

정혜원 | Jeong Hyewon

언어 공부에 대한 집착, 깔끔한 코딩에 대한 갈망, 최상의 결과물을 주는 팀워크



Contact.

- ✓ Email. jaake0704@gmail.com
- ✓ Phone. 010-8954-4608

Channel.

- ✓ Portfolio . [개발하는 너굴맨 포트폴리오](#) 🙌
- ✓ GitHub . <https://github.com/wonny-won>
- ✓ Blog . <https://dev-raccoon-man.tistory.com/>

저는 _____ 한 개발자 입니다.

- 개발시 유지보수, 성능, 확장성, 안정성, 보안을 고려합니다.
- 기술의 동향을 파악하고 새로운 기술을 공부하는 것을 즐깁니다.
- 기존의 기술을 다양한 측면으로 깊게 공부하는 것을 즐깁니다.

저는 이렇게 일해요!

🔧 사용자 경험 향상 측면에 많은 열정을 가지고 있어요.

사용자가 사용해보기 전에 개발자가 직접 동작해보며 '이부분은 어색한데?' 한 부분은 기획자와 디자이너님의 의도를 여쭙어 본 뒤 의견을 제시해 더 나은 방향으로 수정 될 수 있도록 합니다.
지금도 좋지만 개선되면 조금 더 편리하게 사용이 가능한 부분에 도움이 될 수 있습니다.

🔧 성능 최적화는 늘 작은 곳에서 시작되죠.

저의 성능 최적화는 컴포넌트에서부터 시작합니다.
컴포넌트 내의 불필요한 리렌더를 잡아가는 것에서부터 시작하기 때문에 프로젝트의 렌더 성능에 도움을 드릴 수 있습니다.
또한 렌더 성능 최적화 이외에도 필요하다면 주도적으로 공부하고 집착적으로 생각해보기도 합니다.
지난 회사에서는 프로젝트 성능 향상을 위해 여러가지 시도를 해보았으며, 이 경험들이 더 다양한 방면의 창의적인 아이디어로 빛을 발하게 될 것 입니다.

🔧 유지보수를 고려하는건 당연한 일이죠.

저는 개발시 개발 원칙으로 유지보수를 고려하고 있습니다.
예를들어 UI와 데이터 컴포넌트를 분리하는 단일책임 원리, 함수 추상화 적용등 추후 코드 유지보수를 위한 저만의 원칙을 가지고 있습니다.
때문에 일관성있고 코드 유지보수에 용이한 코드 퀄리티를 보여드릴 수 있습니다.


보유 기술

Frontend

현재 업무에 사용중 혹은 사용했던 기술들입니다.

- HTML / CSS
- JavaScript / TypeScript
- React.js / Next.js / svelte
- apollo-client / axios / websocket / webworker / react-query
- recoil / context-api
- emotion / styled-component / tailwind

회사 프로젝트

 를 클릭하시면 상세한 프로젝트 설명 또는 제작한 프로젝트를 보실 수 있습니다.

포페런츠

엔지니어 | FE팀
개발자



2025.04.10 ~

✓ Tech stack

- typescript
- next.js
- tailwind
- emotion
- context-api
...

액티브 시니어를 위한 여행 이커머스 서비스

✓ 개발 시스템 및 문서화 체계 구축 및 서비스, 어드민 제작 및 런칭

- **아너드 자체 웹 서비스 프론트엔드 E2E 단독 개발 및 런칭** 
 - **신한 X 아너드 제휴 서비스 프론트엔드 E2E 단독 개발 및 런칭** 
 - 모노레포 구축
 - Turborepo + pnpm 기반 워크스페이스 설계
 - 서비스/공용 모듈을 분리해 구조화
 - 유지보수, 재사용성을 고려한 데이터/UI 컴포넌트 분리
 - 확장성을 고려한 컴포넌트 설계
 - 전체 삭제 데이터 프로바이더 → 다중 프로세스 처리 프로바이더로 확장
 - 서비스 런칭 전 과정(기획, 디자인, 개발)에 참여해 E2E 개발 프로세스 리딩
- **신규 어드민 서비스 제작**
 - 실무진 니즈를 반영한 핵심 기능 기획 및 구현
 - 기존 리소스를 활용한 MVP 성격의 빠른 제작/배포로 업무 효율성 극대화
 - 개발 일정 리드 및 협업 조율
- **기존 어드민 유지보수 및 신규 기능 추가**
 - 실무진 니즈를 반영한 신규 기능 기획 및 구현
 - 각종 핫픽스 수정
- **개발 시스템 및 문서화 도입**
 - 자체 웹 서비스 릴리즈 및 문서화 체계 구축
 - 개발팀 주요 관리 문서 및 프로세스 도입으로 업무 공유를 원활히 할 수 있도록 기여

- 회의록 / 릴리즈 문서 / 프로젝트 별 진행 상황 / QA 문서 / 온보딩 및 인수인계 문서 등등

팀스톤

주임 | FE 개발자

2023.08 ~

2024.08

✓ Tech stack

- typescript
- svelte
- websocket
- webworker
- tailwind
- echart
- svelte-i18n
- tabulator
- xlsx
- sheetjs
- yup
- vite
- ...

삼성, 메이저 은행권에서 사용하는 시스템 모니터링 서비스

✓ 기존 ontune v4를 웹버전으로 컨버팅해 v5로 출시

• 시스템 성능 분석 위젯 개발

- 시스템 성능 분석 위젯 개발 (총 8개)
- 위젯 커스텀 레이어 팝업 개발
- 워커 스레드를 이용한 위젯 데이터 처리
- 반응형UI 적용 및 유지보수성 고려
- 직접 사용해보며 불편한 부분은 주도적으로 의견을 제시해 사용자 친화적인 UI 구성
- 위젯 성능 개선
 - 위젯 리플로우 영역 최소화로 렌더 최적화 진행
 - interval을 활용한 애니메이션으로 인한 메모리 누수 개선 (requestAnimationFrame 이용)
 - CSS를 이용한 CPU 사용량 개선
 - 데이터 연산은 워커 스레드 이용으로 렌더 최적화 진행
 - 위젯 차트 트리쉐이킹
 - debouncing을 이용한 이벤트 발생빈도 조절

• Host 및 제품 튜닝 페이지 개발

- 데이터 기본 세팅 및 제품 튜닝 페이지 개발 (총 6페이지)
- 페이지 유지보수 및 기능 버그 해결
- 불필요한 리렌더 완화
- 데이터 관련 함수 안정성 검증
- 백-프론트 데이터 이슈 포인트 원활히 소통하며 의견 절충 및 정리

• 시스템 성능 분석 페이지 개발

- 시스템 성능 분석 페이지 개발 (총 2페이지)
- 위젯 지표 선택 레이어 팝업 개발
- 자동 선택 등 사용자 편의 기능 제공
- 메트릭 데이터 변환 공통 함수 개발

• 공통 컴포넌트 및 글로벌 함수 개발

- ReportBaseChart
 - 트리쉐이킹을 위해 기존 BaseChart를 활용해 개발
 - default상태를 지정해 데이터와 차트타입만 넘겨줘도 그려지도록 개발

- 높은 자유도를 보장해 다양한 커스텀이 가능하도록 개발
- WidgetMideaQuery
윈도우 미디어 쿼리를 이용 할 수 없는 상황에서 반응형 UI를 위해 제작하게 된 공통 컴포넌트
 - ResizeObserver api 이용
 - 스타일 자유도를 높여 언제든지 모드 추가가 가능하도록 개발
- BarChart / BarChartTooltip
Common Grid 내부에서 사용되는 BarChart 및 BarChartTooltip.
 - 라이브러리 없이 직접 구현

- 다국어 처리 경험
- 사원으로 입사해 성과를 인정받아 24년도 하반기 주임으로 승진
- 모든 프로젝트 1차 버전 배포 후 1차 QA 참여

딩코

FE 멘토, 개발자


2021.12 ~

2023.04


✓ Tech stack

- typescript
- Next.js
- apollo-client
- emotion
- yup
- recoil
...

인프런에도 올라온 온/오프라인 부트캠프

- 코드캠프 온/오프라인 부트캠프 학습자료 제작
 - 제작해온 학습자료 
 - 각 주제별 핵심 내용이 꼼꼼히 정리 되어있는 학습자료
 - 수강생에게 노션 장인의 자료로 평가받은 학습자료 (링크에 자료평 캡쳐본있습니다.)
- [인프런 X 코드캠프] & 코드캠프 온/오프라인 부트캠프 프론트엔드 멘토
 - 프론트엔드 수강생 일정 관리
 - 멘티 과제 확인 후 로직 피드백 및 질의 응답
 - 과제 피드백 및 질의 응답은 게더와 구글 클래스룸에서 이루어지며, 로직 피드백은 성능상이나 유지보수상 조금 더 나은 방향으로 제시
 - 최종 포트폴리오 피드백
- 코드캠프 광고 페이지 개발
 - SEO를 위한 시멘틱 태그 적용
 - 반응형 웹 제작
- 코드캠프 상담신청 페이지 개발
 - 유지보수성을 고려한 개발
- 딩코 사이트 어드민 & 유저 백업 개발
 - 유저 로그인 로그 데이터 개발

개발 문제 해결 사례

 를 클릭하시면 상세한 개선 사례 설명을 보실 수 있습니다.

1. 모노레포 + 버셀 빌드 및 배포 에러 해결 사례

✓ 특이사항

- 데이터 2초 주기 변경
- 화면 2초 주기 리렌더

✓ [문제]

Vercel 배포 실패 → 빌드 단계에서 모듈을 찾지 못해 에러 발생

✓ [원인]

모노레포 + npm 워크스페이스 환경에서 모듈 호이스팅으로 인한 패키지 경로 인식 실패

- 모듈 호이스팅으로 node_modules가 상위 루트 경로에서만 생성
- 빌드 할 대상은 하위 폴더이기 때문에 루트의 모듈을 읽어올 수 없음

✓ [해결]

npm은 직접적으로 호이스팅을 제어하기 어렵다.

가장 빠르고 확실한 해결 방법은 패키지 매니저를 pnpm으로 바꾸고, 워크스페이스를 pnpm 으로 바꾸는 것이라 생각했음.

따라서 pnpm으로 패키지 매니저 변경.

1. interval을 활용한 애니메이션으로 인한 메모리 누수를 개선한 사례

✓ 특이사항

- 데이터 2초 주기 변경
- 화면 2초 주기 리렌더

✓ [문제]

특정 위젯의 자바스크립트 메모리가 지속적으로 증가하는 메모리 누수 상황 발견

✓ [원인]

애니메이션을 구현하기 위해 사용한 interval이 너무 빈번하게 호출되며 clear 할 틈이 없어 발생한 문제

✓ [해결]

requestAnimationFrame api를 이용한 애니메이션으로 재구현

✓ [평가]

메모리 사용량이 개선되어 누수가 개선되고 더불어 CPU 사용량도 개선되며 화면이 많이 부드러워졌다.

하지만, GPU가 다소 많이 사용되는 것 같다는 상부의 평가가 있었음

✓ [비교]

현재 본인 지식선에선 리소스를 더이상 줄이는 것이 무리라 판단해 애니메이션 제거를 요청했으며, 수락

3. 패널의 active를 수정 시 불필요한 전체 리렌더로 인한 성능저하 개선 사례

✓ [문제]

패널의 active를 수정 시 페이지 전체에 불필요한 리렌더가 빈번히 발생
해당 리렌더는 그리드에 데이터가 많아질 경우 버벅임과 메모리 사용량이 크게 증가함.

✓ [원인]

1. 반응성 부여 시 리렌더를 실행 할 조건을 부여하지 않아 변수 중 하나라도 변경되면 무조건 전체가 리렌더 된다.
2. 페이지 내부에 필요한 대부분의 변수가 외부 파일에 하나의 객체로 묶여 있다.
→ grid의 인스턴트도 분리해 묶여놔서 패널 밖 그리드도 리렌더 발생
→ 해당 객체중 하나의 프로퍼티가 변경되면 객체 관련된건 와르르 리렌더 된다.

✓ [해결]


1. 반응성 부여 시 리렌더 실행 조건 부여,
리렌더를 쪼개서 실행하도록 렌더 시점 분리
2. 파일 분리가 굳이 필요하지 않은 변수들은 컴포넌트 내부에 선언
→ 웬만하면 블록 내부로 선언해 사용후 빠르게 사라지도록 리팩토링

✓ [평가]

버벅임과 메모리는 확실히 개선되었으며, 변수의 선언을 컴포넌트 내부로 일부 옮겨 리딩이 좀 더 쉬워지는 효과가 있었음.

✓ [비교]

사이드 프로젝트


 를 클릭하시면 상세한 프로젝트 설명을 보실 수 있습니다.

Huggie

side | 팀
프론트엔드 개발자

나만의 맛집을 컴팩하게 모아볼 수 있도록 만든 웹앱

✓ 팀원분들의 각자의 개인사정으로 중단된 프로젝트


- 🍏 **Huggie 소개 노션** 
- 지도 페이지 구현
- 카카오 소셜 로그인 구현
- **Tech stack**
typescript, React.js, vite, axios, styled-components, react-query, kakao-map, jest ...

Fitable

side | 개인
프론트엔드 개발자

이직을 위한 포트폴리오와 이력서를 탐색 받을 수 있는 사이트

✓ 배포 후 일정기간 이후 비용 문제로 인스턴스 삭제

- 🧑 **Fitable 소개 노션** 
- dompurify를 이용한 XSS 보안 강화
- 검색엔진 최적화를 위한 시멘틱 태그 적용
- 배포환경 및 운영체제 통일을 위한 Docker 사용
- FileReader, react-hook-form, hooks 분리, 캐시등을 이용한 프로젝트 성능 고려

- **Tech stack**

firebase, typescript, Next.js, react-query, react-hook-form, react-quill, dompurify, react-slick, antd, portone, emotion, yup, AWS, docker ...

여기까지 읽어주셔서 감사합니다.

추가로 궁금하신 점이나 필요하신 서류가 있으시다면

jaake0704@gmail.com 또는 010-8954-4608로 연락 주시기 바랍니다.

좋은 하루 보내세요.☀️