

전국 상조/장례 비교·견적·부고 모바일 웹 MVP — UI/UX 구현 명세 및 디자인 시스템

1. 디자인 토큰 / 컴포넌트 스펙 / 코드 스니펫의 필요 수준

- 예시 수준이 아니라 **실제 산출물로** 필요합니다.
- 이번 프로젝트는 기획 문서가 아닌, **개발·디자인이 그대로 참고하여 구현 가능한 기준 문서**를 만드는 것이 목표입니다.
- 아래 항목들은 **모두 실제 사용 가능한 형태로** 제공되길 원합니다.

필수 산출물

1. 디자인 토큰 표
 2. Color (Primary / Neutral / Semantic / CTA)
 3. Typography (Font size, weight, line-height, usage rule)
 4. Spacing / Radius / Shadow
 5. Light / Dark mode 모두 포함
6. 컴포넌트 스펙 시트
 7. Card, List, Filter, Input, Button, Modal, Bottom Sheet, Toast, Skeleton 등
 8. 사이즈, 여백, 상태(기본/비활성/에러/로딩), 접근성 속성 포함
9. UI 레이아웃 및 화면 규칙
 10. 모바일 웹 기준 IA 구조
 11. 화면별 레이아웃 원칙
 12. 빈 상태 / 오류 / 로딩 상태 가이드
13. **React + Tailwind** 코드 스니펫
 14. 핵심 컴포넌트 단위 예제 코드
 15. 실제 프로젝트에 바로 가져다 쓸 수 있는 수준

2. 업로드된 엑셀 파일의 역할 (없을 경우 포함)

- 현재 **도메인 관련 엑셀 데이터는 제공되지 않은 상황**입니다. (업로드된 파일은 해당 서비스와 무관한 샘플로 간주)
- 따라서 Deep Research에서 다음 내용을 **선제적으로 정의**해주길 바랍니다.
- 상조 상품 / 장례식장 데이터 **표준 스키마**
- 가격 항목 구조 (기본비 / 옵션비 / 변동비)
- 데이터 출처 및 신뢰도 표기 방식

- 추후 데이터를 엑셀로 정리할 예정이므로, **엑셀 템플릿(컬럼 정의)**까지 제안되면 좋겠습니다.
-

3. 우선 정리되길 원하는 산출물의 순서

이번 리서치에서 우선순위는 아래와 같습니다.

1. 모바일 웹 IA 및 사용자 흐름
2. 비교 → 상세 → 견적 → 상담
3. 부고 생성 → 미리보기 → 공유
4. 디자인 시스템 기초
5. 컬러 팔레트
6. 타이포그래피
7. 기본 레이아웃 그리드
8. 핵심 컴포넌트 스펙
9. 비교 카드
10. 견적 Step Form
11. 필터/정렬 UI
12. 부고장 템플릿 UI
13. 구현 기준 문서
14. 컴포넌트 상태 정의
15. 접근성 규칙
16. React + Tailwind 코드 스니펫

결과물은 기획서가 아닌 ‘구현 기준 문서’ 성격을 강하게 갖기를 바랍니다.

최종 목표 요약

- 모바일 웹 MVP 기준
 - 장례/상조 도메인 특성을 고려한 **차분하고 신뢰 중심 UI**
 - 개발자와 디자이너가 **추가 해석 없이 바로 구현 가능한 수준**
 - 문서 + 표 + 코드 스니펫이 함께 제공되는 결과물
-

프로젝트 개요 및 IA(Information Architecture)

본 MVP는 국내 사용자를 위한 상조 상품 및 장례식장 비교·견적·부고 생성 모바일 웹입니다. 사용자는 모바일 기기(화면 폭 375-430px 기준)에서 편리하게:

- 다양한 상조 상품(선불식 상조 등)과 장례식장 정보를 투명하게 비교하고,

- 자신의 조건에 맞춘 **예상 견적**을 계산,
- 가족장/무빈소 등에 맞는 **디지털 부고장**을 생성하여 공유할 수 있습니다.

내비게이션 구조: 하단의 탭 바로 4개 주요 섹션을 제공합니다. Material Design 가이드라인에 따라 **하단 내비게이션 바**에는 3~5개의 주요 목적지를 아이콘+레이블과 함께 배치하여 손쉽게 접근하도록 합니다 ¹. 각 탭은 아래와 같습니다:

- **비교:** 상조 상품/장례식장 **비교 목록** 화면 (기본 홈으로 간주).
- **견적:** 개인 조건 입력을 통한 **장례 비용 견적** 산출 화면.
- **부고:** **디지털 부고장 생성** 및 관리 화면.
- **가이드:** 장례 준비 및 상조 지식 등의 **콘텐츠 가이드** 섹션.

온보딩 절차: 최초 접속 시, 간단한 3가지 문항(지역, 장례 형태, 예산)을 통해 **사용자 맞춤 초기 설정**을 유도합니다. 이 온보딩 답변을 토대로 **비교 목록** 화면의 기본 결과를 개인화합니다. (예: 선택 지역 인근의 장례식장 우선 노출 등.) 온보딩은 **Progressive Disclosure** 원칙에 따라 단계별로 질문하여 사용자의 집중도를 높입니다 ².

주요 사용자 흐름:

- **상조/장례 비교 흐름:**
 - **비교 목록:** 상조상품 또는 장례식장 카드 리스트를 조회하고 필터/정렬 적용.
 - **상세 보기:** 특정 상품/시설 카드를 탭하여 상세정보 페이지 열람.
 - **비교 담기:** 리스트 또는 상세에서 최대 3개의 항목을 선택(add to compare).
 - **비교하기:** 하단 비교 바를 통해 선택한 항목들을 **비교 화면**으로 이동, 스펙과 비용 항목 비교.
 - **상담 연결:** 마음에 드는 항목의 상담 요청 버튼을 눌러 전화 연결 또는 폼 제출.
- **장례 비용 견적 흐름:**
 - **견적 메인:** 주요 변수 입력을 위한 **스텝별 폼** 시작 (Step 1).
 - **Step 1 – 기본 정보:** 지역, 장례 형태(가족장/일반장 등), 빈소 유무, 화장 여부 등 입력.
 - **Step 2 – 규모/기간:** 예상 조문객 인원, 장례 기간(일수), 운구 차량 등 **변동 요인** 입력.
 - **Step 3 – 옵션 선택:** 수의(예단), 제단 꽃 장식, 장례 용품 등 **선택 옵션** 선택. 실시간으로 하단 **요약 패널**에 비용 합계(최소-최대) 갱신.
 - **결과 출력:** 견적 완료 시 **가격 표**를 표시 – 기본비/옵션비/변동비 구분된 **상세 내역**과 합계, **신뢰도 배지** 및 출처/최신 정보 표시.
 - **상담 유도:** 필요 시 결과 화면에서 상담 신청 버튼.
- **디지털 부고장 생성 흐름:**
 - **부고 탭 진입:** 부고 리스트 및 “새 부고 작성” 버튼 노출.
 - **정보 입력:** 고인명, 상주명 및 관계, 장례 일시, 장소(빈소 주소 또는 “무빈소”), 연락처(선택) 등을 입력.
 - **공개 범위 설정:** **기본 비공개**(링크 아는 사람만)로 생성하며, **비밀번호 설정 옵션** 제공.
 - **템플릿 선택:** 3가지 스타일(미니멀/전통/무빈소 전용) 중 하나 선택 → 부고장 미리보기.
 - **공유:** 완성된 부고장을 링크로 공유하거나, 카카오톡 등 SNS에 전송. 공유 시 **접근 권한** 고려 (비밀번호 안내 등).
 - **만료 안내:** 부고장은 생성 후 30일 뒤 자동 삭제되며, 이에 대한 안내를 UI에 표시.
- **콘텐츠 가이드 열람 흐름:**
 - **가이드 메인:** 카테고리별 정보(장례 절차, 비용 절감 팁, 법령/제도 안내 등) 카드 또는 리스트 형태.

- **검색 및 필터:** 키워드 검색이나 주제별 필터로 원하는 정보를 찾기.
- **상세 콘텐츠:** 아티클 상세 페이지 - 본문과 함께 **출처, 최신 갱신일** 표시, 관련 기사 추천.
- **메타 정보:** 각 콘텐츠에 **작성자/검수자, 작성일/갱신일, 리뷰 예정일** 메타데이터를 포함. (예: “갱신됨: 2025-08-12, 다음 검토: 2026-08-12”)

위와 같은 IA 구조를 기반으로, **모든 주요 기능은 하위 메뉴 없이 탭으로 직접 접근** 가능하게 설계했습니다. 햄버거 메뉴 등은 지양하고, 중요한 섹션은 항상 노출함으로써 고령 사용자도 **직관적으로 이동**할 수 있습니다.

디자인 시스템 기초 (Design Tokens)

MVP의 UI는 **차분한 중립색 조합과 신뢰감을 주는 청록/남색 계열**을 바탕으로 합니다. 디자인 토큰은 Light/Dark 모드 모두를 고려하며, **KWCAG 2.2 AA 수준의 접근성**을 준수합니다.

컬러(Color Palette)

라이트 모드와 다크 모드에서 사용할 **컬러 팔레트**를 정의합니다. 저채도의 뉴트럴 컬러와 청남색 계열의 주요 색상, 그리고 CTA 강조색으로 구성했습니다. 모든 텍스트 대비는 최소 4.5:1 이상(소형 텍스트 기준)으로 설정하여 **명도 대비 WCAG 2.2 AA** 기준을 충족합니다 ³ ⁴.

Primary Palette (청남 계열 본색): 신뢰와 안정감을 주는 **남색 계열**을 메인으로 합니다. 예를 들면, Primary 700은 다소 어두운 **네이비 블루** 톤으로 헤더나 강조 텍스트에 사용하고, Primary 300은 이를 연하게 한 톤으로 백그라운드 강조 등에 활용합니다. Primary 계열은 너무 짙한 채도가 아닌 **약간 톤 다운된 블루/틸로** 선정하여, 장례 서비스의 점잖은 분위기를 표현합니다.

Neutral Palette: UI 전반의 배경, 텍스트, 디바이더에 쓰이는 **모노톤 회색 계열**입니다. Neutral 900은 주 텍스트(#212121 등 진한 회색 또는 거의 검정)로 사용하고, Neutral 100은 밝은 배경(#F3F4F6 등)으로 활용합니다. 이 팔레트는 밝은 회색부터 어두운 회색까지 8~10단계로 구성합니다 (예: 50, 100, 200, ..., 900). 라이트 모드에서 Neutral 900 텍스트 vs 흰 배경은 대비 21:1로 충분하며 ⁵, Neutral 500 아이콘 vs Neutral 50 배경 등도 3:1 이상으로 **비텍스트 요소 대비 요건(3:1)**을 충족시킵니다 ⁶.

Semantic Colors: Success / Warning / Error와 같은 상태 색상도 정의합니다.

- Success (성공/신뢰): 진중한 **초록 (#2E7D32)** 계열.
- Warning (경고): 부드러운 **주황 (#ED6C02)** 또는 **노랑 (#F9A825)** 계열.
- Error (오류): **빨강 (#D32F2F)** 계열.

모든 상태 색상의 명도 대비도 텍스트/배경 조합 시 충분히 확보합니다 (예: Error 빨강 배경에 흰 텍스트 대비 4.5:1 이상).

CTA Accent: 주요 콜투액션 버튼 등에 사용할 **고채도 강조색은 청록 (Teal)** 으로 지정합니다. 예를 들어 `--color-cta = #0F766E` (질은 청록). 이 색상은 Neutral/Primary 팔레트보다 포화도가 높아 눈에 띄지만, 충분히 어두운 톤이어서 흰 텍스트와 함께 사용 시 대비 약 5.47:1로 기준을 충족합니다. CTA 버튼의 텍스트는 **항상 흰색**으로 하여 시인성을 극대화하며, 색각 이상자가 구분할 수 있도록 **아이콘 + 텍스트 병행**도 고려합니다.

라이트 모드에서의 컬러 사용 규칙: 배경은 보통 Neutral 50~100 (아주 연한 회색 또는 흰색), 카드 배경은 흰색, 본문 텍스트는 Neutral 900 (거의 검정), 보조 텍스트는 Neutral 600 정도 (중간 회색)으로 설정합니다. Primary 700은 링크나 강조 텍스트, 또는 상단 앱 바 배경 등에 사용되고, CTA Teal은 주요 버튼과 하이라이트에만 제한적으로 사용됩니다 (일관성있는 사용으로 눈에 띄게).

다크 모드에서는 위 팔레트를 반전 활용합니다. Neutral 계열은 900이 배경에 가까운 어두운색(예: #121212)으로 쓰이고 100/50이 텍스트에 사용됩니다. Primary/CTA 색상은 **명도를 높여** 어두운 배경 위에서 충분한 대비 확보 (예: 네이

비->약간 밝은 블루, 청록->밝은 민트 등) 또는 **텍스트/테두리 스타일**로 활용합니다. 다크 모드에서도 모든 요소가 **3:1 이상 명도 대비**를 갖도록 하며 ⑥, UI 컴포넌트 경계 식별이 어려워지지 않게 그림자 대신 **밝은 경계선**을 추가합니다.

컬러 토큰 예시 (Light Mode 기준):

| Token | Light Mode 값 | 용도 | 대비 (예시) |
|----------------------------------|------------------|----------------|---------------------|
| <code>--color-primary-700</code> | #1E3A5F (예시) | 브랜드 주요색 (헤더 등) | 흰색 텍스트: 7:1 이상 (AA) |
| <code>--color-primary-300</code> | #90B4D4 (예시) | 강조 배경, 선택 상태 | 검정 텍스트: 4.5:1 이상 |
| <code>--color-neutral-900</code> | #212121 | 기본 텍스트 | 흰 배경: 21:1 (AAA) |
| <code>--color-neutral-600</code> | #6E6E6E | 보조 텍스트 | 흰 배경: 5:1 (AA) |
| <code>--color-neutral-100</code> | #F3F4F6 | 기본 배경, 카드 배경 | 검정 텍스트: 19:1 (AAA) |
| <code>--color-cta</code> | #0F766E (Teal) | 주요 CTA 버튼 배경 | 흰 텍스트: 5.47:1 (AA) |
| <code>--color-success</code> | #2E7D32 (Green) | 성공 배지/텍스트 | 흰 텍스트: 4.6:1 (AA) |
| <code>--color-warning</code> | #ED6C02 (Orange) | 경고 배지/텍스트 | 흑색 텍스트: 4.8:1 (AA) |
| <code>--color-error</code> | #D32F2F (Red) | 오류 상태 | 흰 텍스트: 5:1 (AA) |

(주: 정확한 HEX 값은 추후 조정 가능. 대비 수치는 WCAG 명도 대비 계산 기반 추정.)

Light/Dark 모드 전환 시에는 디자인 토큰만 교체하면 UI가 적절히 변하도록 설계합니다. 예컨대 `--color-neutral-900` (라이트: 검정에 가까움, 다크: 거의 흰색) 식으로 의미론적 토큰으로 유지합니다. 이렇게 하면 다크 모드에서 텍스트 색상이 자동으로 밝은색으로 적용되어 가독성을 유지합니다.

또한 **색상 사용 지침**: Primary/CTA 색상은 남용하지 않고, **주요 인터랙티브 요소**(버튼, 링크, 선택된 메뉴 등)에만 사용합니다. 넓은 면적 배경에는 Neutral 계열을 써서 **차분한 느낌**을 유지합니다. 강조를 위해 색상을 사용할 때도 **한 화면에 2~3종 이하로 제한**하여 정보 과부하를 막고 통일성을 높입니다.

마지막으로 **컬러 대비 검증**: 모든 텍스트/아이콘 색상 쌍은 디자인 단계에서 Contrast Checker를 통해 검증합니다. 예를 들어 작은 본문 글자는 4.5:1 이상, 18pt 이상 큰 텍스트는 3:1 이상 대비를 지키도록 합니다 ③. 또한 UI 구성 요소 (버튼 테두리 등)와 배경 간에도 3:1 이상의 대비를 확보하여 시력 저하 사용자도 식별 가능하게 합니다 ⑥.

타이포그래피(Typography)

기본 원칙: 고령층 사용자 읽기 쉽도록 **기본 본문 폰트 크기 17px**를 채택합니다 (일반적 웹 표준 16px보다 약간 큰 크기). 경우에 따라 18px까지도 고려하여 반응형으로 조정합니다. 이 크기는 **가독성 향상**에 유리하며, 적절한 행간과 함께 제공 시 편안한 독서를 지원합니다 ⑦ ⑧. 모든 텍스트 스타일은 design token으로 정의해 개발 시 바로 사용하도록 합니다.

폰트 패밀리: 한글 본문은 Noto Sans KR 또는 본고딕 등 **가독성 높은 산세리프 계열**을 사용합니다. 영문/숫자는 해당 폰트의 라틴 글립이나 Helvetica, Arial 등 기본 sans-serif 계열로 통일합니다. 숫자 표시는 **탭 간격 고정폭(tabular lining)** 기능을 적용하여 테이블이나 금액 표시 시 자릿수에 따른 들쭉날쭉함이 없도록 합니다. (CSS `font-variant-numeric: tabular-nums;` 사용)

타입 스케일: 시각적 계층을 위해 6~7단계의 헤드라인과 본문 스타일을 정의합니다. 예를 들면:

- H1: 24px, Bold (주요 화면 제목)
- H2: 20px, Bold (섹션 제목)
- H3: 18px, Semibold (카드 제목 등)
- Body (기본 본문): 17px, Regular
- Small (보조 설명): 15px, Regular
- Caption: 13px, Regular (아이콘 라벨 등 최소 크기)

각 텍스트 스타일에는 적절한 **line-height(줄높이)**를 적용합니다. 일반 본문은 **1.5배 이상**으로(예: 17px 폰트에 26px line-height) 설정하여 줄 간격이 너무 좁지 않게 합니다 ⁹. 이렇게 하면 가독성이 높아지고, 저시력자나 난독증 사용자의 **읽기 부담이 줄어듭니다** ⁹. 헤드라인은 디자인에 따라 1.2~1.4배 정도로 조금 타이트하게 할 수 있지만, 최소 1.2배 이상을 유지합니다. 또한 문단 간 여백(paragraph spacing)은 기본 line-height의 1.5~2배 정도를 추가하여 문단 구분을 명확히 합니다 ¹⁰.

글자 정렬 및 자간: 본문은 좌측 정렬을 기본으로 하고, **한 줄 당 40-60자 내외**로 길이를 제한해 가독성을 높입니다 (반응형 레이아웃에서 패딩을 통해 조정). 자간(letter-spacing)은 기본값을 따르되, Heading 등 대문자만 있는 경우 약간의 자간을 주어 읽기 편하게 조절합니다.

반응형 타이포그래피: 뷰포트 너비가 매우 작을 때(예: <360px) 본문 폰트를 16px로 약간 줄이고, 반대로 큰 태블릿에서는 18px로 키우는 식의 미세 조정을 할 수 있습니다. 그러나 가능한 **디자인 일관성 유지**를 위해 큰 변화는 없도록 합니다.

폰트 두께(weight): Regular(400), Medium(500), Bold(700) 세 가지를 주로 사용합니다. 본문 대부분은 Regular, 중요한 강조 어구나 버튼 레이블 등은 Medium 이상 사용합니다. 너무 많은 굵기 변주를 피하여 **일관성과 성능(폰트 파일 수)**을 확보합니다.

접근성 고려: 텍스트는 이미지로 넣지 않고 실텍스트로 제공하여 사용자 커스터마이징(폰트 크기 확대 등)이 가능하게 합니다 ¹¹. 또한 iOS/안드로이드 **접근성 설정(폰트 크기 변경)**을 해도 UI가 깨지지 않도록 200%까지 확대 시나리오를 테스트합니다 ¹². 키보드 포커스로 요소를 이동할 때, 스크린 리더를 위한 `aria-label` 등을 통해 텍스트 의미를 정확히 전달하도록 레이블링합니다.

공간 및 레이아웃 (Spacing & Layout Grid)

모바일에서 일관된 레이아웃을 위해 **4dp 기준 그리드** 시스템을 채택합니다. 즉, 여백(margin/padding)과 요소 크기는 4px 단위로 떨어지게 정의합니다. 이 미세단위를 바탕으로 8px, 12px, 16px 등의 간격을 조합하여 사용합니다 (Material Design도 8dp grid 권장 ¹³ 하지만 4dp 세분화로 유연하게 적용).

컨테이너 및 그리드: 기본 화면은 세로 스크롤 1열 레이아웃입니다. 화면 여백은 좌우 각 16px를 기본으로 두어 양옆에 충분한 패딩을 줍니다 (카드형 리스트 등에서 화면 양측에 닿지 않게). 주요 섹션 간 상하 마진은 24px 정도로 설정해 구분감을 줍니다. 작은 컴포넌트 간 간격은 8px~12px, 관련 그룹 내부 항목은 4px 간격 등 **계층적인 공간 규칙**을 정의합니다.

예시 공간 단위:

- 4px: 최소 간격 (아이콘과 텍스트 사이, 테두리 안쪽 패딩 등)
- 8px: 작은 컴포넌트 마진 (배지와 텍스트 간격 등)
- 12px: 중간 마진 (리스트 아이템 간격)
- 16px: 카드 내부 패딩, 모달 내부 패딩
- 24px: 섹션 간격, 모달 헤더-내용 간격 등

- 32px: 화면 상단 패딩 (status bar + toolbar margin), 큰 섹션 여백 등

컴포넌트 규격: 각 컴포넌트의 사이징도 표준화합니다. 버튼 높이는 44px (터치 타겟 최소 44px 준수 ¹⁴; WCAG2.2에서는 24px 이상 권장하지만 모바일에서는 44px 이상이 바람직)로 하고, 좌우 패딩 16px씩 줍니다. 입력 필드는 높이 48px, 좌우 패딩 12~16px. 카드 컴포넌트는 width 100% (좌우 16px margin 내장)이고, 내부 padding 16px, 코너 반경 **radius 12px**. 모달, 바텀시트의 코너는 radius 16px으로 조금 더 둥글게 해 상단 다른 화면과 구분합니다.

레이디우스(Radius): 전반적으로 **중간 정도의 둥근 모서리**를 적용합니다. 작은 요소 (버튼, 입력필드)에는 8px radius, 큰 요소 (카드, 모달)는 12~16px radius로 통일합니다. 지나치게 각진 디자인을 피하고, 또 너무 둥글지 않게 하여 **현대적이면서 차별한 인상**을 줍니다.

그림자(Shadow): 재질에 따라 2단계의 음영을 사용합니다:

- **레벨1:** 카드, 버튼 등 살짝 떠있는 요소 - `0 1px 3px rgba(0,0,0,0.1)` 정도의 작은 그림자.
- **레벨2:** 모달, 다이얼로그 등 최상위 오버레이 - `0 4px 12px rgba(0,0,0,0.15)` 정도로 더 크고 soft하게.

그림자는 다크모드에서는 **밝은 음영**(예: white 5% opacity)으로 변환하여, 어두운 배경에서도 부조효과를 비슷하게 냅니다.

반응형 및 Orientation: 본 MVP는 기본적으로 세로 포트레이트 모드를 가정하지만, 가로모드에서도 중요 컴포넌트가 한 줄에 여러 칸 나타나지 않도록 max-width 조정이 필요합니다. 또한 태블릿(예: iPad mini 등)에서 화면이 넓어지면 좌우 패딩을 늘리고 본문 폭을 제한(600px 이내)하여 너무 긴 문장이 되지 않게 합니다.

빈 화면(empty state) 및 오류 레이아웃: 콘텐츠가 없을 때는 기본 레이아웃 내에서 아이콘/이미지 + 안내 텍스트 + 필요한 액션버튼을 중앙 정렬해 표시합니다. 이때도 상하 padding과 정렬을 일관성 있게 (화면 위쪽 여백 25%, 아래 25% 등) 적용해 시각적 균형을 맞춥니다.

정리하면, **모든 UI는 4px 그리드에 맞춰 짜여진 여백과 크기를 가지며**, 이는 개발 시 Tailwind 등의 spacing 단위로 바로 쓰일 수 있게 준비됩니다. 예컨대 `p-4` (16px) 클래스, `px-4` (16px 좌우패딩) 등으로 대응됩니다. 이를 통해 개발자는 시안의 간격값을 해석하기 쉽고, 전체 UI의 **리듬감과 정렬**이 유지됩니다.

모션 & 인터랙션(Motion & Interaction)

MVP에서는 과도한 모션을 지양하고, **필요한 정도의 피드백 애니메이션**만 사용합니다. 인터랙션 모션은 모두 짧게 (150ms~200ms) 부드러운 이징(ease-out 또는 ease-in-out)으로 설정하여 **재빠른 응답**을 체감하게 합니다. 예를 들어:

- 버튼/카드 터치 시 살짝 scale up/down 혹은 배경색 변경: 100~150ms.
- 모달 오픈/닫힘: 200ms 이하 fade & slide up.
- 페이지 전환: 화면 좌->우 슬라이드 등 200ms 정도.

사용자의 **시각적 피로**를 고려해 모션은 절제하며, 동시에 **시스템 상태를 전달**하는 데 활용합니다. 예를 들어 로딩 중에는 스켈레톤이 빛나며(gradient shimmer) 1초 주기로 반복하는 등 너무 빠르지 않은 애니메이션으로 진행을 인식시킵니다.

접근성을 위한 모션 설정: OS의 “Reduce Motion(움직임 감소)” 설정을 감지하여(`prefers-reduced-motion: reduce`) 해당 사용자의 경우 모션 효과를 최소화하거나 아예 제거합니다. 모든 중요한 정보 전달은 모션 없이도 가능해야 하므로, 모션은 **보조적 역할**만 수행합니다. 예: 토스트 알림이 슬라이드로 나타나지만 reduce 모드에서는 페이드만 적용.

또한 포커스 이동시 **스무스 스크롤**로 대상이 화면에 보이게 유도하되 (예: `scroll-behavior: smooth`), 이 역시 `reduce-motion` 환경에서는 즉각 점프 이동하도록 합니다.

以上가 디자인 시스템의 기초 토큰과 스타일 가이드입니다. 다음으로, 이를 바탕으로 한 **핵심 컴포넌트별** 상세 스펙을 정의합니다.

핵심 UI 컴포넌트 스펙 시트

MVP 서비스에 필요한 주요 UI 컴포넌트의 규격과 상호작용, 접근성 속성을 명세합니다. 각 컴포넌트는 Design Token을 활용하여 일관성 있게 디자인되며, **다양한 상태(기본, hover, focus, disabled, loading 등)**에 대한 처리를 포함합니다. 또한 **React + Tailwind**를 가정한 코드 스니펫을 함께 제시하여 구현 레퍼런스로 활용합니다.

1. 카드(Card) 컴포넌트 - 상품/장례식장 목록

용도: 비교 목록 화면에서 각각의 상품 상품이나 장례식장 정보를 요약하여 표시하는 카드. 클릭 시 상세 페이지로 이동. 또한 카드에서 “비교 담기” 선택 기능 제공.

디자인: - 크기: 가로 풀(**width 100%**), 세로는 내용에 따라 변동되나 대략 120~140px. - **레이아웃:** 좌측에 썸네일 이미지(예: 업체 로고 또는 장례식장 사진) 64px 정방형, 우측에 텍스트 정보. 썸네일 모서리 8px radius로 살짝 둥글게. - **내용 구성:** 상단에 **이름(상품명/장례식장명)** - Bold, 16px. 그 아래 **지역/거리** 등 보조 정보 - 14px, Neutral 600. 주요 포함 서비스 요약 또는 특징점 **배지** 나열 - 예: “화장 포함”, “버스 지원” 등 짧은 키워드를 배지 형태로. 우측 아래에는 **총 예상비용** 표시: 최소-최대 금액 범위. 이 금액 텍스트는 강조색 (Primary 700)으로 16px Bold 표시하고, 신뢰도에 따라 옆에 배지 아이콘(예: 공공데이터 출처인 경우, 제휴사 데이터인 경우 등). 카드 우측 상단이나 하단에 **비교담기** 토글 버튼(체크박스 모양 아이콘버튼)을 배치합니다. - **여백:** 카드 내부 패딩 16px. 이미지와 텍스트 사이 12px 간격. 텍스트 블록 간 4px~8px 간격.

상태: - **기본:** 흰색 카드 바탕 + 그림자 레벨1 (아주 약한). - **터치 Hover/Active:** 모바일에서는 hover 대신 **터치 피드백**으로, 살짝 회색 배경 강조 또는 그림자 강도 증가. 예: `active:bg-neutral-100`. 또한 터치 시 안드로이드 물결효과(ripple)를 대체할 수 있도록 적절히 CSS로 효과 부여. - **선택됨(비교담기 체크):** 카드를 약간 하이라이트(배경 light primary 50 tint)하거나, 체크버튼이 채워진 아이콘으로 바뀝니다. 여러 개 선택 가능하므로, 선택된 카드 여러개가 표시될 수 있음.

접근성: - 카드 전체에 `<button>` 이나 `<a>` 로 role을 주어 키보드 포커스 가능하게 합니다. - 카드 내부 섹션을 스크린리더가 잘 읽도록, 예: 이름은 `<h3>` or `aria-label="상품명: ..."` 등, 가격은 `` 식으로 보강. - 비교담기 버튼에는 `aria-pressed` 속성으로 토글상태를 알립니다. 모든 아이콘에는 `` 로 텍스트 라벨을 제공 (예: "비교 목록에 추가").

React + Tailwind 예시 코드:

```
<button className="w-full bg-white rounded-lg shadow-sm p-4 flex items-start focus:outline-none focus:ring-2 focus:ring-primary-300">
  { /* 썸네일 이미지 */ }
  
  { /* 정보 콘텐츠 */ }
  <div className="flex-grow text-left">
    <div className="text-base font-bold text-gray-900">프리드라이프 가족장</div>
    <div className="text-sm text-gray-600 mt-0.5">서울 📍 거리 15km</div>
  </div>
</button>
```



```

    { /* 배지 목록 */ }
    <div className="mt-1 mb-2 space-x-1">
      <span className="inline-block bg-gray-100 text-gray-800 text-xs px-2 py-1 rounded-full">화장 포
      함</span>
      <span className="inline-block bg-gray-100 text-gray-800 text-xs px-2 py-1 rounded-full">버스 지
      원</span>
    </div>
    { /* 비용 및 신뢰도 */ }
    <div className="text-sm">
      <span className="font-bold text-primary-700">₩350만 ~ ₩420만</span>
      <span className="align-middle ml-1 text-[0]" aria-label="공공데이터 기반"></span>
    </div>
  </div>
  { /* 비교 담기 토글버튼 */ }
  <div className="ml-2">
    <input type="checkbox" className="hidden peer" id="compare-1" />
    <label for="compare-1" className="flex items-center justify-center w-6 h-6 border-2 border-
    primary-700 rounded focus:outline-none peer-checked:bg-primary-700 peer-checked:text-white"
    aria-pressed="false">
      <svg className="w-4 h-4 text-primary-700 peer-checked:text-white" fill="currentColor"
      viewBox="0 0 16 16" aria-hidden="true"><path d="M12.97 4.97a.75.75 0 0 0-1.06-1.06L7 8.81 4.53
      6.34a.75.75 0 1 0-1.06 1.06l3 3a.75.75 0 0 0 1.06 0l5.5-5.5z"/></svg>
      <span className="sr-only">비교 목록에 추가</span>
    </label>
  </div>
</button>

```

위 코드에서는 `<button>` 요소가 카드 전체를 clickable하게 감싸고 있습니다. 썸네일 이미지는 `object-cover`로 지정해 비율Crop을 처리하고, 텍스트 정보는 왼쪽 정렬로 쌓았습니다. 배지들은 inline-block으로 작은 pill 스타일을 구현했고, **비교담기**는 `<input type="checkbox">`와 `<label>`을 활용해 커스텀 체크박스를 만들었습니다. Tailwind `peer` 클래스를 이용하여 체크 상태 (`peer-checked`)에 따라 배경색과 아이콘 색이 변합니다. 이 컴포넌트는 React에서 map으로 다수 렌더링하면 리스트를 형성하게 되며, 키보드 접근성과 스크린리더 라벨을 모두 고려했습니다.

설명: 사용자가 카드를 탭하면 상세 페이지로 이동하며 (onClick 이벤트), 카드 우측 상단의 체크박스를 탭하면 **선택 상태**가 되고 하단 비교바에 반영됩니다. 두 가지 인터랙션을 모두 허용하므로 터치 영역을 분리하거나, 체크박스는 `<button>` 내부 별도 stopPropagation 등을 처리해야 합니다. UX 상으로는 카드 전체 탭 → 상세 이동, 체크 아이콘 탭 → 선택이 직관적입니다.

2. 필터 바(Filter Bar) 및 필터 칩(Chips)

용도: 비교 목록 화면 상단에 **필터 바**를 배치해 사용자가 원하는 조건으로 목록을 필터링할 수 있게 합니다. 또한 정렬 옵션도 함께 제공합니다. 필터 바는 horizontal scroll 가능한 **필터 칩**들로 구성되며, “필터” 버튼을 눌러 전체 조건 모달을 띄울 수도 있습니다.

디자인: - **필터 바 컨테이너:** 화면 상단 고정(sticky)되어 스크롤 시에도 상단에 남게 합니다. 배경은 흰색, 약한 bottom border (1px 중립색)로 분리. 높이 약 48px. - 좌측에 **검색 아이콘 + "장례식장 검색" 플레이스홀더**(필요 시), 우측에 **필터/정렬 버튼 아이콘**. 모바일 화면 폭을 고려해, 실제로는 검색창 대신 아이콘 혹은 작은 버튼으로 모아서 **모달로 검색창을 띄우는 방식** 가능. - **필터 칩들:** 주요 범주 (예: 지역, 장례 형태, 가격 범위, 시설규모 등)를 미리 가로 스크롤 칩으

로 보여줍니다. Material Design의 필터 칩을 참고하여, **다중 선택 가능한 칩**으로 구현합니다 15 16 . - 칩 스타일: 높이 32px, 좌우 패딩 12px, radius 16px(완경량), 글자 14px. - 기본: 배경 Neutral 100, 텍스트 Neutral 800. 선택됨: 배경 Primary 100 tint, 텍스트 Primary 700 (또는 배경 CTA색, 흰 텍스트 등 명확히). - 칩 내에 선택 시 X 아이콘 표시하여 개별 해제가 가능함. - **정렬 드롭다운**: 정렬은 보통 "정렬: 비용순 ▲/▼, 거리순, 평점순" 등을 선택하는 메뉴. 모바일에서는 드롭다운보다는 **모달**이나 **액션시트** 형태로 구현하는 것이 누르기 쉽습니다. 따라서 "정렬" 버튼 누르면 바텀시트로 옵션 리스트를 띄우는 디자인으로 권장합니다. 정렬 메뉴에서는 현재 선택된 기준을 체크 표시 등으로 나타냅니다.

상태: - 필터 칩: - 기본(Default): 위 설명같이 중립 스타일. - Hover/Active: (모바일의 경우) 터치되면 약간 더 진한 회색 배경으로 피드백. - Selected: 강조 색상 배경 + 강조 텍스트, 아이콘 (X) 노출. - 필터 바 자체: - 기본: 화면폭을 꽉 채워서 표시. - 사용자가 임의로 스크롤할 수 있게 overflow-x: auto 적용 (칩이 많을 경우). - 고정 시, 상단에 그림자나 경계선이 있어 content와 구분.

접근성: - 키보드로 Tab키를 통해 각 칩에 포커스 가능해야 하므로, 칩은 `<button role="checkbox" aria-checked="true/false">` 형태로 만들어 스크린리더가 멀티셀렉트임을 알 수 있게 합니다 17 . - 바텀시트 정렬 메뉴는 `<ul role="listbox">` 와 `<li role="option">` 으로 구성, 현재 선택은 `aria-selected="true"` . - 색상 외에 **텍스트 변화**(굵기나 아이콘 추가) 등으로 선택 상태를 표현하여 색 구분 어려운 경우도 인지하게 합니다. 예를 들어 선택된 칩은 앞에 체크 아이콘 추가 + 색상변경. - 필터바는 `aria-label="목록 필터 및 정렬"` 등으로 전체 영역 설명을 제공.

React + Tailwind 예시 코드 (필터 칩):

```
<div className="flex gap-2 overflow-x-auto whitespace-nowrap px-4 py-2 bg-white border-b border-gray-200">
  /* 필터 칩 예시들 */
  {filters.map(filter => (
    <button
      key={filter.id}
      onClick={() => toggleFilter(filter.id)}
      className={
        "shrink-0 rounded-full px-3 py-1 text-sm border " +
        (filter.selected
          ? "bg-primary-100 border-primary-300 text-primary-800"
          : "bg-gray-100 border-gray-200 text-gray-800")
        }
      role="checkbox"
      aria-checked={filter.selected}
    >
      {filter.label}
      {filter.selected &&
        <svg aria-hidden="true" className="inline ml-1 w-3 h-3" fill="currentColor" viewBox="0 0 16 16">
          <path d="M3.72 3.72a.75.75 0 0 1 0 1.06L8 6.939 13.22 3.22a.75.75 0 1 1 1.06 1.06L9.061 8 13.22 13.22a.75.75 0 1 1 1.06 1.06L8 16 3.72 13.22a.75.75 0 0 1 0 1.06z"/>
        </svg>
      }
    </button>
  ))}
  /* 정렬 버튼 */
```

```

<button onClick={() => openSortSheet()} className="shrink-0 ml-auto flex items-center text-sm text-gray-700">
  <svg className="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0 20 20" aria-hidden="true"><path
d="M6 10a.75.75 0 0 1 .75-.75h10.19l-2.72-2.72a.75.75 0 1 1 1.06-1.06l4 4a.75.75 0 0 1 1.06-1.06l-4 4a.
75.75 0 1 1-1.06-1.06l2.72-2.72H6.75a.75.75 0 0 1 1.06 1.06z"/></svg>
  정렬
</button>
</div>

```

위 코드에서 `filters` 배열의 각 항목을 칩 버튼으로 렌더링하며, `filter.selected`에 따라 클래스와 `aria-checked`를 동적으로 바꿉니다. 선택된 경우 칩 내부에 X모양 아이콘(SVG)도 보여주고 있습니다. 정렬 버튼은 우측에 `ml-auto`로 밀어 배치하고, 아이콘+텍스트로 구성했습니다. (정렬 아이콘은 예시로 오른쪽 화살표 사용)

필터 칩은 작아서 터치 타겟이 작을 수 있는데, padding을 8px 높이 주어 32px 높이를 확보했고 폭은 label 따라 유동적입니다. 이것으로 **터치 영역 최소 24x24px** 요건도 충족합니다 (실제 32px 높이) ¹⁴.

정렬 바텀시트는 별도 컴포넌트로 구현한다 가정하며, UI 상 간단히 설명하면:

- 상단에 "정렬 기준" 제목과 닫기 버튼.
- 리스트로 "비용 낮은순", "비용 높은순", "거리순", "평점순". 현재 선택 항목에 라디오버튼 체크표시 또는 강조.
- 선택 시 바로 적용 후 바텀시트 닫힘.

UX 패턴: 필터/정렬 기능은 **Nielsen Norman의 검색 및 필터 UX**에 따라, 사용자가 통제감을 갖도록 clear/reset 옵션과 현재 선택 필터를 명확히 보여주는 것이 중요합니다 ^{18 19}. 예컨대 필터를 여러 개 적용했으면 “필터 X개 적용” 같은 요약 배지를 상단에 보여주고 한번에 초기화 가능하게 합니다.

스폰서/제휴 등의 특별 표시도 이 영역과 연계될 수 있습니다 (예: “제휴업체만 보기” 토글). 스폰서 표시 원칙으로, **검색 결과 내 유료 광고/제휴 항목은 별도로 구분**하여 투명하게 표기합니다. 구글 검색 사례처럼 sponsored label을 강조하고 사용자가 원하면 숨길 수 있게 하는 것이 이상적입니다 ^{20 21}. 우리 서비스에서는 제휴사가 제공한 데이터일 경우 카드에 “제휴” 배지를 붙이되, **랭킹에는 가중치 반영 없이** 일반 항목과 섞여 보여줍니다. 필터를 통해 “제휴사만 보기”를 선택할 수 있으나 기본 정렬에는 영향을 주지 않습니다. 또한 가이드 콘텐츠나 상세 페이지에도 “제휴 혜택 안내” 등을 명확히 표시하여 상업적 영향과 정보를 분리합니다.

3. 비교 고정 바(Comparison Bar)

용도: 사용자가 두세 개의 항목을 “비교담기”로 선택하면, 화면 하단에 **비교 바**가 나타나 선택 현황을 보여주고 **비교 화면으로 이동**할 수 있게 합니다.

디자인: - 하단 고정(full width) 바, 높이 약 56px, 배경 Primary 700 (짙은 색) 또는 Neutral 800. 불투명도 95% 정도로 반투명하게 하거나 확실한 색으로. - 내용: “비교 항목 X개 선택됨” 텍스트(14px, 흰색) + 우측에 **비교하기 버튼**. - 선택된 항목이 2개 미만이면 바는 숨겨지고, 2개 이상부터 등장합니다 (최대 3개). 4개 이상 선택 시에는 3개까지만 비교 가능하다는 안내를 띄우거나 자동 3개까지만 허용. - 비교하기 버튼: 바와 일체형 느낌으로 우측에 둥근 모서리 버튼처럼 표현하거나, 텍스트로 “선택 항목 비교”를 명시. 이 버튼 영역을 넓게 잡아 쉽게 누르게 합니다.

상태: - Default: 화면 하단 붙어서 표시, 가려야 할 경우 (예: 모달 올라올 때) z-index 조정. - 나타날 때: fade/slide 애니메이션 200ms로 부드럽게 올라옴. - 사라질 때: 반대로 내려감. - 내부 버튼 기본 활성화 (선택 2개 이상 시 활성화), 2개 미만 시 disabled처리 및 바 자체 hidden.

접근성: - ARIA 라이브 영역으로 설정하여 (aria-live="polite") 스크린리더에게 선택 개수 변화시 알려줄 수 있습니다. 예: “2개 항목 선택됨. 비교 버튼을 눌러 비교하세요.” - 버튼에 `aria-label="선택한 2개 항목 비교하기"` 등 동적 라벨을 적용, 개수 포함. - 키보드 접근은 탭 순서 상 화면상 맨 뒤로 가겠지만, 고정 바 자체를 `tabindex="0"` 로 해서 바로 접근도 고려.

React 구현 고려: 전역 상태 (예: Zustand)에서 선택된 compare 리스트를 관리, length에 따라 바 컴포넌트를 show/hide. 비교하기 버튼 onClick 시 compare 페이지로 라우팅.

4. 견적 스텝폼(Step Form)

용도: 장례 비용 견적을 위해 여러 단계를 나눠 입력하는 폼. 사용자 입력 부담을 줄이고 진행 상황을 알려주는 것이 핵심입니다 18 .

디자인: - 한 화면에 하나의 Step씩 표시. 총 3단계. - 상단에는 **진행 바(progress indicator)**를 표시: “Step 1 of 3” 내지는 점 또는 선형 프로그레스. 여기서는 텍스트로 “1/3” 표시와 함께 가늘은 progress bar (예: 33%, 66%, 100%). 사용자에게 현재 위치와 남은 단계를 명확히 보여줍니다 18 . - 각 Step 화면 구성: - **Step 1: 기본정보** - select 또는 radio 입력들을 카드형태로. 예: 지역 선택 (시/도 드롭다운), 장례 형태 (라디오: 가족장/일반장 등), 빈소 사용 여부 (스위치), 화장 여부. - **Step 2: 인원/일수** - 숫자 입력 혹은 슬라이더: 예상 조문객 수 (ex. 50명), 장례 기간 (2일/3일/4일), 운구 차량 대수 등. 슬라이더에는 현재 값 라벨을 위에 표시. - **Step 3: 옵션 선택** - 몇 가지 optional service의 체크박스 리스트: 수의 선택 (표준/고급), 제단 꽃장식 (유/무), 장례 용품 패키지 등. 항목마다 대략 비용 범위 표시. - **요약 패널:** Step 3 화면 하단이나 별도 섹션에, 지금까지 선택된 항목들로 계산된 **예상 비용 합계**를 보여줍니다. “예상 총액: ₩ ~ ₩” 형식으로 실시간 업데이트. 그리고 그 아래에 “기본 비용 + 옵션 + 변동 비용 포함” 등의 설명과 신뢰도 배지를 표시. - **내비게이션 버튼:** 각 스텝 하단에 “다음” 또는 “이전” 버튼이 고정되어 있음. 이전은 회색, 다음은 CTA색. 마지막 Step3에서는 “견적 결과 보기”로 문구 변경. - 버튼 높이 48px, full-width. 두 버튼 있는 경우 좌우 50%씩 나누거나, 이전은 텍스트버튼로 처리하고 다음만 풀폭 등 UX고려.

상태: - **입력 검증:** 각 스텝에서 필수 입력 미기재 시 “다음” 버튼 disabled. 에러 발생 시 해당 필드 아래 빨간 에러메시지 12px 표시, 해당 step에서 진행 못하게. - **완료 화면:** Step3에서 최종 제출하면 결과 화면으로 넘어가며, progress indicator를 성공 아이콘 혹은 “완료”로 표시. - **중간 저장:** 한 번 입력한 값들은 state로 보존되어 이전단계 돌아가도 유지. (필요 시 localStorage에 임시 저장).

접근성: - Step 전환 시 focus 트래핑: 새 step의 첫 입력으로 포커스 자동 이동, 스크린리더에게 step 변경 알림(ARIA live region 활용 “2단계로 이동”). - 각 step 폼에는 `<fieldset><legend>` 를 사용해 그룹을 묶고 legend에 해당 섹션 제목을 넣어 구조적으로 전달. - 입력 컨트롤마다 `<label for>` 로 라벨 제공. 시각적으로 숨길 수는 있어도 스크린리더용 텍스트는 있어야 함. 예: 아이콘만 있는 toggle에도 sr-only label “빈소 사용 여부”. - 다음/이전 버튼에도 `aria-label` 로 현재 step 정보 부여 가능 (예: “다음 (2/3단계)”).

React + Tailwind 예시 코드 (Step2 일부):

```
// Step 2: 인원/일수 입력 JSX
<div>
  <h2 className="text-xl font-bold mb-4">Step 2 of 3: 규모 및 기간</h2>
  <div className="mb-6">
    <label htmlFor="numPeople" className="block text-sm font-medium text-gray-700">예상 조문객 수</label>
    <div className="flex items-center mt-2">
      <input id="numPeople" type="range" min="0" max="300" step="10" value={numPeople}
        onChange={...} className="flex-grow accent-primary-700 h-1.5 rounded-lg outline-none" />
      <span className="ml-3 text-sm text-gray-800">{numPeople}명</span>
    </div>
  </div>
</div>
```

```

</div>
</div>
<div className="mb-6">
  <label className="block text-sm font-medium text-gray-700 mb-1">장례 기간</label>
  <div className="space-x-3">
    {[ "2일", "3일", "4일" ].map(option => (
      <button key={option} onClick={()=>setDays(option)} className={`px-3 py-2 rounded-lg border $
{days===option ? 'bg-primary-50 border-primary-700 text-primary-800 font-medium' : 'bg-white
border-gray-300 text-gray-700'}`}>
        {option}
      </button>
    ))}
  </div>
</div>
<div className="mb-6">
  <label htmlFor="vehicles" className="block text-sm font-medium text-gray-700">운구 차량 대수</
label>
  <select id="vehicles" value={vehicles} onChange={...} className="mt-1 block w-full rounded-md
border-gray-300 focus:border-primary-500 focus:ring-primary-500 text-sm">
    <option value="0">없음</option>
    <option value="1">1대</option>
    <option value="2">2대</option>
    <option value="3">3대</option>
  </select>
</div>
{/* 네비게이션 버튼들 */}
<div className="flex justify-between mt-8">
  <button type="button" onClick={prevStep} className="text-primary-700 text-sm">이전</button>
  <button type="button" onClick={nextStep} className="bg-primary-700 text-white font-medium
px-5 py-3 rounded-lg disabled:opacity-50" disabled={!canProceed}>다음</button>
</div>
</div>

```

위 예시는 Step2 폼의 핵심 부분을 보여줍니다. 주문객 수는 range input + 실시간 값 표시, 장례 기간은 버튼 토크 (라디오 대응)로 구현, 운구 차량은 select로 구현했습니다. Tailwind `accent-primary-700` 을 통해 range thumb 색상을 지정했고, focus 스타일도 주의깊게 설정합니다.

이처럼 각 Step은 개별 컴포넌트로 만들어지고, `nextStep` 호출 시 현재 입력 검증을 통과하면 `currentStep++` 로 상태 업데이트 하여 다음 컴포넌트를 보여주는 식입니다. Multi-step form의 **Clear Progress Indicators** 원칙에 따라 사용자가 남은 단계를 인지하도록 Step 상단에 “Step X of Y” 텍스트와 progress bar를 두었습니다 18 .

신뢰도 배지 & 공식: 견적 결과 계산은 백엔드 로직이지만, UI에서는 **결과 화면에서 계산 공식의 신뢰도를 전달**합니다. 예컨대 “※ 본 견적은 최근 6개월간 공공데이터 평균을 기반으로 산정되었습니다” 같은 문구나, 각 비용항목 옆에 (공공 데이터) 배지, (업체제공) 배지, (추정) 배지를 붙입니다.

각 배지 디자인은 색상과 아이콘으로 구분: - **공공데이터 기반:** 녹색 체크마크 아이콘 + “공공”. - **제휴사 제공:** 파란 약수 아이콘 + “제휴”. - **사용자 제보:** 주황색 말풍선 아이콘 + “제보”. - **AI추정:** 회색 로봇 아이콘 + “추정”.

이 배지들은 `공공` 형태로 CSS로 스타일링하고, 툴팁 등을 제공하여 의미를 설명할 수 있습니다. 예: 배지에 `title="국가 공공데이터에 근거함"` 속성을 넣거나, Info 아이콘을 눌렀을 때 설명 모달.

공식 및 최소-최대 표시: 기본비, 옵션비, 변동비 각 섹션별로 **최소~최대 범위**를 표시하는 규칙: - 해당 항목의 가격 편차가 큰 경우 “WX ~ WY”로, 편차가 적거나 확정된 경우 “약 WX”, “고정 WX”. - 총합은 항상 범위로 표시. 예: “총 예상비용: ₩500만 ~ ₩650만”. - 만약 신뢰도가 낮은(데이터가 부족한) 항목은 범위를 넓게 잡고 “신뢰도 낮음” 아이콘(△) 표시. - 최소값 기준은 보수적으로(낮은 쪽), 최대값은 높은 쪽으로 책정해 **예산 초과 불만 방지**.

5. 가격 표(견적 결과 상세 내역)

용도: 견적 완료 후 보여지는 상세 **가격 내역 테이블**. 사용자가 입력한 조건에 따라 계산된 각 비용 항목을 기본비 / 옵션비 / 변동비로 구분해 보여줍니다.

디자인: - **섹션 구분:** “기본 비용”, “선택 옵션 비용”, “변동 비용” 세 개의 소제목으로 나누어 표시. 각 소제목은 bold 처리, 아래에 해당 항목 리스트. - **항목 표시:** 항목명 + 금액 범위 + 출처배지. 항목명 좌측에 아이콘(예: 관, 음식, 차량 등)으로 시각적 도움 가능. 금액은 오른쪽 정렬 또는 살짝 강조. - 예: - 기본 비용: 장례 기본 서비스 - ₩300만 (고정) 【배지: 공공】

빈소 임대료 - ₩0 ~ ₩100만 【배지: 추정】

- 선택 옵션 비용: 수의 - ₩50만 ~ ₩100만 【배지: 제휴】

꽃장식 - ₩20만 【배지: 제보】

- 변동 비용: 식사/음료 - ₩80만 ~ ₩120만 【배지: 공공】

버스 임대 - ₩0 ~ ₩30만 【배지: 추정】

- **토달:** 맨 아래 “총 합계: WXXX ~ WYYY”를 굵게 표시 (혹은 상단에 크게). 그 옆에 “(부가세 포함)” 등 안내.
- **출처 타임스탬프:** 각 배지 tooltip 또는 하단에 일괄 “가격 데이터 최신화: 2025-08-01 기준” 식으로 표기.

상태: - 항목이 없는 경우 (예: 무빈소라 빈소 임대료 0원) 해당 항목을 “-” 처리하거나 “해당 없음” 표시. - 오차 범위 큰 경우 색상 강조 (ex: ±50% 넘는 경우 글자 색 경고 등).

접근성: - 표 구조를 사용 `<table>`로 하면 스크린리더가 행/열 맥락을 잘 전달할 수 있습니다. 단 모바일에서는 표 스크롤이 어려울 수 있어 리스트로 구현해도 무방. - `<dl>` 정의리스트로 항목명-dt와 값-dd 쌍으로 만들어도 의미 전달에 유리. - 배지 아이콘에는 `aria-label="공공데이터 출처"` 등 숨겨진 라벨 제공. - 신뢰도 낮음 경고는 `role="alert"` 적절히 사용.

6. 부고장 템플릿 UI

용도: 3가지 스타일의 디지털 부고장 페이지 UI. 사용자가 입력한 부고 정보를 템플릿에 적용해 보여줌.

디자인: - **공통 요소:** 고인 사진(선택적), 고인 이름 및 신상정보(나이 등), 영결식 일시/장소, 상주 명단, 인사말. - **미니멀 템플릿:** 흰 배경에 검은 텍스트 중심 정렬. 불필요한 장식 없고, 모바일 화면에 잘 맞게 폰트 크기 가독성 있게 (본문 16px+, 제목 20px). 현대적인 느낌. - **전통 템플릿:** 전통 문양이나 색채(예: 먹색 배경에 한지 질감, 혹은 매화 문양 등)를 살짝 추가. 폰트도 명조계열 사용할 수 있음. 레이아웃은 미니멀과 비슷하나 스타일만 변화. - **무빈소 템플릿:** 빈소 정보나 조문 관련 내용을 제외한 간략 버전. “○○님 별세, 유족 ○○, 장례 미 실시(무빈소)” 등의 문구로 요약. 디자인은 미니멀에 가깝게. - 각 템플릿은 색상이나 배경 이미지 정도만 다르고 정보 구성은 동일하여, **테마만 변경**하는 형태로 구현 가능 (CSS 클래스 변경으로 적용).

상태 및 상호작용: - 부고장 생성 후 **미리보기 모드**에서 해당 템플릿을 보여주며, 상단에 “다른 템플릿 보기” 슬라이드 버튼들이 있어 좌우 스와이프 혹은 탭으로 3종 미리보기를 전환할 수 있습니다. - 공개 범위 설정이 “비밀번호 보호”인 경우, 미리보기에는 가려지지 않지만 공유된 실제 페이지 접근 시 **비밀번호 입력 모달**이 제일 먼저 뜹니다. 이 UI도 기본

제공. - 부고장 페이지 하단에는 “공유” 버튼과 “열람 현황” 버튼(예: 몇 명이 봤는지) - 이 건 생성자에게만 보이거나 별도 대시보드에서.

접근성: - 부고장 페이지는 시맨틱 마크업으로 구성해 스크린리더가 읽기 쉽게 합니다. 예: 고인명은 <h1>, 일시/장소는 <p>, 상주는 리스트. - 색상 대비도 충분히 (전통 템플릿의 배경색과 글자색도 4.5:1 이상) 유지합니다. - 공유 버튼은 <button>으로 aria-label="공유 옵션 열기".

개인정보/보안 설계: - 부고장은 기본 비공개로 생성되며, URL을 아는 사람만 접근 가능합니다. 리스트에서는 자물쇠 아이콘 등으로 비공개임을 표시하고, 사용자에게 “링크를 아는 사람만 접근 가능” 안내를 합니다. - **비밀번호 설정 옵션:** 사용자가 PW 설정 시, 공유시 반드시 비밀번호 함께 전달하도록 유도(UI에 “비밀번호: _____” 표시 복사 등). - 서버 측에서는 부고장 URL에 GUID를 사용해 추측 어렵게 하며, PW는 해시 저장. UI에서는 3회 PW 오류 시 캡차 등 제안 가능. - **만료:** 부고 생성 시 “본 부고장은 30일 후 (2025-06-01)에 자동 삭제됩니다” 문구를 명시 22 23. 부고장 페이지에도 하단에 “(게시 후 30일 경과 시 자동 삭제)”를 작은 글씨로 표기합니다. 이는 개인정보 최소 보존 원칙 준수 및 오래된 정보 노출 방지 목적입니다. - **열람 통계:** 생성자에게는 몇 명이 이 부고장을 조회했는지 열람 수 통계를 제공합니다. UI에서 부고 리스트에 “조회 123회”처럼 표시하거나, 상세 관리 화면에서 그래프. 단, 열람자 개인 식별 정보는 수집하지 않고 총 횟수만.

공유 흐름: - 공유 버튼 누르면 **시스템 공유 시트**(Web Share API 사용) 또는 자체 모달로 카카오톡, 메시지 등 아이콘 목록을 띄웁니다. - 공유 메시지에는 부고장 URL과 함께 간략한 정보(고인명, 일시 등)를 텍스트에 담아주어 수신자가 내용을 알 수 있게 합니다. - 만약 PW 보호인 경우 “(PW: 1234)” 같이 자동 덧붙이는 옵션도 검토 (그러나 보안상 권장되지 않아, 별도 전달하라는 경고만). - **SNS 미리보기:** 카카오톡 등에서 링크 미리보기 잘 나오도록 Open Graph 태그 (og:title 등)에 고인명/일시 요약을 세팅.

7. 모달 & 바텀시트 (Modal/Bottom Sheet)

용도: 필터 상세설정, 정렬 선택, 공개범위 설정, 안내 메시지 등을 나타낼 때 사용.

디자인: - **모달(Dialog):** 화면 중앙 팝업, 배경 반투명 검은 오버레이. 작은 내용(공지, 확인요청)에 주로 사용. Radius 12px, padding 16px. - **바텀시트:** 화면 하단에서 솟아오르는 시트, 높이는 콘텐츠에 따라 유동적이나 전체 높이의 50~90% 차지 가능. 상단에 작은 drag bar 표시 (width 40px, height 4px, radius 2px, color #ccc) 있어 드래그로 닫기 가능. - 내용 예: 필터 옵션들 (체크박스 리스트 + 적용 버튼), 정렬 옵션 리스트, 부고 공유 방식 등. - 바텀시트에는 헤더바 혹은 제목 텍스트(예: “필터”)가 상단에 굵게 있고, 닫기(X) 아이콘 우측 상단 배치. - 애니메이션: 바텀시트 open 시 0.3s ease-out 위로 슬라이드, close 시 반대.

상태: - 열릴 때 배경 스크롤 고정 (body scroll lock). - 포커스 트랩 활성화: 모달 내부 첫/마지막 요소에서 shift+Tab 순환. - Escape 키 or 오버레이 클릭 시 닫힘 (예외: 필수 입력 상황 모달이면 막음). - 바텀시트의 drag bar를 위로 당기면 전체화면 확장, 아래로 당기면 닫기 (선택 구현).

접근성: - ARIA: <div role="dialog" aria-modal="true" aria-labelledby="modalTitle"> 구조 사용. - 열리면 aria-hidden으로 배경 콘텐츠 비활성. - 첫 포커스는 모달의 닫기버튼으로 보내거나, 폼의 첫 필드로. - 스크린리더에게 모달 열렸음을 알리기 위해 aria-live를 쓰거나, 그냥 focus 전환으로 충분. - 닫기 버튼은 <button aria-label="닫기">xx</button> 이런 식. - 모달창 크기가 화면에 다 안보이면 내부 스크롤 허용하고, 포커스가 안 보이는 영역으로 가지 않도록 관리.

코드 스니펫 (바텀시트 구조):

```
{/* Overlay */}
<div className="fixed inset-0 bg-black bg-opacity-50" onClick={onClose} aria-hidden="true"></div>
{/* Bottom Sheet */}
```



```

<div role="dialog" aria-modal="true" aria-labelledby="sheetTitle" className="fixed inset-x-0
bottom-0 bg-white rounded-t-2xl shadow-lg max-h-[90%] overflow-y-auto focus:outline-none">
  <div className="px-4 py-2 flex items-center justify-between">
    <span id="sheetTitle" className="text-lg font-semibold">필터</span>
    <button onClick={onClose} aria-label="닫기" className="p-2 text-gray-500">
      <svg className="w-6 h-6" fill="currentColor" viewBox="0 0 20 20"><path d="M4.293 4.293a1 1 0
011.414 0L10 8.586l4.293-4.293a1 1 0 011.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10
11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z"/></svg>
    </button>
  </div>
  <hr/>
  <div className="p-4">
    { /* 필터 내용 예시: 체크박스 리스트 */ }
    <div className="mb-3">
      <label className="inline-flex items-center">
        <input type="checkbox" className="form-checkbox text-primary-600" checked={filterA}
onChange={...} />
        <span className="ml-2 text-gray-800">가족장만 보기</span>
      </label>
    </div>
    <div className="mb-3">
      <label className="inline-flex items-center">
        <input type="checkbox" className="form-checkbox text-primary-600" checked={filterB}
onChange={...} />
        <span className="ml-2 text-gray-800">화장 가능한 곳</span>
      </label>
    </div>
    { /* ... more filters ... */ }
    <button onClick={applyFilters} className="mt-4 w-full bg-primary-700 text-white py-3 rounded-
lg">적용</button>
  </div>
</div>

```

위 스니펫은 바텀시트의 구조를 보여줍니다. Overlay 클릭 시 onClose로 시트 닫기. `aria-labelledby`로 제목을 연결하고, 내용 부분에 필터 체크박스를 나열했습니다. Checkbox에는 Tailwind의 `.form-checkbox` 클래스를 사용했는데, 필요에 따라 custom 스타일링 가능합니다.

8. 토스트/스낵바 (Toast/Snackbar)

용도: 사용자에게 잠시 나타났다가 사라지는 메시지로 피드백을 제공. 예: "저장되었습니다", "오류: 네트워크 확인 필요" 등.

디자인: - 화면 하단(또는 상단) 중앙에 작게 뜨는 바 형태. - 배경: 검정에 가깝게 (혹은 Primary 색상), 투명도 80% 정도, 텍스트 흰색. - 패딩: 좌우 16px, 높이 40px 정도. radius 8px. - 아이콘: 성공일 경우 ✓ 체크, 오류 시 ⚠️ 느낌표 등 왼쪽에 배치. - 그림자 약간 추가로 배경과 분리. - 여러 토스트 메시지가 동시에 있으면 스택 쌓지 않고 최신으로 교체하거나, 최대 2개까지만 쌓고 위로 밀려올라가도록.

동작: - 나타날 때 fade-in + 약간 위로 이동 (y축 10px → 0) 0.3s. - 3~5초 후 자동 사라짐 (또는 사용자 스와이프로 즉시 해제 가능). - 새 토스트가 오면 기존 것 밀려나거나 교체.

접근성: - role="status" 또는 role="alert" 사용: - 일반 정보에는 status (읽긴 하지만 인터럽트하지 않음), - 오류는 alert (스크린리더 즉시 공지) ²⁴. - 중요한 정보는 화면에 잠시라도 포커스 받게 하거나 `aria-live="assertive"` 로 설정. - 시각적 파악 어려운 경우 대비, 가능하면 토스트 내용은 **별도 텍스트 영역에도 로깅** (예: 화면Reader-only 영역에 마지막 메시지 유지) - 이는 개발 여력에 따라.

9. 스켈레톤 로딩(Skeleton Loading)

용도: 데이터 로딩 중에 콘텐츠의 대략적 뼈대를 회색 블록으로 표시하여 사용자에게 **컨텐츠 자리**를 미리 보여줌으로써 체감 대기시간을 줄입니다 ²⁵.

디자인: - **리스트형 스켈레톤:** 예를 들어 비교 목록 카드 3개 자리에 회색 네모블록들. - 썸네일 자리: 연한 회색 사각, - 텍스트 줄: 한 카드에 3줄 정도 긴 사각으로. - **상세페이지 스켈레톤:** 제목 큰 사각, 이미지 공간 사각, 본문 여러 줄 등. - **애니메이션:** 진한-연한 회색 그라데이션이 왼->오 흐르는 shimmer 효과 (CSS keyframes) ²⁵. 너무 빠르지 않게, 1.5초 주기.

상태: - 데이터 느리게 올 경우 스켈레톤 유지, 오면 fade-out과 실제 콘텐츠 fade-in 교체. - 만약 에러로 콘텐츠 못 불러올 땐 스켈레톤을 에러 메시지로 전환.

접근성: - 스켈레톤에는 `aria-hidden="true"` 하고, 스크린리더용으로는 로딩 상태를 텍스트로 별도 제공 (ex: 라이브영역에 "목록 정보를 불러오는 중..."). - 너무 오래 로딩되면 (e.g. >8초) 스크린리더에 재공지 또는 대안 경로 안내.

10. 배지(Badge) 컴포넌트

용도: 짧은 상태나 속성을 나타내는 레이블. 예: 제휴, 공공, 신뢰도, 새 소식 등.

디자인: - 작은 **pill 형태** (라운드 풀) 배경에 아이콘+텍스트. 높이 약 20px, 폰트 11~12px. - 색상과 아이콘으로 종류 구분: - 제휴: 파랑 배경 + 아이콘 + 흰 텍스트 "제휴". - 공공: 녹색 배경 + 아이콘 + 흰 텍스트 "공공". - 제보: 주황 배경 + + 흰 텍스트 "제보". - 추정: 회색 배경 + (예측) + 검정 텍스트 "추정". - 페이지 내 여러 요소 옆에 붙을 수 있으므로 너무 튀지 않게, 채도는 중간 정도.

상태: - 배지는 static하게 표시, 클릭되지 않음 (정보 아이콘이 따로 있으면 클릭 가능). - 배지에 마우스오버 시 tooltip으로 상세 설명 가능 (desktop에서는, mobile에선 터치 없음).

접근성: - 색/아이콘 조합이라도 텍스트로도 내용 전달 (배지 자체에 텍스트 "공공" 등 있음). - `aria-label` 또는 screen-reader 전용 설명을 추가해 "공공: 공공데이터 기반" 등 풀어서 제공 가능.

이상 주요 컴포넌트의 디자인과 스펙을 정의했습니다. 각 컴포넌트는 디자인 토큰에 의존하도록 설계되어, 색상/폰트 변경 시 전체적으로 일관되게 적용됩니다.

데이터 및 신뢰성 표기 (Data Sources & Trust UI)

본 서비스는 **데이터의 출처와 최신성**을 투명하게 공개하여 사용자 신뢰를 높이는 것이 중요합니다. 이를 위해 UI 차원에서 다음과 같은 **신뢰 장치**를 설계합니다:

- 데이터 출처 표기:** 모든 가격, 업체 정보 등에는 출처 레이블이나 배지를 표시합니다. 예를 들어 상조회사 순위나 비용 정보가 공정위 공시자료 기반이면 "출처: 공정위 2025-02"를 명시하고, 가격 견적의 각 항목마다 위에서 정의한 배지(공공/제휴/제보/추정)를 붙여서 **한눈에 데이터 신뢰도를 판단**할 수 있게 합니다. 사용자는 이 배지를 통해 어떤 데이터는 공식 기관발표, 어떤 것은 제휴사 제공인지 구분할 수 있습니다.

2. **최신성 타임스탬프:** 콘텐츠와 데이터에는 “최종 업데이트 날짜”를 명기합니다. 가이드 글의 경우 글 상단이나 하단에 “업데이트: 2025년 8월 12일”과 같이 표시해 정보의 시의성을 알립니다 ²⁶ ²⁷. 가격 데이터도 “가격 정보 기준일: 2025-07-01” 등을 표기합니다. 이러한 **콘텐츠 적시성 표시**는 사용자에게 정보의 최신 여부를 전달하고 신뢰를 높이는 데 기여합니다 ²⁸. 또한 정기 검수 필요한 콘텐츠는 메타데이터로 “다음 리뷰 예정: YYYY-MM-DD”를 내부적으로 관리하고, 해당 일 지나면 시스템에 알림 띄어 업데이트하도록 합니다.

3. **오류 신고 기능:** 사용자들이 잘못된 정보나 수정이 필요한 부분을 쉽게 신고할 수 있도록 UI를 제공합니다. 각 데이터 블록 옆에 “정보 오류 신고” 작은 링크를 두거나, 화면 하단에 “정정 요청” 버튼을 배치합니다. 누르면 간단한 폼 모달이 뜹니다:

4. 카테고리 선택 (예: “가격 정보 상이”, “연락처 오류”, “오탈자”, “기타”),
5. 상세 내용 기재 (텍스트박스),
6. (선택) 스크린샷 업로드나 증빙 파일 (가능하면).
7. 제출하면 감사 메세지 “신고해주셔서 감사합니다. 검토 후 반영하겠습니다.”를 토스트로 보여줌.

이 신고 UI는 **익명 제출** 가능하게 하고, 한번에 너무 많은 필드 요구하지 않습니다. 신고된 내역은 운영팀 콘솔에 전달되며, 처리 기간 SLA (예: 48시간 이내 확인)을 안내합니다. 안내 방법: 신고 버튼 옆 “*48시간 이내 검토” 툴팁 또는 FAQ에 명시.

1. **가격 범위 산정 근거:** 견적 결과 등에서 폭이 큰 가격 범위를 제시할 때, “왜 범위가 넓은지” 설명을 제공합니다. 예를 들어 “식사/음료 비용은 메뉴와 인원수에 따라 크게 달라질 수 있어 범위로 표시했습니다.” 같은 문구를 info 아이콘으로 제공. 사용자 입장에서 예상 비용의 **불확실성을 이해**하도록 돕습니다.

2. **신뢰 배지 요약:** 상세 페이지나 결과 페이지 하단에 배지 종류와 의미를 설명하는 legend를 둡니다. 예: 공공 데이터 (정부/공공기관 통계 기반), 제휴사 제공 (파트너사 직접 제공), 제보 (사용자/업계 제보), 추정 (서비스 알고리즘 추산). 이를 통해 배지 아이콘만 보고도 쉽게 유추 가능하지만, 정확한 의미를 누구나 확인 가능하게 합니다.

이러한 장치들은 **정부 사이트의 콘텐츠 신뢰성 표시** 사례를 참고한 것입니다. 예컨대 미국 정부 디지털 지침에서도 **콘텐츠의 게시일/업데이트일**을 명시하여 정보 최신도를 알리는 것을 표준으로 권장합니다 ²⁹ ³⁰. 우리는 이를 적극 도입하고, 더 나아가 **일반 사용자 참여(신고)**까지 유도함으로써 데이터 품질 향상 사이클을 구축합니다.

또한 **CMS 워크플로우**와 연계하여, 데이터와 콘텐츠의 생명주기를 관리합니다:

- **콘텐츠 작성-검수-승인:** 가이드 섹션 등의 콘텐츠는 관리자용 CMS에서 Draft -> Review -> Published 상태로 관리됩니다. UI에는 초안/비공개 글이 노출되지 않고, 승인된 콘텐츠만 사용자에게 보입니다.
- **메타데이터 필드 표준화:** 각 콘텐츠는 작성자, 검수자, 작성일, 수정일, 출처, 만료일 필드를 갖습니다. 일반 사용자는 작성자 대신 “운영팀 제공” 등으로 표시를 간소화할 수 있습니다.
- **표준 템플릿 사용:** 운영자가 새로운 가이드 글 작성 시 일정한 템플릿(예: “배경 - 본문 - 참고 링크”)을 따르게 하여 사용자에게 일관된 읽기 경험을 제공합니다. UI는 템플릿 구조대로 섹션 헤더 등을 스타일링합니다.
- **만료 콘텐츠 표시:** 만료일이 지난 콘텐츠는 자동으로 “(구 정보, 업데이트 예정)” 배너를 달거나 리스트에서 뒤로 밀려납니다. 예를 들어 법령 변경으로 글이 1년마다 재검토 필요하면, 1년 지나면 사용자에게 “최신 정보가 아닐 수 있음” 경고를 표시합니다.

성능 및 품질 (Performance & Core Web Vitals)

MVP 개발 시 **성능 예산(Performance Budget)**을 정의하여 초기 로딩과 상호작용 지연을 최소화합니다. 목표는 Core Web Vitals 지표에서 **모든 페이지 75퍼센타일이 '양호' 등급**을 만족하는 것입니다 ³¹. 구체적인 목표 수치는:

- **LCP (Largest Contentful Paint):** < 2.5초 (모바일, 4G 환경 기준) ³¹. 주요 콘텐츠(영역)의 로딩 완료 시점을 2.5s 이내로 설정. 이미지나 영영 큰 텍스트 블록 등이 늦게 뜨지 않도록 **critical 리소스 우선 로드**전략 적용. 예: 초기 화면에 필요한 이미지들은 프리로드, 나머지는 레이저로드.
- **INP (Interaction to Next Paint):** < 200ms (대부분 상호작용) ³². 즉 버튼 탭 -> 반응되는 시간 <0.2초로 채감될 것. 이를 위해 **초기 JS 번들 최소화**(페이지별 code splitting, 필요 기능만 로드)하고, heavy 작업은 Web Worker나 lazy 처리.
- **CLS (Cumulative Layout Shift):** < 0.1으로 시각적 안정성 확보 ³³. 레이아웃 점프를 막기 위해 이미지/영상 요소에 width/height 미리 지정, 폰트 swap 시 FOIT 방지(font-display: swap), 동적으로 삽입되는 배너나 목록이 있을 경우 placeholder 공간 확보.

성능 예산 구체안: - JS 초기 번들 < 150KB (gzipped). - 이미지 총합 < 300KB (첫 화면), 나머지 지연 로드. - CSS < 100KB (Tailwind JIT 사용 시 실사용만 포함하여 작게). - 폰트: 웹폰트 1~2종만, 서브셋 적용하여 < 100KB. 또는 OS 기본폰트 활용.

저속망 대응: - 네트워크가 느릴 경우 (Navigate Timing API나 NetInfo API로 감지) **스켈레톤 UI** + “네트워크가 느립니다” 안내를 토스트로 표시하거나, 텍스트-only 대체도 고려. - 또한 PWA 기능을 염두에 두고, 중요한 API 콜 실패 시 offline 화면(“현재 오프라인입니다. 캐시된 정보 보여주기/재시도”)도 디자인합니다.

로딩 패턴 최적화: - **Prefetch & Preload:** 사용자가 비교 목록을 본다면 그 다음 상세 진입이 예상되니, 해당 리스트의 첫 몇 개 상품 ID에 대해 상세 API를 미리 살짝 호출해두거나, 상세 페이지 JS chunk를 `<link rel="prefetch">` 해둡니다. - **Code Splitting:** React Router 기준 각 주요 route (비교, 견적, 부고, 가이드)별로 청크 분리. 초기에는 홈/비교만 로드, 나머지는 lazy load. - **이미지 최적화:** WebP/AVIF 포맷 사용 (지원 브라우저 많으므로). ``으로 기기별 해상도 대응. 썸네일 등 작은 이미지는 0.5x~1x로만 로드 (고해상 필요없음). - **캐싱:** 공공데이터 기반 정적 정보 (상조 업체 목록 등)은 build시에 JSON화하거나 CDN 캐시 활용. 사용자별 정보만 API 호출. - **메모리/연산 최적화:** 견적 계산 등은 단순 공식이므로 프론트에서 처리 가능, 속도 영향 미미. 혹시 복잡로직 시 WebWorker 사용.

Core Web Vitals 모니터링: - 런칭 후 **Real User Monitoring(RUM)** 도구를 붙여 Chrome의 field data 수집. LCP, INP, CLS를 크래시틱에 전송. - 구글 서치 콘솔의 Web Vitals 리포트도 주기적으로 확인. - 만약 특정 페이지에서 CLS spikes 발견시 UI 수정 (ex: 폰트 loading 순서). - INP 저하 원인은 FID->INP 전환에 대비해 미리 대응: main thread 블록 분석.

성능 지향 개발 가이드: - React에서는 불필요 리렌더 줄이기 (useMemo 등), - Tailwind JIT로 클래스 조합 최적화, - Lighthouse 100점 목표로 (이미지 사이즈 올바른지, 사용하지 않는 CSS 제거 등).

요약하면, **사용자는 앱 사용 시 딜레이나 버벅임 없이 부드러운 경험**을 하게 하는 것이 목표입니다. 2.5초 안에 콘텐츠 제공, 모든 인터랙션 즉시 반응, 화면 요소 예측 불가 움직임 없음으로 **UX 품질**을 담보합니다 ³¹ ³³.

측정 및 분석 (Analytics & Metrics Dashboard)

제품 개선을 위해 **UX 행동 데이터 수집**과 **성과 지표 대시보드**를 설계합니다. 개인정보는 익명/집계 수준으로만 활용하며, 중요한 이벤트들을 정의하여 퍼널/전환 분석에 사용합니다. **주요 측정 이벤트 시퀀스:**

1. **view_result** - 사용자가 비교 리스트 (검색 결과 화면)을 조회한 횟수. 매 리스트 노출 시 발생. 파라미터: 검색어/필터 조건, 결과 개수, etc.

2. **open_detail** - 특정 상품/시설 상세페이지 열람 이벤트. 파라미터: item_id, from_position (목록 내 순번), from_filter (필터조건) 등.
3. **start_quote** - 견적 스텝1 시작 이벤트. (사용자가 견적 탭 클릭 또는 온보딩 3문항 완료 후 자동 진입 등).
4. **quote_complete** - 견적 3단계 입력 완료하여 결과 본 횟수. 파라미터: 주요 선택 옵션(지역/형태 등) 기록.
5. **consult_submit** - 상담 연결/문의 이벤트. 사용자가 상담 폼 제출 또는 전화연결 버튼 클릭 시. 파라미터: item_id (상세에서 했다면), consultation_type (call/form).
6. **obituary_share** - 부고장 공유 이벤트. 파라미터: template_type, share_channel (kakao, link 등), whether_pw (비번설정여부).

이 외에도 콘텐츠 가이드 관련: - **guide_read** - 가이드 글 읽음 완료 (스크롤 90% 이상) 이벤트.

KPI 지표: - **전환율(Conversion Rate):** view_result 대비 consult_submit 비율. 이를 퍼널로 보면 **목록뷰** -> 상세뷰 -> 견적시도 -> 견적완료 -> 상담제출 흐름으로 전환율을 추적합니다. - **완료율(Completion Rate):** 견적 start 대비 complete 비율 (폼 이탈률로 UX 문제 진단). - **공유율(Share Rate):** 부고 생성 대비 실제 공유 실행 비율. - **활성도:** 일일/월간 활성 사용자 (DAU/MAU), 견적 도구 재사용률 등.

대시보드 설계: - **Funnel 차트:** 주요 전환 퍼널 단계를 시각화 (ex: 1000명 목록조회 -> 300명 상세 -> 100명 견적 -> 50명 상담). Drop-off 많은 구간 색상 강조. - **시계열(Line chart):** 일별 consult_submit 건수, quote_complete 건수 등 추이. - **누적 지표:** MAU, 누적 부고 공유 수 등 card 형태 표기. - **분석 예시:** 필터별 conversion 차이 (예: 특정 지역에서 상담률 높다면 강조), A/B 테스트 결과.

이벤트 명세: - 각 이벤트는 **명확한 스키마**로 로그됩니다. (예: `event: "consult_submit", item_type: "funeral_home", item_id: 123, user_id: anon, timestamp: ...`). - 이벤트들은 GA4, Amplitude 등 연동. 중요한 이벤트인 consult_submit은 GTM 커스텀 목표로 등록해 광고 효율 추적도.

개인정보 고려: - 로그인 없으므로 user_id는 세션ID 또는 익명ID. - 개인정보(전화번호 등) 입력하는 상담 폼 제출은 이벤트로 카운트만 하고 내용은 저장 안 함 (백엔드 email전달정도). - 쿠키 동의 받아야 하면, 분석 스크립트 로드 전에 opt-in 체크.

마지막으로, 이 분석 데이터는 **UI 개선에 피드백 loop**로 활용됩니다. 예를 들어, 견적 폼 Step2에서 이탈자가 많으면 UI를 개선하거나 도움말을 추가하는 등의 액션을 취할 것입니다.

以上으로 모바일 웹 MVP의 UI/UX 명세 전반 (IA, 디자인 시스템, 컴포넌트 스펙, 데이터 신뢰, 성능, 측정)을 모두 상세히 정리했습니다. 이 문서는 기획, 디자인, 개발팀이 **공통 기준으로 삼아 바로 구현에 착수할 수 있는 수준의 구체적인 가이드**를 제공하며, 향후 기능 확장과 유지보수에도 일관된 사용자 경험을 유지하는 기반이 될 것입니다.

모든 내용은 최신의 웹 접근성 가이드라인(KWCAG/WCAG 2.2)와 모범 디자인 시스템(Material, Fluent, GOV.UK 등) 참고를 통해 수립되었으며, 상조/장례라는 특수 도메인에 맞춰 **신뢰성과 사용자 배려**를 최우선으로 고려했습니다.

향후 이 명세에 따라 실제 디자인 작업(Figma 라이브러리 생성)과 React 컴포넌트 개발을 진행하면, 제품의 일관성 있는 경험과 높은 완성도를 담보할 수 있을 것입니다. 필요한 경우 본 문서를 기반으로 추가 UX 테스트와 피드백을 반영해 지속 개선해 나갈 계획입니다.

1 Bottom navigation - Material Design

<https://m2.material.io/components/bottom-navigation>

2 18 19 24 8 Best Multi-Step Form Examples in 2025 + Best Practices

<https://www.webstacks.com/blog/multi-step-form>

3 5 6 10 11 12 한국형 웹 콘텐츠 접근성 지침(KWCAG) 2.2

<https://a11ykr.github.io/kwcag22/>

4 7 8 9 Mastering typography in design systems with semantic tokens and responsive scaling | by Oluwatosin Obalana | UX Collective

<https://uxdesign.cc/mastering-typography-in-design-systems-with-semantic-tokens-and-responsive-scaling-6ccd598d9f21?gi=00ca67a441cb>

13 6 Items in Mobile App Navigation- Bad idea? : r/UXDesign - Reddit

https://www.reddit.com/r/UXDesign/comments/p71ggp/6_items_in_mobile_app_navigation_bad_idea/

14 Understanding Interaction to Next Paint (INP) | BrowserStack

<https://www.browserstack.com/guide/interaction-to-next-paint-inp>

15 Chips - Material Design

<https://m2.material.io/components/chips>

16 Material.Chip.Filter - material-components-web-elm 1.0.0

<https://package.elm-lang.org/packages/Dacit/material-components-web-elm/latest/Material.Chip.Filter>

17 Accessibility - Chips – Material Design 3

<https://m3.material.io/components/chips/accessibility>

20 21 New sponsored results label in Google Search

<https://blog.google/products/ads-commerce/google-search-sponsored-results-label/>

22 23 26 27 28 29 30 Content timeliness indicator | Federal website standards

<https://standards.digital.gov/standards/content-timeliness-indicator/>

25 Skeleton Screens vs. Progress Bars vs. Spinners (Video) - NN/G

<https://www.nngroup.com/videos/skeleton-screens-vs-progress-bars-vs-spinners/>

31 Largest Contentful Paint (LCP) | Articles | web.dev

<https://web.dev/articles/lcp>

32 Interaction to Next Paint (INP) | Articles - web.dev

<https://web.dev/articles/inp>

33 Cumulative Layout Shift (CLS) | Articles - web.dev

<https://web.dev/articles/cls>