AIO 1기 · FINAL PROJECT · GUIDE

기획서 잘 쓰는 법

읽는 사람의 질문에 미리 답하는 문서 만들기

문제 → 사용자 → 범위 → 설계 → 검증

WHAT IS IT

기획서는 결정을 돕는 문서입니다

읽는 사람이 “이걸 왜, 무엇을, 어떻게, 언제까지 만들고, 성공은 어떻게 판단하나?”를 한 번에 판단할 수 있어야 합니다.

설득 문서

이 프로젝트가 할 만한 가치가 있다는 근거

팀의 설계도

무엇을 만들고 무엇을 안 만들지에 대한 합의

판단의 기준

나중에 범위·우선순위를 바꿀 때 돌아올 곳

WHO READS IT

읽는 사람은 세 명입니다

평가자

강사 · 멘토 · 심사위원

- 문제가 진짜인가?
- 해법이 논리적인가?
- 기간 안에 가능한가?

팀원

함께 만드는 사람

- 내가 만들 것은 무엇인가?
- 어디까지가 완료인가?
- 언제까지 해야 하나?

미래의 우리

5주 뒤, 면접장에서

- 그때 왜 이렇게 정했지?
- 무엇을 일부러 뺐지?
- 이 프로젝트를 어떻게 설명하지?

세 사람의 질문 중 하나라도 답이 없으면, 그 부분이 기획서의 빈칸입니다.

GOOD PROPOSALS

좋은 기획서의 다섯 가지 특징

01	문제에서 출발한다	기술이 아니라 누군가의 불편에서 시작
02	구체적이다	누가 · 언제 · 무엇을이 이름과 숫자로 보인다
03	한 줄로 이어진다	문제 → 해법 → 기능 → 일정이 서로 근거가 된다
04	확인할 수 있다	완료와 성공을 무엇으로 판단할지 적혀 있다
05	현실적이다	인원 · 기간 · 데이터로 가능한 범위만 약속한다

COMMON MISTAKES

흔한 실수와 고쳐 쓴 문장

실수	이렇게 쓰면 약하다	이렇게 고친다
기술부터 시작	LangGraph로 멀티 에이전트를 구현한다	장애 대응에 [N시간]이 걸리는 운영자의 수작업을 줄인다
모호한 목표	AI로 업무를 효율화한다	일정 충돌 확인 시간을 [N분]에서 [M분]으로 줄인다
기능 나열	검색, 요약, 분석, 보고서, 알림…	사용자 한 명이 끝까지 가는 시나리오 1개로 기능을 묶는다
전부 다 하기	Must 기능 15개	Must 3~5개 · 나머지는 Should·Could로
근거 없는 주장	많은 사람이 불편을 겪고 있다	출처가 있는 자료나 인터뷰 [N명] 결과를 붙인다

대괄호 [ ]는 팀이 조사해서 채울 자리입니다 — 숫자를 지어내지 마세요

STRUCTURE

기획서의 뼈대는 다섯 개의 질문

WHY
왜
배경 · 문제 · 타깃 사용자

WHAT
무엇을
솔루션 · 시나리오 · 기능 · 요구사항

HOW
어떻게
아키텍처 · 기술 스택 · 데이터

WHEN · WHO
언제 · 누가
일정 · 마일스톤 · 역할

HOW WELL
얼마나
성공 지표 · 리스크와 대응

‘왜’에 가장 많은 시간을 쓰세요. ‘왜’가 약하면 나머지가 아무리 화려해도 설득력이 없습니다.

WHY · PROBLEM

문제를 한 문장으로 진술하기

문제 진술 공식

[누가] [어떤 상황에서] [무엇 때문에] [어떤 손해]를 겪는다

예시

소규모 개발팀의 운영 담당자는 야간 장애가 나면 로그를 일일이 뒤져 원인을 찾느라 복구까지 [N시간]이 걸리고, 그동안 서비스가 멈춘다.

‘왜?’를 다섯 번

눈에 보이는 불편에서 “왜 그런가?”를 반복해 근본 원인까지 내려가면, 해결해야 할 진짜 문제가 보입니다.

WHY · USER

사용자를 한 사람으로 그리기 — 페르소나

PERSONA CARD · 예시

김운영 (32) · 스타트업 인프라 담당

목표 — 장애를 빨리 끝내고 재발을 막고 싶다

불편 — 혼자서 여러 서버를 보고, 원인 찾기가 오래 걸린다

지금 쓰는 도구 — 모니터링 알림, 로그 검색, 메신저

성공의 모습 — 알림을 받으면 원인과 조치가 이미 정리되어 있다

왜 한 사람으로?

- “모든 사용자”를 위한 서비스는 아무도 만족시키지 못한다
- 기능을 넣을지 말지 “김운영에게 필요한가?”로 판단
- 팀 전원이 같은 사람을 떠올리며 만든다

대표 페르소나 1명 + 보조 1명이면 충분합니다

WHAT · SCENARIO

Before / After 시나리오로 보여주기

BEFORE · 지금

새벽 장애 알림 → 로그 직접 검색 → 원인 추측 · 재시도 → [N시간] 뒤 복구

AFTER · 우리 서비스

이상 감지 → 진단 에이전트 분석 → 자동 복구 실행 → 결과 보고만 확인

시나리오 한 줄이 곧 **시연 대본**이 됩니다. 발표 데모에서 이 흐름을 그대로 보여주세요.

WHAT · DIFFERENTIATION

기존 방식과 비교해 차별점 보여주기

비교 기준	수작업	기존 도구 [이름]	우리 서비스
원인 파악	사람이 로그 검색	알림만 제공	에이전트가 진단
조치	수동 실행	수동 실행	자동 복구 + 승인
오류 시 대응	사람이 재시도	없음	자기 성찰 후 재시도
과정 확인	기록 없음	일부 로그	사고 과정 대시보드

비교 기준은 **사용자가 신경 쓰는 것**으로 고르세요. 기술 이름이 아니라 결과의 차이를 적습니다.

WHAT · SCOPE

기능에 우선순위 매기기 — MoSCoW

MUST

없으면 실패

핵심 시나리오가 끝까지 돌아가는 데 꼭 필요한 것 · 3~5개

SHOULD

있으면 좋음

가치는 크지만 없어도 시연은 가능한 것

COULD

여유 있으면

일정이 남으면 추가할 것

WON'T

이번엔 안 함

일부러 뺀 것 — 적어 두면 범위 논쟁이 줄어듭니다

시나리오에 안 나오는 기능 → Could 이하

Must가 5개를 넘으면 범위가 과하다는 신호

WHAT · REQUIREMENTS

요구사항은 확인할 수 있게 씁니다

확인할 수 없는 문장	확인할 수 있는 문장
빠르게 응답한다	첫 응답 토큰이 [N초] 안에 화면에 표시된다
안전하게 처리한다	로그인하지 않은 요청은 401로 거부한다
오류에 강하다	외부 API 실패 시 [N회] 재시도 후 대체 응답을 준다
사용하기 쉽다	처음 쓰는 사용자가 안내 없이 [핵심 작업]을 끝낸다

주어 — 시스템이 / 사용자가

조건 — 언제 · 어떤 입력에서

기준 — 숫자 · 상태 · 결과

HOW · ARCHITECTURE

구성도는 한 장으로, 흐름이 보이게



- 왼쪽에서 오른쪽으로, 요청이 흐르는 방향대로
- 외부 서비스는 점선 등으로 구분
- 색과 선 모양의 범례 달기
- 화살표 옆에 오가는 데이터 이름 적기
- 박스마다 쓰는 기술 이름 한 줄
- 박스는 10개 안팎까지

HOW WELL · METRICS

성공을 무엇으로 판단할지 미리 정하기

사용자 가치 지표

- 작업 처리 시간 (Before → After)
- 사람이 개입한 횟수
- 사용자 만족도 · 재사용 의향

시스템 품질 지표

- 응답 지연 시간
- 자동 복구 성공률 · 평균 복구 시간
- 보안 테스트 차단 결과

좋은 지표의 조건

측정 방법이 정해져 있고 · 기준값과 목표값이 있고 · 시연이나 로그로 보여줄 수 있다

WHEN · WHO

일정은 마감에서 거꾸로 세웁니다

최종 발표 → 리허설 · 버퍼 → 통합 테스트 → 기능 구현 → 오늘

마일스톤마다 결과물

“무엇이 돌아가는가”로 정의

작업은 1~3일 단위

더 크면 쪼개야 진행이 보입니다

담당자는 한 명

“다 같이”는 아무도 안 한다는 뜻

버퍼를 남긴다

예상 못 한 문제는 반드시 생깁니다

HOW WELL · RISKS

리스크는 미리 적어 두면 관리가 됩니다

리스크	가능성	영향	대응 계획
외부 API 비용·호출 제한	중간	높음	캐시 · 모의 응답 · 사용량 모니터링
필요한 데이터를 못 구함	중간	높음	공개 데이터 · 합성 데이터로 대체
새 기술 학습이 오래 걸림	높음	중간	첫 주에 작은 기술 검증(PoC) 먼저
팀원 일정 공백	낮음	중간	보조 담당자 지정 · 매일 기록 공유
시연 중 오류	중간	높음	시연 영상 백업 · 고정된 시연 데이터

가능성·영향은 예시 판단입니다 — 팀 상황에 맞게 다시 평가하세요

WRITING TIPS

읽히는 문서를 만드는 여섯 가지 습관

두괄식
결론을 먼저, 근거는 그다음

한 장 한 메시지
제목만 읽어도 흐름이 이어지게

숫자와 그림
설명 세 줄보다 표 하나, 그림 하나

용어 통일
같은 것은 끝까지 같은 이름으로

출처 표시
인용한 자료와 수치에는 출처를

가정은 가정이라고
확인 못 한 내용은 '가정'으로 명시

AI 도구로 초안을 만들었다면, 사실과 수치는 반드시 직접 확인하고 우리 팀의 말로 다시 쓰세요.

SELF CHECK

제출 전 셀프 체크리스트

- ✓ 문제를 한 문장으로 말할 수 있다
- ✓ 타깃 사용자가 구체적인 한 사람이다
- ✓ 수치와 주장에 근거나 출처가 있다
- ✓ Before / After 시나리오가 있다
- ✓ 기존 방식과의 차이가 표로 보인다
- ✓ Must 기능이 5개 이하다
- ✓ 요구사항마다 확인 기준이 있다
- ✓ 구성도 한 장으로 흐름이 보인다
- ✓ 성공 지표와 측정 방법이 있다
- ✓ 일정·담당자·리스크 대응이 있다

마지막 확인: 팀 밖의 사람에게 5분 설명했을 때, 질문 없이 이해했는가?

좋은 기획서는
읽는 사람의 질문에
미리 답합니다

왜 | 무엇을 | 어떻게 | 언제 · 누가 | 얼마나

Q & A