

[주니어 개발자 채용] 실전 과제 평가기준 작성

평가기준



PR 평가 기준 (LoopOS 주니어 Flutter 개발자)

본 평가지표는 **Clean Architecture** 완성도보다

“빠른 실행력 · 우선순위 판단 · 사용자 흐름 · 기술 기본기”를 보기 위한 기준입니다.

1) 빠른 실행력 (Speed & Delivery) — 30%

확인 포인트

- 3~4시간 안에서 **동작하는 흐름(MVP)**이 구현되었는가?
- 최소 요구사항(타이머/기록 등) 중 **1~2개 이상**이 동작하는가?
- 과제 범위를 스스로 축소하고 핵심을 우선 구현했는가?
- 불필요한 구조/최적화에 빠지지 않았는가?

합격 신호

- 타이머 MVP라도 정상 동작
- 메트로놈/기록 화면 등 일부라도 실행됨
- 시간 안에서 적절히 타협한 흔적
- README에 “어디까지 했고 어디는 못 했는지” 명확히 기술됨

탈락 신호

- 아무 기능도 동작하지 않음

- 구조 세팅만 있음 (UX 흐름 없음)
 - 사소한 디테일에 시간을 과도하게 투자
 - 결과물이 시간 대비 불합리하게 빈약함
-

2) 우선순위 판단 (Prioritization) — 30%

확인 포인트

- README에 무엇을 먼저 만들었는지 / 왜 그렇게 결정했는지 설명되어 있는가?
- LoopOS 철학(빠른 실험 → 가설 검증)에 맞는 결정인가?
- 핵심 기능 흐름을 먼저 잡았는가?
- 과제 범위를 스스로 합리적으로 축소했는가?

합격 신호

- 구현/포기 부분이 논리적으로 설명됨
- 중요한 기능부터 순서 있게 개발됨
- 우선순위 판단 기준이 명확함
- AI 도구 활용 이유가 설명됨

탈락 신호

- 전부 하려다 전부 미완성
 - 우선순위 사고가 전혀 보이지 않음
 - README가 템플릿 수준이며 사고 부재
 - 비본질적인 영역에 과몰입
-

3) 사용자 경험 흐름 (UX Flow Sense) — 20%

확인 포인트

- 타이머 → 기록 → 저장 일련의 플로우가 끊김 없이 연결되는가?
- 사용자 관점에서 최소한의 자연스러운 흐름이 구현되었는가?
- 버튼/네비게이션의 위치/흐름이 직관적인가?
- 기록 UI가 과하게 복잡하지 않은가?

합격 신호

- 타이머 종료 → 기록 화면 이동이 자연스럽다
- 저장 완료 후 피드백이 있다
- 최소 UX지만 맥락이 맞고 불편하지 않음

탈락 신호

- 화면 간 이동이 어색하거나 단절됨
- UX가 개발자 중심(사용자 맥락 無)
- 기록 UI가 지나치게 무겁거나 접근 어려움

4) 기술 기본기 (Technical Foundation) — 20%

확인 포인트 (Flutter 모를 때도 쉽게 판단 가능)

- 파일 구조가 불필요하게 비대하지 않은가? (한 파일 300줄↑는 위험)
- 위젯/함수/로직이 적절히 분리되어 있는가?
- 네이밍이 역할을 명확히 설명하는가?
- 상태관리를 일관된 방식으로 사용했는가?
- 비동기 처리(async/await) 흐름이 자연스러운가?

합격 신호

- 폴더/파일 분리가 깔끔함

- 위젯과 로직이 느슨한 결합
- 의미 있는 변수/함수명
- 구조는 거칠지만 논리 일관성 존재

탈락 신호

- 중첩 구조 5단계 이상의 스파게티 코드
 - temp/test/new 같은 네이밍
 - UI/상태/로직이 한 파일에 뒤섞임
 - 실행은 되지만 구조적으로 유지불가
-

5) 보너스 평가 (양수 시그널)

AI 활용 능력

- AI가 작성한 코드를 이해/수정/최적화한 흔적이 보이면 가점

README 품질

- 사고 흐름·우선순위·판단 기준이 잘 드러나면 +기본기 +리더십 잠재력으로 판단

실험정신

- “시간 부족으로 여기까지 했지만, 다음 단계는 이렇게 갈 수 있습니다” 같은 진술은 매우 긍정적
-

🔥 최종 합격 판단 기준

✓ 합격

- 동작하는 흐름이 있고
- 우선순위 판단이 논리적이며
- 사용자 중심 흐름이 자연스럽고

- 코드 구조가 미래의 개발 속도에 방해 안 되는 수준이면 합격

✓ 보류/불합격

- 아무 기능도 동작하지 않음
- README에 사고 과정이 없음
- 스파게티 코드 + 설명 부재
- UX 흐름이 연결되지 않음
- 우선순위 판단 능력 부재



PR 리뷰 시 체크리스트 (은향 전용)

- 동작 데모 또는 흐름 영상/스크린샷 존재
- 타이머 → 기록 플로우 자연스러움
- README에 “Why / 선택 이유 / 우선순위 판단” 정리됨
- 기능 구현 범위가 시간 대비 합리적
- 파일 구조가 과도하게 무너지지 않음
- 네이밍이 의미를 갖고 있음
- UX가 사용자 맥락에 맞음
- 학습/실험 태도가 보임

안녕하세요, 단님!

기술 과제를 안내드립니다.

과제 개요

실제 꼭꼭 서비스의 핵심 흐름인 **"타이머 → 식사 완료 → 감정/포만감 기록 → 저장"** 을
3-4시간 안에 MVP로 구현해보는 과제입니다.

> 가장 보고 싶은 것은 **"어떤 것을 먼저 만들고, 어떤 것을 과감히 포기했는지"**입니다.

🔗 저장소

<https://github.com/Eunhyang/kkokkkok-assignment>

🕒 과제 시간

3-4시간 (모든 요구사항 100% 구현할 필요 없음)

💎 기본 요구사항 (최소 1-2개 완성)

1. 카운트다운 타이머 (25분 기본)
2. 씹는 간격 시작적 피드백 (메트로놈)
3. 타이머 종료 → 식사 기록 화면 (감정/포만감)
4. 기록 저장 후 완료 표시

🚧 기술적 안내

- **AI 도구 사용 가능** (v0, Claude, Copilot 등) - 면접 시 설명 필요
- **상태관리 자유** (setState, Provider, Riverpod 등)
- **Clean Architecture 필수 아님**

📁 제출 방법

1. 저장소 클론
 2. `feature/timer-[이름]` 브랜치 생성
 3. 구현 후 PR 생성
 4. `SUBMISSION_README.md` 작성 (소요시간, 우선순위 판단 기준 포함)
-

자세한 내용은 저장소 **README.md**를 참고해주세요.
질문 있으시면 언제든 연락주세요!

<https://github.com/Eunhyang/kkokkkok-assignment/settings/access>

단님, 과제 시작 시 접속하실 수 있도록
GitHub private repo에 Collaborator 초대드렸습니다.

GitHub username: urbosaDev 로 등록했고,
수락하시면 feature/timer-[이름] 브랜치 만들어서 바로 작업 시작하시면 됩니다 :)