리얼 벤치마킹 PBL

학생용 자기주도 워크북

Jumia · YouTube · Jiji의 성장 구조를 관찰하고
문서화 · AI 구현 · 비교 평가까지 완주하는 4개 미션

발행일 2026-10-27
대구 DIP SW 랜딩 PBL

목차

각 항목을 클릭하면 해당 장으로 이동합니다.

리얼 벤치마킹 PBL 학생 안내서 5

최종 목표 . . . 5
시작 전에 준비할 것 . . . 5
진행 순서 . . . 5
미션 통과 규칙 . . . 5
혼자 진행할 때의 작업 리듬 . . . 6
공통 안전 규칙 . . . 6
막혔을 때 . . . 6
참고용 완성 문서 . . . 6

미션 1. 실제 서비스의 성장 구조를 벤치마킹하라 8

1. 미션 카드 . . . 8
2. 서비스 하나 선택하기 . . . 8
3. 프로젝트 폴더 만들기 . . . 8
4. 관찰 규칙 익히기 . . . 9
5. 관찰 세션 3회 수행하기 . . . 9
6. 관찰 로그 10건 작성하기 . . . 10
7. AARRR 퍼널로 분류하기 . . . 10
8. 성장 루프 그리기 . . . 11
9. 실험 가설 5개 만들기 . . . 11
10. AI에게 받을 수 있는 도움 . . . 11
11. 제출물 . . . 12
12. 제출 전 자기점검 . . . 12
13. 미션 1 채점표, 25점 . . . 12
14. 공식 참고 링크 . . . 12

미션 2. 관찰 결과를 3종 문서로 역설계하라 14

1. 미션 카드 . . . 14
2. 시작 조건 확인하기 . . . 14
3. 먼저 근거 추적표 만들기 . . . 14
4. 문서 A: 제품 기회 정의 . . . 14
5. 문서 B: 비즈니스 모델 정의 . . . 15
6. 문서 C: 개발 기본설계 . . . 16
7. MVP 범위 자르기 . . . 16
8. 분석 이벤트 정의하기 . . . 16
9. AI로 세 문서 교차 검토하기 . . . 17

10. 제출물 . . . 17
11. 제출 전 자기점검 . . . 17
12. 미션 2 채점표, 25점 . . . 18
13. 참고 링크 . . . 18
미션 3. 문서를 기준으로 AI와 사이트를 구현하라 19
1. 미션 카드 . . . 19
2. 시작 전 환경 점검 . . . 19
3. 구현 기준선 고정하기 . . . 19
4. Next.js 앱 초기화하기 . . . 20
5. 구현 폴더 확인하기 . . . 20
6. 세 번의 수직 구현 사이클 . . . 20
7. AI 작업 요청 방식 . . . 21
8. AI 작업 로그 작성하기 . . . 22
9. 기능 테스트하기 . . . 22
10. 접근성·성능 점검하기 . . . 23
11. 고장 해결 순서 . . . 23
12. 제출물 . . . 23
13. 제출 전 자기점검 . . . 24
14. 미션 3 채점표, 30점 . . . 24
15. 공식 참고 링크 . . . 24
미션 4. 원본 서비스와 구현 결과의 차이를 증명하라 25
1. 미션 카드 . . . 25
2. 비교 전에 기준선 고정하기 . . . 25
3. 공통 대표 과업 만들기 . . . 25
4. 혼자 수행하는 3회 반복 테스트 . . . 26
5. 7개 영역 비교하기 . . . 26
6. 기술 지표를 같은 조건에서 측정하기 . . . 27
7. 사용성 휴리스틱 점검하기 . . . 27
8. AI의 판단을 평가하기 . . . 27
9. 다음 성장 실험 3개 결정하기 . . . 28
10. 비교 보고서 작성하기 . . . 28
11. AI로 결론의 과장을 검사하기 . . . 28
12. 최종 제출 패키지 . . . 28
13. 제출 전 자기점검 . . . 29
14. 미션 4 채점표, 20점 . . . 29
15. 참고 링크 . . . 29
리얼 벤치마킹 PBL 평가 루브릭 30

1. 평가 방식 . . . 30
2. 미션별 배점 . . . 30
3. 상세 루브릭 . . . 30
4. 감점과 통과 제한 . . . 32
5. 최종 통과 조건 . . . 32
6. 미션별 통과 게이트 . . . 32
7. 최종 자기평가표 . . . 33
8. 발표 체크리스트 . . . 33
관련 문서 . . . 33
기회 정의 34
1. 문서 목적 . . . 34
2. 기회 가설 . . . 34
3. 핵심 고객 . . . 34
4. 해결할 문제 . . . 34
5. 제품 약속 . . . 35
6. MVP 사용자 흐름 . . . 35
7. MVP 범위 . . . 35
8. 성공 지표 초안 . . . 36
9. 검증 계획 . . . 36
10. 결정이 필요한 항목 . . . 36
관련 문서 . . . 36
비즈니스 모델 정의 37
1. 모델 요약 . . . 37
2. 이해관계자별 가치 . . . 37
3. Business Model Canvas . . . 37
4. 수익 모델 가설 . . . 37
5. 핵심 단위 경제성 . . . 38
6. 운영 퍼널 . . . 38
7. 신뢰와 정책 원칙 . . . 38
8. 90일 검증 로드맵 . . . 38
9. Go / Pivot / Stop 기준 . . . 39
관련 문서 . . . 39
개발 기본설계 40
1. 설계 목표 . . . 40
2. 제안 기술 스택 . . . 40
3. 시스템 경계 . . . 40
4. 개발 폴더 구조 . . . 40

5. 화면 구성 . . . 41
6. 콘텐츠 모델 . . . 41
7. 상담 처리 흐름 . . . 42
8. 분석 이벤트 초안 . . . 42
9. 비기능 요구사항 . . . 42
10. 테스트 전략 . . . 42
11. 환경 구분 . . . 43
12. 기술 결정 기록 . . . 43
13. 구현 전 확인 목록 . . . 43
관련 문서 . . . 43
개발 작업 안내 44
시작 전 확인 . . . 44
작업 단위 . . . 44
경계 규칙 . . . 44
완료 정의 . . . 44
아직 생성하지 않은 항목 . . . 45

리얼 벤치마킹 PBL 학생 안내서

Jumia, YouTube, Jiji 중 하나를 선택해 성장 구조를 관찰하고, 문서로 역설계하고, AI로 구현한 뒤 원본과 비교하는 4단계 개인 프로젝트입니다. 교사의 추가 설명 없이도 아래 순서대로 진행할 수 있습니다.

최종 목표

> 성공한 서비스의 화면을 흉내 내는 데서 끝나지 않고, 성장을 만드는 사용자 행동을 근거로 설명해 새로운 웹서비스로 구현한다.

시작 전에 준비할 것

- Chrome 또는 Edge 브라우저
- 화면 캡처 도구
- Markdown을 편집할 수 있는 에디터
- Git과 Node.js LTS
- 코딩을 도와줄 AI 도구
- 총 16~24시간의 개인 작업 시간

설치가 어렵다면 미션 3의 환경 점검부터 확인합니다.

진행 순서

순서	미션	권장 시간	완료 결과	배점
1	성장 구조 벤치마킹	4~6시간	관찰 10건, 성장 루프, 실험 가설	25
2	3종 문서 역설계	4~6시간	제품·BM·개발설계 문서	25
3	AI 기반 사이트 구현	6~10시간	실행 가능한 사이트와 테스트	30
4	원본과 결과 비교	3~5시간	비교 보고서와 다음 실험	20

총점과 평가 방법은 평가 루브릭에 있습니다.

미션 통과 규칙

미션마다 마지막의 `제출 전 자기점검` 을 완료합니다. 통과 조건을 충족하지 못했다면 다음 미션으로 넘어가지 않습니다.

```
미션 1 근거 OBS-###
  → 미션 2 요구사항 PRD-### / 가설 BM-### / 결정 ADR-###
    → 미션 3 구현 파일 / 테스트 / AI-### 기록
      → 미션 4 비교 근거 CMP-### / 다음 실험 EXP-###
```

혼자 진행할 때의 작업 리듬

1. 한 단계의 설명을 읽습니다.
2. 제시된 파일에 직접 기록합니다.
3. AI에게 검토를 요청하되, 사실을 새로 만들지 못하게 제한합니다.
4. 자기점검 표에서 증거 파일 위치를 적습니다.
5. Git을 사용한다면 단계가 끝날 때 커밋합니다.

권장 커밋 예시:

```
docs: complete mission 1 benchmark evidence
docs: complete product business and engineering specs
feat: implement primary user journey
docs: compare benchmark and implementation
```

공통 안전 규칙

- 공개 화면과 공식 공개 자료만 사용합니다.
- 인증 우회, 비공개 API 접근, 대량 자동 수집을 하지 않습니다.
- 원본 소스 코드, 상호, 로고, 문구, 이미지, 영상은 복사하지 않습니다.
- 구현물은 새로운 서비스 이름과 합법적으로 사용할 수 있는 자산을 사용합니다.
- 테스트에 실제 개인정보와 결제 정보를 입력하지 않습니다.
- AI가 말한 내용은 근거가 확인될 때만 사실로 사용합니다.

안전 규칙을 위반하면 점수와 관계없이 미통과할 수 있습니다.

막혔을 때

- 무엇을 분석해야 할지 모르겠다면 미션 1의 서비스별 추천 흐름 중 하나를 고릅니다.
- 문서가 너무 커지면 미션 2의 `MVP 범위 자르기` 규칙을 적용합니다.
- 코드가 실행되지 않으면 미션 3의 `고장 해결 순서` 를 그대로 따릅니다.
- 비교 기준이 모호하면 미션 4의 공통 과업표를 먼저 작성합니다.

참고용 완성 문서

아래 문서는 형식 참고용입니다. 내용을 그대로 복사하지 말고 자신이 선택한 서비스와 근거에 맞춰 다시 작성합니다.

- 제품 기회 정의 예시
- 비즈니스 모델 예시

- 개발 기본설계 예시
- 개발 작업 안내

미션 1. 실제 서비스의 성장 구조를 벤치마킹하라

1. 미션 카드

항목	내용
목표	공개 화면에서 사용자의 다음 행동을 만드는 장치를 찾아 성장 루프로 설명한다
선택 대상	Jumia, YouTube, Jiji 중 하나
권장 시간	4~6시간
배점	25점
입력	브라우저에서 직접 관찰한 화면과 공식 공개 자료
출력	프로젝트 폴더, 관찰 10건 이상, 퍼널, 성장 루프, 실험 가설 5건
완료 신호	모든 핵심 주장에 `OBS-###` 근거가 연결됨

여기서 그로스 해킹은 시스템에 침입하는 행위가 아닙니다. 사용자가 유입되고, 첫 가치를 경험하고, 다시 방문하거나 다른 사용자를 부르는 구조를 관찰해 실험 가설로 만드는 활동입니다.

2. 서비스 하나 선택하기

아래 질문에 가장 답하고 싶은 서비스를 선택합니다.

서비스	추천 질문	추천 대표 흐름	공식 시작 링크
Jumia	평점·후기·공식 스토어가 구매 의도에 어떻게 연결되는가?	상품 발견 → 상세 탐색 → 신뢰 확인 → 구매 의도	Jumia, Jumia 기업 소개
YouTube	한 편의 영상이 다음 시청과 재방문을 어떻게 만드는가?	콘텐츠 발견 → 시청 → 다음 영상 → 구독·재방문	YouTube, How YouTube Works
Jiji	지역성과 판매자 신뢰가 연락 의도를	지역 선택 → 매물 발견 → 신뢰 확인 → 판매자 연락 의도	Jiji, Jiji 소개

선택 문장을 프로젝트 README에 작성합니다.

```
선택 서비스: 〔서비스명〕
대표 사용자: 〔누가〕
대표 과업: 〔무엇을 완료하려 하는가〕
선택 이유: 〔왜 이 흐름이 성장과 연결된다고 보는가〕
관찰 기준일: 〔YYYY-MM-DD〕
```

3. 프로젝트 폴더 만들기

`{team-name}` 은 개인이면 영문 이름이나 학번 별칭을 사용합니다. `{service-name}` 은 `jumia` , `youtube` , `jiji` 중 하나입니다.

```
teams/{team-name}/{service-name}/
├── README.md
├── research/
│   ├── BENCHMARK_LOG.md
│   ├── GROWTH_LOOP.md
│   └── evidence/
├── docs/
│   ├── 01-product/OPPORTUNITY.md
│   ├── 02-business/BUSINESS_MODEL.md
│   └── 03-engineering/BASE_ARCHITECTURE.md
├── development/
│   ├── apps/web/
│   ├── packages/ui/src/
│   ├── packages/config/
│   └── tests/e2e/
└── reports/
    ├── AI_BUILD_LOG.md
    └── COMPARISON.md
```

터미널을 사용할 수 있다면 프로젝트 루트에서 실행합니다. 예시 이름은 반드시 자신의 값으로 바꿉니다.

```
mkdir -p teams/my-name/jumia/{research/evidence,docs/01-product,docs/02-
business,docs/03-
engineering,development/apps/web,development/packages/ui/src,development/packages/conf
ig,development/tests/e2e,reports}
```

터미널이 어렵다면 파일 탐색기에서 같은 구조로 폴더와 빈 Markdown 파일을 만듭니다.

4. 관찰 규칙 익히기

관찰과 해석을 섞지 않습니다.

- 관찰: “상품 카드에 후기 수와 평점이 함께 표시된다.”
- 해석: “구매 전 불확실성을 줄여 상세 진입을 돕는 것으로 보인다.”
- 잘못된 단정: “후기 표시 때문에 매출이 30% 증가한다.”

화면에 보이지 않는 알고리즘, 매출, 전환율은 공식 출처가 없으면 가설입니다. YouTube 추천 시스템을 분석한다면 먼저 YouTube의 공식 추천 설명을 읽고, 공식 설명과 자신의 화면 관찰을 구분합니다.

5. 관찰 세션 3회 수행하기

한 번에 보지 말고 목적이 다른 세션을 진행합니다.

세션 A: 첫 방문, 30분

1. 로그아웃 또는 시크릿 창에서 시작합니다.
2. 첫 화면에서 보이는 가치, 선택지, CTA를 기록합니다.
3. 5초 안에 이해한 내용을 한 문장으로 씁니다.
4. 다음 행동을 유도한 요소를 캡처합니다.

세션 B: 대표 과업, 45분

1. 선택한 대표 과업을 시작합니다.
2. 클릭·스크롤·입력 단계 수를 셉니다.
3. 각 단계의 기대, 실제 결과, 망설임을 기록합니다.
4. 신뢰 요소와 이탈 요소를 각각 찾습니다.

세션 C: 재방문 장치, 30분

1. 다음 콘텐츠, 알림, 저장, 구독, 리뷰, 공유 장치를 찾습니다.
2. 사용자의 활동이 다른 사용자에게 가치를 만드는 지점을 찾습니다.
3. 재방문이 일어나려면 어떤 조건이 필요한지 가설로 씁니다.

로그인이나 개인정보가 필요한 단계에서는 실제 데이터를 제출하지 않아도 됩니다. 버튼 직전까지 관찰하고 제한 사항을 기록합니다.

6. 관찰 로그 10건 작성하기

`research/BENCHMARK_LOG.md` 에 아래 형식을 복사해 `OBS-001` 부터 작성합니다.

```
## OBS-001 — 〔짧은 관찰 제목〕

- 관찰 일시: YYYY-MM-DD HH:mm
- URL 또는 화면 위치:
- 환경: 모바일/데스크톱, 로그인/비로그인
- 사용자 목표:
- 관찰한 사실:
- 팀의 해석:
- 다음 행동에 미치는 영향:
- AARRR 단계: 획득/활성화/유지/추천/수익
- 근거 파일: research/evidence/OBS-001.png
- 불확실성 또는 추가 확인:
```

파일명은 `OBS-001-home-cta.png` 처럼 ID와 내용을 함께 씁니다. 캡처에는 계정명, 이메일, 위치 등 개인정보가 보이지 않게 합니다.

7. AARRR 퍼널로 분류하기

단계	학생이 답할 질문	관찰 예시
획득	사용자는 어디서 처음 들어오는가?	검색, 공유 링크, 추천 피드, 캠페인
활성화	첫 번째 가치 경험은 무엇인가?	원하는 상품·영상·게시물을 발견
유지	다시 돌아올 이유는 무엇인가?	새 콘텐츠, 알림, 저장 목록, 개인화
추천	다른 사용자를 부르는 행동은 무엇인가?	공유, 후기, 콘텐츠 등록, 초대
수익	어떤 가치 교환이 수익과 연결되는가?	거래, 광고, 구독, 판매 수수료

`research/GROWTH_LOOP.md` 에 각 단계별 `OBS-###` 를 연결합니다. 근거가 없는 칸은 억지로 채우지 말고 `미관찰` 이라고 씁니다.

8. 성장 루프 그리기

다음 문장을 완성한 뒤 화살표로 그립니다.

```
〔사용자 행동〕
 → 〔서비스 안의 가치가 증가〕
 → 〔다른 사용자가 발견하거나 반응〕
 → 〔신규 유입 또는 기존 사용자 재방문〕
 → 〔첫 행동이 더 많이 발생〕
```

성장 루프 아래에 반드시 작성합니다.

- 가장 강한 근거 3개: `OBS-___` , `OBS-___` , `OBS-___`
- 아직 보이지 않는 연결:
- 이 루프가 멈추는 조건:
- 새 서비스에 가져올 원리:
- 그대로 복제하면 안 되는 요소:

9. 실험 가설 5개 만들기

```
## EXP-001 — 〔실험명〕
- 연결 근거: OBS-___
- 대상 사용자:
- 가설: 만약 〔변화〕를 제공하면 〔이유〕 때문에 〔지표〕가 변할 것이다.
- 1차 지표:
- 부작용 확인 지표:
- 성공 기준:
- 가장 작은 검증 방법:
- 영향도 1~5:
- 근거 수준 1~5:
- 구현 난이도 1~5: # 낮을수록 쉬움
- 우선순위 점수: (영향도 × 근거 수준) ÷ 구현 난이도
```

점수가 가장 높은 실험 하나가 미션 2와 3의 중심이 됩니다.

10. AI에게 받을 수 있는 도움

관찰 기록 전체를 주지 않아도 됩니다. 개인정보를 제거한 3~5개 관찰만 입력합니다.

```
나는 〔서비스〕의 공개 화면을 관찰했다. 아래 내용에서 사실과 해석을 분리하고,
근거 없이 단정한 문장을 찾아 질문으로 바꿔줘. 새로운 사실은 만들지 마.

〔OBS 기록 붙여넣기〕

출력 형식:
1. 확인 가능한 사실
2. 해석 또는 가설
3. 근거가 부족한 주장
4. 다음 관찰에서 확인할 질문
```

AI 답변은 관찰 근거가 아닙니다. 채택한 수정만 자신의 문서에 반영합니다.

11. 제출물

- 프로젝트 `README.md`
- `research/BENCHMARK_LOG.md` : `OBS-###` 10건 이상
- `research/GROWTH_LOOP.md` : AARRR, 성장 루프, `EXP-###` 5건 이상
- `research/evidence/` : 관찰과 연결된 캡처

12. 제출 전 자기점검

- 〔 〕 서비스, 사용자, 대표 과업이 한 문장으로 정의되었다.
- 〔 〕 폴더 구조가 요구사항과 일치한다.
- 〔 〕 10개 이상의 관찰에 날짜·URL·환경이 있다.
- 〔 〕 사실과 해석을 다른 항목에 적었다.
- 〔 〕 모든 주요 주장에 `OBS-###`가 연결되었다.
- 〔 〕 성장 루프에 멈추는 조건까지 적었다.
- 〔 〕 5개 실험에 지표와 성공 기준이 있다.
- 〔 〕 개인정보와 원본 저작물을 제출물에 포함하지 않았다.

13. 미션 1 채점표, 25점

평가 요소	배점	만점 기준
폴더와 기록 구조	5	모든 산출물이 정해진 위치에 있고 README에서 찾을 수 있음
관찰 근거	8	10건 이상이며 날짜·URL·환경·캡처가 주장과 연결됨
성장 퍼널과 루프	7	반복 행동, 가치 증가, 재유입, 정지 조건을 근거로 설명함
실험 가설	5	5개 이상이며 지표·성공 기준·우선순위가 있음

세부 성취 수준은 전체 평가 루브릭을 사용합니다.

14. 공식 참고 링크

- Jumia: 온라인 스토어, 기업 소개, Jumia Nigeria 소개
- YouTube: 서비스, How YouTube Works, 추천 작동 방식
- Jiji: 서비스, 공식 소개, 안전 거래 팁, 이용 약관
- 사용성 관찰: Nielsen Norman Group의 10가지 사용성 휴리스틱

학생 안내서로 돌아가기 · 다음 미션

미션 2. 관찰 결과를 3종 문서로 역설계하라

1. 미션 카드

항목	내용
목표	미션 1의 근거를 제품, 비즈니스, 기술의 실행 가능한 문서로 바꾼다
권장 시간	4~6시간
배점	25점
입력	`BENCHMARK_LOG.md` , `GROWTH_LOOP.md` , 최우선 실험 가설
출력	`OPPORTUNITY.md` , `BUSINESS_MODEL.md` , `BASE_ARCHITECTURE.md`
완료 신호	`OBS→PRD→BM→ADR` 연결표에 끊긴 항목이 없음

2. 시작 조건 확인하기

다음 세 항목에 모두 답할 수 있을 때 시작합니다.

1. 선택한 사용자는 누구인가?
2. 그 사용자가 완료할 대표 과업은 무엇인가?
3. 어떤 `OBS-###` 와 `EXP-###` 가 구현의 출발점인가?

답할 수 없다면 미션 1로 돌아가 관찰 범위를 줄입니다.

3. 먼저 근거 추적표 만들기

세 문서 중 하나에 아래 표를 만들고 작업 중 계속 갱신합니다.

관찰	관찰에서 얻은 원리	제품 요구사항	BM 가설	기술 결정	상태
OBS-001	예: 행동 전 신뢰 근거 노출	PRD-001	BM-001	ADR-003	초안

ID 규칙:

- `OBS-###` : 직접 관찰한 근거
- `PRD-###` : 사용자가 경험해야 하는 제품 요구사항
- `BM-###` : 가치 교환이나 지속 가능성 가설
- `ADR-###` : 구현 방법과 이유를 담은 기술 결정

4. 문서 A: 제품 기회 정의

파일: `docs/01-product/OPPORTUNITY.md`

아래 목차를 그대로 복사해 작성합니다.

```
# 제품 기회 정의
## 1. 한 문장 기회
## 2. 대상 사용자와 상황
## 3. 현재 행동과 문제
## 4. 벤치마크에서 발견한 원리
## 5. 새 서비스의 가치 제안
## 6. 대표 사용자 여정
## 7. 제품 요구사항 PRD-###
## 8. MVP 포함 범위
## 9. MVP 제외 범위
## 10. 성공 지표와 검증 방법
## 11. 위험과 미확정 질문
## 12. 관련 근거와 문서 링크
```

PRD 작성 형식

```
### PRD-001 — 〔사용자가 얻을 결과〕
- 연결 근거: OBS-___, EXP-___
- 사용자 스토리: 〔사용자〕는 〔목표〕를 위해 〔행동〕할 수 있어야 한다.
- 수용 기준:
  - 〔 〕 화면에서 확인 가능한 조건
  - 〔 〕 실패 시 사용자에게 보이는 피드백
  - 〔 〕 모바일과 키보드 사용 조건
- 제외 사항:
- 성공 지표:
```

"추천 기능 구현"처럼 해결책만 쓰지 않습니다. 사용자가 얻는 결과와 완료 여부를 확인할 기준을 씁니다.

5. 문서 B: 비즈니스 모델 정의

파일: `docs/02-business/BUSINESS_MODEL.md`

```
# 비즈니스 모델 정의
## 1. 모델 한 문장 요약
## 2. 이해관계자와 각자의 목표
## 3. 주고받는 가치
## 4. 획득·활성화·유지·추천·수익 구조
## 5. Business Model Canvas 9개 블록
## 6. BM-### 가설
## 7. 비용과 단위 경제성
## 8. 신뢰·공정성·개인정보 위험
## 9. 검증 순서와 중단 기준
## 10. 관련 근거와 문서 링크
```

Business Model Canvas는 사업이 누구에게 어떤 가치를 어떤 구조로 제공하고 비용과 수익을 어떻게 만드는지 한 페이지에서 연결하는 도구입니다. 처음이라면 Strategyzer 공식 설명을 읽습니다.

BM 가설 작성 형식

```
### BM-001 — 〔가설 이름〕
- 연결 근거: OBS-___, PRD-___
- 지불하거나 기여하는 주체:
- 얻는 가치:
- 서비스가 받는 가치:
```

```
- 핵심 지표:
- 검증 방법:
- 성공 기준:
- 실패 시 변경할 내용:
```

수익 모델이 명확하지 않다면 광고나 구독을 억지로 넣지 않습니다. 콘텐츠 생산, 데이터 제공, 재방문, 운영비 절감처럼 서비스가 지속되는 데 필요한 가치도 BM 가설이 될 수 있습니다.

6. 문서 C: 개발 기본설계

파일: `docs/03-engineering/BASE_ARCHITECTURE.md`

```
# 개발 기본설계
## 1. 구현 목표와 대표 사용자 흐름
## 2. MVP 화면과 라우트
## 3. 시스템 경계와 데이터 흐름
## 4. 기술 스택과 선택 이유
## 5. 폴더와 컴포넌트 구조
## 6. 콘텐츠 및 데이터 모델
## 7. 분석 이벤트
## 8. 오류와 빈 상태
## 9. 접근성·반응형·성능·보안 기준
## 10. 테스트 전략
## 11. ADR-### 기술 결정
## 12. 관련 근거와 문서 링크
```

ADR 작성 형식

```
### ADR-001 — 〔결정 제목〕
- 상태: 제안/채택/폐기
- 연결 요구사항: PRD-___, BM-___
- 문제:
- 선택:
- 선택 이유:
- 포기한 대안:
- 예상 부작용:
- 다시 검토할 조건:
```

7. MVP 범위 자르기

다음 질문에 `예` 라고 답할 수 없는 기능은 미션 3에서 제외합니다.

- 대표 사용자가 한 번의 방문에서 경험할 수 있는가?
- 최우선 성장 가설을 보여 주는가?
- 6~10시간 안에 기본 흐름을 구현하고 테스트할 수 있는가?
- 외부 서비스나 실제 개인정보 없이 데모할 수 있는가?

권장 화면은 1~3개, 핵심 사용자 흐름은 1개입니다. 로그인, 결제, 실시간 채팅, 추천 알고리즘은 목업 데이터로 대체할 수 있습니다.

8. 분석 이벤트 정의하기

이벤트	발생 조건	연결 지표	포함 데이터	금지 데이터
primary_cta_click	핵심 CTA 클릭	의도율	CTA 위치	이름, 이메일
primary_flow_complete	대표 과업 완료	완료율	흐름 버전	자유 입력 원문

이벤트는 화면 조회보다 사용자 행동을 우선합니다. 이벤트 개념이 낯설면 Google Analytics 공식 이벤트 설명을 참고합니다.

9. AI로 세 문서 교차 검토하기

각 문서를 따로 생성시키지 말고 작성 후 모순 검토에 사용합니다.

```
아래 세 문서는 같은 서비스의 제품·비즈니스·기술 정의다.
새로운 기능이나 사실을 만들지 말고 다음만 찾아줘.

1. OBS 근거가 없는 PRD/BM/ADR
2. 같은 용어를 다르게 사용한 부분
3. 제품 범위와 개발 범위가 충돌하는 부분
4. 측정할 수 없는 성공 기준
5. 개인정보·저작권·보안 위험

결과는 문제 위치, 이유, 최소 수정안 표로 작성해줘.

〔세 문서 붙여넣기〕
```

AI가 지적한 항목마다 `수정` , `보류` , `기각` 중 하나를 정하고 이유를 남깁니다.

10. 제출물

- `docs/01-product/OPPORTUNITY.md`
- `docs/02-business/BUSINESS_MODEL.md`
- `docs/03-engineering/BASE_ARCHITECTURE.md`
- `OBS→PRD→BM→ADR` 근거 추적표

형식이 막히면 다음 예시를 참고합니다.

- 제품 기회 정의 예시
- 비즈니스 모델 예시
- 개발 기본설계 예시

11. 제출 전 자기점검

- 〔 〕 세 문서가 같은 사용자와 대표 과업을 설명한다.
- 〔 〕 모든 핵심 요구사항에 `OBS-###` 가 있다.
- 〔 〕 PRD마다 화면에서 확인 가능한 수용 기준이 있다.
- 〔 〕 BM 가설에 검증 방법과 실패 기준이 있다.

- 〔 〕 ADR에 선택 이유와 포기한 대안이 있다.
- 〔 〕 MVP 포함·제외 범위가 명시되어 있다.
- 〔 〕 분석 이벤트에 개인정보가 포함되지 않는다.
- 〔 〕 세 문서가 서로 링크되어 있다.

12. 미션 2 채점표, 25점

평가 요소	배점	만점 기준
제품 문서	8	문제·여정·PRD·범위·지표가 근거와 함께 구체적임
비즈니스 문서	7	이해관계자·가치 교환·BM 가설·검증 기준이 연결됨
개발설계	7	흐름·데이터·품질·테스트·ADR이 구현 가능한 수준임
문서 추적성	3	`OBS→PRD→BM→ADR` 와 문서 링크에 단절이 없음

세부 성취 수준은 전체 평가 루브릭을 사용합니다.

13. 참고 링크

- Business Model Canvas 공식 설명
- Google Analytics 이벤트 개념
- WCAG 2.2 빠른 참조
- Nielsen Norman Group 사용성 원칙

이전 미션 · 학생 안내서 · 다음 미션

미션 3. 문서를 기준으로 AI와 사이트를 구현하라

1. 미션 카드

항목	내용
목표	미션 2의 문서를 구현 계약으로 사용해 대표 사용자 흐름을 완성한다
권장 시간	6~10시간
배점	30점
입력	`OPPORTUNITY.md` , `BUSINESS_MODEL.md` , `BASE_ARCHITECTURE.md`
출력	실행 가능한 웹사이트, 테스트, README, AI 작업 기록
완료 신호	새 환경에서 실행되고 대표 흐름과 실패 흐름이 검증됨

AI가 많은 코드를 만들었다는 사실은 평가 대상이 아닙니다. 문서와 일치하고, 직접 실행해 확인했으며, 문제가 생겼을 때 원인을 설명할 수 있어야 합니다.

2. 시작 전 환경 점검

터미널에서 다음 명령을 실행합니다.

```
node --version
npm --version
git --version
```

명령을 찾을 수 없으면 다음 공식 문서를 사용합니다.

- Node.js 다운로드
- Git 설치
- GitHub 저장소 빠른 시작

Next.js의 현재 시스템 요구사항과 생성 명령은 공식 설치 문서를 우선합니다. 이 미션의 명령과 공식 문서가 다르면 공식 문서를 따르고 `ADR-###` 에 변경 이유를 적습니다.

3. 구현 기준선 고정하기

개발 전에 `development/README.md` 에 아래 표를 작성합니다.

구현 대상	연결 ID	완료 확인 방법	상태
대표 흐름 시작 화면	PRD-___	첫 화면에서 대상과 CTA 확인	예정
핵심 상호작용	PRD-___ / BM-___	클릭 또는 입력 후 상태 변경	예정

구현 대상	연결 ID	완료 확인 방법	상태
완료 피드백	PRD-___	완료 메시지와 다음 행동 표시	예정
분석 이벤트	ADR-___	개발 로그 또는 테스트로 발생 확인	예정

이번 미션에서 구현하지 않을 항목도 적습니다. AI가 범위를 임의로 확대하지 못하게 하기 위한 장치입니다.

4. Next.js 앱 초기화하기

프로젝트 루트에서 실행합니다. `development/apps/web` 폴더 안에 중요한 파일이 있다면 먼저 백업하고, 비어 있는지 확인합니다.

```
npx create-next-app@latest development/apps/web
```

질문이 나오면 다음을 권장합니다.

```
TypeScript: Yes
Linter: ESLint
Tailwind CSS: Yes
src directory: Yes
App Router: Yes
Import alias: @/*
```

설치 후 실행합니다.

```
cd development/apps/web
npm run dev
```

브라우저에서 `http://localhost:3000` 을 열어 첫 화면이 보이면 설치 성공입니다. 터미널에 오류가 있다면 다음 단계로 넘어가지 않습니다.

5. 구현 폴더 확인하기

```
development/apps/web/
├── public/images/              # 직접 제작했거나 사용 권한이 있는 이미지
└── src/
    ├── app/                    # 페이지, 레이아웃, 메타데이터
    ├── components/common/      # 버튼, 헤더, 상태 UI
    ├── components/sections/    # 서비스 섹션 단위
    ├── lib/                    # 데이터, 검증, 분석 이벤트
    └── styles/                 # 전역 스타일과 토큰
```

폴더가 설계와 달라졌다면 코드를 억지로 옮기기 전에 `BASE_ARCHITECTURE.md` 의 ADR을 갱신합니다.

6. 세 번의 수직 구현 사이클

각 사이클은 `계획 → AI 요청 → 코드 검토 → 실행 → 테스트 → 기록` 순서로 진행합니다.

사이클 1: 가치 이해

완료 조건:

- 사용자가 5초 안에 서비스 대상과 가치를 설명할 수 있습니다.
- 대표 행동으로 가는 CTA가 하나 명확하게 보입니다.
- 모바일 360px에서 제목과 CTA가 잘리지 않습니다.

사이클 2: 핵심 행동

완료 조건:

- 사용자가 콘텐츠를 탐색하거나 입력해 상태를 바꿀 수 있습니다.
- 로딩, 빈 결과, 잘못된 입력 중 필요한 상태를 구현합니다.
- 행동 결과가 화면에 즉시 설명됩니다.

사이클 3: 완료와 다음 행동

완료 조건:

- 대표 과업의 완료 상태가 명확합니다.
- 재방문, 저장, 공유, 다음 탐색 중 성장 가설과 연결된 행동이 있습니다.
- `primary_flow_complete` 같은 핵심 이벤트가 정의됩니다.

7. AI 작업 요청 방식

한 번에 사이트 전체를 요청하지 않습니다. 한 사이클씩 다음 형식을 사용합니다.

```
역할: Next.js 개발 파트너

목표:
〔이번 사이클의 사용자 결과〕

반드시 지킬 문서:
- PRD-___:〔요약〕
- BM-___:〔요약〕
- ADR-___:〔요약〕

수정 가능한 경로:
〔파일 또는 폴더〕

완료 기준:
-〔화면에서 확인할 조건〕
-〔실패 상태〕
-〔모바일/접근성 조건〕
-〔테스트 조건〕

금지:
- 새 기능 추가
- 원본 브랜드·문구·이미지 복제
- 확인하지 않은 패키지 설치
- 비밀 값이나 실제 개인정보 사용

먼저 작업 계획과 수정 파일을 제안하고, 내가 확인한 뒤 구현해줘.
```

AI가 제안한 명령은 목적과 현재 위치를 확인한 뒤 실행합니다. 파일 삭제, 강제 초기화, 비밀 값 출력 명령은 실행하지 않습니다.

8. AI 작업 로그 작성하기

각 작업 후 `reports/AI_BUILD_LOG.md` 에 기록합니다.

```
## AI-001 — 〔작업 제목〕
- 일시:
- 연결 ID: PRD-___ / BM-___ / ADR-___
- 사용 도구:
- 요청 목표와 제약:
- 변경 파일:
- 실행한 검증:
- AI 결과의 오류 또는 누락:
- 사람이 직접 수정한 내용:
- 채택하지 않은 제안과 이유:
```

최소 5개의 작업 기록이 있어야 하며, “문제 없음”만 반복해서 쓰면 검증 기록으로 인정하지 않습니다.

9. 기능 테스트하기

수동 테스트

- 〔 〕 첫 화면에서 대표 사용자가 누구인지 알 수 있다.
- 〔 〕 CTA를 키보드 `Tab` 과 `Enter` 로 사용할 수 있다.
- 〔 〕 핵심 흐름을 새로고침 후 다시 완료할 수 있다.
- 〔 〕 잘못된 입력 또는 빈 상태에 해결 방법이 표시된다.
- 〔 〕 360px와 1440px에서 가로 스크롤이 생기지 않는다.
- 〔 〕 콘솔에 설명하지 못하는 오류가 없다.

코드 검사

```
npm run lint
npm run build
```

두 명령이 성공해야 합니다. 실패했다면 오류의 첫 원인을 해결한 뒤 다시 실행합니다.

E2E 테스트 선택 사항

대표 흐름을 자동화하려면 Playwright 테스트 작성 안내를 사용합니다. 테스트는 사용자가 보는 역할, 이름, 텍스트를 기준으로 작성하고 내부 클래스명에 의존하지 않습니다.

최소 테스트 시나리오:

```
Given 사용자가 첫 화면을 연다
When 핵심 CTA를 선택하고 대표 과업을 수행한다
Then 완료 피드백과 다음 행동이 보인다
```

10. 접근성·성능 점검하기

Chrome DevTools에서 Lighthouse를 실행합니다. Lighthouse 공식 안내를 참고합니다.

기록할 항목:

- 측정 일시와 URL
- 모바일 또는 데스크톱 조건
- Performance, Accessibility, Best Practices, SEO 점수
- 가장 큰 문제 3개
- 수정 전·후 결과

점수 100 자체가 목표는 아닙니다. 대표 흐름을 방해하는 문제를 설명하고 개선하는 것이 목표입니다. 접근성 판단에는 WCAG 2.2 빠른 참조를 사용합니다.

11. 고장 해결 순서

1. 실행한 명령과 현재 폴더를 확인합니다.
2. 오류 메시지의 첫 번째 원인을 읽습니다.
3. 마지막으로 바꾼 파일의 diff를 확인합니다.
4. 최소 재현 절차를 적습니다.
5. AI에는 전체 프로젝트 대신 오류, 관련 파일, 재현 절차를 제공합니다.
6. 수정 후 `lint` , `build` , 대표 흐름을 다시 검증합니다.

AI 질문 예시:

```
아래 오류의 원인을 먼저 설명해줘. 아직 코드를 바꾸지 마.
재현 명령, 관련 파일, 오류 원문을 기준으로 가능한 원인을 최대 3개로 좁히고,
각 원인을 확인할 가장 작은 진단 방법을 알려줘.
```

12. 제출물

- `development/apps/web/` 실행 가능한 소스
- `development/README.md` : 설치, 실행, 검사, 테스트 방법
- 대표 흐름 테스트 또는 재현 가능한 수동 테스트 기록
- `reports/AI_BUILD_LOG.md` : `AI-###` 5건 이상
- Lighthouse 전·후 기록

README에는 최소한 다음 명령이 있어야 합니다.

```
npm install
npm run dev
```

```
npm run lint
npm run build
```

13. 제출 전 자기점검

- 〔 〕 새 터미널에서 README만 보고 실행할 수 있다.
- 〔 〕 대표 흐름을 처음부터 끝까지 완료할 수 있다.
- 〔 〕 실패·빈·완료 상태 중 필요한 상태를 구현했다.
- 〔 〕 모바일과 키보드로 핵심 기능을 사용할 수 있다.
- 〔 〕 `npm run lint` 와 `npm run build` 가 성공한다.
- 〔 〕 코드와 테스트가 PRD/BM/ADR에 연결된다.
- 〔 〕 AI 작업 5건 이상에 사람의 검증과 수정이 기록되었다.
- 〔 〕 원본 브랜드와 저작물을 복제하지 않았다.

14. 미션 3 채점표, 30점

평가 요소	배점	만점 기준
대표 사용자 흐름	10	시작·핵심 행동·완료·오류 복구가 안정적으로 동작
문서 일치	6	구현과 테스트가 PRD/BM/ADR에 연결되고 변경 이유가 기록됨
품질	6	반응형·접근성·성능·오류 상태를 직접 검증하고 개선함
테스트	4	핵심 정상·실패 흐름이 자동 또는 재현 가능한 방식으로 검증됨
AI 사용 책임성	4	작업·오류·사람의 검증·기각 결정이 구체적으로 기록됨

세부 성취 수준은 전체 평가 루브릭을 사용합니다.

15. 공식 참고 링크

- Next.js 설치와 시작
- Node.js 다운로드
- Git 설치
- GitHub 저장소 빠른 시작
- Playwright 테스트 작성
- Lighthouse
- Core Web Vitals
- WCAG 2.2 빠른 참조

이전 미션 · 학생 안내서 · 다음 미션

미션 4. 원본 서비스와 구현 결과의 차이를 증명하라

1. 미션 카드

항목	내용
목표	원본과 구현물을 같은 과업·조건·기준으로 비교하고 다음 실험을 결정한다
권장 시간	3~5시간
배점	20점
입력	미션 1 기준 자료, 미션 2 문서, 미션 3 사이트와 테스트 기록
출력	`COMPARISON.md` , 비교 근거, 다음 실험 3개, 최종 회고
완료 신호	모든 결론에 `CMP-###` 근거가 있고 다음 실험에 판정 기준이 있음

이 미션은 “원본과 얼마나 비슷한가”를 평가하지 않습니다. 원본에서 발견한 성장 원리가 새로운 서비스에서 어떻게 유지·변형·손실되었는지 평가합니다.

2. 비교 전에 기준선 고정하기

`reports/COMPARISON.md` 첫 부분에 작성합니다.

```
## 비교 기준선
- 원본 서비스:
- 원본 관찰 일시:
- 구현 사이트 버전 또는 커밋:
- 구현 사이트 측정 일시:
- 기기와 브라우저:
- 화면 크기:
- 로그인 상태:
- 공통 대표 과업:
- 비교 제한 사항:
```

원본 서비스는 미션 1 이후 바뀔 수 있습니다. 바뀐 화면은 새 근거로 캡처하고 “기준선 변경”이라고 표시합니다.

3. 공통 대표 과업 만들기

두 사이트에서 의미가 같은 시작과 끝을 정의합니다.

```
시작: 첫 화면을 처음 본 상태
과업: 〔관심 콘텐츠를 발견하고 신뢰 근거를 확인한다〕
끝: 〔주요 행동 버튼을 선택하거나 완료 상태를 확인한다〕
```

원본에서는 구매, 구독, 채팅 직전까지만 수행해도 됩니다. 실제 결제나 타인에게 메시지를 보내지 않습니다.

4. 혼자 수행하는 3회 반복 테스트

같은 사람의 반복은 실제 사용자 테스트를 대신하지 못하지만, 절차와 기술 차이를 비교하는 근거가 됩니다.

1. 원본 서비스의 공통 과업을 1회 수행합니다.
2. 브라우저 상태를 초기화하고 구현 사이트에서 수행합니다.
3. 순서를 바꿔 2회 더 반복합니다.
4. 각 실행의 시간, 행동 수, 막힘, 오류를 기록합니다.

실행	대상	완료	시간	클릭·입력 수	막힘 수	오류 수	메모
1	원본	예/아니오					
2	구현	예/아니오					

가능하면 3명의 동료에게 같은 과업을 요청해 별도 기록합니다. 이는 권장 사항이며 혼자 진행하는 학생의 필수 조건은 아닙니다.

5. 7개 영역 비교하기

각 판단에 `CMP-001` 부터 ID를 붙입니다.

영역	비교 질문	근거
가치 이해	첫 화면에서 대상·가치·다음 행동을 이해하는가?	5초 기록, 제목·CTA 캡처
과업 효율	대표 행동을 얼마나 적은 마찰로 완료하는가?	시간, 단계, 오류 수
성장 구조	다음 행동·재방문·공유를 만드는 장치가 있는가?	성장 루프와 화면 연결
신뢰	행동 전 불안을 줄이는 근거가 있는가?	후기, 정책, 출처, 상태 피드백
BM 연결	화면의 행동이 가치 교환·지속 가능성과 연결되는가?	BM-###와 UI 연결
기술 품질	접근성·반응형·성능·복구가 충분한가?	Lighthouse, 빌드, 테스트
독창성	원리를 새 맥락에 맞게 바꿨는가?	ADR과 변경 이유

비교 기록 형식

```
## CMP-001 — {비교 제목}
- 영역:
- 연결 근거: OBS-___ / PRD-___ / BM-___ / ADR-___
- 원본에서 확인한 내용:
- 구현물에서 확인한 내용:
- 분류: 유지/개선/손실/의도적 변경
- 근거 파일 또는 측정값:
- 차이가 생긴 이유:
- 사용자에게 미치는 영향:
```

최소 10개의 `CMP-###` 를 작성합니다.

6. 기술 지표를 같은 조건에서 측정하기

Lighthouse를 원본과 구현물에 같은 브라우저·기기 모드로 실행합니다. 로그인이나 개인화 때문에 원본 측정이 불가능하면 그 이유를 적고 비교에서 제외합니다.

지표	원본	구현	차이	해석 가능 여부
Performance				
Accessibility				
Best Practices				
SEO				

Lighthouse 공식 안내와 Core Web Vitals 학습 자료를 기준으로 사용합니다. 서로 다른 네트워크 조건에서 나온 숫자를 단순 비교하지 않습니다.

7. 사용성 휴리스틱 점검하기

Nielsen Norman Group의 10가지 사용성 휴리스틱 중 아래 5개를 필수로 확인합니다.

- 시스템 상태가 보이는가?
- 사용자의 언어와 실제 세계의 개념을 사용하는가?
- 사용자가 취소하거나 이전 상태로 돌아갈 수 있는가?
- 오류를 예방하고 복구 방법을 설명하는가?
- 필요한 도움말이나 안내가 있는가?

각 문제에는 심각도를 붙입니다.

심각도	의미
0	문제가 아님
1	외형 수준의 작은 문제
2	일부 사용자를 지연시키는 문제
3	대표 과업을 어렵게 만드는 큰 문제
4	대표 과업을 완료할 수 없는 문제

8. AI의 판단을 평가하기

`AI_BUILD_LOG.md` 에서 최소 3개의 결정을 골라 작성합니다.

AI 작업 ID	AI가 제안한 것	결과	사람의 수정	배운 점
AI-001		채택/수정/기각		

AI가 잘한 것만 쓰지 않습니다. 사실을 만들었거나 범위를 넓혔거나 접근성을 놓친 사례도 기록합니다.

9. 다음 성장 실험 3개 결정하기

```
## EXP-NEXT-001 — 〔실험 제목〕
- 발견 근거: CMP-___
- 대상 사용자:
- 바꿀 한 가지:
- 1차 지표:
- 부작용 확인 지표:
- 성공 기준:
- 실패 기준:
- 가장 작은 구현 범위:
- 우선순위 이유:
```

“디자인 개선”처럼 범위가 넓은 항목은 실험이 아닙니다. 한 번에 한 가지 변화와 한 가지 주요 지표를 둡니다.

10. 비교 보고서 작성하기

`reports/COMPARISON.md` 목차:

```
# 원본 서비스와 구현 사이트 비교
## 1. 비교 대상과 기준선
## 2. 공통 대표 과업
## 3. 반복 테스트 결과
## 4. 제품 경험 차이
## 5. 성장 루프 차이
## 6. 비즈니스 모델 차이
## 7. 기술·품질 차이
## 8. 유지·개선·손실·의도적 변경
## 9. AI가 잘한 판단과 사람이 수정한 판단
## 10. 다음 성장 실험 3개
## 11. 한계와 결론
```

11. AI로 결론의 과장을 검사하기

```
아래 비교 보고서에서 근거보다 강하게 말한 결론을 찾아줘.
새로운 사실은 추가하지 말고 다음 기준으로만 검사해줘.

1. 측정 조건이 다른 수치 비교
2. 한 사람의 테스트를 전체 사용자에게 일반화
3. 상관관계를 인과관계로 표현
4. 원본 서비스의 내부 데이터 추측
5. 장점만 있고 손실이나 한계가 없는 결론

각 항목에 문제 문장, 문제 이유, 더 정확한 수정 문장을 제안해줘.
```

수정 후에도 보고서의 한계 항목은 삭제하지 않습니다.

12. 최종 제출 패키지

```
README.md
research/BENCHMARK_LOG.md
research/GROWTH_LOOP.md
research/evidence/
docs/01-product/OPPORTUNITY.md
docs/02-business/BUSINESS_MODEL.md
```

```
docs/03-engineering/BASE_ARCHITECTURE.md
development/apps/web/
development/README.md
reports/AI_BUILD_LOG.md
reports/COMPARISON.md
```

최종 README 첫 화면에 실행 방법, 대표 흐름, 문서 링크, 비교 보고서 링크를 둡니다.

13. 제출 전 자기점검

- 〔 〕 원본과 구현물의 과업과 측정 조건이 같다.
- 〔 〕 반복 테스트 결과가 표로 남아 있다.
- 〔 〕 `CMP-###` 10건 이상이 근거 파일과 연결된다.
- 〔 〕 차이를 유지·개선·손실·의도적 변경으로 분류했다.
- 〔 〕 기술 지표의 측정 조건과 한계를 기록했다.
- 〔 〕 AI의 채택·수정·기각 사례를 모두 포함했다.
- 〔 〕 다음 실험 3개에 성공·실패 기준이 있다.
- 〔 〕 원본보다 낫다는 근거 없는 결론을 쓰지 않았다.

14. 미션 4 채점표, 20점

평가 요소	배점	만점 기준
비교 설계	6	공통 과업·조건·측정 방법이 고정되고 반복 가능함
비교 증거	5	10개 이상 CMP에 화면·시간·단계·품질 지표가 연결됨
비판적 회고	5	개선과 손실, AI 오류, 비교 한계를 모두 설명함
다음 실험	4	3개 실험에 지표·성공·실패 기준과 우선순위가 있음

세부 성취 수준은 전체 평가 루브릭을 사용합니다.

15. 참고 링크

- Lighthouse 공식 안내
- Core Web Vitals
- WCAG 2.2 빠른 참조
- 10가지 사용성 휴리스틱
- Playwright 테스트 모범 사례

이전 미션 · 학생 안내서 · 평가 루브릭

리얼 벤치마킹 PBL 평가 루브릭

1. 평가 방식

총점은 100점입니다. 결과물의 외형만 평가하지 않고, 관찰 근거가 문서와 구현을 거쳐 최종 비교까지 추적되는지를 평가합니다.

학생은 제출 전에 스스로 먼저 채점합니다. 각 평가 요소에 대해 아래 성취 수준을 고르고 근거 파일과 위치를 적습니다.

수준	의미	배점 환산
4	요구사항을 모두 충족하고 근거·예외·한계까지 설명	해당 항목 배점의 100%
3	핵심 요구사항과 근거를 충족하나 일부 예외·깊이가 부족	해당 항목 배점의 80%
2	결과물은 있으나 근거 연결이나 검증이 부분적	해당 항목 배점의 60%
1	초안 수준이며 재현·검증하기 어려움	해당 항목 배점의 30%
0	미제출, 확인 불가 또는 필수 안전 규칙 위반	0점

계산 예시: 배점 8점 항목에서 수준 3이면 `8 × 0.8 = 6.4점` 입니다. 총점은 소수점 첫째 자리까지 계산합니다.

자기채점 기록 형식

```
| 평가 요소 | 배점 | 자기 수준 | 자기 점수 | 근거 파일과 위치 | 보완할 점 |
|---|---|---|---|---|---|
| 관찰 근거 | 8 | 3 | 6.4 | research/BENCHMARK_LOG.md, OBS-001~012 | URL 누락 1건 보완 |
```

2. 미션별 배점

영역	배점	핵심 평가 질문
미션 1: 조사와 성장 가설	25	실제 관찰을 바탕으로 성장 루프를 설명했는가?
미션 2: 3종 문서	25	제품·BM·기술 문서가 일관되고 추적 가능한가?
미션 3: AI 기반 구현	30	문서와 일치하는 작동 가능한 흐름을 만들고 검증했는가?
미션 4: 비교와 학습	20	동일 기준으로 차이를 증명하고 다음 실험을 제안했는가?

3. 상세 루브릭

미션 1: 조사와 성장 가설, 25점

기준	우수	보통	미흡	배점
폴더와 기록 구조	산출물 경계와 링크가 명확하고 바로 탐색 가능	기본 구조는 있으나 일부 파일 목적이나 링크가 불명확	자료와 코드가 섞이고 주요 파일을 찾기 어려움	5

기준	우수	보통	미흡	배점
관찰 근거	날짜·URL·사용자 상태·캡처가 핵심 주장마다 연결	근거는 있으나 일부 주장과 연결이 약함	기억이나 일반론 중심이며 근거가 거의 없음	8
성장 루프	반복 행동과 가치 증가 구조를 명확히 설명	퍼널은 있으나 반복 루프 설명이 약함	기능 목록이나 화면 묘사에 머묾	7
실험 가설	지표와 판정 기준이 있는 가설 5개 이상	가설은 있으나 측정 기준이 일부 없음	해결책 아이디어만 나열	5

미션 2: 3종 문서, 25점

기준	우수	보통	미흡	배점
제품 문서	고객 문제, 여정, 범위, 지표가 구체적	주요 항목은 있으나 범위 또는 검증 기준이 모호	기능 요구만 나열	8
BM 문서	가치 교환, 성장, 수익, 위험이 연결됨	Canvas는 있으나 검증 가설이 약함	수익 아이디어만 제시	7
개발설계	사용자 흐름, 데이터, 품질, 테스트가 구현 가능 수준	기술 스택과 화면은 있으나 품질·데이터 정의가 부족	도구 이름만 나열	7
추적성	`OBS→PRD→BM→ADR` 연결이 일관됨	일부 연결이 누락됨	문서가 서로 다른 제품을 설명함	3

미션 3: AI 기반 구현, 30점

기준	우수	보통	미흡	배점
대표 흐름	처음부터 완료 피드백까지 안정적으로 동작	핵심 흐름은 동작하나 일부 오류 복구가 부족	데모용 정적 화면에 머묾	10
문서 일치	구현과 테스트가 요구사항 ID에 연결됨	주요 기능은 일치하나 변경 근거가 일부 없음	문서와 실제 동작이 다름	6
품질	반응형, 접근성, 성능, 오류 처리를 검증	일부 항목만 검증	시각 결과 외 품질 검증이 없음	6
테스트	핵심·실패 흐름을 자동 또는 재현 가능한 방식으로 검증	정상 흐름 위주로 검증	재현 가능한 테스트가 없음	4
AI 사용 책임성	입력·결과·검증·수정 기록이 구체적	사용 기록은 있으나 사람의 검증이 불명확	생성 결과를 검증 없이 제출	4

미션 4: 비교와 학습, 20점

기준	우수	보통	미흡	배점
비교 설계	동일 과업과 조건으로 원본·구현물을 비교	비교는 했으나 조건 통제가 약함	서로 다른 기준으로 인상만 비교	6
증거	사용자 기록과 측정값이 결론에 연결	일부 결론만 근거가 있음	근거 없는 자기평가 중심	5
비판적 회고	개선·손실·AI 오류를 모두 구체적으로 설명	장단점은 있으나 원인 분석이 얕음	성공 사례만 소개	5
다음 실험	지표와 판정 기준이 있는 우선 실험 3개	실험은 있으나 기준이 일부 모호	일반적인 개선 의견만 제시	4

4. 감점과 통과 제한

아래 항목은 총점과 별도로 적용합니다.

상황	처리
출처 또는 관찰 근거 없이 원본 서비스의 내부 전략·성과를 사실로 단정	해당 근거 항목 0점
원본 브랜드 자산이나 저작물을 무단 복제	최대 20점 감점 및 수정 후 재제출
비공개 데이터 접근, 인증 우회, 이용약관을 위반하는 수집	프로젝트 미통과
실행 불가능한 사이트 제출	미션 3 대표 흐름 항목 0점
AI 생성물을 검증 없이 제출하거나 사용 기록 누락	AI 사용 책임성 항목 0점
실제 개인정보나 비밀 값 노출	프로젝트 미통과 및 즉시 제거

제출 전 자동 미통과 점검

- 〔 〕 원본 서비스의 계정·인증을 우회하지 않았다.
- 〔 〕 비공개 API나 자동 대량 수집을 사용하지 않았다.
- 〔 〕 원본 로고·이미지·영상·문구·소스 코드를 복제하지 않았다.
- 〔 〕 실제 이름·이메일·전화번호·위치·결제 정보를 저장하지 않았다.
- 〔 〕 AI가 생성한 주장과 코드를 사람이 검증했다.
- 〔 〕 사이트가 원본 서비스의 공식 결과물로 오해되지 않도록 이름과 자산을 바꿨다.

5. 최종 통과 조건

- 총점 70점 이상
- 네 개 미션을 모두 제출
- 미션 3의 대표 사용자 흐름이 실행 가능
- 윤리·법적 경계의 미통과 사유가 없음
- 각 미션에서 해당 배점의 50% 이상 획득

6. 미션별 통과 게이트

미션	다음 단계로 가기 전 필수 조건
미션 1	OBS 10건, 성장 루프 1개, 측정 가능한 실험 가설 5개
미션 2	3종 문서, MVP 포함·제외 범위, OBS→PRD→BM→ADR 추적표
미션 3	실행 가능한 대표 흐름, lint/build 성공, 실패 상태, AI 작업 기록 5건
미션 4	동일 조건 비교, CMP 10건, 한계, 다음 실험 3개

7. 최종 자기평가표

```
# 최종 자기평가

| 미션 | 배점 | 자기 점수 | 가장 강한 근거 | 가장 약한 부분 |
|---|---:|---:|---|---|
| 미션 1 | 25 | | | |
| 미션 2 | 25 | | | |
| 미션 3 | 30 | | | |
| 미션 4 | 20 | | | |
| 합계 | 100 | | | |

## 내가 직접 결정한 것
-

## AI 제안을 수정하거나 거절한 것
-

## 다시 한다면 가장 먼저 바꿀 것
-
```

8. 발표 체크리스트

- 〔 〕 30초 안에 선택 서비스, 사용자, 대표 흐름을 설명한다.
- 〔 〕 핵심 성장 루프를 하나의 그림으로 보여 준다.
- 〔 〕 관찰 근거가 3종 문서의 어떤 결정으로 이어졌는지 보여 준다.
- 〔 〕 사이트에서 대표 사용자 흐름을 직접 시연한다.
- 〔 〕 원본과 구현물의 차이를 같은 기준으로 비교한다.
- 〔 〕 AI가 만든 오류와 사람이 바꾼 결정을 하나 이상 공개한다.
- 〔 〕 다음 실험 3개와 성공 판정 기준을 제시한다.

관련 문서

- PBL 미션 안내
- 미션 1: 성장 구조 벤치마킹
- 미션 2: 3종 문서 역설계
- 미션 3: AI 기반 사이트 구현
- 미션 4: 원본과 결과 비교
- 기회 정의 예시
- 비즈니스 모델 예시
- 개발 기본설계 예시

기회 정의

1. 문서 목적

이 문서는 SW 랜딩 서비스가 어떤 문제를 해결하고, 누구에게 어떤 결과를 제공할지 정의합니다. 아래 내용은 초기 가설이며 사용자 인터뷰와 운영 데이터로 수정해야 합니다.

2. 기회 가설

지역의 SW 기업과 공급 역량은 여러 기관, 기업 소개서, 지원사업 페이지에 흩어져 있습니다. 수요자는 자신의 과제에 맞는 기업을 빠르게 비교하기 어렵고, 공급자는 실제 상담으로 이어지는 방식으로 역량을 설명하기 어렵습니다.

서비스의 기회는 단순한 홍보 페이지가 아니라 다음 흐름을 짧게 만드는 데 있습니다.

```
수요자가 문제를 인식함
  → 적합한 SW 역량과 사례를 확인함
  → 신뢰할 근거를 검토함
  → 상담 또는 연결을 요청함
  → 운영자가 적합한 공급자와 연결함
```

3. 핵심 고객

3.1 1차 고객: SW 도입 수요 기업

- 디지털 전환이나 신규 서비스 개발이 필요하지만 적절한 공급자를 모르는 기업
- 비교 가능한 정보, 실제 구축 사례, 예상 진행 절차가 필요한 실무 책임자
- 긴 제안 절차 전에 적합성을 빠르게 판단하고 싶은 의사결정자

3.2 2차 고객: 지역 SW 공급 기업

- 보유 기술과 레퍼런스를 잠재 고객에게 알리고 싶은 기업
- 단순 노출보다 조건이 맞는 상담 기회를 원하는 영업 담당자
- 공공 및 민간 협업 기회를 찾는 기술 기업

3.3 운영 고객

- 기업 정보와 사례를 관리하고 연결 성과를 추적하는 운영 담당자
- 유입, 문의, 매칭, 후속 성과를 보고해야 하는 사업 책임자

4. 해결할 문제

ID	문제	현재 대안의 한계	기대 결과
PRD-001	수요자가 적합한 SW 기업을 찾기 어렵다	검색 결과와 기업 소개 자료가 흩어져 비교가 어렵다	분야와 사례를 기준으로 후보를 빠르게 이해한다
PRD-002	공급자의 역량 설명이 기술 나열에 머문다	고객 문제와 성과 중심의 설명이 부족하다	문제, 해결 방식, 성과가 연결된 사례를 본다
PRD-003	관심이 생겨도 다음 행동이 불명확하다	문의 채널과 필요 정보가 제각각이다	한 개의 명확한 상담 흐름으로 전환한다
PRD-004	운영자가 랜딩 성과를 확인하기 어렵다	페이지 조회와 실제 매칭 성과가 분리된다	문의부터 연결 결과까지 단계별로 측정한다

5. 제품 약속

> 수요 기업이 자신의 문제에 맞는 지역 SW 역량을 이해하고, 신뢰할 근거를 확인한 뒤, 짧은 절차로 상담을 요청하게 한다.

6. MVP 사용자 흐름

1. 방문자가 첫 화면에서 서비스 대상과 가치를 5초 안에 이해한다.
2. 해결 분야 또는 대표 사례를 탐색한다.
3. 참여 기업과 지원 근거를 확인한다.
4. 상담 신청 버튼을 누른다.
5. 최소 정보만 입력해 신청한다.
6. 운영자는 신청 내용을 확인하고 후속 상태를 기록한다.

7. MVP 범위

포함

- 가치 제안이 명확한 히어로 섹션
- 해결 분야 또는 제공 역량 소개
- 대표 성공 사례
- 참여 기업 또는 파트너 신뢰 요소
- 진행 절차와 자주 묻는 질문
- 상담 신청 폼 및 개인정보 동의
- 기본 분석 이벤트와 접근성, 반응형 레이아웃

제외

- 공개 기업 검색·필터 플랫폼
- 회원가입과 개인 대시보드

- 계약, 결제, 정산
- 자동 추천 알고리즘
- 운영자용 대규모 CMS

8. 성공 지표 초안

단계	지표	초기 판정 기준	비고
도달	유효 방문 수	기준선 수집	캠페인별 구분 필요
관심	핵심 섹션 도달률	50% 이상 가설	스크롤 이벤트
의도	주요 CTA 클릭률	5% 이상 가설	위치별 비교
전환	상담 신청 완료율	2% 이상 가설	유효 방문 대비
품질	유효 리드 비율	50% 이상 가설	운영 판정 기준 필요
성과	실제 연결 비율	기준선 수집	후속 CRM 또는 시트 연동

9. 검증 계획

가설	검증 방법	판정 기준
고객은 기업 목록보다 문제 해결 사례를 먼저 원한다	수요 기업 5곳 인터뷰와 두 가지 메시지 테스트	사례 중심 안을 선호한 응답이 과반
짧은 폼이 상담 의도를 충분히 포착한다	필드별 이탈률 및 운영자의 추가 질문 기록	완료율을 해치지 않으며 후속 연락에 필요한 정보 확보
지역성과 공공기관 신뢰가 전환에 도움을 준다	신뢰 요소 노출 전후 CTA 클릭 비교	클릭 또는 신청률 개선

10. 결정이 필요한 항목

- 서비스의 공식 명칭과 운영 주체
- 주요 수요 산업군과 지역 범위
- 소개할 SW 분야와 기업 선정 기준
- 상담 접수 후 응답 SLA
- 개인정보 보관 기간과 담당자
- 핵심 사례에 사용할 실제 성과 수치

관련 문서

- 비즈니스 모델
- 개발 기본설계

비즈니스 모델 정의

1. 모델 요약

이 서비스는 SW 수요 기업의 탐색 비용을 줄이고, 지역 SW 공급 기업에는 조건이 맞는 상담 기회를 제공하며, 운영 기관에는 연결 성과를 측정할 수 있는 채널을 제공합니다.

초기 BM은 매출 극대화보다 검증 가능한 매칭 성과에 초점을 둡니다. 수요와 공급의 품질이 확인된 뒤 유료 모델을 단계적으로 검토합니다.

2. 이해관계자별 가치

이해관계자	해결할 일	제공 가치	기대 행동
수요 기업	적합한 SW 파트너 탐색	사례 기반 비교, 신뢰 근거, 간단한 상담 신청	상담 요청
공급 기업	유효한 영업 기회 확보	역량과 사례 노출, 적합 리드 연결	정보 제공 및 상담 대응
운영 기관	사업 성과 창출과 보고	유입부터 연결까지의 측정 데이터	품질 관리와 후속 조정
협력 파트너	공동 사업 기회 발굴	검증된 네트워크와 수요 신호	공동 프로그램 참여

3. Business Model Canvas

영역	초기 정의
고객 세그먼트	SW 도입 수요 기업, 지역 SW 공급 기업, 사업 운영 기관
가치 제안	문제에 맞는 SW 역량을 사례로 이해하고 빠르게 연결
채널	랜딩 페이지, 검색, 기관 캠페인, 행사, 파트너 추천
고객 관계	셀프 탐색 후 운영자 기반 상담·매칭
핵심 활동	콘텐츠 검수, 사례 큐레이션, 상담 접수, 매칭, 성과 추적
핵심 자원	기업·사례 데이터, 운영 네트워크, 브랜드 신뢰, 분석 데이터
핵심 파트너	지역 SW 기업, 산업별 협회, 지원기관, 전문 컨설턴트
비용 구조	기획·개발, 콘텐츠 제작, 캠페인, 상담 운영, 데이터 관리
수익원	초기에는 사업 예산, 이후 유료 노출·검증·성공 기반 모델 검토

4. 수익 모델 가설

수익 모델은 리드 품질과 이해관계자 신뢰를 훼손하지 않는 순서로 검증합니다.

ID	모델	지불 주체	제공 단위	검증할 위험
BM-001	기관 운영형	운영·지원 기관	연간 구축 및 운영	예산 종료 후 지속 가능성

ID	모델	지불 주체	제공 단위	검증할 위험
BM-002	검증 프로필형	공급 기업	검수된 기업·사례 프로필	유료 노출이 신뢰를 훼손할 가능성
BM-003	캠페인형	공급 기업 또는 파트너	산업별 공동 캠페인	리드 품질과 비용 대비 효과
BM-004	성과형	공급 기업	유효 상담 또는 계약 성과	성과 귀속과 추적의 어려움

권장 검증 순서

1. `BM-001` 로 운영과 수요를 검증합니다.
2. 유효 리드 기준이 안정되면 `BM-002` 또는 `BM-003` 을 소규모로 시험합니다.
3. 계약 추적과 이해관계자 합의가 갖춰진 뒤에만 `BM-004` 를 검토합니다.

5. 핵심 단위 경제성

아래 값은 실제 데이터를 입력할 수 있도록 남겨 둔 계산 구조입니다.

```
유효 리드당 비용(CPL) = 총 획득 비용 / 유효 리드 수
매칭당 비용          = 총 운영 비용 / 실제 매칭 수
리드→매칭 전환율      = 실제 매칭 수 / 유효 리드 수
매칭→계약 전환율      = 확인된 계약 수 / 실제 매칭 수
공급자 유지율         = 다음 기간 재참여 공급자 / 직전 기간 참여 공급자
```

6. 운영 퍼널

```
방문 → 관심 콘텐츠 확인 → CTA 클릭 → 상담 신청
    → 유효성 검토 → 공급자 매칭 → 상담 진행 → 성과 확인
```

각 단계에는 날짜, 유입 경로, 담당자, 상태 변경 사유를 남겨야 합니다. MVP에서는 분석 도구와 운영 시트를 연결하고, 반복 업무가 확인된 뒤 전용 관리 기능을 만듭니다.

7. 신뢰와 정책 원칙

- 유료 참여 여부와 기업 추천 순위를 혼동시키지 않습니다.
- 사례의 고객명, 성과 수치, 로고는 사용 권한을 확인합니다.
- 상담 정보는 명시한 목적과 기간 안에서만 보관합니다.
- 공급 기업 선정 및 제외 기준을 내부 운영 문서로 남깁니다.
- 성과 수치는 `문의` , `유효 리드` , `매칭` , `계약` 을 구분해 보고합니다.

8. 90일 검증 로드맵

1~30일: 문제와 메시지

- 수요 기업 인터뷰 5건 이상

- 공급 기업 인터뷰 5건 이상
- 핵심 사례 3건의 사용 권한과 근거 확보
- 메시지와 CTA 문구 검증

31~60일: MVP와 운영

- 랜딩 MVP 공개
- 상담 처리 기준과 담당자 확정
- 이벤트 및 운영 퍼널 측정
- 첫 유효 상담 데이터 수집

61~90일: 채널과 BM

- 채널별 유효 리드 비용 비교
- 공급 기업의 리드 품질 평가
- `BM-002` 또는 `BM-003` 가격 인터뷰
- 다음 분기 투자 또는 중단 기준 결정

9. Go / Pivot / Stop 기준

판단	기준 예시
Go	반복 유입 채널과 유효 매칭이 확인되고 공급자의 재참여 의사가 있음
Pivot	문의는 발생하지만 대상·메시지·매칭 품질 중 한 축이 지속적으로 낮음
Stop	인터뷰와 캠페인 후에도 문제 강도와 지불 의사가 확인되지 않음

관련 문서

- 기회 정의
- 개발 기본설계

개발 기본설계

1. 설계 목표

초기에는 빠르게 배포할 수 있는 단일 웹 애플리케이션으로 시작하되, 콘텐츠와 상담 데이터가 늘어날 때 무리 없이 분리할 수 있게 설계합니다.

2. 제안 기술 스택

영역	제안	이유	확정 시점
웹	Next.js App Router + TypeScript	SEO, 서버 렌더링, 폼 처리, 배포 생태계	구현 시작 전
스타일	Tailwind CSS 또는 CSS Modules	작은 팀의 빠른 UI 반복	디자인 시스템 확정 시
콘텐츠	코드 기반 정적 콘텐츠	초기 변경 이력과 검수가 단순함	MVP
상담 저장	서버 액션/API + 외부 시트·CRM 또는 DB	운영 방식에 따라 교체 가능	개인정보 검토 후
검증	Vitest, Testing Library, Playwright	단위·컴포넌트·핵심 흐름 분리	초기 설정 시
배포	Vercel 또는 조직 표준 환경	Next.js 운영 단순화	보안·조달 검토 후
분석	조직 승인 분석 도구	퍼널 이벤트 측정	개인정보 정책 확정 후

기술 이름은 제안이며 확정 사항이 아닙니다. 조직의 보안, 조달, 운영 제약이 우선합니다.

3. 시스템 경계

```
〔방문자 브라우저〕
      |
      v
〔Next.js 웹 애플리케이션〕
  |      |        |
  |      |        +--〉〔분석 서비스〕
  |      +--------------〉〔콘텐츠/이미지〕
  +----------------------〉〔상담 접수 API〕
                         |
                         +--〉〔메일/알림〕
                         +--〉〔운영 시트·CRM 또는 DB〕
```

4. 개발 폴더 구조

```
development/
├── README.md
├── apps/
│   └── web/
│       ├── public/
│       │   └── images/
│       └── src/
│           ├── app/            # 라우트, 메타데이터, 서버 엔드포인트
```

```
|           ├── components/
|           |   ├── common/       # 버튼, 헤더, 푸터 등 공통 UI
|           |   └── sections/     # 히어로, 사례, FAQ, CTA 등 랜딩 섹션
|           ├── lib/              # 검증, 분석, 외부 연동, 공통 함수
|           └── styles/           # 전역 토큰과 스타일
├── packages/
|   ├── config/                   # 공유 lint, TypeScript, 환경 설정
|   └── ui/
|       └── src/                  # 재사용 가능한 디자인 시스템 요소
└── tests/
    └── e2e/                    # 방문→상담 신청 핵심 흐름
```

하나의 랜딩 사이트만 운영하는 동안에는 `packages/ui`를 억지로 별도 패키지로 배포하지 않습니다. 두 번째 앱이나 실제 재사용 요구가 생길 때 독립 패키지로 승격합니다.

5. 화면 구성

순서	섹션	연결 요구사항	핵심 역할
1	Header	PRD-003	주요 CTA를 항상 찾을 수 있게 함
2	Hero	PRD-001	대상, 문제, 제공 가치를 즉시 설명
3	Problem / Solution	PRD-001	방문자의 상황과 제공 역량을 연결
4	Capabilities	PRD-001	지원 가능한 SW 분야를 구조화
5	Case Studies	PRD-002	실제 문제, 실행, 성과를 증명
6	Partners / Trust	PRD-002	운영 주체와 참여 근거를 제공
7	Process	PRD-003	상담 이후의 절차와 예상 시간을 설명
8	FAQ	PRD-003	주요 불확실성과 이탈 요인을 해소
9	Contact CTA / Form	PRD-003	상담 의도를 수집
10	Footer	정책	연락처와 법적 문서 연결

6. 콘텐츠 모델

```ts
type CaseStudy = {
  id: string;
  title: string;
  industry: string;
  problem: string;
  solution: string;
  outcomes: Array<{ label: string; value: string; evidence?: string }>;
  image: { src: string; alt: string };
  published: boolean;
};

type Inquiry = {
  companyName: string;
  contactName: string;
  email: string;
  phone?: string;
  need: string;
  consent: boolean;
  source?: string;
};
```

실제 저장 전에 입력 길이, 허용 형식, 필수 항목, 보관 기간을 개인정보 담당자와 확정해야 합니다.

7. 상담 처리 흐름

1. 클라이언트와 서버에서 입력을 검증합니다.
2. 봇 방지 정책을 적용합니다.
3. 동의 시점과 정책 버전을 함께 기록합니다.
4. 저장 또는 외부 시스템 전달 성공 후에만 완료 화면을 표시합니다.
5. 실패 시 사용자가 입력을 잃지 않도록 복구 가능한 오류를 보여 줍니다.
6. 민감한 입력값은 로그와 분석 이벤트에 남기지 않습니다.

8. 분석 이벤트 초안

이벤트	발생 조건	금지 데이터
page_view	페이지 유효 조회	개인 식별 정보
section_view	핵심 섹션이 화면에 노출	자유 입력 내용
cta_click	위치별 주요 CTA 클릭	연락처
form_start	첫 폼 상호작용	입력값
form_submit_success	서버 접수 성공	이름, 이메일, 전화번호
form_submit_error	접수 실패	원문 오류나 입력값

9. 비기능 요구사항

- 모바일 360px부터 데스크톱까지 주요 흐름이 끊기지 않아야 합니다.
- 키보드만으로 탐색하고 폼을 제출할 수 있어야 합니다.
- 이미지에는 의미 있는 대체 텍스트를 제공합니다.
- 목표 성능은 Core Web Vitals의 `good` 범위를 기준으로 잡습니다.
- 비밀 값은 환경 변수로 관리하며 브라우저 번들에 포함하지 않습니다.
- 의존성 오류, 폼 실패, 외부 연동 실패를 관찰할 수 있어야 합니다.
- 개인정보를 로그, URL, 분석 속성에 포함하지 않습니다.

10. 테스트 전략

층	대상	예시
정적 검사	타입, lint, 형식	잘못된 콘텐츠 필드 차단
단위 테스트	검증 및 변환 함수	필수 값, 길이, 잘못된 이메일
컴포넌트 테스트	폼과 상호작용 UI	오류 메시지, 키보드 사용

층	대상	예시
E2E	핵심 사용자 흐름	방문, CTA, 유효 신청, 실패 복구
수동 검수	반응형·콘텐츠·접근성	실제 기기와 주요 브라우저

11. 환경 구분

- `local` : 개발자 로컬, 테스트 데이터만 사용
- `preview` : 변경 검토, 실제 개인정보 수집 금지
- `production` : 승인된 도메인과 저장소만 사용

환경 변수의 이름과 책임자는 실제 외부 연동 선택 후 `.env.example` 에 문서화합니다. 비밀 값 자체는 저장소에 커밋하지 않습니다.

12. 기술 결정 기록

ID	결정	상태
ADR-001	제품 문서와 실제 소스를 분리한다	채택
ADR-002	MVP는 단일 Next.js 웹 앱으로 시작한다	제안
ADR-003	초기 콘텐츠는 코드 기반으로 관리한다	제안
ADR-004	운영 흐름이 검증될 때까지 대규모 CMS를 도입하지 않는다	제안
ADR-005	개인정보 저장소는 정책 검토 후 선택한다	보류

13. 구현 전 확인 목록

- 브랜드, 카피, 로고와 이미지 사용 권한
- 폼 필드, 동의 문구, 보관 기간
- 상담 저장소와 알림 수신자
- 분석 도구와 쿠키 정책
- 배포 도메인, 보안 헤더, 장애 대응 담당자
- 브라우저 지원 범위와 성능 기준

관련 문서

- 기회 정의
- 비즈니스 모델
- 개발 작업 안내

개발 작업 안내

이 디렉터리는 실제 사이트 코드, 공유 UI, 설정, 테스트를 보관합니다. 상위 `docs/` 는 제품과 사업의 기준이고, 이곳은 그 기준을 실행 가능한 결과로 바꾸는 공간입니다.

시작 전 확인

구현을 시작하기 전에 다음 항목을 확정합니다.

1. 기회 정의의 MVP 범위
2. 비즈니스 모델의 운영 퍼널
3. 개발 기본설계의 제안 기술 스택
4. 브랜드 자산, 실제 사례, 개인정보 처리 기준

작업 단위

권장 구현 순서는 다음과 같습니다.

1. 워크스페이스와 Next.js 앱 초기화
2. 디자인 토큰, 공통 레이아웃, 접근성 기준 설정
3. 정적 랜딩 섹션 구현
4. 사례와 기업 콘텐츠 모델 연결
5. 상담 폼의 검증, 저장, 오류 복구 구현
6. 분석 이벤트 연결
7. 단위·컴포넌트·E2E 테스트 추가
8. preview 검수 후 production 배포

경계 규칙

- `apps/web/src/app` 은 라우팅과 페이지 조합을 담당합니다.
- `components/sections` 는 랜딩의 의미 단위 섹션을 담당합니다.
- `components/common` 은 앱 내부에서 반복되는 UI를 담당합니다.
- `packages/ui` 는 실제로 여러 위치에서 재사용되는 요소만 둡니다.
- `lib` 은 외부 연동, 검증, 분석처럼 UI와 분리할 로직을 둡니다.
- `tests/e2e` 는 구현 세부사항보다 사용자의 완료 가능한 흐름을 검증합니다.

완료 정의

기능은 다음 조건을 모두 만족할 때 완료로 봅니다.

- 연결된 `PRD-###`, `BM-###`, `ADR-###`가 명확합니다.
- 모바일과 데스크톱에서 동작합니다.
- 키보드 탐색과 오류 메시지를 포함한 접근성을 확인했습니다.
- 정상 흐름과 실패 흐름을 테스트했습니다.
- 분석 이벤트에 개인정보가 포함되지 않습니다.
- 문서와 실제 동작이 일치합니다.

아직 생성하지 않은 항목

현재는 구조 정의 단계이므로 `package.json`, 프레임워크 설정, 배포 설정은 만들지 않았습니다. 기술 스택 확정 후 `development/`를 워크스페이스 루트로 초기화합니다.