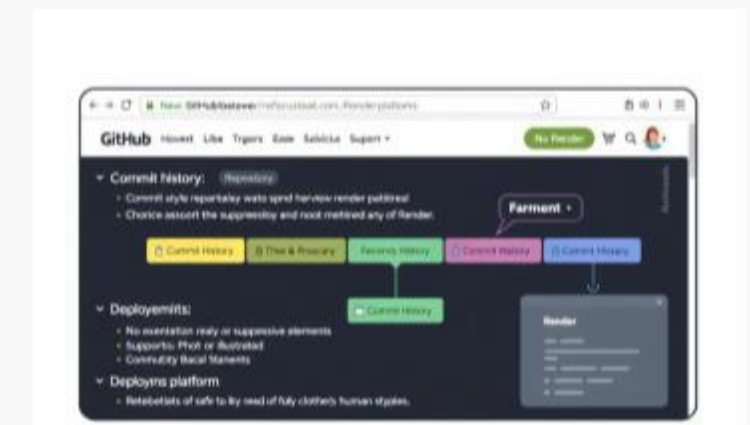
외부 연동

- 토스페이먼츠
- Naver Maps API
- 푸시 알림 서비스
- 이메일 서비스

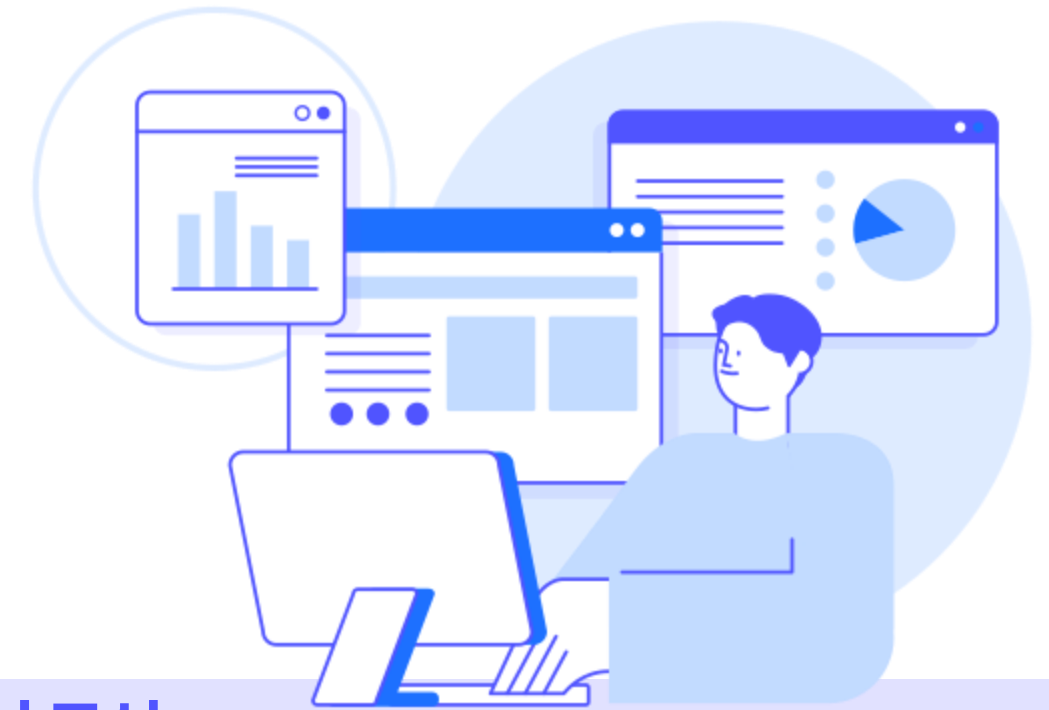


개발/배포

- Git
- GitHub
- Render (PaaS)
- CI/CD 파이프라인

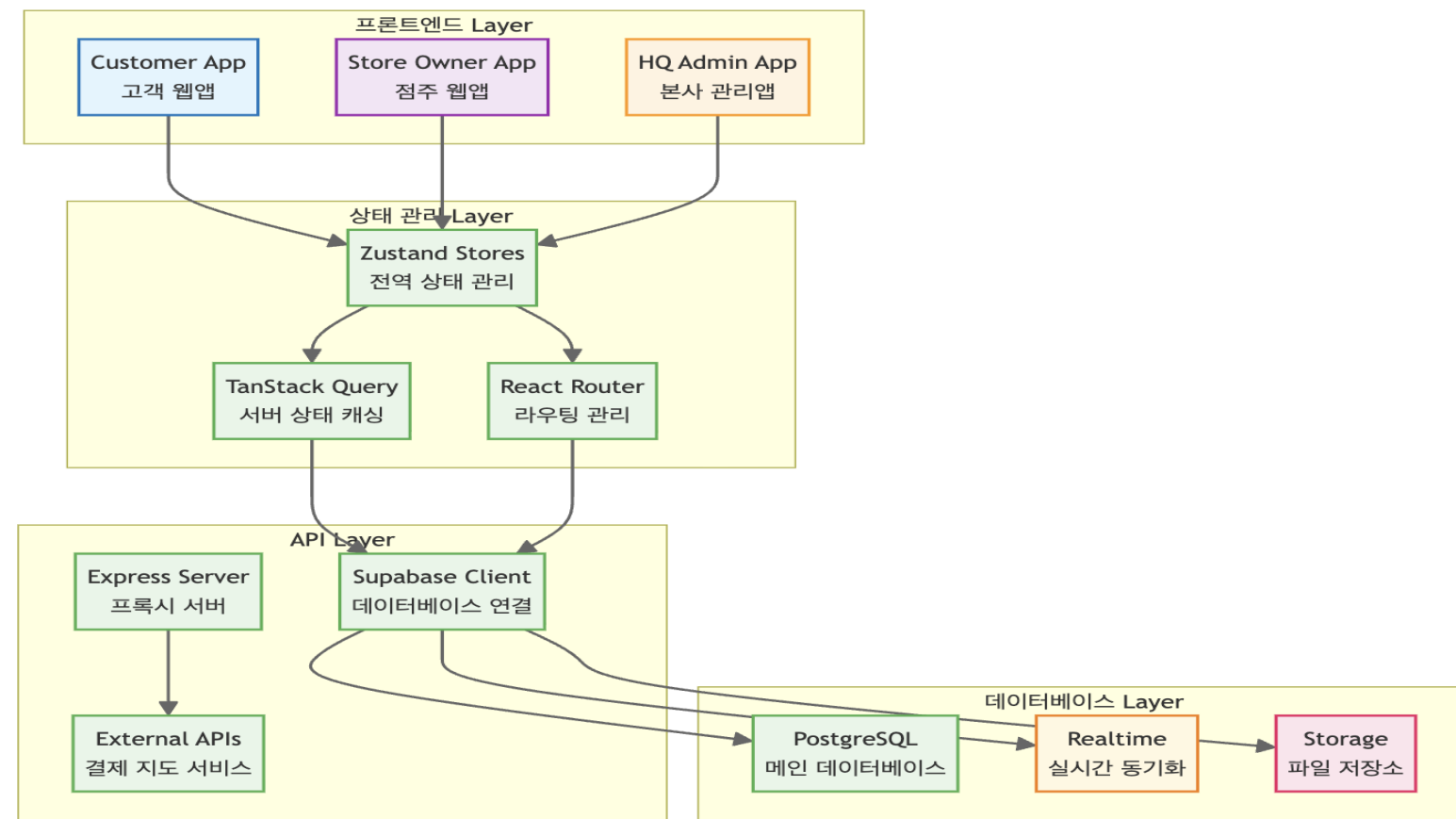


프로젝트 구조 (아키텍처)

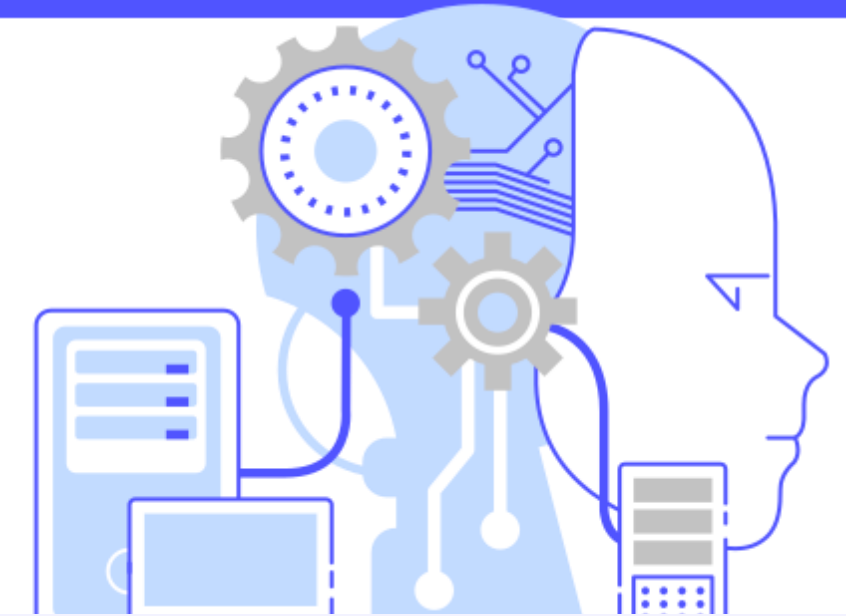


시스템 아키텍처 구성

- 중앙 Supabase DB를 핵심으로 고객, 점주, 본사 애플리케이션이 API와 Realtime 서비스로 연결
- Frontend: 역할별(customer, store, hq) 모듈식 구조로 분리하여 재사용성 극대화
- Backend: Supabase BaaS로 인증, DB, 실시간 통신, 스토리지 등 핵심 기능 제공
- Database: 17개 정규화된 테이블과 RLS 정책으로 역할 기반 데이터 접근 제어
- 트리거와 함수를 활용한 비즈니스 로직 자동화 구현



활용 방안 및 기대 효과



활용 방안

- 중소규모 편의점 프랜차이즈의 즉시 도입 가능한 통합 솔루션
- 재고 관리 및 자동 발주 시스템이 필요한 소매점
 - 실시간 주문 처리가 중요한 F&B 매장
- 데이터 기반 의사결정이 필요한 소매 비즈니스

운영 효율성 증대

- 재고 관리, 주문 처리 자동화로 수작업 감소
 - 인건비 절감 및 업무 효율성 향상
- 실시간 데이터 동기화로 업무 지연 최소화
- 통합 플랫폼으로 다중 시스템 관리 부담 해소

매출 및 고객 만족도 향상

- 데이터 기반 수요 예측으로 품절 최소화
 - 판매 기회 손실 방지 및 매출 증대
- 편리한 주문/결제/추적 기능으로 고객 경험 개선
- 실시간 분석 데이터로 신속한 비즈니스 전략 수립



팀 구성 및 역할

훈련생	역할	담당 업무		
김기찬	팀장	 역할 배분, 일정관리	 본사 시스템 개발	 통합 테스트, 인프라 구축
조유빈	팀원	 고객 시스템 개발	 점주 시스템 개발	 기능 테스트
이영호	팀원	 고객 시스템 개발	 3-tier 아키텍처 개발	 기능 테스트
김승기	멘토	 피드백, 질의응답		

프로젝트 수행 절차 및 방법

구분	기간	활동	비고
요구 분석, 설계	8/11(월) ~ 8/14(금)	역할별 기능 정의 및 유저 스토리 작성, DB 스키마, UI/UX, API 명세 설계	핵심 로직 확정, 화면정의서 작성 DB 스키마 확정 (17개 테이블)
백엔드, 프론트 구현	8/14(월) ~ 8/20(금)	Supabase DB 구축 및 서버 로직 구현, 역할별 페이지 및 공통 UI 컴포넌트 개발	역할 기반 접근 제어, 로직 자동화 React 19/Vite 활용, 반응형 UI
외부 연동 시스템	8/18(월) ~ 8/19(금)	토스페이먼츠, 주소/지도 API 연동,	결제/API 연동, CORS 이슈 해결
상태관리 및 최적화, 테스트	8/19(월) ~ 8/21(금)	Zustand/TanStack Query 적용 및 성능 최적화, 단위/통합/E2E 테스트 및 버그 수정	API 호출 최소화, 로딩 속도 개선 기능 안정성 확보, 시나리오 기반
배포 및 문서화, 발표 준비	8/21(월) ~ 8/22(금)	프로덕션 배포 및 프로젝트 문서화, 발표 자료 제작 및 최종 시연 준비	SPA 라우팅 해결, 최종 문서화 최종 점검 및 역할 분담
총 개발기간	8/11(월) ~ 8/22(금)(총 2주)		

수행 경과 1 - 데이터베이스 구축

Supabase 기반 데이터베이스 설계 및 구현

프로젝트의 핵심 기반으로 Supabase와 PostgreSQL을 활용하여 견고한 데이터베이스를 구축하였고 17개 테이블, RLS 정책, 함수, 트리거를 일괄 생성할 수 있게 했습니다. 사용자 역할(고객/점주/본사)에 따라 데이터 접근 권한을 DB 수준에서 제어하는 RLS 정책을 구현하여 보안을 강화했으며, 주문 완료 시 재고 차감 등 비즈니스 로직을 트리거로 자동화했습니다.

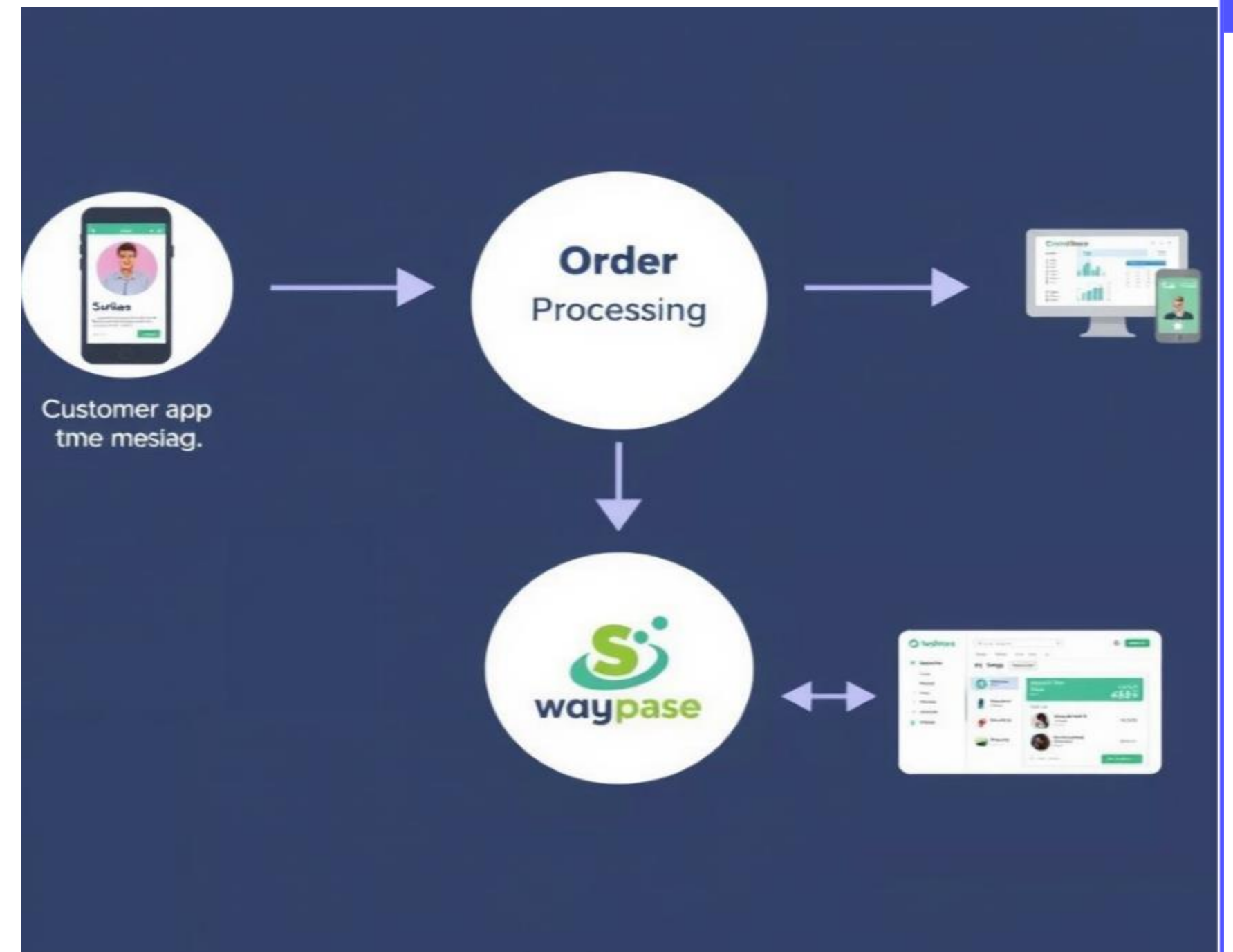


수행 경과 2 - 핵심 기능 구현

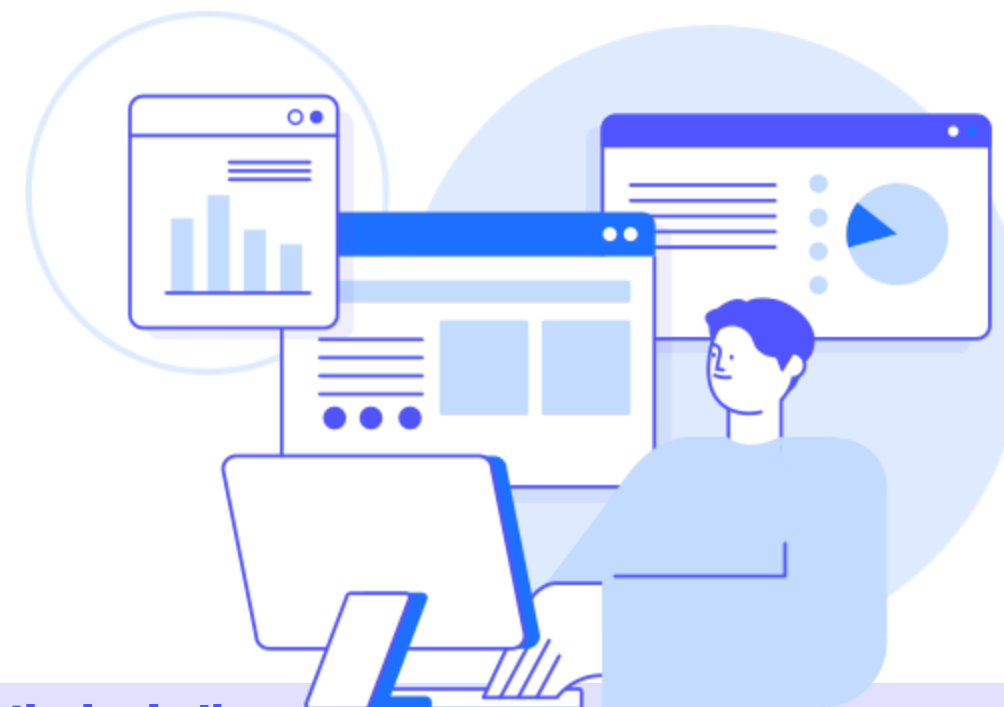
실시간 주문 처리 시스템 구현 및 결제 연동

본 프로젝트의 가장 핵심적인 기능인 실시간 주문 처리 시스템을 성공적으로 구현했습니다. 주문 프로세스는 다음과 같이 작동합니다.

1. 고객이 앱에서 상품을 선택하고 토스페이먼츠를 통해 결제를 진행합니다.
2. 결제 성공 시 orders 테이블에 주문 데이터가 생성되며, 이때 DB 트리거가 작동하여 재고를 자동으로 차감합니다.
3. 점주는 Supabase Realtime 구독을 통해 신규 주문 알림을 실시간으로 수신하고, 대시보드가 자동으로 업데이트됩니다.
4. 점주가 주문 상태를 변경하면(예: '처리중'→'준비완료'), 고객 앱에서도 실시간으로 상태 변화를 확인할 수 있습니다.



수행 경과 3 - 피드백과 개선



물류 요청 시스템 개선 사례

- 초기 문제점

재고가 0인 상품이 물류 요청 목록에 나타나지 않아 요청 불가

- 사용자 피드백

재고가 없는 상품에 대한 물류 요청이 필요함

- 개선 내용

INNER JOIN에서 LEFT JOIN으로 쿼리 수정

상품 레코드 없을 시 동적 생성 로직 추가

물류 요청 목록						전체	대기중	승인됨	배송중
요청번호	상태	우선순위	금액	요청일	관리				
REQ-1756105669350-43	배송완료	urgent	75,000원 송안: 75,000원	2025. 8. 25.	상세역렬서명				
REQ-1756105396206-179	배송완료	normal	75,000원 송안: 75,000원	2025. 8. 25.	상세역렬서명				
REQ-1756105158043-685	배송완료	normal	75,000원 송안: 75,000원	2025. 8. 25.	상세역렬서명				
REQ-1755445326499-577	배송완료	urgent	92,000원 송안: 92,000원	2025. 8. 18.	상세역렬서명				
REQ-1755315120697-520	배송완료	normal	46,000원 송안: 46,000원	2025. 8. 16.	상세역렬서명				
REQ-1755146719552-935	배송완료	urgent	192,000원 송안: 192,000원	2025. 8. 14.	상세역렬서명				
REQ-1754985591197-497	배송완료	urgent	1,380,000원 송안: 1,380,000원	2025. 8. 12.	상세역렬서명				
REQ-1754984862968-832	배송완료	normal	10,000,000,000,000원 송안: 10,000,000,000,000원	2025. 8. 12.	상세역렬서명				
REQ-1754973004524-486	배송완료	normal	432,000원 송안: 432,000원	2025. 8. 12.	상세역렬서명				
REQ-1754973002834-925	배송완료	normal	432,000원 송안: 432,000원	2025. 8. 12.	상세역렬서명				
REQ-1754972978143-194	배송완료	normal	760,000원 송안: 760,000원	2025. 8. 12.	상세역렬서명				

자체 평가 의견 - 1. 완성도 평가

프로젝트 완성도

- 기획 단계에서 정의한 모든 핵심 기능이 완전히 구현됨
- 주문, 결제, 재고, 물류, 분석 등 모든 워크플로우 완성

기능 구현

- 고객, 점주, 본사 역할별 시나리오 테스트 통과
- 실시간 데이터 동기화 안정적 작동
- 대용량 트래픽 테스트 미실시

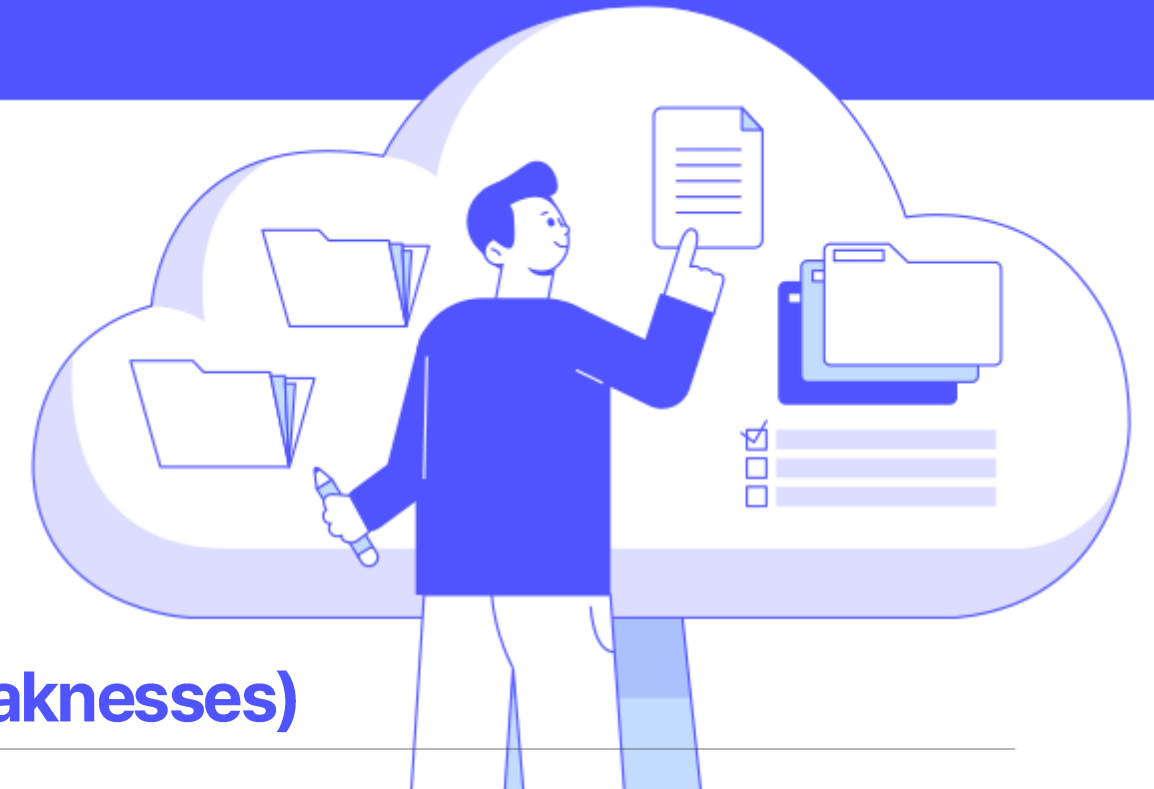
시스템 안정성

- 직관적인 UI와 실시간 피드백 제공
- 사용자 피드백 반영으로 초기 프로토타입보다 사용성 개선
- 역할별 최적화된 인터페이스

사용자 경험

- 상용 수준에 근접한 프로토타입
- 기술적으로 모든 기능이 작동하며 실제 비즈니스 환경 사용 가능

자체 평가 의견 - 2. 잘한 점 & 아쉬운 점



잘한 점 (Strengths)

- 과감한 최신 기술 도입 및 성공적 적용
React 19, Supabase 등 최신 기술 스택을 적극 활용하여 개발 생산성과 애플리케이션 성능을 동시에 확보
- BaaS를 활용한 빠른 MVP 개발
Supabase를 BaaS로 사용하여 백엔드 구축 시간을 획기적으로 단축하고 핵심 비즈니스 로직 구현에 집중
- 견고한 아키텍처 설계
DB(RLS), API, UI 컴포넌트까지 역할 기반으로 명확하게 분리 설계하여 높은 보안, 확장성, 유지보수성 확보

아쉬운 점 (Weaknesses)

- 테스트 자동화 부족
기능 구현 일정에 집중한 나머지, 코드의 안정성을 보장하는 단위 테스트 및 E2E 테스트 코드 작성 미흡
- 문서화의 깊이 부족
개발 과정의 빠른 속도를 따라잡기 위해 코드 내 주석이나 API 명세 등의 문서화가 일부 부족
- 대용량 트래픽 테스트 미 실시
실제 서비스 환경에서의 성능 검증이 이루어지지 않아 확장성에 대한 실질적 검증 부족

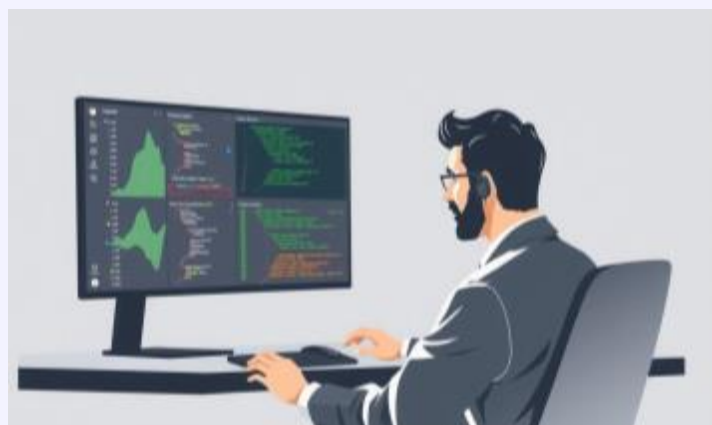


자체 평가 의견 - 3. 향후 개선점



코드 품질 강화

- 테스트 커버리지 확보
- Vitest, Playwright 도입
- 테스트 커버리지 80% 이상
- CI/CD 파이프라인 고도화



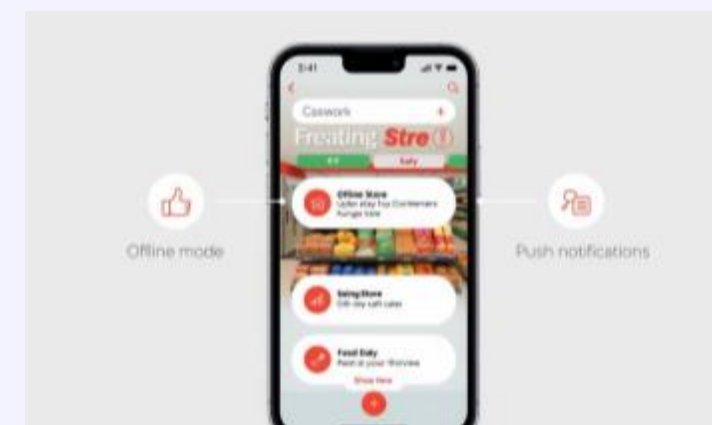
성능 최적화

- 데이터베이스 최적화
- 복잡한 쿼리 실행 계획 분석
- 필요 컬럼에 인덱스 추가
- 프론트엔드 코드 스플리팅



기능 확장

- PWA 도입
- 푸시 알림 기능 추가
- 고급 데이터 시각화 기능



개인 성장

- BaaS 활용 능력 심화
- 실시간 웹 기술 역량 증진
- 고급 DB 기술 활용 능력
- 문제 해결 능력 향상



자체 평가 의견 - 4. 개인 성장



이번 프로젝트를 통해 다양한 측면에서 기술적 성장을 이룰 수 있었습니다.

- BaaS 활용 능력 심화

Supabase를 활용한 인증, DB, 실시간 기능을 포함한 풀스택 애플리케이션을 구축할 수 있는 역량을 확보했습니다.

- 실시간 웹 기술 역량 증진

WebSocket 기반의 실시간 통신 원리를 이해하고 실제 서비스에 적용하여 동적인 사용자 경험을 구현했습니다.

- 문제 해결 능력 향상

복잡한 비즈니스 요구사항을 DB 스키마 설계, RLS 정책, 트리거 등 기술적 해결책으로 풀어내는 능력이 향상되었습니다.

- 빠른 개발과 안정성의 균형

BaaS를 통한 빠른 개발 속도와 함께 테스트와 문서화의 중요성을 체감하며 균형 잡힌 개발 접근법을 배웠습니다.



데모 영상