취업준비생을 위한

MVP

Minimum Viable Product

최소 기능 제품으로 프로젝트 완성하기

MVP 란 무엇인가 ?

MVP = Minimum Viable Product

사용자에게 가치를 전달할 수 있는 **가장 작은 버전**의 제품

핵심 원칙

- 핵심 기능만 포함
- " 완벽한 제품 " 이 아닌 " 검증 가능한 제품 "
- 빠르게 만들고 → 피드백 → 개선

MVP 의 목표

검증

아이디어가 맞는지 빠르게 확인

쉬운 비유 : 카페 창업

❌

완벽주의 접근

- 인테리어 완벽하게
- 메뉴 30 개 준비
- 직원 5 명 고용

결과 : 6 개월 준비
실패 시 : 큰 손실

✅

MVP 접근

- 푸드트럭에서 시작
- 아메리카노 1 종만
- 혼자서 운영

결과 : 2 주 준비
장점 : 반응 보고 확장 결정

취준생에게 MVP 가 중요한 이유

71%
프로젝트 일정 초과

89%
평균 일정 초과율

❌ 흔한 문제

- " 이것도 넣고 저것도 넣자 "
- → 기간 내 완성 못함
- → 포트폴리오에 못 넣음
- " 왜 완성 못했나요 ?" 질문

✅ MVP 사고방식

완성된 결과물 확보

전체 흐름 경험 (기획 → 배포)

확장 가능성 보여주기

- " 우선순위 정해 완성 " 스토리

MVP vs 프로토타입 vs 완제품

프로토타입

목적
아이디어 시연

특징
동작 안 해도 됨
외형만 구현

예시
Figma 목업

MVP ⭐

목적
핵심 가치 검증

특징
실제로 동작함!
핵심 기능만 포함

예시
동작하는 웹앱

완제품

목적
시장 출시

특징
모든 기능 포함
완전한 서비스

예시
(여기까진 안 가도 됨 !)

MVP 범위 설정 : MoSCoW 기법

Must 반드시 - 이거 없으면 의미 없음

Should 해야 함 - 중요하지만 없어도 동작

Could 할 수 있음 - 시간 남으면 추가

Won't 안 함 - 이번엔 안 함 (다음에)

⚠️ 핵심 원칙

"Must" 는

3 개 이하

로 제한하라

자문 질문 :

" 이 기능 없으면 프로젝트를 버려야 하나 ?"

예시 : To-Do 앱 MVP 설계

✅ MVP 범위 (16 시간)

Must (반드시)

- 할 일 추가
- 할 일 삭제
- 완료 / 미완료 토글

Should (시간 되면)

- 할 일 수정
- 필터 (전체 / 완료 / 미완료)

예상 시간 : **약 16 시간**

❌ Won't (이번엔 안 함)

- 로그인 / 회원가입
- 마감일 설정 + 알림
- 드래그로 순서 변경
- 다른 사람과 공유
- 카테고리 분류

💡 **백엔드 확장 포인트**

Java Spring Boot 로 동일 기능의 REST API 구축 가능

MVP 적합성 테스트 질문

1

"1 주일 안에 , 혼자서 완성할 수 있나요 ?"

Yes → 적절한 MVP | No → 범위 축소 필요

2

" 이 MVP 만 있어도 사용자가 뭔가를 할 수 있나요 ?"

Yes → MVP 성립 | No → 핵심 기능이 빠진 것

3

" 백엔드를 연결하면 더 좋아질 기능이 있나요 ?"

Yes → 좋은 MVP (풀스택 스토리) | 해당 없음 → 기획 재검토

흔한 실수 4 가지

❌ 기능 과잉 (Feature Creep)

" 이것도 넣으면 좋겠다 " → 끝없이 추가

해결 : Must 만 고수 , 나머지는 "v2 에서 "

❌ 기술 스택 잡착

"React + TypeScript + Redux..." → 학습에만 시간

해결 : 아는 기술로 먼저 만들기

❌ 완벽주의 함정

"CSS 가 아직 안 예뻐서 ..." → 배포 미룸

해결 : " 동작하면 배포한다 " 원칙

❌ 범위 축소 공포

" 이렇게 작으면 포폴이 안 될 것 같아요 "

해결 : 작아도 **완성된** 프로젝트가 낫다

핵심 정리 : 3 가지 원칙

1

완성이 완벽보다 낫다

Done is better than perfect

2

핵심만 남기고 버려라

80% 를 버려야 20% 가 빛난다

3

빠르게 만들고 , 피드백 , 개선

Build → Measure → Learn