

1. 핵심 사용자 플로우 테스트

- 장소 탐색 → 목록 → 지도 → 외부 지도 딥링크
- 동선: 필터/검색으로 장소 찾기 → 리스트에서 장소 선택 → 지도에서 마커/프리뷰 확인 → 버튼 클릭 시 네이버/카카오 앱/웹으로 정상 이동.
- PC/모바일 차이:
 - PC: 웹 지도 새 탭 오픈이 정상 동작하는지.
 - 모바일: 앱 딥링크 시도 후 700ms 내/후에 웹 fallback 잘 되는지, 자동 리다이렉트 없이 반드시 탭 이벤트에서만 발생하는지.
- placeId / 좌표 fallback:
 - place_id/naiver_place_id 있는 경우와 없는 경우 모두 테스트.
 - 좌표 기반 링크 생성 시 주소/좌표가 깨지지 않는지.

2. LLM 자연어 검색 플로우 (TopSearchBar, /api/search)

- LLMQuery 스키마 매핑 검증
 - region, subRegion, categoryMain, categorySub, excludeCategoryMain, keywords, constraints, sort 값이 의도대로 파싱되는지.
 - “용산 혼밥 맛집”, “강남 디저트 조용한 곳” 등 대표 쿼리로 실제 필터링 결과 확인.
 - 검색 파이프라인 규칙 준수
 - LLM이 장소를 생성하지 않고 실제 locations 테이블 데이터만 반환하는지.
 - 결과 없음일 때: 빈 배열 + “용산에는 아직 등록된 곳이 없어요” 유형의 메시지가 올바르게 노출되는지.
- 정렬 로직
 - 태그/키워드 매칭 시 태그 매칭 > 평점 우선 정렬이 제대로 반영되는지.
 - 평점만으로 정렬되는 잘못된 케이스가 없는지 확인.
- 성능/로깅
 - search_logs에 query, parsed, result_count, llm_ms, db_ms, total_ms가 제대로 저장되는지.
 - LLM 응답 실패/타임아웃 시 UI 피드백(로딩, 에러 토스트 등)이 적절한지.

3. 필터와 카테고리/지역 캐스케이드 (FilterSection, 태입 location.ts)

- 지역 계층
 - Province → Region → SubRegion의 단계적 선택이 올바르게 동작하는지.
 - 새로 추가된/옛지 케이스 지역(예: “인천”, “경기 외곽”) 선택 시도 시도.
- 카테고리 계층
 - CategoryMain / CategorySub 선택 흐름이 UX 상 헷갈리지 않는지.

- `categorySub`에서 `categoryMain`을 추론하는 로직이 문제 없이 동작하는지 (`getMainFromSub` 류).
- **제외 카테고리 / 태그 필터**
- `excludeCategoryMain`, `eventTags`, 일반 태그 필터를 조합했을 때 예상대로 결과가 줄어드는지.
- **필터와 LLM 검색 모드 전환**
- `isSearchMode` 플래그로 필터 모드 ↔ 검색 결과 모드 전환이 명확하게 작동하는지 (UI 전환, 상태 초기화).

4. 지도 연동 (`Map.tsx`)

- **마커 렌더링**
- `categoryMain`별로 올바른 마커 SVG (`rice_marker.svg`, `beef_marker.svg` 등)가 매핑되는지.
- `categoryMain`이 없거나 레거시 `category`만 있는 데이터에 대한 fallback 동작.
- **마커 인터랙션**
- 마커 클릭 시 `PlacePreview`/`PlaceDetail`과의 상호작용(선택 상태, 하이라이트, 스크롤 연동) 테스트.
- 현재 선택된 장소가 바뀔 때 지도 중심/줌 레벨이 자연스럽게 이동하는지.
- **반응형 동작**
- 모바일에서 지도 영역 높이, 바텀시트 (`BottomSheet`) 오픈/닫힘에 따라 지도 리사이즈가 제대로 되어서 마커가 깨지지 않는지.

5. 장소 CRUD 플로우 (`AddLocationModal`, `LocationCard`, `supabase.ts`)

- **장소 추가**
- 필수 필드(이름, 주소, 좌표, 카테고리 등) 미입력 시 검증/에러 메시지.
- 작성 후 실제 `locations` 테이블에 데이터가 잘 들어가고, 새로고침 없이 리스트/지도에 반영되는지.
- `features`, `tags`, `price_level`, `event_tags`, `visit_date` 등 옵션 필드가 Supabase JSONB/배열 ↔ 앱 타입 간에 정상 매핑되는지 (`snake_case` ↔ `camelCase`).
- **장소 수정/삭제**
- `LocationCard`에서의 편집/삭제 UI.
- 삭제 시 연관 리뷰(`reviews`) 처리 전략(남겨두기, 연쇄 삭제, 필터링 제외 등)이 의도대로 구현되었는지.
- **데이터 일관성**
- 이미지 URL 필드: `image_url` vs `imageUrl` 둘 다 처리되는지.
- `category_main`, `category_sub`, `province`, `region`, `sub_region` 간의 불일치 케이스가 없도록 검증.

6. 리뷰 플로우 (AddReviewModal, CommunityReviews, reviews 테이블)

- **리뷰 작성**

- `one_liner, visit_type, features`(조용한, 혼밥, 데이트 등) 입력과 검증.
- `location_id` 연관성이 올바른지, 다른 장소에 잘못 붙지 않는지.

- **리뷰 표시**

- 최신순 정렬, 일부만 표시 후 “더보기” 패턴이 있다면 동작 확인.
- 장소 삭제/비활성화 시 리뷰 노출 전략 (숨김, 그대로 등).

- **Features 매핑**

- 장소의 `features`와 리뷰의 `features`가 구분되어 있는지, UI에서 헷갈리지 않게 잘 표현되는지.

7. 이미지 업로드 (ImageUpload, Cloudinary 연동)

- **업로드 플로우**

- 파일 선택 → 프리뷰 → Cloudinary 업로드 → 업로드 실패/성공 시 UI 반응.
- 큰 이미지/네트워크 느린 상황에서 로딩 인디케이터, 중복 클릭 방지.

- **최적화**

- 업로드 후 받아오는 URL에 리사이즈/포맷 파라미터가 붙어 있는지(Cloudinary 최적화).
- 리스트/지도/상세 모달에서 이미지 로딩 성능(레이아웃 시프트, skeleton 등).

8. 레이아웃 & 반응형

(MobileLayout, DesktopLayout, Sidebar, BottomSheet, useBreakpoint)

- **뷰포트 기준 전환**

- 768px 기준에서 모바일 ↔ 데스크탑 레이아웃 전환 시 UI 깨짐/상태 초기화 문제 여부.

- 브라우저 리사이즈 시 지도, 리스트, 바텀시트, 사이드바가 자연스럽게 재배치되는지.

- **바텀시트 제스처 (useBottomSheet)**

- 드래그로 열기/닫기/중간 단계가 부드럽게 동작하는지.
- 지도와 스크롤 제스처 충돌(스크롤이 안 되거나 지도가 움직이지 않는 문제) 여부.

- **스크롤 동작**

- 장소 리스트/리뷰 리스트의 스크롤이 모바일/데스크탑에서 잘 구분되어 작동하는지(바디 스크롤 잠금, 오버스크롤 이슈 등).

9. 이벤트/태그 기능 (EventBanner, EventTagFilter 등)

- **이벤트 태그 UI**
- 특정 `eventTags`를 가진 장소만 필터링하는 동작.
- 이벤트 배너 클릭 시 특정 필터/검색으로 연결되는지.
- **기간/노출 전략**
- 이벤트가 종료되었거나 태그가 없는 데이터에 대한 처리 (배너 비노출, 필터 비활성화 등).

10. 공유/외부 기능 (`kakaoShare.ts`, SNS 공유 등)

- **카카오/기타 공유**
- 공유 버튼 클릭 시 카카오톡/다른 플랫폼 공유가 실제로 열리는지.
- 모바일/PC 환경에서의 공유 플로우 차이, 팝업 차단 이슈.
- **URL & 메타데이터**
- 공유 시 포함되는 제목, 설명, 이미지가 의도한 것과 일치하는지.

11. 오류/엣지 케이스 & 회복력

- **네트워크 오류**
- Supabase, LLM API, Cloudinary 요청 실패 시 각각의 UI 처리.
- 재시도 버튼, 사용자 안내 문구 (“잠시 후 다시 시도해주세요” 등)의 명확성.
- **데이터 없는 상태**
- 지역/카테고리/검색 결과가 모두 0일 때 UX (비어 있는 상태, 추천 액션 힌트).
- **잘못된 데이터**
- 좌표가 잘못된 장소, 카테고리 없는 장소, 이미지 URL 깨진 경우 등에서 UI가 망가지지 않는지.

12. 성능 및 UX 품질

- **초기 로딩 시간**
- 코드 스플리팅/지연 로딩이 적용된 부분(지도, 상세 모달, LLM 검색 등)이 실제로 분리 로딩되는지.
- 초기 번들 용량, 첫 화면 렌더링까지 걸리는 시간 체감.
- **리렌더링 최적화**
- 지도/리스트/필터 상태 변화 시 불필요한 리렌더링이 있는지 (React DevTools Profiler 기반으로 확인해볼 가치 있음).
- **스크롤/애니메이션 부드러움**
- 모바일 환경에서 바텀시트 이동, 목록 스크롤, 지도 줌이 끊기지 않는지.

13. 로깅/분석 (`search_logs`, `click_logs`)

- **검색 로그**
- 각 검색 요청마다 `search_logs`에 올바른 JSON이 쓰이는지, `parsed` 컬럼 구

조가 LLMQuery와 일치하는지.

- **클릭 로그**
- 장소 클릭/지도 딥링크/리뷰 보기 등 핵심 액션이 `click_logs`에 기록되는지.
- `search_id`와 `location_id` 연관이 올바른지.