Email	seunghwan7700@gmail.com
Mobile	010-5752-7305
Github	link
Blog	link

이승환

Frontend Developer

소개

📌 주요 개발 내용

- Next를 사용한 프로젝트를 배포, 운영한 경험이 있습니다.
- CI/CD 파이프라인을 구축해 배포 자동화를 한 경험이 있습니다.
- 개발하는 것을 좋아해서 연말 기념으로 동아리내에서 사용할 이벤트 웹사이트를 구축한 경험이 있습니다.

📌 원리를 분석하고 기록하는 것을 좋아합니다. [📝버츄얼 돔] [📝V8엔진의 GC] [📝상태 관리 내부구조]

어떤 도구를 쓸 때 스스로 **왜? 라는 질문을 하면서 그 이유를 찾아보는것을 좋아합니다.**

실제로 virtual dom의 동작 과정에 대한 원리를 분석해 블로그에 포스팅을 하였으며, 2년동안 활동하고 있는

개발 동아리에서 virtual dom을 주제로 발표를 한 경험이 있습니다.

또한 전역 상태관리에 관심이 생겨 라이브러리를 만들어 배포한 경험이 있습니다. [📝react-state-shed]

📌 눈으로 확인한 최적화가 진짜 최적화라고 생각합니다. [📝리액트 쿼리 최적화] [📝이미지 최적화]

최적화를 할 때 실제로 제 눈으로 확인되지 않는다면 최적화를 했다고 말할 수 없다고 생각합니다.

저는 개발자 도구, lighthouse을 보고 얼마나 어떻게 최적화되었는지에 대해 확인하며

리액트 쿼리, 이미지, 무한 스크롤 관련된 최적화를 진행한 경험이 있습니다.

📌 사용자를 공감하며 문제를 해결합니다.

내 서비스는 어떤 사람이 쓰고 어떤 환경에서 쓰이고 있을까? 를 생각하면서 개발을 진행합니다.

실제로 운동 측정 및 커뮤니티를 주제로한 웹앱 기반 어플리케이션에서 사용자가 운동을 하다가 핸드폰이 꺼져

운동 측정한 데이터가 사라지는 문제를 localStorage를 사용해 해결한 경험이 있습니다.

프로젝트

Commit Body / Frontend Engineer (Frontend: 2명, Backend: 1명, Designer: 1명) (2024.08 ~ 2025.01)

사용자가 자신만의 운동 루틴을 만들고 기록을 관리하며, 운동 성과를 공유하고 커뮤니티에서 소통할 수 있는 서비스	
기술 스택	NextJs, Tanstack-Query, Zustand, Tailwind, ShadnUI, PWA, Docker, Github Actions

[반복되는 서버 요청에 대한 캐싱을 위해 리액트 쿼리 사용] [blog]

→ **이슈** : 자주 변경되지 않는 데이터의 반복되는 요청으로 인해 불필요한 네트워크 트래픽 발생

→ **해결** : staleTime과 gcTime을 사용해 데이터를 캐시해 **서버요청 최소화**

[깃허브 액션을 사용해 배포 자동화]

→ **이슈** : 배포를 하기위해 항상 shell환경에서 반복되는 코드를 치는것에 불편함을 느낌

→ **해결** : 메인 브랜치 푸시 시 CI/CD 파이프라인을 통해 Docker 이미지 빌드,
Docker Hub 배포 및 Shell 환경에서 컨테이너 자동 배포 프로세스 구축

[서버컴포넌트를 활용해 초기 렌더링 개선]

→ **이슈** : CSR환경에서 운동 리스트API를 호출하면서 첫 렌더링시간이 **1.36초**로 지연

→ **해결** : 서버컴포넌트로 전환 후 첫 운동 리스트API를 prefetch, 그 결과 첫 렌더링시간을 **1.36초 → 0.5초**로 단축

→ **고민** : 서버컴포넌트를 사용해 JS번들 크기와 초기 렌더링속도를 감소 시켰지만 상황에 따라 **서버부담이 커진다고 판단**

→ **해결** : SSR환경의 fetch에서 제공하는 **캐시**를 활용해 서버부담을 최소화

[스켈레톤 UI를 사용해 사용자 경험 향상]

→ **상황** : 데이터를 로딩하는 동안 빈 화면 대신 Skeleton UI를 표시하도록 설계하여 사용자 경험 향상을 기대

→ **이슈** : 데이터 로딩 속도가 매우 빨라 Skeleton UI가 깜빡이는 현상이 발생하며 오히려 사용자 경험 저하

→ **해결** : 데이터가 로딩되는 시간이 **200ms이하면 빈화면**을 보여주고 그 이상이면 스켈레톤 UI를
조건부로 보여주는 **ConditionalSkeleton 컴포넌트** 구현

→ **2차 이슈** : 필터 버튼을 통해 상태 변경 시, ConditionalSkeleton의 isDeferred 상태가초기화되지
않아 Skeleton UI가 다시 깜빡이는 문제 발생.

→ **해결** : React의 **key** 속성을 활용하여 ConditionalSkeleton를 강제로 리렌더링시켜 isDeferred 상태를 초기화

[서버 부하 최소화 및 사용자 입력 처리 최적화를 위한 Debounce 기능 구현] [blog]

→ **이슈** : 검색창에 검색어 입력 시 API 요청이 무분별하게 발생하는 문제 발생

→ **해결** : 사용자 입력 처리 과정에서 서버 부하를 줄이기 위해 입력 텍스트에 500ms의 **디바운스** 적용

[단계별로 이루어진 회원가입 폼 개발]

→ **1차 이슈** : 작성해야 하는 폼이 많아 사용자가 도중 이탈할 수 있다고 예상

→ **해결** : 회의를 통해 **단계별로 이루어진 회원가입 폼을 개발**하여 UX를 개선하고 도중 이탈을 방지하도록 변경

→ **2차 이슈** : 회원가입 절차가 여러 페이지로 나누어지면서 데이터 관리와 디버깅의 어려움이 생기는 문제 발생

→ **해결** : **Funnel패턴**을 사용해 한 페이지에서 데이터 관리

[사용자의 간편한 로그인을 위해 OAuth 사용]

→ **이슈** : 사용자 인증 과정에서 복잡한 로그인 절차로 인한 사용자 경험 저하

→ **고민** : 간편하고 안전한 인증 방식을 제공하기 위해 어떤 방법을 도입할지

→ **해결** : OAuth(next-auth)를 사용해 소셜 로그인 구현, 인증 절차를 간소화하고 사용자 경험을 개선

[좋아요 버튼에 낙관적 업데이트 적용]

→ **이슈** : 좋아요 API 요청을 보냈을 때 최신화를 위해 리스트를 매번 refetch 해줘야하는 문제 발생

→ **고민** : 방법 1. 데이터 최신화를 위해 refetch시 좋아요API 요청을 하면 운동 리스트도 매번 다시 가져와야해서 http 요청을 두번하는건 부적절하다고 판단

방법 2. 좋아요API가 성공한다면 운동 리스트의 캐시를 조작, http 요청은 한번에 끝나지만 좋아요 API 응답이 늦어지면 사용자 경험 저하

방법 3. 좋아요API 실행 전 캐시를 미리 업데이트해 사용자에게 변경된 UI제공, 실패한다면 롤백

→ **해결** : 좋아요 버튼 같은 경우 실패하기 어렵고 빠른 응답이 이상적이기때문에 낙관적 업데이트 (방법 3) 선택

그 결과 약 **0.7s** 딜레이 단축으로 사용자 경험 증가

피그마	**Figma**
깃허브	**GitHub**
배포 링크	**Link**

Auction / Frontend Engineer (Frontend: 1명, Backend: 1명)

(2023.12 ~ 2024.04)

중고 거래 서비스

기술 스택	Next.js, TypeScript, React Hook Form, Scss, Tanstack query, Recoil, RTL, Cypress

[Next/image의 최적화 방식 알아보기] [blog]

→ **고민 :** Next/image를 사용하면 어떻게 최적화되는지 호기심이 생김

→ **해결 :** 차이를 알아보고 Next/image를 사용해 이미지 용량을 줄여 lighthouse 기준 **LCP 5초 → 1.1초로 최적화**

[프론트엔드 개발 환경 개선을 위해 MSW도입] [blog]

→ **이슈** : 백엔드 개발이 늦어져 API 연동이 늦어지는 상황 발생

→ **해결** : **MSW**를 도입해 API 개발 속도에 영향 받지 않고 프론트엔드 개발 스프린트 진행

[댓글 무한 스크롤 개발] [blog]

→ **1차 이슈** : onScroll방식을 사용해 무분별하게 이벤트가 호출되는 문제 발생

→ **해결** : **Throttling**을 적용해 이벤트 호출 최소화

→ **2차 이슈** : **Throttling** 적용 후에도 이벤트 호출이 잦다고 판단

→ **해결** : onScroll방식보다 성능이 좋은 **Intersection Observer**를 활용해 개발

[합성 컴포넌트 패턴을 사용해 컴포넌트 재사용] [blog]

→ **이슈** : 비슷한 Input UI가 많아서 컴포넌트로 만들면 props가 많아져 확장성이 떨어지는 상황 발생

→ **해결** : **합성 컴포넌트 패턴**을 사용해 Input UI컴포넌트를 만들어 재사용

깃허브	**GitHub**

기능 구현 챌린지/ Frontend Engineer (개인)

(2024.03 ~ 2024.04)

한국임상정보 페이지의 검색영역을 클론하기

기술 스택	Vite, React, TypeScript, Tanstack query v5, Styled-components, Vitest, StoryBook, Github actions

[검색 결과리스트 무한스크롤 구현]

→ **1차 이슈** : 검색 결과리스트의 요소가 많아 한번에 데이터를 불러왔을 때 페이지 로딩 시간이 길어짐

→ **해결** : **Intersection observer**를 활용해 무한스크롤 적용, 사용자가 스크롤을 바닥까지 내렸을 때 데이터를 부분적으로 가져와 페이지 로딩 시간 개선

→ **2차 이슈** : 스크롤을 내릴 때마다 **DOM요소가 누적되어 성능 문제 발생**

→ **해결** : react-virtuoso 라이브러리를 사용해 **가상 스크롤 구현**

→ **3차 이슈** : 스크롤을 내린 뒤 다른 페이지를 갔다가 돌아왔을 때 **스크롤 위치가 초기화되어 사용자 경험 저하**

→ **해결 방안** : 브라우저의 **세션 스토리지를 활용해서** 사용자의 **스크롤 위치를 저장**하고, 페이지 재방문 시 해당 위치를 복원

[Render props 패턴을 사용해 컴포넌트 재사용]

→ **이슈** : 동일한 UI지만 다른 기능을 하는 요소가 있어서 재사용하기 어려운 상황 발생

→ **해결** : **Render props 패턴**을 사용해 다른 기능을 가진 UI만 props로 넘겨줘서 해결

깃허브	GitHub

DeepBlue / Frontend Engineer (Frontend: 3명, Backend: 4명) (2022.12 ~ 2023.1)

AI를 이용해 사진 한 장으로 물고기 어종을 판별할 수 있는 서비스

기술 스택	Vite, React, TypeScript, Recoil, Tanstack query, MSW, Scss, Aws, Docker
개발 내용	• **API 연동** • **Figma를 사용해 디자인 설계**
깃허브	GitHub
데모	Demo

TikiTaka / Frontend Engineer (Frontend: 2명, Backend: 4명) (2022.08 ~ 2022.10)

익명으로 질문을 주고 받는 서비스

	React, React-Native, Styled-Components
개발 내용	• **API 연동** • **Figma를 사용해 디자인 구축**
깃허브	GitHub
데모	Demo

기술 스택

FrontEnd	BackEnd
Language : JavaScript, TypeScript **Framework :** NextJs **Library :** React, Recoil, Zustand, React-Query **Style :** Styled-components, Scss, Tailwind	**Language :** JavaScript, TypeScript **Framework :** Nest **Devops :** Docker

경험

- 성결대학교 **(컴퓨터공학과)** 2018 ~ 2024.02 **(졸업)**
- [티타임즈 x techeer]실리콘밸리 sw 부트캠프 **2023.07 - 2023.08**
- 원티드 프리온보딩 프론트엔드 인턴쉽 과정 수료 **2023.04 - 2023.05** → 관련자료
- 실리콘밸리 개발자 멘토링 프로그램 Techeer 4 기 **2022.09 - 현재**
  - 실리콘밸리 엔지니어의 SW 개발자 커리어 그룹
    기술 세션, 프로젝트, 스터디 등 다양한 개발 및 네트워킹 활동에 집중하는 그룹
- [티타임즈 x techeer]실리콘밸리 sw 부트캠프 **2022.12 - 2023.2**
- [티타임즈 x techeer]실리콘밸리 sw 부트캠프 **2022.08 - 2022.10**