



# 이동훈

## 이력서

## Key Achievements

- 대학생 IT 연합 동아리 UMC
  - 5기, 6기 웹 파트장
  - 2023.09 ~ 2024.08
- (주)블록웨이브랩스
  - 프로덕트팀 FE/Blockchain (인턴)
  - 2024.09 ~

## Skills

### Programming Language

- Typescript
- Dart
- Java
- Go
- Rust

### Framework/ Library

- React / Next.js
- Flutter
- React Native
- Spring Boot
- Flask

### Tooling/ DevOps

- AWS(Ec2, S3, CloudFront)
- Docker

### ETC

- Git/Github
- Figma
- Notion
- Slack



LinkedIn

[linkedin.com/in/동훈-이-7778a5315](https://www.linkedin.com/in/동훈-이-7778a5315)



Github

[github.com/promlee](https://github.com/promlee)



Portfolio Website

[promleeblog.com/aboutme](https://promleeblog.com/aboutme)

## Frontend Developer

| Name | Birth      | Phone Number  | Email                 | Address                           |
|------|------------|---------------|-----------------------|-----------------------------------|
| 이동훈  | 1999.10.02 | 010-2561-5813 | donghoon099@naver.com | 서울특별시 동작구 사당로 23바길 9<br>112동 404호 |

저는 주로 Typescript, Dart 언어를 사용한 Frontend 개발을 진행해 왔습니다. Next.js 기반의 풀스택 개인 블로그를 만들어 로드 최적화와 반응형, SEO 적용 경험을 쌓았으며 졸업 프로젝트로 Flutter앱에 기반한 실내 길찾기 어플리케이션을 제작했습니다. 블록체인 스타트업인 (주)블록웨이브랩스에서 인턴을 진행하며 Web3 기반의 프로젝트를 4개 이상 맡아 다양한 경험도 쌓았습니다.

언어와 프레임워크를 가리지 않고 1순위로 사용자 경험을 우선시하는, **도전하는 개발자**가 되고 싶습니다.

## Experience

### (주)블록웨이브랩스 인턴

프로덕트 팀 FE/Blockchan 파트

(2024.09 - 현재)

- WEB3 기반의 프로젝트를 개발하는 스타트업에서 2024년 9월부터 4개월간 산학 인턴십, 이후 일반 인턴으로 근무 하고 있습니다.
- میم코인 런치패드 프로젝트(Next.js), 미션 기반 NFT생성 어플리케이션(Flutter, Next.js), 다날이 서비스하는 블록체인 지갑 및 결제 어플리케이션(React Native) 등의 프론트엔드 파트를 맡아 진행했습니다.
- Neutron 체인의 스마트 컨트랙트(Rust), 블록체인 코어 개발(GoLang) 등에도 참여하며 다양한 경험과 넓은 시야를 얻었습니다.

### UMC (앱, 웹 서비스 런칭 대학생 IT 연합 동아리)

5기, 6기 참여

(2023.09 - 2024.08)

- 5기의 웹 파트원, 6기의 웹(React) 파트장이자 서버(Spring Boot) 파트원으로 참여하였습니다.
- 5기에서 제로 웨이트 매장 관리 서비스 [Map2Zero](#), 6기에서 기업의 프로토타입 제품 체험 플랫폼 [PROTOTYPE](#)을 런칭하였습니다.
- 두 번의 서비스 런칭 모두 각각 프론트엔드 팀장, 서버 파트 팀장으로써 개발을 이끌었습니다.
- Neodinary Demoday 출품으로 실 사용자 100명을 유치했습니다.

## Certificates

| 이름          | 등급 | 합격 일자      | 합격 번호        | 발급처      |
|-------------|----|------------|--------------|----------|
| 정보처리기사(검정형) | 기사 | 2024.12.11 | 24203011231W | 한국산업인력공단 |
| OPIC (영어)   | IH | 2025.03.22 | 2A0482306532 | ACTFL    |

## Education

### 한국외국어대학교

공과대학 컴퓨터전자시스템공학부 (2학년 수료)

2018.03 - 2023.02 (편입학으로 인한 중퇴)

### 중앙대학교

소프트웨어대학 소프트웨어학부 (학사, 편입학)

2023.03 - 2025.08 (4학년 수료, 졸업예정)

# 경력기술서

## Info

| 이름  | 기업명         | 부서명   | 직급     | 기간              |                                                                                                    |
|-----|-------------|-------|--------|-----------------|----------------------------------------------------------------------------------------------------|
| 이동훈 | (주) 블록웨이브랩스 | 프로덕트팀 | 팀원(인턴) | 2024.08.26 ~ 현재 |  Blockwave Labs |

블록웨이브랩스는 블록체인 기술을 기반으로 다양한 Web3 서비스를 개발하는 스타트업으로, 저는 2024년 9월부터 산학 협력 인턴으로 합류하였으며 계약 만료 시점인 2025년 1월부터는 별도의 인턴 계약으로 프로덕트 팀에 소속되어 근무 중입니다. 주요 업무는 프론트엔드 개발, 블록체인 스마트 컨트랙트 작성, 운영 자동화 도구 개발 등 서비스의 기획부터 배포까지 폭넓은 실무 수행입니다. React, Next.js, Flutter, Rust, Go 등 다양한 기술 스택을 활용하여 제품의 사용자 경험과 기능 안정성을 책임졌으며, 각 프로젝트에서 핵심 기능 구현을 주도적으로 수행했습니다.

## Projects



### 찍고머니 (사용자의 활동을 보상으로 이어주는 리워드 앱) Flutter, Next.js, firebase

2024.12.01 ~ 2025.2.28 [Google Playstore - https://play.google.com/store/apps/details?id=com.jeff.liite](https://play.google.com/store/apps/details?id=com.jeff.liite)  
[Apple App Store - https://apps.apple.com/kr/app/찍고머니-사진만-찍으면-보상이-쏟아지는-리워드-앱/id6740126637](https://apps.apple.com/kr/app/찍고머니-사진만-찍으면-보상이-쏟아지는-리워드-앱/id6740126637)

- Flutter를 이용한 앱, Nextjs를 이용한 어드민 페이지 전담 프론트엔드 개발 업무를 수행했습니다. (FE 1, PM 1, BE 1, Design 1)
- 화면 라이프사이클 관리, 서버 부하 최소화 등을 고려한 핵심 화면 개발을 수행했습니다.
- Riverpod와 GetX를 사용한 하이브리드 상태관리를 통한 종속성 관리를 효율적으로 진행하였습니다.
- 서버 요청 시 Retrofit + Dio 를 이용한 타입 안정성과 에러 처리에 일관성을 부여하였습니다.
- MVVM 아키텍처 패턴을 채택하여 데이터, 비즈니스, UI 로직을 분리하여 유지보수의 용이성을 대비한 모듈화를 진행했습니다.
- 토큰 갱신 시 중복 요청 방지 매커니즘을 적용하여 기존 대비 요청 횟수가 횟수당 약 60% 감소하였습니다. (5회 → 2회)
- 관리자 페이지에서의 앱 소개, 약관 페이지를 SSG 방식의 pre-renders 페이지로 구현하였습니다.
- 기타 기능: Firebase Analytics를 통한 사용자 행동 분석, Firebase Messaging을 통한 푸시/인앱 알림, 카카오, 네이버, 애플 소셜 로그인, 네이버 지도 SDK, Admob 지원, easy\_localization을 통한 다국어 지원.



### 사슬 (블록체인 코어 개발) Golang, Docker, Mysql, AWS EC2, PHP

2024.10.01 ~ 2024.12.20 <https://saseul.com>

- 블록체인 네트워크 '사슬'의 동작 코어 개발에 참여하였습니다. (Core 개발 및 설계 3, PM 1)
- 주 업무는 PHP 환경을 Golang으로 마이그레이션, 기존 네트워크의 취약점 파악입니다.
- Golang의 논리적 멀티프로세스인 goroutine을 사용하여 노드간 데이터 동기화 속도를 기존 대비 30% 개선하여, 블록체인의 처리 효율과 안정성을 향상시켰습니다.
- ForceSync 로직의 병목 현상을 효과적으로 해소하여 시스템의 신뢰성을 확보했고, 테스트넷 환경 구축 및 운영을 통해 실제 운영 시나리오를 검증하였습니다.
- 시스템 안정화와 데이터 처리 성능 향상으로 블록체인 서비스 운영 효율성을 높이는 데 크게 기여하였습니다.



### 프로톤 (토큰 스테이킹 서비스) Rust, wasm, typescript

2024.10.01 ~ 2024.11.15 <https://www.proton.cool/>

- Neutron 체인 기반의 Web3 스테이킹 서비스 개발에 참여하였습니다. (Smart Contract 1, PM 1, FE 1, BE 1, Design 1)
- Rust 및 CosmWasm 기반의 스마트 컨트랙트 개발 및 테스트 를 진행하였습니다.
- qNeutron 토큰 발행 및 CW20 표준에 따른 토큰 로직 설계 스테이킹 및 언스테이킹 로직 구현, 배포 작업을 진행했습니다.
- TVL(Total Value Locked) 자동 수집 및 검증 스크립트(ts) 작성을 통해 운영 데이터의 정확성과 관리 효율성을 확보하였습니다.
- Rust 언어와 CosmWasm 프레임워크와 블록체인 기반 실무 경험을 통해 시스템 설계 역량 향상에 도움이 되었습니다.



### میمزل롯 (토큰 런치패드) Next.js, tailwindcss

2024.8.26 ~ 2024.10.1 <https://www.memeslot.me/>  
<https://www.kaiseki.fun/>

- 블록체인 입문자를 위한 토큰 생성 런치패드 프론트엔드 개발을 담당하였습니다. (FE 1, PM 1, BE 1, Design 1, Smart Contract 1)
- 인수인계 받은 코드를 기반으로 반응형 전환, api 호출 병렬화를 통해 초기 로드 속도를 2.5초 → 1초로 감소시켰습니다.
- 트레이딩 뷰 차트 적용을 위한 최적화된 데이터 가공, 렌더링 성능 최적화를 위한 데이터 업데이트 주기 관리로 기존 대비 향상된 속도로 차트 표시가 가능해졌습니다.
- 실시간 정보 수신을 위해 주기적 폴링 방식에서 웹소켓 방식으로 전환하여 실시간성을 높이고, 서버 부하를 줄였습니다. 이를 통해 실시간 가격 데이터의 지연시간을 최대 5초에서 평균 100ms 이하로 단축하여 트레이딩 시스템의 반응성을 50배 이상 개선하였고, 서버 부하는 기존 대비 70% 감소시켰습니다.

# 개인 프로젝트



## Promleeblog.com (개인 블로그)

Next.js, Supabase, postgresql, AWS(S3, Cloudfront, Lambda), Github Actions

2024.04.20 ~

<https://www.promleeblog.com>

<https://github.com/PROMLEE/Promleeblog.com>

- 협업을 우선시했던 팀 프로젝트에서 벗어나 내가 쓰고 싶은 기술들을 마음껏 사용해보자는 마음에서 연습용으로 시작한 사이트입니다.
- Supabase에서 가져오는 데이터를 캐싱하고, fallback, revalidate, mutate 등 옵션을 적극 활용하여 유저 체감 속도를 높였습니다. 초창기에 비해 페이지 렌더링 속도가 평균 35% 개선되었고, 실시간 데이터 갱신이 필요한 홈 화면에서도 UX 이탈률이 18% 감소했습니다. (Vercel Speed Insights 기반)
- 이미지 최적화 시스템 구축과 CDN 캐싱 전략 개선
  - **Cache-Control, ETag, immutable** 설정을 커스터마이징해 CloudFront 이미지 캐싱 효율을 극대화했습니다. 그 결과, CDN Hit Ratio가 97% 이상을 유지했고, 평균 이미지 응답 시간은 **1.2초 → 180ms**로 줄어 들었습니다.
  - Lambda@Edge와 S3를 조합해 요청 시 WebP 이미지를 동적으로 변환하고 캐싱 헤더를 추가해 S3, CloudFront에서 모두 캐싱하는 시스템을 직접 구현했습니다. 이 구조로 변경한 후, 평균 이미지 용량이 **65%** 감소, LCP가 **1.4초 → 0.9초**로 단축되었으며, 로딩 중 사용자 이탈률이 눈에 띄게 줄었습니다.
- Core Web Vitals(LCP, CLS 등) 최적화
  - Lazy load, prefetch, Skeleton UI, dynamic import 등을 활용해 블로그의 LCP 점수를 모바일 기준 **56 → 91점**, CLS는 **0.23 → 0.01**로 개선했습니다. Google Lighthouse 기준 SEO 점수는 100점, Performance 점수는 90점 이상 유지 중입니다.
- 페이지 성격에 따라 정적(SSG), 동적(SSR), 점진적(ISR) 렌더링 전략을 나누고, 각 방식의 장단점을 비교하여 적용했습니다. 검색 유입이 필요한 페이지는 SSG/ISR, 관리자 페이지 기능은 SSR로 처리해 SEO와 보안 모두를 확보했습니다.
- Tailwind & Shadcn/UI로 일관적인 디자인 시스템, 3단 반응형 UI, 라이트/다크 모드를 구축하였습니다.
- 현재 평일 기준 하루 **1000회** 이상의 구글 검색 노출과 **CTR 5%** 평균 **300회** 이상의 조회수를 기록하고 있습니다.



## Wear Os 워치페이스 개발

Kotlin

2022.11.04 ~ 2023.06

[https://github.com/PROMLEE/Wear\\_Os\\_WatchFace](https://github.com/PROMLEE/Wear_Os_WatchFace)

- 프론트엔드에 경험이 적었던 시절 사용중이던 스마트워치의 배경화면 중 마음에 드는 것이 없어 직접 제작해보고자 시작한 프로젝트입니다.
- WatchFace API와 **ComplicationSlotsManager**를 사용하여 시계 화면에 다양한 정보를 표시할 수 있는 기능을 구현했으며, **CurrentUserStyleRepository**를 통해 사용자가 직접 워치페이스의 스타일을 커스터마이징할 수 있도록 했습니다. 코드 품질 관리를 위해 Spotless와 ktlint를 도입하여 일관된 코드 스타일을 유지하고, 모듈화된 구조로 프로젝트를 설계하여 유지보수성을 높였습니다.
- Wear OS에 대한 레퍼런스가 거의 없었지만 결과를 내겠다는 집념으로 다른 외국 개발자들의 오픈소스를 하나하나 뜯어보며 시간을 투자했고 결국 3년째 사용중인 나만의 워치페이스가 나오게 되었습니다.
- 비슷한 어려움이 닥쳐도 끈기를 가지고 노력한다면 해낼 수 있다는 자신감을 얻게 된 소중한 경험입니다.

# 팀 프로젝트



## Indoor Map (딥러닝 실내 길찾기 어플리케이션)

Python(TensorFlow, Flask), Flutter, React, AWS(EC2)

2023.09.01 ~ 2023.12.15

[https://github.com/PROMLEE/Indoor\\_map](https://github.com/PROMLEE/Indoor_map)

- 복잡한 실내 환경에서 길을 찾을 수 있게 도와주는 어플리케이션 개발 프로젝트에 참여하였습니다. (**AI + FE+ BE 1**, FE 1, Algorithm 1)
- 딥러닝 Image Segmentation 모델(DeepLab V3+)을 적용해 건물 도면을 자동 인식 및 분석했습니다.
  - 실내 도면을 AI로 해석하기 위해 U-Net 기반 이미지 세분화 모델을 설계하고 학습하였습니다.
  - 약 3,000장의 도면 데이터를 수집 및 라벨링해 평균 **IoU 0.84**, **Pixel Accuracy 92.3%**를 달성했습니다.
- 실내 도면 이미지를 벡터화하고 경로를 SVG 형태로 렌더링하는 기능을 통해 확장성과 시각적 일관성을 확보했으며, 동적 경로 렌더링을 구현하였습니다.
- Flutter 기반 실내 길찾기 앱을 iOS/Android 동시 대응으로 개발, 하나의 코드베이스로 멀티플랫폼 앱 구현에 성공했습니다.
- Flask 기반 Python API 서버를 구축, Flutter 클라이언트와 RESTful 통신으로 데이터 흐름을 안정적으로 처리했습니다.
- AI + 프론트엔드 중심의 복합 기술 프로젝트를 기획부터 배포까지 주도적으로 완수하였습니다.

# 팀 프로젝트



## Map: 2 Zero (제로 웨이스트 매장 정보 제공 서비스) React

2024.01.05 ~ 2024.02.25 <https://github.com/PROMLEE/map2zero-frontend>  
<https://www.promleeblog.com/aboutme/map2zero>

- 제로 웨이스트 샵 정보 플랫폼 개발에 프론트엔드 팀장으로 개발에 참여하였습니다. (FE 4, BE 4, PM 1, Design 1)
- 팀장으로서 git-workflow 관리, 코드/커밋 컨벤션 지정, 코드리뷰, 팀 간 소통 등을 진행하였습니다.
- React 컴포넌트를 기능별로 모듈화하고, TypeScript를 통해 타입 안정성과 개발 생산성을 높였습니다.
- 약 50개 이상의 제로 웨이스트 샵 데이터를 시각화하고, 사용자 위치 기반 매장 추천 기능을 구현했습니다.
  - 서울시 스마트서울맵 API 연동으로 실시간 매장 위치 제공하고, API의 비동기 응답에 따른 예외 처리를 통해 사용자 위치 기반 탐색 시 오류를 없도록 하였습니다.
- UMC 공식 데모데이에 출품하여 Map: 2 Zero의 기술적 구조와 ESG 가치 전달 목표를 설명하고, 약 100명 이상의 사용자를 유치하였습니다.



## PROTOTYNE (시제품 체험 플랫폼) Spring Boot, MySQL, React, AWS(EC2)

2024.09.01 ~ 2024.12.15 [https://github.com/PROMLEE/PROTOTYNE\\_BE](https://github.com/PROMLEE/PROTOTYNE_BE)

- 유저는 시제품을 체험해볼 수 있는 기회를 얻고, 기업은 후기와 마케팅에 사용 가능한 정보를 얻는 윈-윈 플랫폼 개발에 참여하였습니다. (BE 4, FE 4, PM 1)
- 총 4개월 간 진행된 프로젝트에서 백엔드 팀장 역할을 맡아 전체 아키텍처 설계, 코드 리뷰, CI/CD 구성, 이슈 관리까지 총괄하였습니다.
  - 기능 분담과 브랜치 전략, 커밋 컨벤션을 직접 설계해 개발 효율성과 품질을 동시에 확보하였습니다.
- QueryDSL을 도입하여 복잡한 쿼리 실행 시간을 평균 30% 단축했으며, Redis 캐싱을 통해 반복 조회 요청의 응답 시간을 200ms에서 50ms로 75% 개선했습니다.
- Docker를 사용하여 애플리케이션을 컨테이너화 함으로써 일관된 실행 환경을 보장했으며, Alpine 리눅스 기반 Docker 이미지를 사용하여 컨테이너 크기를 기존 400MB에서 120MB로 70% 축소했습니다.
  - EC2 기반 서버 운영, S3 정적 파일 저장, Route53 도메인 연결을 통해 실제 서비스 배포 환경을 구성하였으며, GitHub Actions + Docker 을 활용한 자동화된 CI/CD 파이프라인을 직접 설계하고 배포 자동화까지 구현하였습니다.
- 카카오 소셜 로그인, 신제품 체험 신청, 피드백 등록 및 관리, 체험 히스토리 조회 등 20개 이상의 핵심 기능 API를 RESTful 설계를 기반으로 직관적이고 확장성 높은 API 구조로 개발하였습니다.
  - 카카오 OAuth2 연동을 통해 간편 로그인 기능을 제공하고, JWT 토큰을 활용해 사용자 권한(관리자/일반 사용자)을 분리하였습니다.
- UMC 공식 데모데이에 출품하여 PROTOTYNE의 기술적 구조와 실현 가능성, 목적을 설명하고, 약 60명의 사용자를 유치하였습니다.



## 이슈역 (이슈 관리 서비스) Spring Boot, MySQL, React, AWS(EC2)

2024.05.01 ~ 2024.06.10 <https://github.com/PROMLEE/Issue-Station>

- 소프트웨어 공학적 설계를 기반으로 한 이슈 관리 서비스 제작 프로젝트입니다. (FE + BE 1, FE 1, BE 2, Docs 1)
- 다양한 역할(Role)에 따라 접근 권한이 분리되는 이슈 관리 시스템으로써 이슈 등록, 댓글 작성, 상태 변경, 조회 등 프로젝트 협업에서 필요한 핵심 기능을 제공했습니다.
- Spring Boot API 서버, React 기반 웹 UI, AWS 배포까지 전반적인 개발을 주도하며, 단순 구현을 넘어 설계-문서화-검증 전 과정을 경험했습니다.
- 설계 단계에서 유스케이스 다이어그램, 도메인 모델링, API 스펙 정의 문서 등을 직접 작성하고, Notion에 문서화했습니다. 이로 인해 웹, GUI 개발자와 백엔드 개발자 간 소통이 원활해졌고, 기능 추가 시 설계 변경 없이 연속 개발이 가능했습니다.
- 이슈역은 Web UI와 별도로 GUI 클라이언트를 제공하는 구조로 설계되었으며, API 레이어의 재사용성과 RESTful 설계를 통해 클라이언트 이중화 구조에서의 설계 타당성을 직접 검증했습니다. 이 경험을 통해 멀티 인터페이스 환경에서도 유연하게 대응할 수 있는 구조적 사고 역량을 키웠습니다.