

이직/구직 중이에요

김태현

프론트엔드 개발자



반갑습니다, 주니어 프론트엔드 개발자 김태현입니다.

자기소개

- 혼자가 아닌 함께 성장하는 것을 가치 있게 생각합니다. 여럿이 함께 문제를 해결하며 성취감을 느낍니다.
- 요구사항과 문제 정의에 집중하여 기획 문서를 기반으로 적극적인 질문과 효과적인 소통으로 기획과 개발 간의 간극을 줄여QA시 오류를 최소화하고 프로젝트 효율성을 향상시켰습니다.
- 스스로의 실력을 객관화 하고 업무 마감일을 준수하여 분기별 평가에서 좋은 평가를 받았습니다.
- 질문을 두려워하지 않고 배우는 것을 즐깁니다. 스택오버플로우나 깃허브 이슈 등 직접 질문을 남기고 답변을 받아 문제를 해결 한 경험이 있습니다.
- 팀 리더로 팀원들의 업무 분배와 일정 조율을 효율적으로 이끌어내어, 프로젝트 진행에 도움이 되었습니다. 이에 따라 분기별 리뷰에서는 팀원들로부터 긍정적인 피드백을 받았습니다.

기술 스택

HTML/CSS, JavaScript, React, Next.js, Redux, redux-saga, react-query, Webpack, Git, GitHub, github-actions, VSCode, TypeScript

경력

주식회사 라이트하우스

프론트 팀 리더 | 웹 개발

2021.12. ~ 2024.01. (2년 2개월)

추천해요 🍷 | 제안하기 ✉ | 댓글 💬

회사 소개 Landing Site 개발

자사 서비스 오늘케어 개발 및 유지보수

자사 서비스 데일리북 개발 및 유지보수

자사 서비스 장지 헬프콜 개발 및 유지보수

협업 프로젝트 패밀리케어 개발 및 유지보수

팀 내 코딩 컨벤션 및 코드리뷰 도입

테크 스펙 작성을 통한 문서화 작업

신입 프론트엔드 개발자 업무 분배 및 일정 산출 도움

프로젝트

패밀리케어

라이트하우스

2023.03. ~ 2024.01.

장기요양기관의 운영 / 관리 프로그램과 보호자와 소통이 가능한 시니어 플랫폼

키즈노트와의 협업 프로젝트

- 코드 개선
 - 깃허브 액션 사용 중 공통적으로 불필요한 설치를 반복하며 워크플로우 시간이 길어져 반복되는 코드를 공통 코드로 묶어 유지보수성을 향상시키고 실행시간을 7분에서 5분으로 단축시킴
 - 이전 개발사의 코드가 하나의 컴포넌트에 비즈니스 로직과 UI 코드가 모두 모여있어 유지보수성이 매우 낮아 관심사별로 로직을 분리함으로써 유지보수성을 높임
- 기능 구현
 - canvas를 이용해 사용자가 그린 서명을 이미지화하고 확대 축소 초기화 기능을 통해 전자서명 기능을 구현
 - 서비스의 타입과 사용자의 타입에 따라 메뉴의 권한을 부여하는 기능 구현
 - 타사 CMS 서비스와의 연동을 통해 관리 페이지 및 api 구축
- 프로젝트 프론트 팀 리드
 - 장기요양 특성상 생소하고 이해하기 다소 어려운 기획을 먼저 분석 후 어떤 식으로 개발을 해야하는지 테크 스펙을 작성하여 공유함으로 팀원들의 업무 이해도를 높임
 - 팀원들과의 의견 조율로 PR 작성 이후 코드리뷰를 하고 머지해주는 문화를 도입했으며, 신입개발자들의 코드를 리뷰하며 코드의 일관성과 품질 향상에 기여하며 해당 프로젝트 이후 동일 적용 시킴
 - 주어진 업무를 각각 분석, 설계, 개발, QA, 배포로 나누고 분석과 설계를 돕고 개발 기간을 산정해줌으로써 일정을 준수하며 프로젝트를 관리하는 능력을 향상시킴

데일리북

장애인 주간보호센터에 입소한 이용인의 보호자를 대상으로 텍스트, 사진 기반의 알림장을 발송하고 수신할 수 있는 웹 서비스와 시설 관리자를 위한 ERP 서비스 제공

- 기능 구현
 - 시설 관리자가 이용인의 보호자를 대상으로 텍스트, 사진 및 PDF를 업로드할 수 있는 게시판 구현
 - 선택한 이용인의 관찰일지를 주간 단위로 확인 할 수 있는 테이블 및 모달을 통해 일지 등록과 수정 기능을 구현
 - 댓글과 대댓글 작성 기능을 개발하며, 여러 커뮤니티의 정책 및 가이드라인을 참고하여 유저 경험을 높임
 - 문서 출력 이전 출력 될 UI를 보여주는 공통 컴포넌트 개발
- 코드 개선
 - 메뉴 추가 시 발생하는 반복적인 메뉴를 배열 안에 객체로 변환 시켜 코드의 재사용성과 유지보수성을 향상 시킴
 - 알림장 컴포넌트에서만 사용되던 유저 선택 모달을 유저 타입에 따라 선택이 가능하도록 공용 컴포넌트로 확장
 - 유틸리티 타입과 제너릭을 통해 기존의 중복되던 타입 개선함으로써 재사용을 높임
 - 전역으로 관리하는 데이터의 부재로 인해 기존에 사용중이던 Redux Saga에서 React-Query로 마이그레이션
 - 프로젝트에서 코드를 분할하여 사용자가 필요로 하는 시점에만 로드되도록 지연로딩을 통해 전반적인 성능을 최적화 함
 - 앨범형 알림장 이미지의 최적화를 위해 지연로딩을 적용시킴으로 사용자 경험을 향상시킴
- 프로젝트 프론트 팀 리드
 - 팀원들과의 원활한 협업을 위해 코드 컨벤션 문서를 작성하고 코드 리뷰를 도입함으로써 자유롭게 의견을 교류할 수 있는 팀 분위기와 환경을 만듦
 - 낙후된 라이브러리들의 버전 체크 및 업그레이드 이유와 내용을 문서로 작성하여 팀원들에게 공유
 - 리액트 공식문서를 기반으로 사내 스터디를 주관함으로 공식문서를 자주 읽는 습관과 함께 기존에 놓치고있던 이론들을 재점검함으로써 리액트를 보다 효율적으로 사용

장지 헬프콜

장지를 찾는 고객과 장지 업체를 운영하는 사용자간의 상호 수요를 매칭하는 반응형 웹 서비스

- 코드 개선
 - Media-Query 모듈화
 - 기존에 정의해 둔 디바이스별 사이즈를 가지고 styled-components의 `css` 유틸리티 함수로 구현
 - 모듈화를 통해 여러곳에서 동일한 미디어 쿼리를 적용함으로써 중복을 방지하여 코드 재사용성을 높임
- 기능 구현

- 지도 검색 구현
- 지도 내에 화면을 움직일 경우 자동으로 재검색 시켜주는 로직 구현
- 라이브러리 공식문서의 인포창 커스텀 예시가 간단한 HTML Tag로 되어있어 PR을 통해 리액트에 맞게 커스텀 하는 방법을 안내 받아 적용시킴
- 요청서 구현
- 요청서에 입력되는 정보 값의 유효성 검사를 직접 구현함으로써 다양한 케이스에 대응하는 법을 배우고 유효성 검사를 지원하는 라이브러리와 테스트 라이브러리의 필요함을 느낌
- 기술 도입과 활용
 - React-Query 도입
 - 데이터의 전역 관리가 불필요해지며 Redux의 보일러 플레이트와 깊어지는 구조를 컴포넌트 내부에서 간단하고 직관적으로 API를 사용할 수 있는 이점
 - 공공API와 지도 API를 통해 받아오는 데이터의 fetching과 캐싱으로 지속적인 실시간 업데이트와 빠른 로딩 속도를 제공함으로써 보다 좋은 사용자 경험을 제공할 수 있는 이점

오늘케어

라이트하우스

2022.02. ~ 2023.01.

노인장기요양시설과, 해당 시설에 입소한 수급자의 보호자를 대상으로 텍스트, 사진 기반의 알림장을 발송하고 수신할 수 있는 웹 서비스 제공

- 기능구현
 - 기존 MPA 프로젝트를 SPA로 재설계 및 구현
 - 보호자 등록하기 기능
 - 사용자가 업로드 한 엑셀 파일을 라이브러리를 통해 JSON 데이터로 변환하고 유효성 검사를 거쳐 서버에 등록하는 기능 구현, 이 과정에서 유효성 검사 로직을 직접 구현하며 테스트 코드의 필요성에 대해 느낌
 - 사용자의 엑셀 버전에 따라 간헐적으로 라이브러리가 실행되지않아 라이브러리 공식문서를 통해 지원하는 엑셀 버전 확인 및 낮은 버전을 지원하는 다른 라이브러리를 조합하여 문제를 해결
 - Class를 활용한 HTTP 통신 구현
 - 반복되는 HTTP 메서드를 간편하게 활용할 수 있고 공통적인 메서드를 부모 클래스로 구현하여 각 API에서 상속받아 사용
 - 중복되어 작성된 HTTP 메서드 코드 제거 및 가독성과 신규 API 추가 시 유지보수성 향상
- 코드 개선
 - 상수화를 통한 파일 관리
 - 자주 사용되는 PATH경로, 에러처리 문구, KEY값등을 상수화 시켜 별도의 파일로 관리함으로 유지보수성 향상
 - 환경변수를 통한 프로젝트 분리
 - 오늘케어와 동일한 기능을 제공하는 데일리북 구현 시 환경 변수를 통해 서비스명과 몇가지 단어를 수정하는 방식으로 구현
 - Webpack 설정
 - DotEnv를 활용하여 dev환경과, prod 환경의 설정 분리

- HtmlWebpackPlugin을 통한 html 템플릿화 및 SEO작업
- BundleAnalyzerPlugin을 사용하여 번들 크기 확인 후 최적화 진행
 - 더 이상 사용하지 않는 라이브러리 제거로 번들 크기를 1.78MB에서 1.47MB로 줄임
- CleanWebpackPlugin으로 빌드 된 결과물을 자동 정리하여 최적화 진행
- 기술 도입과 활용
 - 런타임 에러방지와 안전한 프로젝트를 위하여 TypeScript 도입
 - Typescript도입에 대한 테크스팩 작성
 - Utility Type을 통해 반복되는 코드 제거 및 안전한 타입 설계
 - Atomic Design Pattern 도입
 - 작은 단위인 'atom'부터 'templates'까지 구성하며 조합하는 방식으로 디자인의 재사용성을 높임
 - 프로젝트 전반에 걸쳐 일관된 디자인을 통해 개발 효율성이 향상됨
 - Sentry 도입
 - 테스트와 QA에 걸리지 않은 에러 추적과 신속한 대응을 통해 서비스 안정성 향상
 - 빠른 버그 발견으로 인해 고객 문의가 들어오기 전 해결한 경험

회사 소개 Landing Site

라이트하우스

2021.12. ~ 2022.01.

(주) 라이트하우스 소개 웹 사이트 제작

- 기능 구현
 - 검색엔진 최적화(SEO)를 위해 시멘틱 마크업을 준수하여 작성
 - XMLHttpRequest를 사용한 HTTP Request 통신 구현
 - 단 하나의 POST요청을 위해 라이브러리 설치보단 기존에 제공하는 fetch와 XMLHttpRequest중 Axios가 XMLHttpRequest를 베이스로 사용하기에 XMLHttpRequest선택하여 구현
- 코드 개선
 - Styled-Components의 글로벌 스타일과 theme을 활용하여 반복되어 사용되는 스타일을 정리
 - 컴포넌트를 import시 상대경로로 지정하는 방식이 번거롭기도 하고 구조가 복잡해지면 코드가 길어지기에 절대 경로 설정

구독 서비스 스케줄러 "구딩"

사이드프로젝트

2022.08. ~ 2022.08.

다양한 종류의 구독 서비스를 관리할 수 있는 서비스 제공

깃허브 : <https://github.com/Princess-Teo-And-The-Seven-Frontend/gooding>

프로젝트 : <https://gooding.vercel.app/>

- 다른 직군과의 협업을 경험하면서 정작 프론트엔드 개발자와의 협업 경험이 상대적으로 부족하여 이를 보완하고자 프론트엔드 개발자로만 이루어진 사이드 프로젝트를 진행

- 7일간 구글 스프린트 방식으로 기획부터 디자인, 개발 그리고 배포까지 진행하며 코드 컨벤션과 PR 템플릿, 코드리뷰를 통하여 협업을 진행
- React로도 검색 엔진 최적화를 할 수 있지만 NextJS가 SSR으로 별다른 라이브러리 없이 검색 엔진 최적화를 할 수 있기에 NextJS를 선택하고, 가볍고 쉽게 배울 수 있다는 장점의 Recoil을 선택하여 유저정보와 모달의 상태를 관리