성민석

Frontend Engineer

안녕하세요, 사용자 경험을 최우선으로 생각하는 프론트엔드 개발자 성민석입니다.

개인적으로 사용했던 앱들의 복잡한 UI로 인해 불편함을 겪으며, 사용자 경험의 중요성을 깨달았습니다. 이러한 경험을 바탕으로 실제 프로젝트에서 베타 테스터들의 피드백을 적극 수용하여 UI를 개선했고, 그 결과 출시 2주 만에 400명의 활성 사용자를 확보할 수 있었습니다.

디자이너와의 협업 과정에서 재사용 가능한 컴포넌트의 중요성을 인식했고, 이를 실천하기 위해 컴파운드 패턴과 헤드리스 UI를 적용했습니다. 이를 통해 유지보수가 용이하고 확장 가능한 컴포넌트를 구축했으며, 개발 생산성과 프로젝트의 UI 일관성을 크게 향상시켰습니다.

또한, Next.js, React, TypeScript의 공부내용을 개인 블로그에 꾸준히 기록하며 지식을 공유하고 있습니다.

앞으로도 최신 프론트엔드 기술을 익히고 적용하여 사용자들에게 직관적이고 매력적인 서비스를 제공하는 개발자로 성장하고 싶습니다. 감사합니다.

기술

HTML5 & CSS3 ： WebP와 <picture> 태그를 활용한 이미지 최적화 및 크로스브라우징 대응으로 LCP 개선 , aria-label 속성을 활용한 웹 접근성 준수 경험, Flexbox를 활용한 반응형 레이아웃 구현 경험

JavaScript (ES6+) - async/defer를 활용한 스크립트 로딩 최적화 경험, 실행 컨텍스트, 클로저, 스코프 체인, this 바인딩에 대한 깊은 이해, Promise, async/await를 활용한 비동기 프로그래밍 구현 경험

TypeScript: 타입 시스템을 활용하여 코드의 안정성과 가독성을 높이기 위해 노력하고 있으며, 타입 연산자에 대해 학습하였습니다.

React: React Hook의 동작 원리에 대한 이해, 커스텀 훅을 활용한 비즈니스 로직 모듈화 경험, React Query를 활용한 서버 상태 관리 구현 경험

etc: Git, Styled-components, Axios, React Query, React Hook Form,Zustand

—

프로젝트

승리요정 / 프로그래머스 데브코스

2024.07. ~ 2024.08.
디자이너 2명 / FE 3명 / BE 2명

프로젝트 설명

프로젝트 링크 (ID :imuchgabisss@naver.com PW : Test123@)

야구 직관 기록 서비스 '승리요정'입니다. 사용자는 야구 직관을 기록하고, 개인 승률을 확인할 수 있으며, 야구장별 주차장 정보와 팀/선수 응원가를 제공합니다.

사용 기술

React, TypeScript, Styled-Component, Zustand, React-Hook-Form, Zod, React Query

주요내용

성능 및 웹 접근성 개선

문제: 초기 **Lighthouse** 점수가 성능 46점, 접근성 67점으로 낮았습니다. 특히 LCP와 FCP가 개선이 필요했습니다.

해결:

1. Lazy Loading 및 코드 스플리팅
    - **Lazy Loading**과 코드 스플리팅을 통해 필요한 리소스만 로드하여 초기 로딩 속도를 향상시켰습니다.
2. 이미지 최적화
    - 이미지의 바이트 크기를 줄이기 위해 **WebP** 포맷으로 변환하고, 이를 **preload**하여 빠른 표시가 가능하도록 하였습니다.
    - 브라우저 호환성을 고려하여 **WebP**를 지원하지 않는 경우 **PNG** 포맷을 대체로 보여줄 수 있도록 설정하였습니다.
3. 웹 접근성 개선
    - 버튼에 **aria-label**을 추가하여 스크린 리더 사용자가 더 쉽게 이해하고 사용할 수 있도록 개선하였습니다.
4. 폰트 최적화
    - **preload**된 폰트를 **`font-display: swap`**을 통해 로드 전 시스템 폰트를 적용하여 사용자 경험을 개선하였습니다.

정보페이지 개발**(**주차장페이지**,** 응원가페이지**)**

1. 주차장 페이지 개발

   문제: 구글 애널리틱스 데이터를 분석한 결과, 주차장 페이지의 방문율이 낮다는 사실을 확인했습니다. 이로 인해 지도 로딩 방식을 개선할 필요가 있었습니다.

   해결:

   - 비동기 로딩 방식을 도입하여 `index.html`에서 지도를 직접 로드하는 대신, **async**를 활용했습니다. 이를 통해 페이지 로딩 성능을 개선하고 사용자가 지도가 필요할 때만 로드되도록 하였습니다.
   - 이 과정에서 **async**와 **defer**의 차이를 이해하게 되었으며, 비동기 방식이 페이지 초기 로딩 속도에 긍정적인 영향을 미친다는 점을 확인했습니다.

2. 응원가 검색 기능 개발

   문제: 응원가 검색 기능에서 타이핑이 발생할 때마다 서버로 요청을 보내어 불필요한 요청이 발생하고 있었습니다.

   해결: 디바운스**(debounce)** 개념을 적용하여 이벤트를 그룹화하였습니다. 이를 통해 마지막 이벤트가 끝난 후에 한 번만 호출되도록 설정하여 잦은 렌더링과 API 호출을 줄일 수 있었습니다.

React Hook Form 및 **Zod** 도입

해결

- **React Hook Form**을 통해 제어 컴포넌트와 비제어 컴포넌트의 장점을 결합하여 폼 상태를 간편하게 관리하였습니다.
- 또한, **Zod**를 활용하여 유효성 검사를 함수가 아닌 스키마로 정의함으로써 코드의 가독성을 높이고, 유효성 검사 로직을 간편하게 유지할 수 있었습니다.

에러바운더리 도입

문제: axios인터셉터와, api요청 에러처리 외의 렌더링에러, 네트워크에러 처리 방법 필요했습니다

해결

- 에러 바운더리를 도입하여 컴포넌트 레벨에서 발생하는 오류와 네트워크 오류를 잡았습니다.
- 이를 통해 사용자에게 친숙한 **fallback UI**를 표시하여 더 나은 사용자 경험을 제공했습니다

회원가입, 로그인 구현

1. 멀티 스텝 관리

   문제: 폼 단계를 전역 상태로 관리할지, 중첩 라우팅을 사용할지 고민이 있었습니다.

   비교:

   - 전역 상태 관리: 단계 간 데이터 공유가 용이하며, 상태 변경 시 전체 폼의 흐름을 쉽게 제어할 수 있습니다.
   - 중첩 라우팅: 새로고침 시에도 현재 단계가 유지되어 접근성이 높아지고 SEO에 장점이 있지만, 사용자가 URL을 직접 변경해 스텝을 건너뛸 위험이 있습니다.

   해결 : **Zustand** 라이브러리를 사용해 전역상태로 폼 단계를 관리했습니다

2. 리프레쉬 토큰, 액세스 토큰 관리 및 갱신

   문제: 리프레시 토큰과 액세스 토큰 관리 및 갱신 방법을 어떻게 할지 고민이 있었습니다.

   해결: **Axios** 인터셉터를 활용해 모든 요청을 가로채 토큰 만료 여부를 확인하고, 만료 시 자동으로 갱신될 수 있도록 공통 처리했습니다. 이를 통해 토큰 관리 로직을 인터셉터로 간소화하였습니다.

리액트쿼리 도입

리액트 쿼리를 도입하여 데이터 패칭과 캐싱을 효율적으로 관리했습니다. 도입한 이유는 다음과 같습니다

1. 캐시 기능: 자주 변경되지 않는 데이터를 캐시하여 불필요한 요청을 줄이고 서버 부하를 완화했습니다.
2. 무한 스크롤 기능: 무한 스크롤을 간편하게 구현할 수 있기에 사용하였습니다

자주 쓰이는 **ui**만들어보기 **/** 개인

2024.10

프로젝트 설명

프로젝트 링크

중첩모달,토스트 등 다양한 UI 컴포넌트를 라이브러리 없이 직접 개발해보는 프로젝트입니다. 컴포넌트 설계, 상태 관리, 재사용성, 디자인 패턴에 대해 학습하며 총 11개 컴포넌트를 개발하였습니다.

주요 내용

IntersectionObserver 사용

문제**:** 사용자가 모든 이미지를 한 번에 로드하면 초기 로딩 시간이 길어지고 성능 저하가 발생합니다.

해결**:** IntersectionObserver를 통해 무한 스크롤과 이미지 레이지 로딩을 구현했습니다. 이 기술을 사용함으로써 사용자가 스크롤할 때 필요한 이미지들만 로드하여 성능을 개선했습니다.

포탈**(Portal)** 활용

문제**:** 모달, 툴팁, 팝오버 컴포넌트가 부모 태그의 overflow 속성으로 인해 UI가 잘리는 문제가 발생했습니다.

해결**:** 포탈을 활용하여 컴포넌트를 독립된 **DOM**에서 렌더링하였습니다. 이를 통해 UI가 다른 콘텐츠와 겹치지 않도록 하고, getBoundingClientRect()를 사용하여 정확한 위치를 계산할 수 있게 되었습니다.

컴파운드 패턴**(Compound Pattern)** 적용

문제**:** 복잡한 UI 컴포넌트의 재사용성과 유지보수성이 낮아지는 문제가 있었습니다.

해결**:** 드랍다운과 모달 UI를 컴파운드 패턴으로 개발하여 각 컴포넌트의 유연성을 높이고, 변화에 적응하기 쉬운 구조로 개선하였습니다.

헤드리스 **UI** 컴포넌트 사용

문제**:** UI와 로직이 결합되어 있으면 기능적인 부분의 재사용성이 떨어지는 문제가 발생했습니다.

해결**:** 헤드리스 UI 컴포넌트를 도입하여 UI와 로직을 분리했습니다. 이로 인해 기능적인 부분은 공통적으로 제공하면서도 UI는 개발자가 원하는 대로 커스터마이즈 할 수 있도록 하여 유연성을 높였습니다.

교육

영진 전문대학교

종류 | 전공
대학교(전문학사) | 컴퓨터정보계열

재학 기간 | 재학 상태
2018.03. ~ 2024.02. | 졸업
졸업학점 : 3.7

프로그래머스 풀스택 데브코스 **2**기 수료

JavaScript: "모던 자바스크립트 딥 다이브"스터디를 진행하여 ES6+ 문법과 비동기 프로그래밍, 실행 컨텍스트 및 이벤트 루프에 대한 깊이 있는 이해를 얻었습니다.

React: 공식 문서 스터디를 통해 React 훅(useState, useEffect 등)의 동작 원리를 학습하고, 상태 관리 및 컴포넌트 생명주기에 대한 이해를 높였습니다.

프론트엔드 개발: TypeScript와 React를 활용하여 사용자 인터페이스를 개발하는 방법을 학습하였습니다.

백엔드 개발: Node.js, Express, MariaDB를 활용하여 REST API를 설계하고, 데이터베이스와의 연동 경험을 통해 전체적인 웹 애플리케이션 구조를 이해하였습니다.

지식 공유 및 협업: 매일 아침 10시에 스크럼을 통해 동료들과의 지식 공유를 진행하였고, 팀 프로젝트에서의 협업 개발 경험을 통해 커뮤니케이션의 중요성을 깨달았습니다.

.

자격증

정보처리산업기사

발급기관
한국산업인력공단

취득년월
2023.09.

url

깃허브 : https://github.com/minseoook

블로그 : https://velog.io/@minseoook/posts

포트폴리오 :
https://cdn.rallit.com/attachment/2024-10-21/Y6qsvGtjftwOufYCswqfT/%ED%8F%AC%ED%8A%B8%ED%8F%B4%EB%A6%AC%EC%98%A4-%EC%84%B1%EB%AF%BC%EC%84%9D.pdf