

이승환

Frontend Engineer

Email seunghwan7700@gmail.com

Mobile 010-5752-7305

Github [link](#)

Blog [link](#)

소개

- 좋은 코드란 무엇일까? 라는 생각을 끊임없이하며 더 나은 개발자가 되기위해 성장을 추구합니다.
- 비즈니스 로직과 뷰 로직 분리 및 추상화의 중요성을 인식하여 개발자 커뮤니티에서 발표를 진행한 경험이 있습니다.
- 확장성과 유지보수를 위해 상황에 맞는 다양한 컴포넌트 패턴을 사용하도록 노력하고 있습니다.

프로젝트

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

(2023.12 ~ 2024.4)

물품 거래 서비스

기술 스택	Next.js, TypeScript, React Hook Form, Scss, Tanstack query, Recoil, RTL, Cypress
개발내용	<p>[Stepper 기반 회원가입 폼 개발]</p> <ul style="list-style-type: none">→ 1차 이슈 : 회원가입을 위해 작성해야하는 폼이 많아 사용자가 도중이탈할 가능성이 있다고 판단→ 해결 : Stepper 기반으로 회원가입 폼 개발, UX를 높여 도중이탈 방지→ 2차 이슈 : 여러 페이지로 나누어지면서 데이터 관리와 디버깅의 어려움이 생기는 문제 발생→ 해결 : Funnel패턴을 사용해서 한 페이지에서 데이터를 관리, 추가로 재사용을 위해 useFunnel 커스텀 훅 개발→ 3차 이슈 : 비슷한 Input UI가 많아서 컴포넌트로 만들면 props가 많아져 확장성이 떨어지는 상황 발생→ 해결 : 합성 컴포넌트 패턴을 사용해 Input UI컴포넌트를 만들어 재사용→ 4차 이슈 : 많은 Input의 유효성 검사의 어려움과 불필요한 리렌더링이 비효율적이라고 판단→ 해결 : 비제어 컴포넌트를 사용하는 react hook form을 활용해 유효성검사 및 불필요한 리렌더링 최소화 <p>[댓글 무한 스크롤 개발] [blog]</p> <ul style="list-style-type: none">→ 1차 이슈 : onScroll방식을 사용해 무분별하게 이벤트가 호출되는 문제 발생→ 해결 : Throttling을 적용해 이벤트 호출 최소화→ 2차 이슈 : Throttling 적용 후에도 이벤트 호출이 잦다고 판단→ 해결 : onScroll방식보다 성능이 좋은 Intersection Observer를 활용해 개발 <p>[좋아요 버튼을 누를 때 사용자에게 즉각적으로 응답이 오는 것처럼 적용하기 위한 낙관적 업데이트 적용] [blog]</p> <ul style="list-style-type: none">→ 이슈 : 좋아요 버튼을 클릭 후 서버에서 가져오는 데이터가 늦어 약 0.7s 딜레이 발생→ 해결 : React-Query를 사용해 캐싱되어 있는 데이터를 선 업데이트 하여 네트워크 속도와 상관없이 즉각적인 UI 업데이트 처리 <p>[서버 부하 최소화 및 사용자 입력 처리 최적화를 위한 Debounce 기능 구현]</p> <ul style="list-style-type: none">→ 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 <p>[비즈니스 로직과 뷰 로직 분리]</p> <ul style="list-style-type: none">→ 이슈 : 관심사의 분리가 안되어있어 컴포넌트가 UI에 집중할 수 없다고 판단→ 고민 : 커스텀 훅 패턴, presentation container 패턴 중 컴포넌트가 많이 나오지않는 커스텀 훅 패턴 채택→ 해결 : 비즈니스 로직과 뷰 로직을 분리하기 위해 커스텀 훅 개발 <p>[프론트엔드 개발 환경 개선을 위해 MSW도입] [blog]</p> <ul style="list-style-type: none">→ 이슈 : 백엔드 개발이 늦어져 API 연동이 늦어지는 상황 발생→ 해결 : MSW를 도입해 API 개발 속도에 영향 받지 않고 프론트엔드 개발 스프린트 진행
깃허브	GitHub

AI를 이용해 사진 한 장으로 물고기 어종을 판별할 수 있는 서비스

기술 스택	Vite, React, TypeScript, Recoil, Tanstack query, MSW, Sass, Aws, Docker
개발 내용	<p>[JWT방식으로 회원가입, 로그인 서비스 개발]</p> <p>→ 고민 : 서버부하없이 token하나만으로 사용자를 식별할수 있고 서버 메모리에 저장 공간을 확보하지 않아도 되기때문에 JWT방식을 채택</p> <p>→ 이슈 : 브라우저를 새로고침했을 때 저장된 로그인 정보를 잃어버리는 상황 발생</p> <p>→ 해결 : Recoil-persist를 활용해 브라우저를 새로고침해도 저장된 로그인 정보를 이용해 자동 로그인을 요청 · 사용자가 브라우저에서 나가도 로그인을 유지할 수 있게 sessionStorage가 아닌 localStorage에 저장</p> <p>[Route Level에서 Code Splitting을 적용해 초기 로딩 속도 개선]</p> <p>→ 이슈 : 초기 로딩속도 2초</p> <p>→ 원인 : 페이지들이 추가되면서 번들 사이즈 증가</p> <p>→ 해결 : 메인 페이지 접근 시, 불필요한 페이지들은 lazy로딩을 적용하여 2초 → 0.5s로 초기 로딩속도 개선</p> <p>[배포 환경 구축]</p> <p>→ 이슈 : 배포 후 도커를 구동하기 위해 여러 컨테이너를 하나씩 돌려야하는 상황 발생</p> <p>→ 해결 : 여러 컨테이너를 한번에 빌드하기 위해 Docker-compose 를 활용해 AWS EC2 에 배포</p> <p>• ApexCharts를 이용한 데이터 시각화</p>
데모	GitHub
깃허브	Demo

챌린지

동아리에서 진행한 기능 구현 챌린지/ Frontend Engineer (개인)

(2024.05.03 ~ 2024.05.18)

[한국임상정보](#) 페이지의 검색영역을 클론하기

기술 스택	Vite, React, TypeScript, Tanstack query v5, Styled-components, Vitest, StoryBook
개발 내용	<p>[검색창에서 키보드만으로 추천, 최근검색어들로 이동 가능하도록 구현]</p> <p>→ 고민 : 드롭박스로 구성된 추천검색어, 최근검색어를 마우스로 접근하는게 사용자 경험에 안좋다고 판단</p> <p>→ 해결 : 키보드를 활용해 추천,최근 검색어들로 이동 가능하게 구현해 사용자 경험 향상</p> <p>[검색 결과리스트 무한스크롤 구현]</p> <p>→ 이슈 : 검색 결과리스트의 요소가 많아 한번에 데이터를 불러왔을 때 페이지 로딩 시간이 길어짐</p> <p>→ 해결 : Intersection observer를 활용한 무한스크롤을 적용해 사용자 화면의 위치에 따라 데이터를 부분적으로 가져와 페이지 로딩 시간을 단축</p> <p>[스토리북을 활용한 UI 테스트]</p> <p>→ 이슈 : 새로운 기능이나 변경된 기능이 도입되었을 때, 예상치 못한 버그로 일관된 UI를 제공할 수 없을 수도 있다고 판단</p> <p>→ 해결 : 스토리북을 적용해 컴포넌트의 동작을 쉽게 확인하고 디버깅 및 시각화</p> <p>[Vitest를 통한 비즈니스 로직 테스트]</p> <p>→ 고민 : 기능 변경 및 새로운 기능을 도입할 때 예상치 못한 오류가 발생할 수 있다고 판단</p> <p>→ 해결 : 비즈니스 로직을 테스트해 기능 구현 단계에서 안정성 향상</p> <p>[Error Boundary를 활용한 안정성 확보]</p> <p>→ 고민 : 에러가 발생했을 때 웹페이지가 다운될 수 있다고 판단</p> <p>→ 해결 : 에러 바운더리를 활용해 부분적으로 에러 fallbackUI를 보여줌으로써 다운방지 및 사용자 경험 향상</p>
깃허브	GitHub

FrontEnd

- JavaScript, TypeScript
- React, React-native, NextJs
- Recoil, React-Query

BackEnd

- JavaScript, TypeScript
- Node, Nest

DevOps

- Docker AWS

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**