



# 유주섭

사용자의 입장에서 문제를 바라보고 해결하는 것을 중요하게 생각하는 개발자

이메일

juseob159@gmail.com

깃허브

[@yujuseop](https://github.com/yujuseop)

블로그

[JuseopSite](https://juseopSite)

전화번호

010-4431-6831

## 소개

병원에서 근무하던 당시, 환자의 보험 자격을 조회하려면 버튼을 세 번이나 눌러야 하는 비효율적인 전산 시스템을 사용하면서, '왜 이렇게 불편할까? 직접 고칠 수는 없을까?'라는 생각이 들었습니다.

이 경험을 계기로 사용자 중심의 더 나은 프로그램을 만들고 싶다는 목표가 생겼고, 병원을 퇴사한 후 본격적으로 개발 공부를 시작했습니다.

JavaScript, React, TypeScript, Next.js를 중심으로 기능 구현 역량을 키워왔으며, 환자와 의료진의 입장을 이해하며 소통했던 경험은 개발 협업 과정에서도 유용하게 작용했고, 사용자와 팀의 관점을 모두 고려한 개발을 지향하는 데 밑바탕이 되었습니다.

## 경력

**PECSPERT(프론트엔드/인턴)** “발화에 어려움을 겪는 자폐 아동을 위한 보완대체의사소통(AAC) 앱”서비스를 개발하였습니다. 아이들이 단어 카드를 선택하면 음성으로 출력되어 자신의 감정이나 욕구를 쉽게 표현할 수 있으며, 상대방과의 원활한 소통을 돋는 서비스입니다.  
2025.06 ~ 2025.08

ReactNative    React    Expo    Reacthook    TypeScript

팀 프로젝트: FE(6명), BE(2명)

- **발화 아동을 위한 시스템 및 음성적용 시스템 개발**
  - 음성 설정의 직관성 부족 및 설정값 초기화 문제를 개선하기 위해
    - 전역 상태 관리(Context API)를 도입하여 앱 전역에서 설정값 일관성 유지
    - Expo Audio 기반 실시간 미리듣기 및 7단계 속도 조절 기능 개발
  - 사용자 설정 관리 시 Firestore(클라우드 동기화)+AsyncStorage(로컬 저장소)를 병행 활용
    - 앱 재실행 시 즉시 설정값 적용
    - 다중 디바이스 간 설정 동기화 보장
    - 오프라인 환경에서도 안정적 동작  
→ 결과적으로 일관된 사용자 경험(UX) 제공
  - 디자이너·기획자와 협업하여 디자인 시스템 구축
    - 슬라이더·버튼 등 공용 컴포넌트 개발  
→ 공용 컴포넌트 재사용으로 팀 개발 속도 향상
- **태블릿 전용 앱을 모바일 디바이스까지 지원하도록 확장**
  - 고정 너비 레이아웃을 비율 기반 반응형 레이아웃으로 전환
  - 기기 해상도별 텍스트 크기 자동 조절 로직 구현  
→ 다양한 디바이스 환경에서도 일관된 사용자 경험 제공
- **렌더링 성능 및 코드 유지보수성 개선**
  - 음성 설정 UI 컴포넌트의 불필요한 리렌더링 문제  
→ 공통 UI 분리 및 useMemo, useCallback 최적화 적용
  - 렌더링 성능 향상, 코드 수정 범위 30% 이상 축소 → 유지보수성 개선
- **Tech PM적 역할로 기획-디자인-프론트 협업 조율**
  - 구독권 시스템 구현 과정에서 발생한 문제를 주도적으로 조율·해결
    - 디자이너, 프론트엔드, 백엔드 간 상이한 인식과 범위 차이를 조율
  - 팀 내 합의점을 도출하여 프로젝트 진행 지연 방지 및 기능 완성

## 김해바른병원(원무과)

2022.03 ~ 2024.06

(2년 3개월)

- 환자 응대 및 행정 지원

- 환자 응대를 통해 상황별 문제 해결 능력과 원활한 커뮤니케이션 역량을 강화
- 다양한 요구와 민원을 신속하게 파악하고 대응하면서 사용자 중심의 사고와 공감능력을 습득
- 타부서와 협력하여 업무를 처리하며 협업 능력과 조율 능력을 향상

## 프로젝트

### Epigram

2025.03 ~ 2025.04 (4주)

팀 프로젝트: FE(6명)

[Github](#) · [Link](#)

감정을 기록하고 명언을 남길 수 있는 감성 기록 플랫폼입니다.

명언 공유와 댓글 소통, 감정 통계를 통해 공감과 감정 변화를 시각화하여 보여주는 웹 사이트입니다.

Next.js

NextAuth

Tailwind

TypeScript

zod

React hook form

- 협업 프로세스 최적화: 빌드 오류 70% 감소 경험

- 협업 체계 부재로 인한 기능 누락과 일정 지연 문제
  - Notion&Tldraw 기반 협업 문서화 체계 설계, Git Flow 전략 및 Vercel 자동 배포 환경 구축
  - 빌드 오류 70% 이상 감소, 팀 협업 프로세스 안정화

- 불안정한 인증 흐름을 NextAuth로 리빌드 : 보안&유지보수성 UP

- 커스텀 인증 로직의 세션 불안정 및 소셜 로그인 불가 문제
  - NextAuth(JWT 기반) 도입 및 Google OAuth 통합, 기존 로그인 흐름과 커스터마이징된 통합 적용
  - 인증 관련 코드 40% 이상 감소, 로그인 실패율 감소 및 유지보수성 강화

- 효율적인 감정 인식 인터페이스 구축

- 텍스트 기반 감정 데이터 UI의 인지 부담 및 새로고침 시 데이터 유실 문제
  - Recharts 기반 PieChart&리스트 시각화 UI 도입, localStorage 상태 지속성 및 Toastify 알림 기능 추가
  - 사용성 및 감정 인식 효율 개선, 사용자 만족도 향상

### TripView

2025.09 ~ 2025.10(2주)

개인 프로젝트

[Github](#) · [Link](#)

여행을 기록하고 리뷰를 할 수 있는 여행 리뷰 플랫폼입니다.

사용자는 날짜별로 여행지를 입력해 리뷰를 작성하고, 공개/비공개 설정으로 기록을 관리할 수 있습니다.

Next.js

Tailwind

TypeScript

Supabase

ReactQuery

React Hook Form

- Supabase Auth 기반 인증 시스템 구축 : 통합성과 유지보수성 UP

- Supabase SSR 인증 동기화 리빌드
  - Supabase를 백엔드 서버로 활용하면서, 동일 생태계 내 Auth 기능을 통합적으로 사용
  - createServerClient + cookies 기반 SSR 인증 구조로 세션 동기화 안정화
  - 별도 서버 구축 없이 인증 + 세션 + 데이터 베이스 간 호환성 확보
- 보안 및 유지보수성 향상
  - 백엔드-프론트엔드 통합 아키텍처 설계 및 유지보수 효율성 강화
  - 미들웨어 기반의 인증 라우팅 추가로 접근 제어 및 권한 관리 강화

- React Hook Form + Zod 기반 폼 검증 시스템 구축 : 안정성과 생산성 UP

- 타입 안전한 입력 검증 구조
  - React Hook Form과 Zod를 결합해 폼 입력 데이터의 타입 안정성 확보
  - 여행지 추가/리뷰 작성 시 실시간 유효성 검사 및 에러 피드백 구현
  - 폼 상태를 커스텀 흙으로 모듈화해 유지 보수성 향상

- Toast 시스템 제작 및 통합 : 커스터마이징 자유도 UP

- CSS 자동 주입이 가능한 Toast 알림 라이브러리 ([js-toastify](#)) 개발
  - 기존 Toast 라이브러리의 한계(스타일 수정 불가, 외부 CSS 의존) 해결
  - 프레임워크 무관하게 재사용 가능한 독립형 UI 알림 시스템 구축
  - 경량화된 모듈 구조로 빌드 크기 최소화 및 유지보수성 향상

# 경험/활동

## 코딩 테스트 스터디

2025.01 ~ 현재

## 코딩 테스트 문제풀이 및 알고리즘 스터디 [Github](#)

프로그래머스 및 백준 플랫폼의 코딩 테스트 문제를 주 1~2회 풀이하고, 문제에 사용된 알고리즘(스택, DP, 이분 탐색 등)을 함께 학습하며 스터디원과 풀이 방식 및 개념을 공유하는 스터디에 참여

## React 스터디

2024.11 - 2025.03 (20주)

## React 쇼핑몰 프로젝트 기반 스터디

리액트 쇼핑몰 프로젝트 책을 통하여 스터디를 하며 실무 중심의 React 개발 역량을 키웠고, 컴포넌트 설계, API 연동, 상태관리 등 웹 앱 전반의 구조를 경험하며 문제 해결력과 협업 능력을 향상.

# 교육

## 코드잇 스프린트 12기 수료

2024.10 - 2025.04 (6개월)

- JavaScript, TypeScript, React, Next.js 실무 기반의 체계적인 학습.
- GitHub를 활용한 브랜치 전략, 코드 리뷰, 이슈 관리 등 협업 방식 실습.
- 4~6인 팀으로 구성된 총 3개의 협업 프로젝트 수행(기획 → 개발 → 배포 전 과정 경험).
- Vercel을 활용해 총 3개 프로젝틀을 배포, 배포 자동화 및 환경 변수 설정 경험.