



김종민

프론트엔드 개발자

+82 01040236518 • yung3152@gmail.com

안녕하세요. 사용자 경험과 성능 최적화를 중시하는 3년차 프론트엔드 개발자 김종민입니다.

React와 Next.js를 주력으로 다양한 프로젝트를 진행했으며 안정적이고 사용자가 만족할 수 있는 제품을 개발해 왔습니다.

지속적인 모니터링을 통해 오류 최소화를 위해 노력하고 있고 성능 최적화를 위한 코드 스플리팅, Lazy Loading, Dynamic Import 등을 통해 로딩 속도를 개선한 경험이 있습니다.

개발 환경 개선을 위해 프로젝트 내에서는 코드 품질 향상을 위해 공통 ESLint 룰과 Prettier를 도입하고, 테스트 환경 구축을 위해 Vitest 기반의 단위 테스트 환경을 구성했습니다. 또한, GitLab 및 EC2 기반의 CI/CD 구축 경험이 있습니다.

사이드 프로젝트로는 Node.js를 활용한 백엔드 기능 구현에 도전하며 풀스택 개발 경험을 쌓고 있습니다. 이를 통해 프론트엔드와 백엔드 간의 통신 및 구조에 대한 이해를 넓혀가고 있습니다.

기술 스택

Next.js

React

TypeScript

react-query

zustand

vitest

styled-components

TailwindCSS

Node.js

경력

팀원 • 개발팀

주식회사엘케이벤처스서울본부

2024.04. ~ 2024.12. (9개월 | 정규직)

**고용승계로 입사했습니다.*

[인생네켓 점주 앱]

Next.js & Flutter 웹뷰 환경으로 구성된 앱을 개발했습니다.

- **Vitest를 사용한 단위 테스트 환경을 구축**

- 점주 전용 앱이다 보니 금액을 계산하는 부분에서 철저한 검증이 필요했습니다.
- 이전부터 테스트코드 작성 필요성을 느껴 개발 처음부터 테스트 코드를 작성했습니다.
- 테스트 코드 때문에 개발 속도는 저하 됐지만 안정적인 로직 구성이 가능했으며 QA도 성공적으로 통과했습니다.

- **SVG를 사용한 차트 구현**

- 차트 라이브러리로 기획서의 디자인을 구현하기 힘들어 SVG를 사용하여 차트를 직접 구현했습니다.

- **CI/CD환경 구축**

- GitLab Runner를 EC2에 등록하여 CI/CD환경을 구축했습니다.
- 코드가 커밋 되거나 배포에 실패하면 바로 확인 가능하도록 Slack과 연동 시켰습니다.
- 개발서버는 자동배포까지 되도록 설정했고
- 운영서버는 포트를 두개로 나누어 스테이징 환경도 구현했습니다.

[QR전용 사이트]

next-i18next 라이브러리를 사용하여 다국어 사이트를 개발했으며 MAU 10만명을 달성했습니다.

- **파일 다운로드 기능 구현**

- 회사 보안 규정상 브라우저에서 바로 다운하지 않고 Next.js 서버를 통해 다운로드 로직을 구현했습니다.
- 처음엔 파일리더로 구현했으나, 성능문제가 발견되어 스트림 방식으로 변경하여 성능개선을 했습니다.

- **Canvas API를 사용한 이미지 합성 기능 구현**

- Canvas API를 사용해서 사용자가 찍은 사진과 템플릿을 합성하는 기능을 구현했습니다.
- 이 과정에서 대용량파일과 Canvas 크기 때문에 모바일 사파리 브라우저에서 메모리제한 오류가 발생했습니다.
- Canvas와 이미지크기를 스케일링하고 사용하지 않는 Canvas객체는 바로 null 처리하여 메모리를 조금 더 확보하는 방법으로 오류를 해결했습니다.

[인생네켓 국내 앱]

인생네켓 국내앱 유지보수 및 기능개발을 담당했습니다.

• 이미지 업로드 기능 개선

- HEIC파일이 업로드되면 서버에서 처리가 불가능한 오류를 발견했습니다.
- HEIC파일 자체가 브라우저 호환성도 낮아 heic2any 라이브러리를 사용해서 변환 기능을 구현했습니다.
- 이 과정에서 AOS 버전에 따라 HEIC파일 처리가 달라 발생하는 트러블 슈팅을 경험했습니다.

• 베트남 버전 개발

- 베트남 전용 이미지 합성 사이트를 개발했습니다.
- 외부 API와 Next.js API Routes를 활용하여 IP 기반 국가 접속 제한 기능을 구현했습니다.

팀원 • 개발팀

(주)버추어라이브

2023.05. ~ 2024.04. (1년 | 정규직)

[Betrip]

Next.js & Flutter 웹뷰 환경으로 유지보수 및 기능 추가를 담당했습니다.

• API요청 횟수 75% 감소

- 유저정보와 배너 같이 변동성이 적은 데이터의 API를 계속 호출하는 것을 발견했고
- TanStack Query의 staleTime을 사용해서 데이터마다 캐싱 기간을 다르게 적용시켰습니다.

• Presentational-Container 패턴을 Atomic 패턴으로 변경

- 기존 프로젝트는 logic과 view를 분리하기 위해 Presentational-Container 디자인 패턴을 사용했습니다.
- 하지만 logic은 hook으로 관리하고 있었고 깔끔한 폴더 구조로 변경 필요성을 느껴, 팀원과 논의하여 Atomic 패턴을 커스텀해서 도입했습니다.
- Atomic 패턴으로 변경 후, props drilling이 조금 더 발생했다는 단점이 있었지만 깔끔한 폴더구조로 유지보수에 용이했습니다.

• TypeScript 레거시 코드 정리와 ESLint 룰 도입

- TypeScript로 작성된 코드가 any 사용이 많았으며 재사용 불가능한 Type이 많았습니다.
- Sentry에 타입 오류들이 지속적으로 올라와서 전체적인 레거시 코드를 정리했습니다.
- 또한 기존 개발 환경에선 공통된 ESLint 룰이 없어서 통일된 개발 환경이 구성되지 않았습니다.

- 이 문제를 해결하기 위해 Airbnb를 도입하여 개발 환경 개선에 성공했습니다.

- **평균 번들 크기를 130KB 감소**

- 유니코드를 판별하는 기능에 xregexp 라이브러리를 사용하고 있었습니다.
- 하지만 해당 라이브러리는 번들사이즈가 컸으며 유니코드만 판별하는 기능은 직접 구현 가능하다고 판단되어 라이브러리 소스를 분석해 정규식 기능을 직접 구현했습니다.
- 또한, Dynamic Import와 package.json의 sideEffects옵션을 사용해서 번들 크기를 감소시켰습니다.
- 이 과정에서 sideEffects 옵션을 false로 사용해서 에러가 발생했지만 파일 분석 후 css파일과 같이 sideEffects가 발생하는 파일은 제외시키는 트러블슈팅 경험이 있었습니다.

팀원 • 개발팀

(주)헬로비즈

2022.02. ~ 2023.02. (1년 1개월 | 정규직)

외주 전문 개발업체에서 클라이언트와 직접 소통하여 7개의 MVP제품을 개발했습니다.

- **Redux → TanStack Query & Zustand 마이그레이션 추진**

- 기존에는 Redux만 사용하여 서버와 클라이언트 상태를 모두 관리하고 있었습니다
- 이 구조에선 코드의 복잡함 때문에 유지보수가 힘들다 생각이 들어 TanStack Query와 Zustand로 마이그레이션을 추진했습니다.
- 마이그레이션 후 서버 상태를 손쉽게 캐싱 할 수 있었고 줄어든 코드량 덕분에 개발환경 개선에 성공했습니다.

- **React로 구성된 컴포넌트를 HTML과 JS 파일로 변환**

- 카카오 지도를 사용하던 공용 컴포넌트를 안드로이드 앱에서 사용해야 하는 일이 있었습니다.
- 안드로이드 앱에서 컴포넌트를 직접 접근할 수 없으므로 index.html 형태로 변환이 필요했습니다.
- 변환 작업을 진행하면서 React로는 간단하게 구현 가능했던 기능을 다시 구현하면서 느슨했던 JS지식을 되돌아보는 기회가 되었습니다.

- **DB구조 분석 및 API 설계서 작성**

- 회사 내부 사정으로 백엔드 인력이 부족하여 프로젝트마다 API 설계서를 작성해야 했습니다.
- 다행히 백엔드는 Node.js로 구성되어 있고 사이드프로젝트로 백엔드 경험이 있어서 API 설계서 작성에 문제는 없었습니다.
- 하지만 실무에서 사용하는 DB구조와 쿼리는 처음 접해서 시행착오가 있었지만 좋은 경험이었습니다.

포트폴리오

URL

 [깃허브](#)

 [블로그](#)

교육

강원대학교

2013.03. ~ 2020.02.

졸업

대학교(학사) | 국제무역학과

Powered by Rallit.