

# 이력서부터 시작하는 커리어 셀프 브랜딩

## 무난하게 쓰고 무난하게 탈락하기

# 발표자 소개

## Software Enginner at ProtoPie

- 5년차 프론트엔드 개발자
- 프론트엔드 교육 경력 2년
  - 300명 이상의 주니어들 멘토링
- 경력직 이력서 피드백 경험 多
- 얼떨결에 CDC 운영진
- 에디터 개발자 모임 Editorium 운영



# 일반적인 이력서

## 왜 대부분 비슷할까?

빠른 학습과 높은 생산성을 바탕으로 협업을 중시하는 프론트엔드 개발자입니다.

### 요약(About)

협업을 중요하게 생각하며 문제 해결을 즐기는 개발자입니다.  
새로운 기술을 빠르게 습득하고 맡은 일을 책임감 있게 완수합니다.  
사용자 관점에서 더 나은 경험을 제공하기 위해 꾸준히 고민합니다.

### 핵심 역량(Skills)

- JavaScript / TypeScript / React
- Next.js / HTML / CSS
- 성능 최적화 / 코드 리팩토링

### 경력(Experience)

#### A사 | Frontend Developer | 2023.01 ~ 2025.12

- 서비스 프론트엔드 개발 및 유지보수
- 신규 기능 개발 및 UI 개선
- API 연동 및 상태 관리 구현
- 성능 개선과 코드 리팩토링 진행
- 팀 내 협업 프로세스 개선에 참여

#### B사 | Frontend Developer | 2021.06 ~ 2022.12

- 운영 중인 웹 서비스 기능 개발
- 공통 컴포넌트 개발
- QA 대응 및 버그 수정

# 내 이력서가 읽히는 현실

얼마나 꼼꼼하게 읽을까? 과연 사람이 읽기는 할까?

채용 담당자가 이력서를 읽는 시간

“ 7분 이상 읽지 않아요. ”

“ 이력서가 이번에 400장 들어왔어요. ”

“ 전부 다 AI 자동화 돌려서 1차로 걸렀어요. ”

# 이력서의 본질

한 번 생각해보자



# 이력서의 본질

한 번 생각해보자

이력서는 목적을 가진 짧은 글

그 중에서도 예술이 아닌 실용 문서

목적 : 면접으로 넘어가게 만드는 통과 문서

# 그렇다면?

## 내가 채용 담당자라면?



하루 온종일 꼼꼼하게 읽어야 하는 이력서

VS



일단 쪽 보니까 대충 괜찮아서 면접에서 검증하면 될 것 같은 이력서

# 대충 봐도 좋으려면 어떻게 해야할까?

한 번 생각해보자



# 강점을 살린 이력서 = 좋은 이력서

결국 이력서는 내 강점 뽑내기 문서



# 강점을 잘 살린다는게 뭘까?

## 한 번 생각해보자

빠른 학습과 높은 생산성을 바탕으로 협업을 중시하는 프론트엔드 개발자입니다.

### 요약(About)

협업을 중요하게 생각하며 문제 해결을 즐기는 개발자입니다.  
새로운 기술을 빠르게 습득하고 맡은 일을 책임감 있게 완수합니다.  
사용자 관점에서 더 나은 경험을 제공하기 위해 꾸준히 고민합니다.

### 핵심 역량(Skills)

- JavaScript / TypeScript / React
- Next.js / HTML / CSS
- 성능 최적화 / 코드 리팩토링

### 경력(Experience)

#### A사 | Frontend Developer | 2023.01 ~ 2025.12

- 서비스 프론트엔드 개발 및 유지보수
- 신규 기능 개발 및 UI 개선
- API 연동 및 상태 관리 구현
- 성능 개선과 코드 리팩토링 진행
- 팀 내 협업 프로세스 개선에 참여

#### B사 | Frontend Developer | 2021.06 ~ 2022.12

- 운영 중인 웹 서비스 기능 개발
- 공통 컴포넌트 개발

주도적으로 문제를 해결하고 원활한 협업을 지향하는 프론트엔드 개발자입니다.

사용자 중심의 서비스를 만들기 위해 노력하는 프론트엔드 개발자입니다.  
팀과의 커뮤니케이션을 중요하게 생각하며 주어진 목표를 꾸준히 달성해왔습니다.  
새로운 기술과 도구를 빠르게 학습하고 업무에 적용하는 것을 좋아합니다.

### 경력(Experience)

#### C사 | Frontend Engineer | 2022.03 ~ 2025.11

- B2C 웹 서비스 기능 개발 및 운영
- UI/UX 개선 작업 진행
- 성능 개선 30~40%
- API 연동 및 상태 관리 구현
- 운영 이슈 대응 및 버그 수정
- 코드 리팩토링 및 품질 개선 작업 수행

#### D사 | Frontend Developer | 2020.07 ~ 2022.02

- 신규 서비스 프론트엔드 개발 참여
- 관리자 페이지 기능 구현
- 재사용 가능한 컴포넌트 개발
- 디자인 시스템 적용 및 유지보수
- 배포/운영 과정 지원 및 이슈 대응

### 프로젝트(Projects)

프로젝트 1: 개인 블로그 시스템 개발

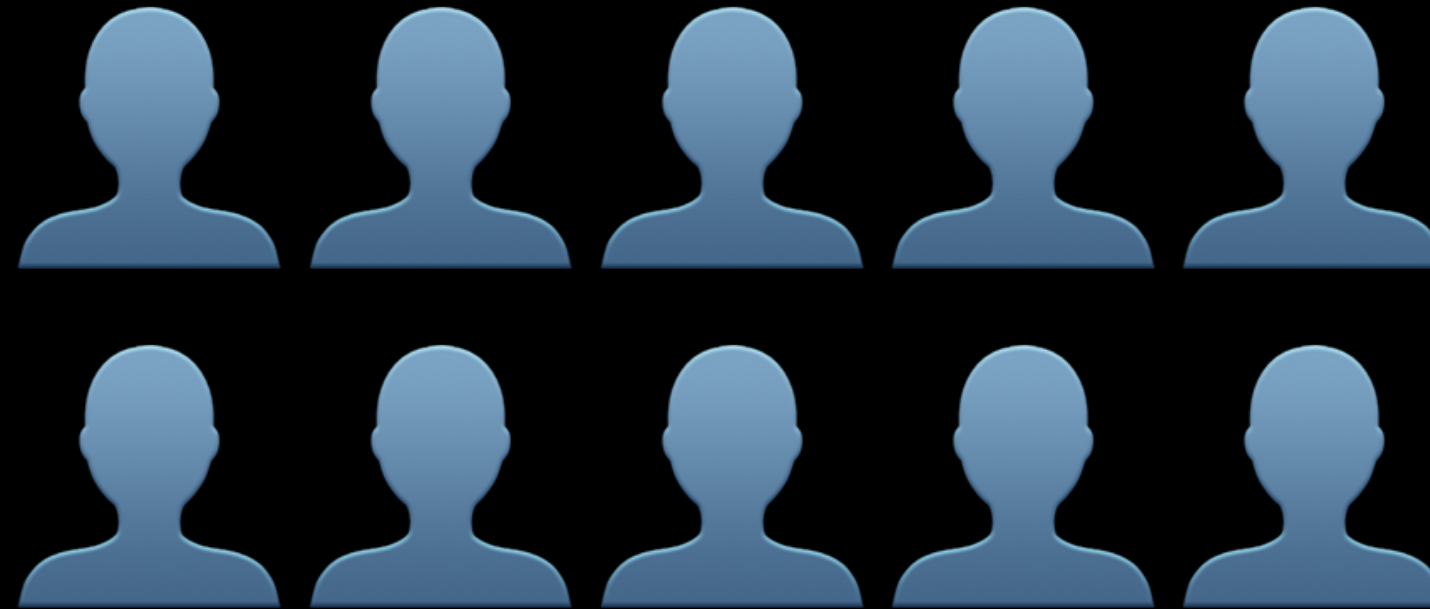
# 우리나라 사람들은 자기 PR을 못한다

## 너무 겸손한 우리나라 사람들

“이 정도는 다 해요.”

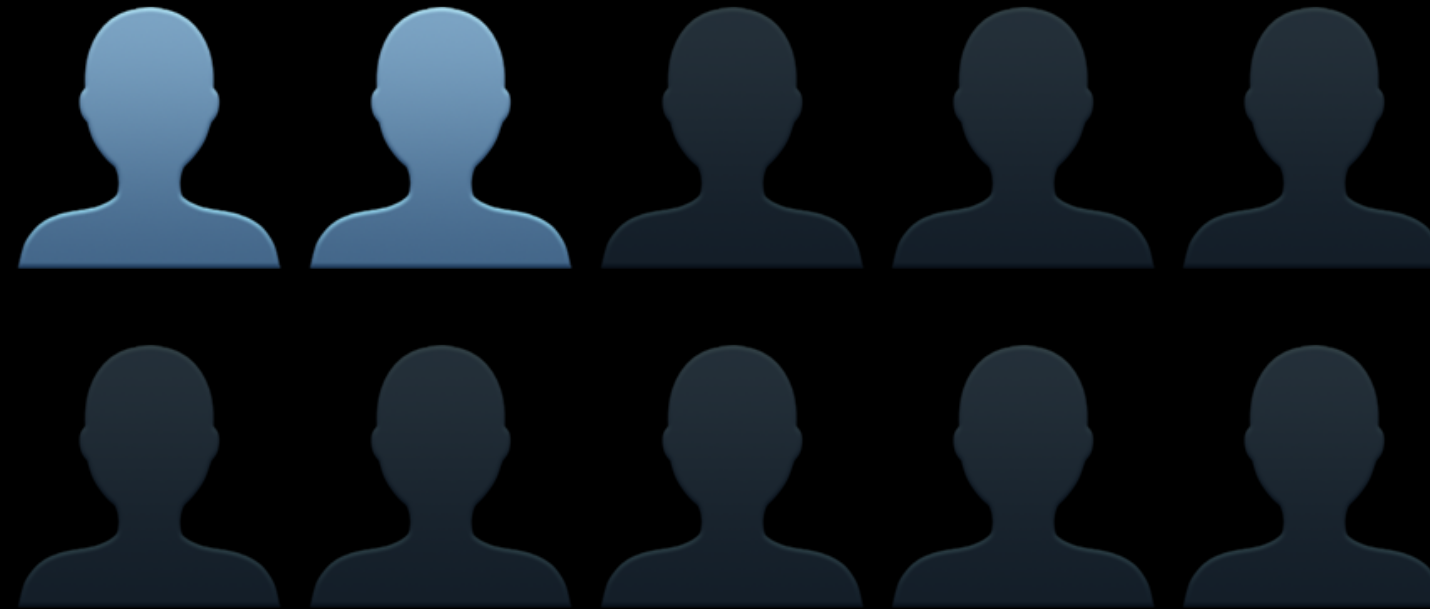
# 3등급 이야기

## 인식을 전환해보자



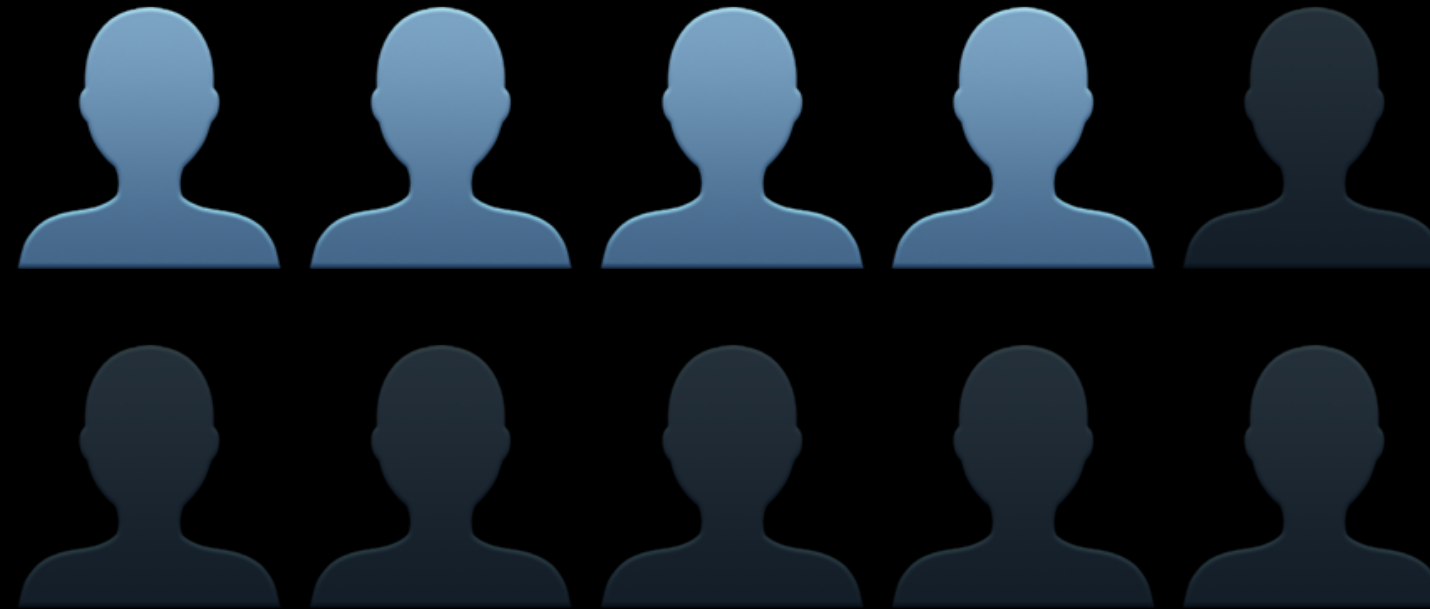
# 3등급 이야기

## 인식을 전환해보자



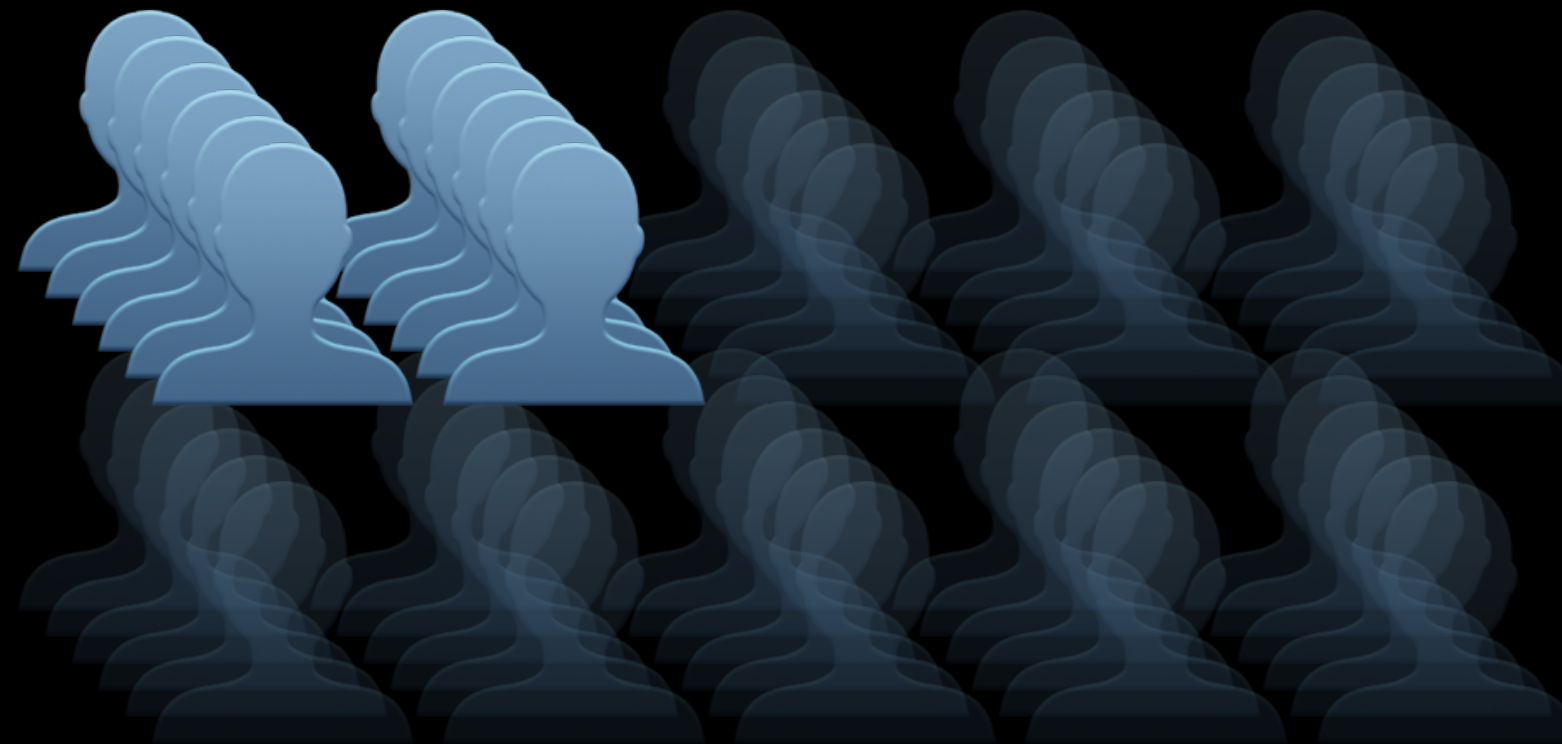
# 3등급 이야기

## 인식을 전환해보자



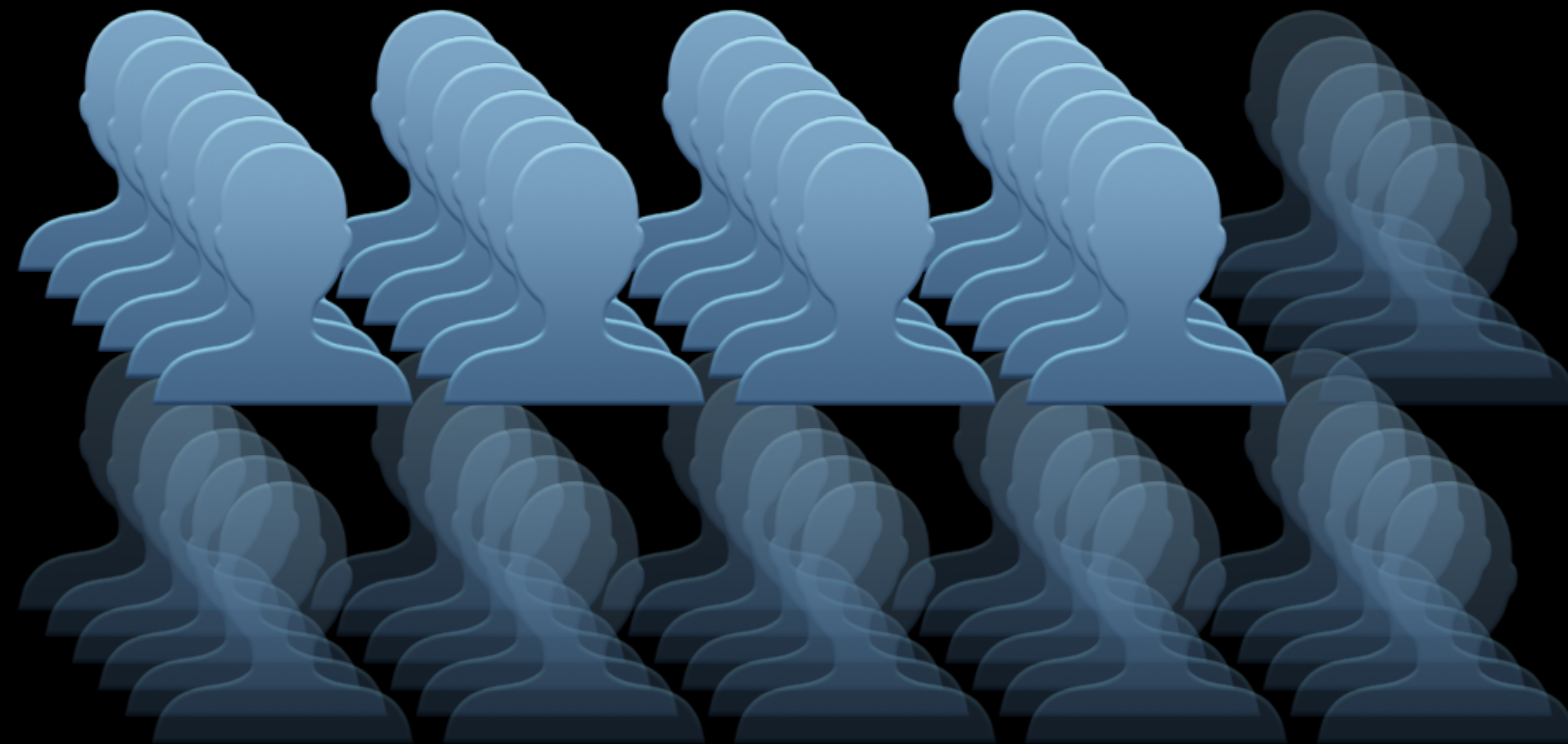
# 3등급 이야기

## 인식을 전환해보자



# 3등급 이야기

## 인식을 전환해보자



# 우리나라 사람들은 자기 PR을 못한다

너무 겸손한 우리나라 사람들

겸손/완곡한 표현이 습관 → 강점 숨겨버림

결국 오히려 다 비슷해지는 결과

# 만약 절대적으로 못한다고 느껴진다면?

그래도 내가 잘하는건 존재한다

강점은 꼭 절대적일 필요가 없다.

결국 “나” 라는 사람이 잘하는게 뭘까?

# 주장하자

“나 오늘 따듯한거 먹고 싶고, 매운 것도 괜찮고, 국물도 좋고, 면도 좋아.”

“사람 많지 않은데, 자연도 좋고, 조용하고, 카페도 있으면 좋겠어.”

어떻게 하면 더 잘보이게 말할 수 있을까?



# 어떻게 하면 더 잘보이게 말할 수 있을까?

“나 오늘 샤브샤브 먹고 싶어. 가볍게 따뜻한 국물로.”

“이번엔 양평 쪽 숲 있는 조용한 곳 가고 싶어. 카페는 한두 군데만 있어도 돼.”

# 어떻게 하면 더 잘보이게 말할 수 있을까?

협업을 중시하고 빠르게 학습하며 문제 해결을 잘합니다.



누구나 쓰는 말 / 검증 포인트 없음 / 해석 필요



# 어떻게 하면 더 잘보이게 말할 수 있을까?

(주장) 저는 문제를 선제적으로 발견하고 구조화해 개선하는 개발자입니다.

(근거) 실제로 [프로젝트] 에서 [문제] 를 [접근] 으로 해결해 [결과]를 만들었습니다.



# 어떻게 하면 더 잘보이게 말할 수 있을까?

## 추상어 바꾸기

생산성이 높습니다.



(행동) 저는 일을 작은 단위로 쪼개 빠르게 반복 배포합니다.

(결과) 그리고 의사결정/리스크를 앞단에서 정리해 대기 시간을 줄입니다.

# 어떻게 하면 더 잘보이게 말할 수 있을까?

## 추상어 바꾸기

협업을 중요하게 생각합니다.



(행동) 의사결정의 기준과 트레이드오프를 문서화해 팀의 합의를 빠르게 만듭니다.

(결과) 그 결과 같은 논의를 반복하지 않게 합니다.

# 어떻게 하면 더 잘보이게 말할 수 있을까?

추상어 바꾸기

문제 해결을 잘합니다.



(행동) 문제를 재현 → 원인 분해 → 해결안 비교 → 롤백 플랜 순서로 구조화합니다.

(결과) 그래서 급한 상황에서도 리스크를 통제하면서 해결합니다.

진짜 잘 쓴 이력서 인가? 보다 중요한 것  
좋은 글인가? 보다 ‘전략과 의도가 읽히는가’가 먼저다

내 의도/강점이 글에서 바로 파악되는가

그 다음이 취향(기호) 영역: “좋다/별로다”는 사람마다 달라짐

**진짜 잘 쓴 이력서 인가? 보다 중요한 것**  
**좋은 글인가? 보다 ‘전략과 의도가 읽히는가’가 먼저다**

결국, 내가 의도한 메시지가 오해 없이 전달되게 만들기

# 신입 / 경력직 차이

같은 경험이라도 보여줘야하는 형태는 다르다

## 신입

“밀도”가 핵심: 성과가 없어도 열심히 살아온 증거 (촘촘함)

프로젝트만 쓰지 말고: 인턴/부트캠프/동아리/개인 프로젝트/블로그 등

메시지: “짧은 시간에 이만큼 왔다”

# 신입 / 경력직 차이

같은 경험이라도 보여줘야하는 형태는 다르다

경력

“Why”가 핵심: 왜 그렇게 했는지, 협업/의사결정의 방식

기술 스택 나열보다: 기준/트레이드오프/문제 구조화

메시지: “나는 이런 기준으로 일해서 **재현 가능한 성과를 낸다**”

# 어렵다면 코드를 작성한다고 생각해보자

이력서도 코드처럼: 추상화 / 가독성 / 트리쉐이킹

Dev 일단 다 쓰기(경험/근거 덕지덕지)



# 어렵다면 코드를 작성한다고 생각해보자

이력서도 코드처럼: 추상화 / 가독성 / 트리셰이킹

**Prod** 추상화 - 핵심 강점으로 구조 잡기  
가독성(스캔)을 위해 섹션/볼드 정리하고  
트리셰이킹 (중복/이상한 표현 제거)

# 여러분의 이력서는 어떤 모습인가요



# 여러분의 이력서는 어떤 모습인가요



# 여러분의 이력서는 어떤 모습인가요



이력서부터 시작하는 커리어 셀프 브랜딩

# 마지막으로

## 이재상 이력서

### Instructor

끊임없이 학습하고 나누는 것을 즐기는 강사 이재상입니다.

다양한 도메인에서 쌓은 개발 경험과 교육·멘토링 노하우를 결합해 “왜”와 “어떻게”를 함께 전달하는 수업을 만듭니다.

### 핵심 역량

#### Leadership

- 교육과 학습, 문화 및 리더십에 관심이 많습니다.
  - 다양한 교육기관에서 React, Next.js, Database, Test 자동화 등 다양한 강의 및 멘토링 진행
  - 풍부한 라이브 강의, VOD 강의 촬영 경험
  - 다양한 기술 세미나 및 웨비나 주도
- 업무 문화와 협업의 가치를 중요하게 생각합니다.
  - 기술 세미나 및 스터디 조직을 통한 조직 내 기술 공유 활성화

#### Process

- CI/CD & Agile 프로세스 환경에서의 개발 경험이 많고 익숙합니다.
  - MUSINSA - 코드 리뷰 프로세스 개선 및 Self Approved PR 도입
  - Xinapse - 전 계열사 Engineering Meet-up 주관

#### Technical

- Frontend Architecture, 구조, 패러다임에 관심이 많습니다.
  - Kotech Hub - Next.js 12 → 14 마이그레이션 및 시스템 확장성과 유지보수성을 위한 Domain 기반 Monorepo 전환 주도
  - 29CM - Micro Frontend, MFA 구조 설계 및 복잡한 애플리케이션 아키텍처 구현
- Cross-Platform 에 대한 이해도와 숙련도가 높습니다.
  - 29CM - Webview 환경의 커머스 프로덕트 경험
  - Lying Snakers - Flutter App(DAU 1000, 다운로드 20K+) 배포 및 2년간의 운영 경험
- Next.js, RSC, TurboRepo, Nx 등 Modern Stack에 대한 이해도가 높습니다.

### 교육 및 멘토링 경력

Sparta Coding Club | Full Time Tutor (2023.05 ~ 2025.05)

핵심 성과

**감사합니다**

**Q&A 는 없겠죠?**