

# 떡담 (Tteokdam)

떡 중개 판매 플랫폼

개발 계획서

떡 + 담(談) = "떡 이야기를 나누는 공간"

떡 + 담(擔) = "여러 업체의 떡을 한곳에 담는 플랫폼"

## 서브에이전트 팀 구조

총괄 (Manager) — 계획 수립 · 작업 분배 · 통합 관리

—— 디자인 Agent — tailwind 테마, 색상, 글로벌 스타일

—— 백엔드 Agent — 데이터(lib/), Context, 설정 파일

—— 프론트 Agent — 컴포넌트(26개), 페이지(13개)

—— 테스트 Agent — 빌드 검증, 오류 수정

## 기술 스택

Next.js 14 (App Router) + React + Tailwind CSS

디자인 컨셉: "韓모던 (Han-Modern)"

# 목차

---

1. 브랜드 & 컨셉
2. 서브에이전트 팀 구조
3. 전체 파일 구조
4. 페이지별 역할
5. Context 설계
6. Tailwind 커스텀 테마 설계
7. 구현 단계 (Phase별)
8. 검증 방법

# 1. 브랜드 & 컨셉

---

브랜드명: 떡담 (Tteokdam)

떡 + 담(談) = "떡 이야기" — 떡에 대한 이야기를 나누는 공간

떡 + 담(擔) = "떡을 담다" — 여러 업체의 떡을 한곳에 담는 플랫폼

짧고 기억하기 쉽우며, 전통적이면서도 현대적인 느낌

디자인 컨셉: "韓모던 (Han-Modern)"

전통 한국의 따뜻함과 현대적 미니멀리즘의 조화

미색/쌀색 배경, 밤색 텍스트, 팡색·엷색·깊색 포인트

핵심 차별점: 떡 중개 마켓플레이스

- 여러 떡 업체의 상품을 한곳에 모아 판매
- 업체별 상품 비교 기능 제공
- 업체 입점 신청 시스템
- 중개수수료 기반 비즈니스 모델

## 2. 서브에이전트 팀 구조

에이전트	담당 영역	주요 산출물
총괄	계획 수립, 작업 분배, 통합 관리	전체 프로젝트 조율
디자인	tailwind.config.js, 색상/폰트/그림자	디자인 시스템 파일
백엔드	lib/ 데이터, contexts/, hooks/	데이터/상태/설정 파일
프론트	components/, app/ 페이지 전체	컴포넌트/페이지 파일
테스트	빌드 검증, 오류 수정, 최종 점검	빌드 성공 확인

### 병렬 실행 전략

독립적인 작업은 병렬로 실행하여 개발 속도를 극대화합니다:

- Phase 1: 디자인 Agent + 백엔드 Agent 동시 진행
- Phase 2~4: 프론트 Agent 2개를 동시 투입
- Phase 6: 테스트 Agent 단독 검증

### 3. 전체 파일 구조

```
agents_homepage/
+-- app/
|   +-- layout.js      <- 전체 레이아웃
|   +-- page.js        <- 홈 페이지
|   +-- globals.css    <- 글로벌 스타일
|   +-- products/
|       +-- page.js    <- 상품 목록
|       +-- [id]/page.js <- 상품 상세
|   +-- compare/page.js <- 상품 비교
|   +-- cart/page.js   <- 장바구니
|   +-- checkout/page.js <- 주문서
|   +-- auth/page.js   <- 로그인/회원가입
|   +-- mypage/page.js <- 마이페이지
|   +-- vendors/
|       +-- page.js    <- 입점 업체 목록
|       +-- [id]/page.js <- 업체 상세
|   +-- vendor-register/page.js <- 업체 등록
|   +-- about/page.js   <- 소개
|   +-- notice/page.js <- 공지사항
+-- components/
|   +-- ui/     Button, Input, Card, Badge, Modal, Toast
|   +-- layout/  Header, Footer, MobileMenu, Container
|   +-- product/ ProductCard, ProductGrid, ProductFilter
|   +-- home/   HeroBanner, PopularProducts, CategorySection
|   +-- vendor/ VendorCard, VendorGrid, VendorRegisterForm
|   +-- cart/   CartItem, CartSummary
+-- contexts/  Cart, Auth, Wishlist, Compare, Theme
+-- lib/       products, vendors, categories, utils
+-- tailwind.config.js
+-- next.config.js
+-- package.json
```

## 4. 페이지별 역할

페이지	경로	설명
홈	/	히어로 배너, 인기 상품, 카테고리, 업체
상품 목록	/products	전체 떡 상품 + 카테고리/업체별 필터
상품 상세	/products/[id]	상품 정보, 판매 업체, 비교 링크
상품 비교	/compare	선택 상품 나란히 비교
장바구니	/cart	담은 상품, 수량 변경, 합계
주문서	/checkout	배송지 입력, 주문 요약
로그인/가입	/auth	로그인/회원가입 (localStorage)
마이페이지	/mypage	주문내역, 찜 목록, 회원정보
업체 목록	/vendors	입점 업체 카드 목록
업체 상세	/vendors/[id]	업체 정보, 해당 업체 상품
업체 등록	/vendor-register	업체 입점 신청 폼
소개	/about	떡담 서비스 소개
공지사항	/notice	공지사항 목록

## 5. Context 설계

Context	localStorage 키	역할
CartContext	tteokdam_cart	장바구니 CRUD, 수량 변경, 합계 계산
AuthContext	tteokdam_user	로그인/로그아웃/회원가입
WishlistContext	tteokdam_wishlist	찜 목록 추가/삭제
CompareContext	tteokdam_compare	비교 목록 (최대 4개)
ThemeContext	tteokdam_theme	라이트/다크 모드 전환

## 6. Tailwind 커스텀 테마 설계

### 색상 (전통 한국 + 모던)

이름	색상값	용도
cream	#FFF8F0	미색 (배경)
rice	#F5EDE0	쌀색 (보조 배경)
chestnut	#5C3A1E	밤색 (주요 텍스트)
chestnut-light	#8B6242	밤색 밝은 (보조 텍스트)
redbean	#8B3A3A	팡색 (포인트)
sesame	#2C2C2C	참깨색 (다크 텍스트)
mugwort	#4A7C59	깻색 (자연/신선)
honey	#D4A847	꿀색 (CTA 버튼, 강조)
injeolmi	#E8D5B7	인절미색 (카드 배경)
songpyeon	#7BA784	송편색 (보조 강조)
caramel	#C4956A	캬라멜 (테두리, 호버)

### 폰트

- font-display: 'Noto Serif KR' — 제목/강조 (전통 세리프)
- font-body: 'Noto Sans KR' — 본문/UI
- font-accent: 'DM Sans' — 영문/숫자 보조

## 7. 구현 단계 (Phase별 서브에이전트 투입)

---

### ▶ Phase 1: 프로젝트 초기 설정

- 총괄: Next.js 프로젝트 생성 (create-next-app)
- 디자인 Agent: tailwind.config.js + globals.css
- 백엔드 Agent: lib/ 데이터 + contexts/ + next.config.js

>> 디자인 +

### ▶ Phase 2: 기반 컴포넌트

- 프론트 Agent 1: ui/ 컴포넌트 (Button, Input, Card...)
- 프론트 Agent 2: layout/ 컴포넌트 + app/layout.js

>> 두 프론트

### ▶ Phase 3: 홈 + 상품 페이지

- 프론트 Agent 1: home/ 컴포넌트 + 홈 페이지
- 프론트 Agent 2: product/ 컴포넌트 + 상품 목록/상세

>> 두 작업 분할

### ▶ Phase 4: 업체 + 장바구니 + 비교

- 프론트 Agent 1: vendor/ 컴포넌트 + 업체 페이지
- 프론트 Agent 2: cart/ + 장바구니/체크아웃/비교

>> 두 작업 병행

### ▶ Phase 5: 나머지 페이지

- 프론트 Agent: 인증, 마이페이지, 소개, 공지사항

### ▶ Phase 6: 통합 테스트

- 테스트 Agent: 빌드 → 오류 수집 → 수정 → 재빌드 → 성공

### ▶ Phase 7: 커밋 & 푸시

- 총괄: 최종 git commit + push

## 8. 검증 방법

---

1. npm run build 성공 확인 (빌드 에러 0)
2. npm run dev로 모든 페이지 접근 가능 확인
3. 장바구니 추가/삭제, 비교 목록, 짐 기능 동작 확인
4. 모바일 반응형 확인

## 5. 다크모드 전환 확인