



{ TRY } && { GROW }

안녕하세요!

오늘의 시도로 내일 더 성장하는 개발자
황연주입니다.

E-mail. hyjgood2000@gmail.com

이수 교육.

2024.07 ~ 2025.06 삼성 청년 소프트웨어 아카데미 (SSAFY) 수료

2023.02 ~ 2023.06 NCS 기반 UI/UX 웹 디자인 & 웹 퍼블리싱 과정 수료

경력.

2023.11 ~ 2024.04 웹 퍼블리셔 & 웹 디자이너

2018.09 ~ 2023.02 화장품 제형 연구원



INDEX

제가 쌓아온 기술과 프로젝트 경험을 소개합니다.

1 SKILL page 3

2 PROJECTS page 4 - 13

PROJECT 1. VOPL

PROJECT 2. YOOHOO

PROJECT 3. Qgen

PROJECT 4. JustOn

3 UI/UX page 14



SKILL

기술에 대한 **빠른 적응력을 키우자**는 마인드로 다양한 기술을 도전하고 습득해왔습니다.

기술 수준



React

상
A horizontal progress bar with five segments, all of which are dark blue.

useState, useEffect 등 다양한 Hooks를 사용해 상태를 관리하고, 상태 관리 라이브러리로 복잡한 상태를 효율적으로 관리할 수 있습니다.

Next.js

중
A horizontal progress bar with five segments. The first three segments are dark blue, and the last two are light blue.

CSR, SSR, SSG의 차이점을 이해하고 적절히 활용할 수 있으며, API 라우트를 활용한 서버리스 함수 구현이 가능합니다.

Vue.js

중
A horizontal progress bar with five segments. The first three segments are dark blue, and the last two are light blue.

Composition API를 활용해 재사용 가능한 로직을 구현할 수 있고, Pinia를 활용한 상태 관리를 할 수 있습니다.

TypeScript

중상
A horizontal progress bar with five segments. The first four segments are dark blue, and the last segment is light blue.

유니온 타입, 인터섹션 타입 등의 타입 연산자와 유틸리티, 고급 타입에 대해 이해하고 있으며, 함수형 프로그래밍과 ES6+ 문법을 알고 있습니다.

JavaScript

상
A horizontal progress bar with five segments, all of which are dark blue.

프로토타입 기반 객체지향 프로그래밍을 이해하고 있으며, 비동기 프로그래밍 (Promise, async/await)과 클로저와 스코프에 대해 이해하고 있습니다.

Git & GitHub

중
A horizontal progress bar with five segments. The first three segments are dark blue, and the last two are light blue.

Git을 사용하여 매번 프로젝트 협업을 진행하였으며, Git-Flow에 익숙합니다. 팀원들과 code-review를 통해 코드를 개선한 경험이 있습니다.

HTML

상
A horizontal progress bar with five segments, all of which are dark blue.

시맨틱 태그를 활용해 웹 접근성을 고려한 마크업을 작성합니다. Keyframes를 사용하여 슬라이드, 글리치 등 여러 애니메이션 작업을 할 수 있습니다.

CSS

상
A horizontal progress bar with five segments, all of which are dark blue.

Flexbox와 Grid를 활용한 레이아웃 구성에 능숙하며, SCSS 등 CSS 전처리기 및 CSS-in-JS 기법을 활용하여 동적인 스타일링을 구현할 수 있습니다.

PROJECTS

직접 설계한 UI/UX와 학습한 기술을 적용하여 구현한 저의 **프로젝트**를 소개합니다.



프로젝트 경진대회 수상작

PROJECT 1. VOPLÉ



중·고등학생을 위한 선거 유세 및 관리 플랫폼으로 프론트엔드를 맡아 WebSocket을 활용한 실시간 채팅과 라이브 영상 송출에 대한 상태값 관리 및 이에 따른 렌더링 처리를 구현하였습니다.

PROJECT 2. YOOHOO



유기견 후원 단체들의 후원금 사용 내역을 투명하게 보여주는 서비스로 Chart.js를 사용하여 단체별 후원 내역을 시각화하고, 결제 서비스 단계를 사용자 친화적으로 개선하였습니다.

PROJECT 3. Qgen



AI와 RAG 기술을 활용하여, 사용자가 입력한 데이터를 기반으로 맞춤형 문제집/시험지 리스 나만의 시험지 문제를 생성해주는 서비스 입니다. 문제집/시험지 리스트 관리 페이지를 구현하고 데이터 정형화를 맡아 진행하였습니다.

PROJECT 4. JustOn



사용자가 기록한 운동 일기를 기반으로 운동 전 필요한 운동 부위를 추천해주는 서비스 입니다. 운동 일기에 대한 서버 로직을 구현하고, 메인 및 운동 일기 페이지와 운동 추천 슬롯머신 UI를 구현하였습니다.

PROJECT 1

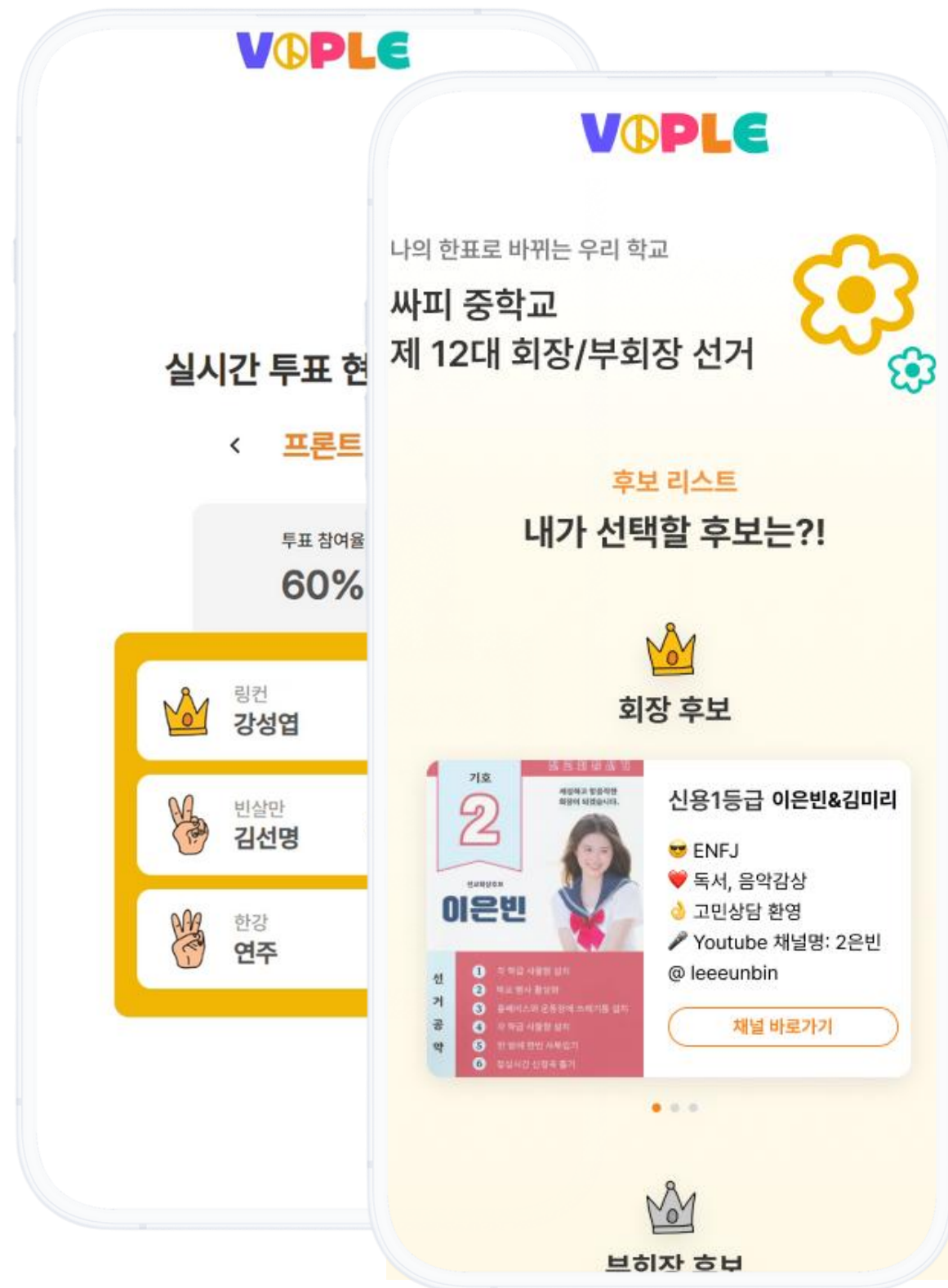
중·고등학생을 위한 선거 관리 플랫폼



삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 3위 수상작

기간 2025.01.05 ~ 2025.02.21 | 참여인원 6명

중고등학생을 위한 선거 관리 서비스로, React와 TypeScript를 활용하고, Storybook을 활용해 컴포넌트를 문서화하며 Atomic Design Pattern을 적용해 개발했습니다. 실시간 투표 현황을 직관적으로 보여주는 시각화 컴포넌트와 라이브 선거 화면을 구현했습니다.



담당 업무

프론트엔드 개발

UI/UX 디자인 및 설계 / 실시간 데이터 시각화
라이브 스트리밍 및 실시간 채팅 화면 구현

주요 기능

실시간 투표 결과 시각화

실시간 채팅 기능 (선거 유세, 투표 결과)

실시간 선거 유세 라이브 스트리밍

트러블 슈팅 사례

- 1) 라이브 선거 화면에서 후보자와 유권자에 따 화면 렌더링 분기 처리 문제 해결
- 2) 라이브 선거 화면의 실시간 채팅창의 탭 이동 시 초기화되는 현상 해결

주요 기술 스택



React



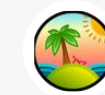
TypeScript



Storybook



Zustand



TanStack Query



MSW



Vite



yarn



Sass

VOPLE 트러블 슈팅 사례 1)

발생한 문제

실시간 라이브 선거 유세 페이지에서 후보자 화면과 투표자 화면이 스트리밍 중단 여부에 따라 적절한 화면 전환이 이루어지지 않음

원인 분석 방법

- 1) console.log를 통한 WebSocket 이벤트 수신 로그 확인
- 2) React DevTools를 활용한 컴포넌트 트리과 상태 변화 모니터링
- 3) 유사 사례 검색을 통한 원인 분석

파악한 원인

UI 전환 로직에 중첩된 조건문이 복잡하게 얹혀 있어 렌더링 순서가 명확하지 않고, isStreaming, isWaiting 등 여러 상태 변수가 개별 관리되어 상태 간 충돌 발생

해결한 방법

- 1) 분산된 상태 통합 : 개별적이었던 isStreaming, isWaiting 등의 state 변수들을 **streamingStatus**라는 단일 객체로 통합하여 상태 간 충돌을 방지했습니다.
- 2) 컴포넌트 분리 : 조건부 렌더링이 더 독립적으로 이루어지도록 **LiveStream**, **WaitingScreen**과 같은 별도 컴포넌트로 분리했습니다.
- 3) 상태 전환 로직 단순화 : 상태 값에 따라 명확하게 분기 처리하여 렌더링 프로세스를 예측 가능하게 변경했습니다.

스트리밍 상태 및 사용자 권한별 화면

	후보자(Candidate)	투표자(Voter)
isStreaming = true; (라이브 시작 후)		
isStreaming = false; (라이브 시작 전)		

VOPLLE 트러블 슈팅 사례 2)

발생한 문제

- 1) 실시간 라이브 선거 유세 페이지에서 하단 탭 변환 시 실시간 채팅창이 계속해서 재랜더링 및 새로고침 되는 현상 발생
- 2) 메시지 전송 시, 메시지 리스트 순서가 역순으로 입력됨

원인 분석 방법

- 1) 브라우저 개발자 도구를 활용한 단계별 디버깅
- 2) WebSocket 이벤트 로깅 시스템 구축
- 3) 컴포넌트 마운트/언마운트 추적

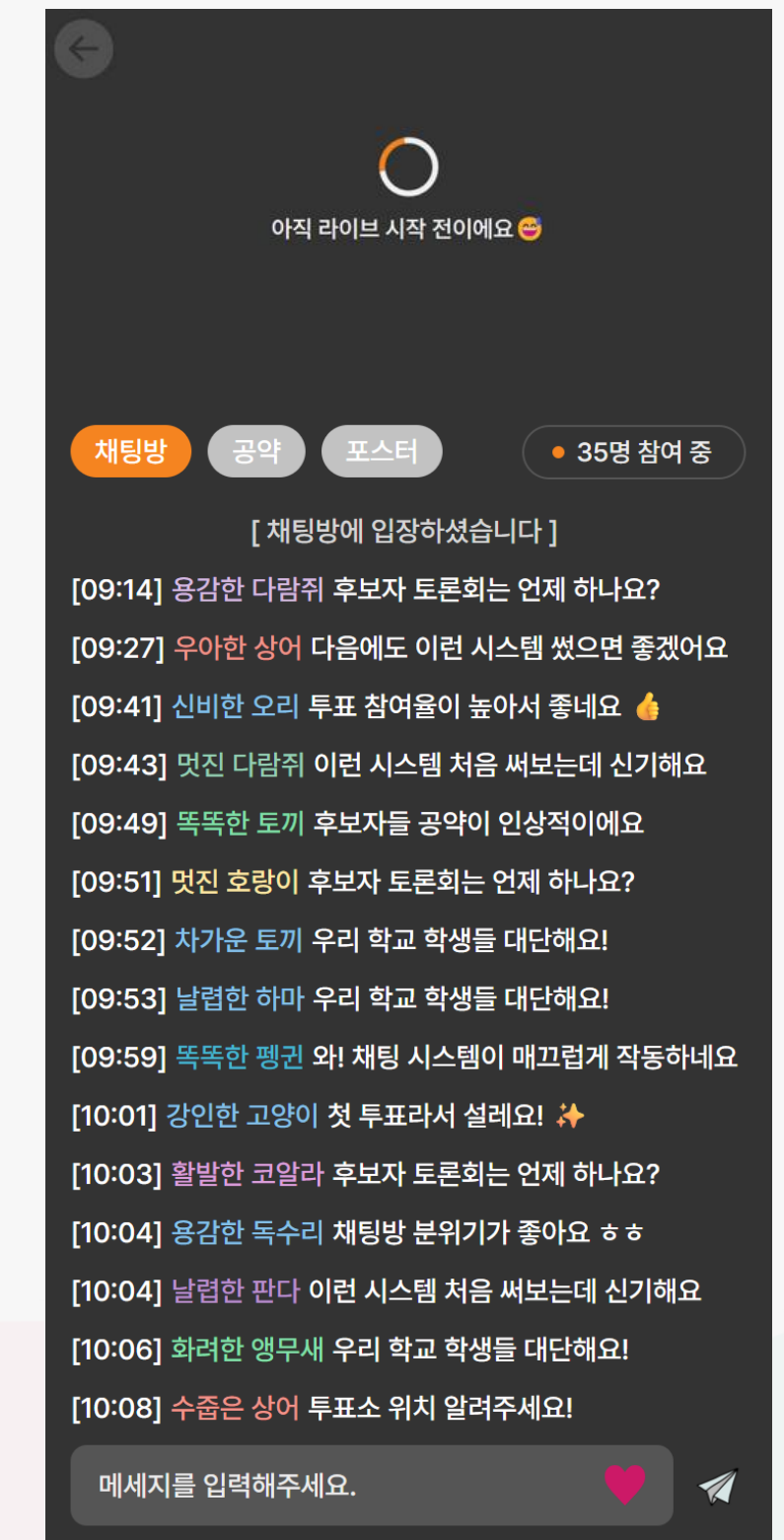
파악한 원인

탭 전환 시, 채팅창 컴포넌트가 재마운트 되면서 WebSocket 연결이 재시작되고 기존 채팅 기록이 초기화됨.
WebSocket 연결이 컴포넌트와 강하게 결합되어 있어, UI 변경 시마다 연결이 끊어지고 재연결됨.
메시지 배열에 새 메시지를 추가할 때 순서가 일관되지 않음.

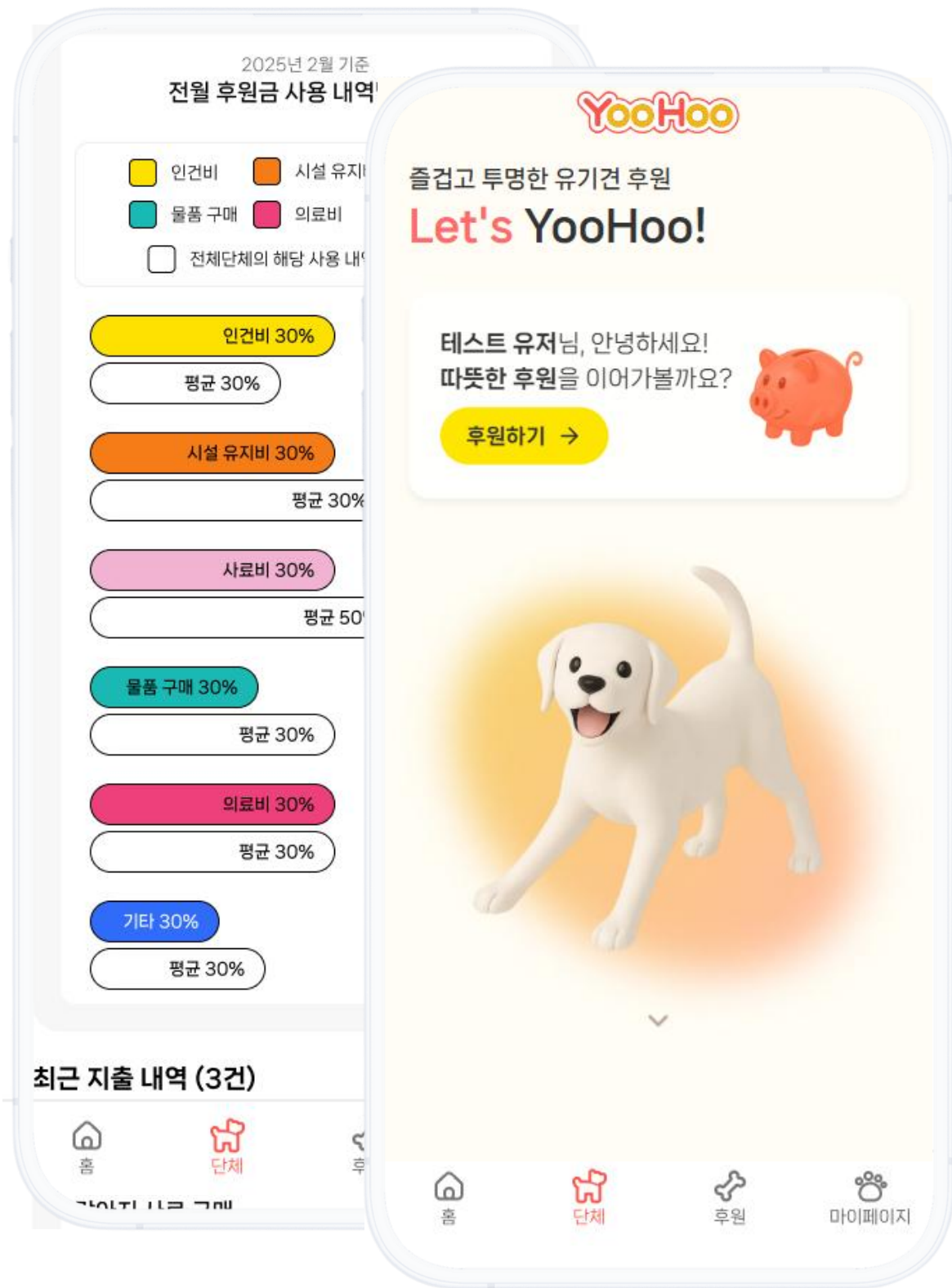
해결한 방법

- 1) 상위 레벨에서 WebSocket 연결 관리 : WebSocket 연결을 UI 컴포넌트와 분리하여 전역 상태로 관리
- 2) 컴포넌트 조건부 표시 방식 개선 : 컴포넌트를 언마운트하지 않고 CSS로 숨김/표시 처리
- 3) ChatBoard 컴포넌트 최적화 : 상태 구독만 하고 WebSocket 관리는 하지 않음
- 4) 메시지 배열 추가 방식 개선 및 타임스탬프 기반 정렬: 배열 추가 방식을 메시지 렌더링 데이터와 통일하고 항상 배열 끝에 추가되도록 변경 및 타임 스탬프 기반으로 정렬되도록 코드 개선. 서버 쪽에도 동일하게 반영.

실시간 선거 유세 화면



PROJECT 2



즐겁고 투명한 유기견 후원



기간 2025.02.24 ~ 2025.04.11 | 참여인원 6명

유기견 후원을 위한 플랫폼으로, 후원금 내역을 투명하게 확인하고 쉽게 후원할 수 있는 서비스입니다. React와 TypeScript를 사용했으며, Chart.js로 후원금 사용 내역 대시보드를 개발했습니다.



삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 2위 수상작

주요 기능

프론트엔드 개발 (팀장)
UI/UX 디자인 및 설계
단체의 후원금 사용 내역 시각화

담당 업무

강아지 후원 및 응원 메시지 전달
투명한 후원금 사용 내역 시각화
사용자 친화적 결제 인터페이스

트러블 슈팅 사례

후원금 사용 내역 Chart 조회 시 대시보드 렌더링 지연 문제 해결

주요 기술 스택



Next



TypeScript



Storybook



Zustand



TanStack Query



MSW



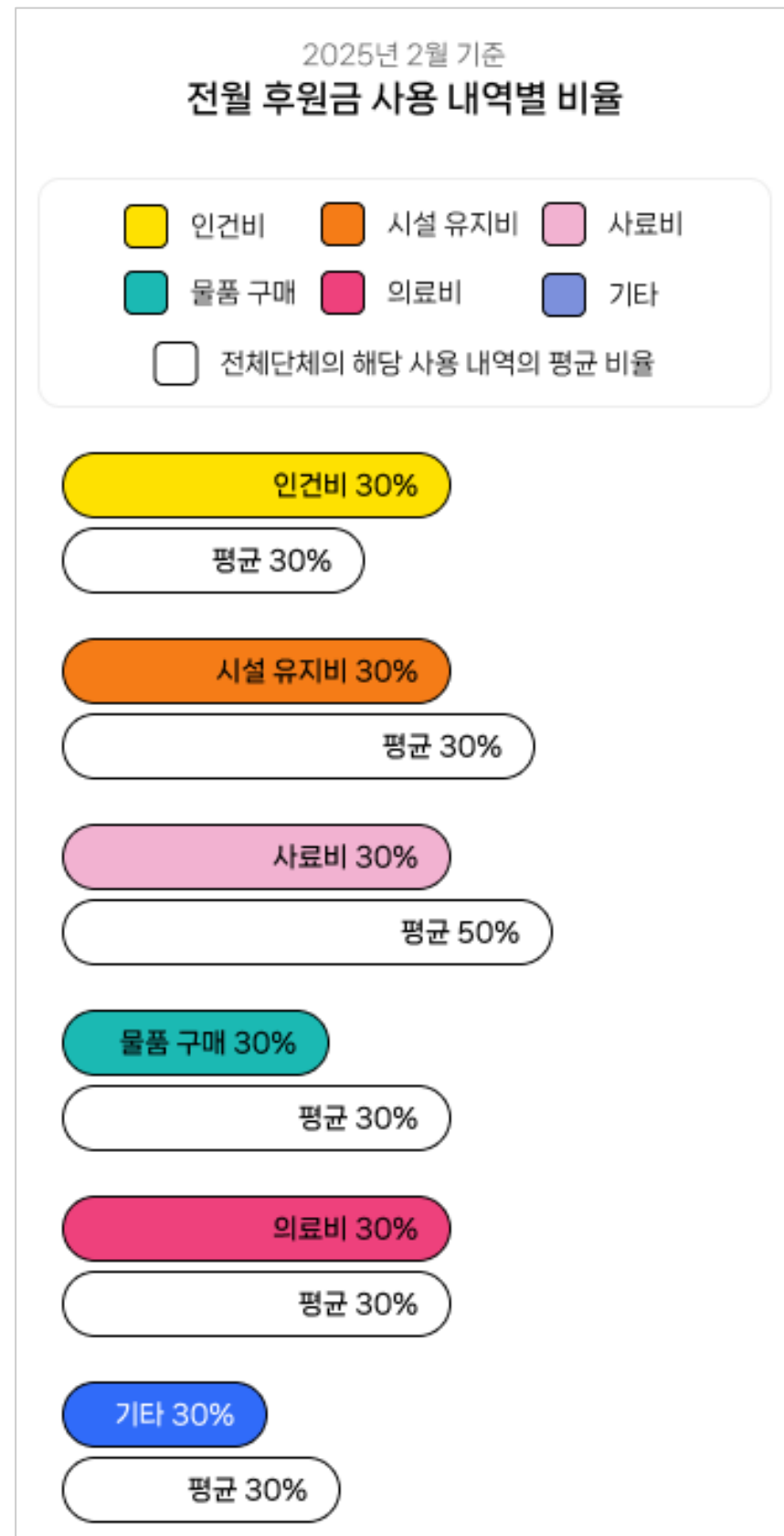
Vite



yarn



Sass



YooHoo 트러블 슈팅 사례

발생한 문제

후원금 사용 내역 대시보드 렌더링 시 성능 저하 문제 발생
대량의 후원 데이터가 있을 경우 페이지 로딩 시간이 4초 이상 소요됨

원인 분석 방법

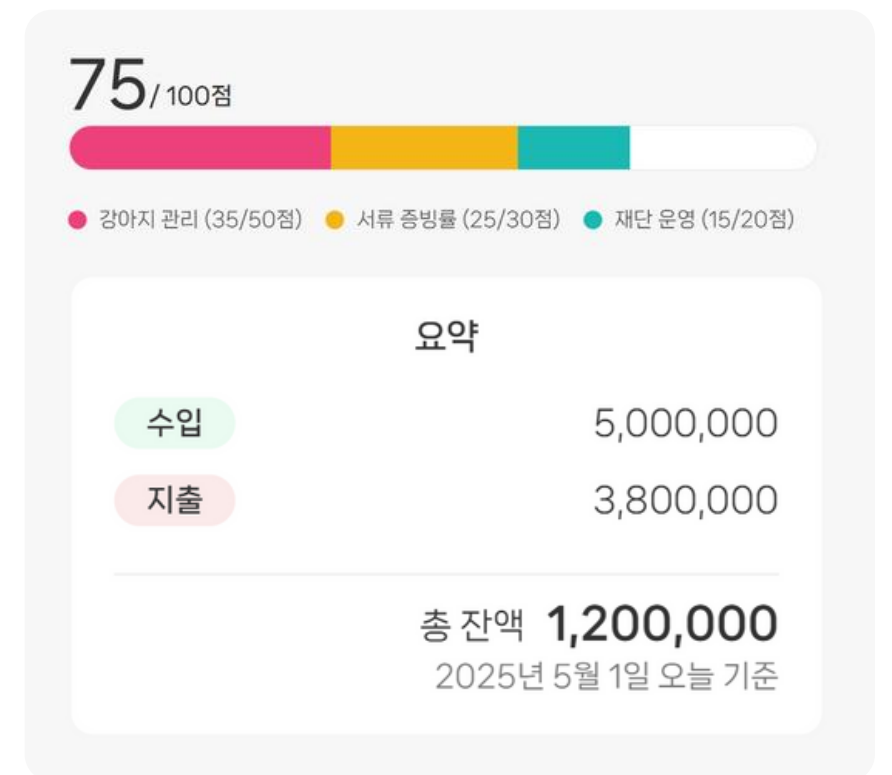
- 1) Chrome DevTools Performance 탭을 활용한 렌더링 병목 지점 파악
- 2) **React Profiler**를 사용하여 컴포넌트별 렌더링 시간 측정
- 3) Chart.js 렌더링 옵션 검토 및 **최적화 방안 조사**

파악한 원인

후원금 데이터를 시각화하는 Chart.js 컴포넌트가 탭 변경 시 불필요하게 리렌더링되는 현상 발견
후원 내역 필터링 시 **클라이언트에서 전체 데이터를 처리**하여 메모리 사용량 과다
각 차트 컴포넌트가 독립적으로 동일한 **API를 중복 호출**하여 네트워크 부하 증가

해결한 방법

- 1) **메모이제이션 적용** : React.memo와 useMemo 혹은 활용하여 차트 컴포넌트의 불필요한 리렌더링을 방지했습니다.
- 2) **데이터 요청 최적화** : 중복 API 호출을 방지하기 위해 React-Query를 도입하여 데이터 캐싱 및 상태 관리를 개선했습니다.
- 3) **서버 사이드 필터링** : 데이터 필터링 로직을 클라이언트에서 서버로 이동시켜 필요한 데이터만 전송받도록 개선했습니다.



PROJECT 3



삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 2위 수상작

기간 2025.04.14 ~ 2025.05.27 | 참여인원 6명

AI의 RAG를 기반으로 자격증 CBT 시험 대비를 위한 문제를 생성해주는 서비스로, React와 TypeScript를 활용하고, Storybook을 활용해 컴포넌트를 문서화하고 Atomic Design Pattern과 FSD 구조를 병용해 개발했습니다. 도메인 주소(http://q-generator.com)를 적용해 배포하여 실서비스화 하였습니다.

주요 기능

팀장 / 프론트엔드 개발 / UI·UX설계
시험지/문제집 리스트 페이지 구현
자료 업로드, 문제 생성 페이지 구현
사용자 및 문제 생성 모달 구현
RAG 기반 데이터 스플릿 및 임베딩 처리

담당 업무

다양한 유형의 사용자 자료 업로드 지원
사용자 업로드 자료와의 유사도 검색 기반
RAG 문제 생성 및 PDF 시험지 생성
문제 풀기 및 AI 기반 자세한 해설 제공 문제
풀기 이력 관리 및 복습

주요 기술 스택



React



TypeScript



Storybook



Zustand



TanStack Query



Vite



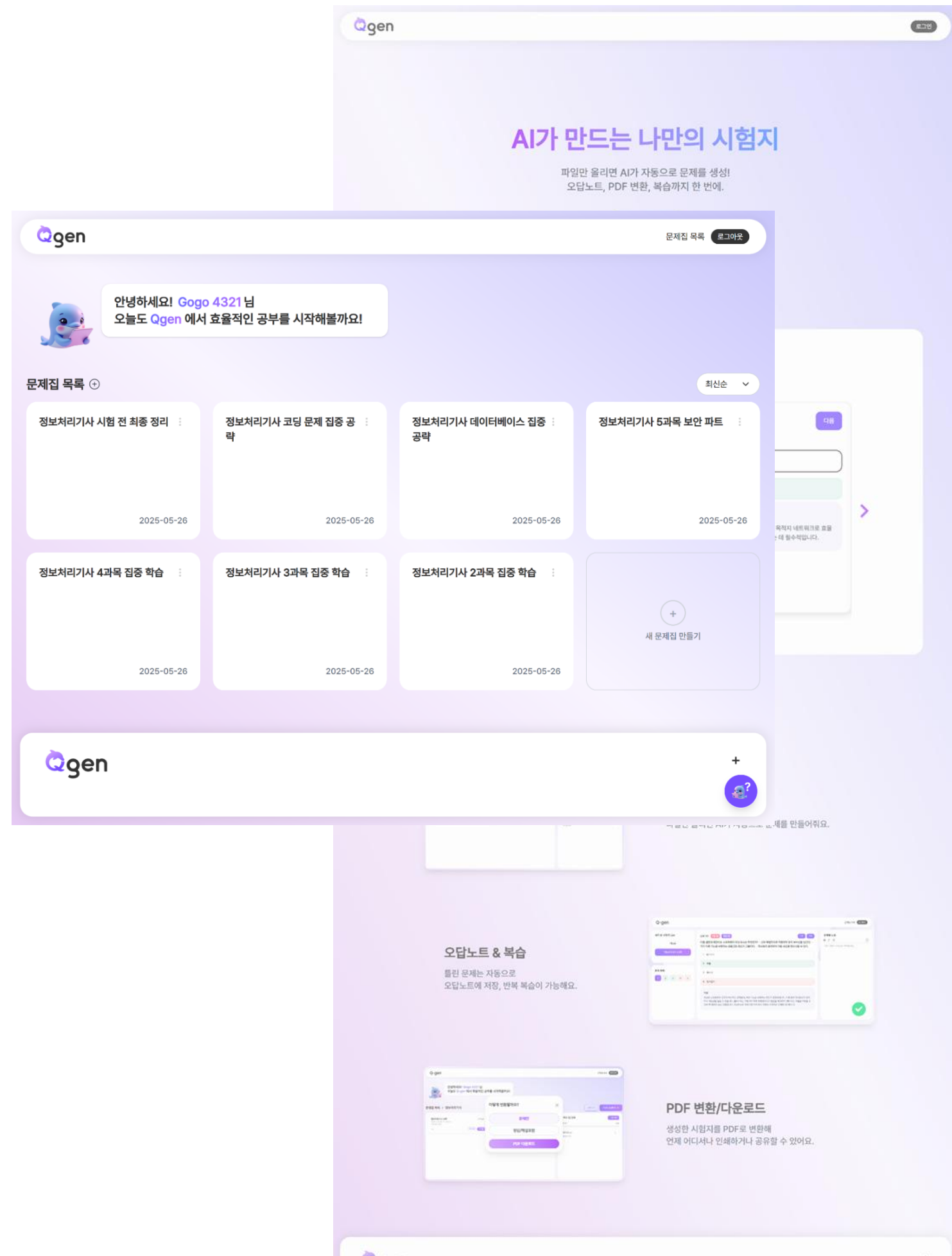
yarn



Tailwind CSS

트러블 슈팅 사례

가이드 안내 모달 간 display 상태 충돌 문제 해결



Qgen 트러블 슈팅 사례

발생한 문제

LocalStorage에 저장된 설정으로 자동 표시되는 가이드 모달과 사용자가 수동으로 여는 기본 가이드 모달이 동시에 표시되어 UI 겹침 현상 발생

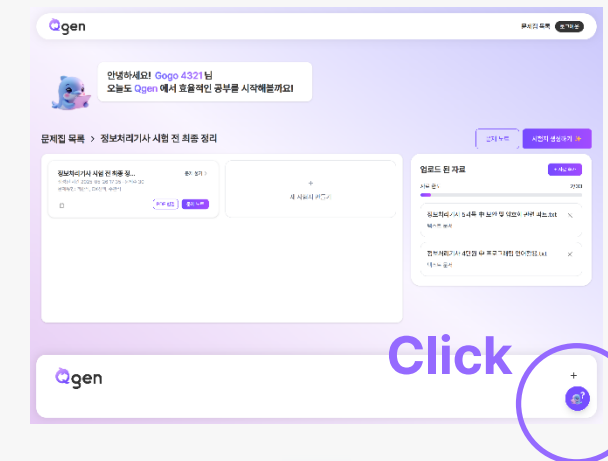
파악한 원인

- 1) 모달 상태 분기 처리 불명확 : 자동 표시 모달과 수동 표시 모달의 조건이 겹치는 상황 발생
- 2) LocalStorage 체크 로직 : 최초 방문자와 기존 사용자 구분 로직이 불명확
- 3) 모달 인스턴스 중복 : 동일한 GuideModal 컴포넌트가 서로 다른 목적으로 두 번 렌더링

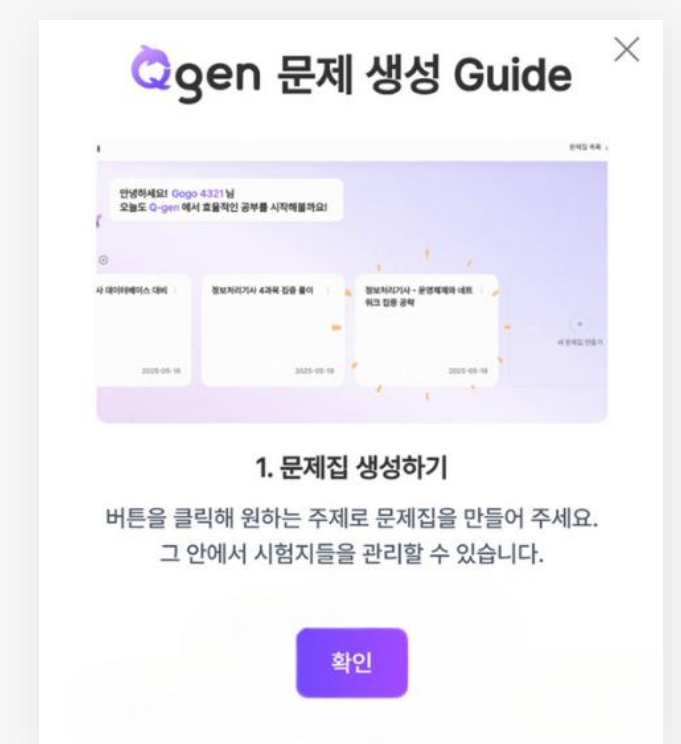
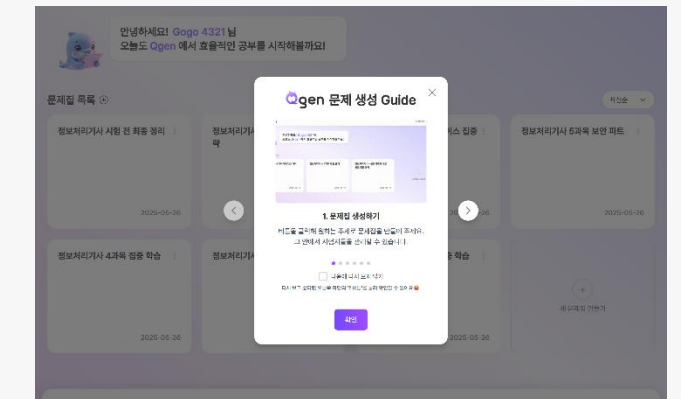
해결한 방법

- 1) 모달 상태 분기 처리 코드 분리 : 조건문을 분리하여 자동 표시와 수동 표시 모달이 독립적으로 동작하도록 개선
- 2) 상태값 전역 관리 : Zustand로 LocalStorage 에 저장되는 모달의 상태값을 전역적으로 관리하여 Props Drilling 개선과 코드 가독성 향상
- 3) 컴포넌트 분리 : Guide Modal 컴포넌트를 mode 속성으로 단일 컴포넌트에서 명확히 분기되어 렌더링 되도록 개선

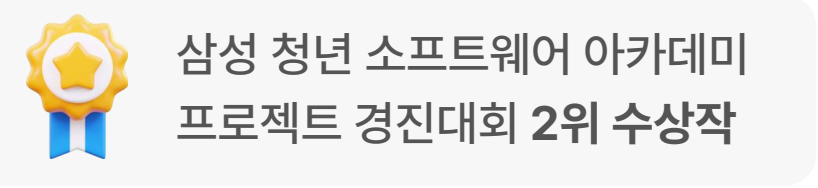
Guide 버튼 클릭 시 보여지는 Guide 모달



메인 페이지에서 띄워지는 Guide 모달



PROJECT 4



기간 2024.10.24 ~ 2024.11.25 | 참여인원 2명

운동 습관 형성을 돕는 홈트레이닝 지원 서비스입니다. YouTube API를 활용해 사용자가 선호하는 운동 영상과 음악을 등록하고 관리할 수 있으며, 운동 기록을 분석해 부족한 부위를 자동으로 추천해주는 기능을 구현했습니다. 백엔드는 **Java Spring Boot**, 프론트엔드는 **Vue.js**, **Tailwind**를 사용했습니다.



담당 업무

백엔드 & 프론트엔드 개발
API 설계 / DB 설계
다이어리 CRUD 구현
UI/UX 설계 및 구현

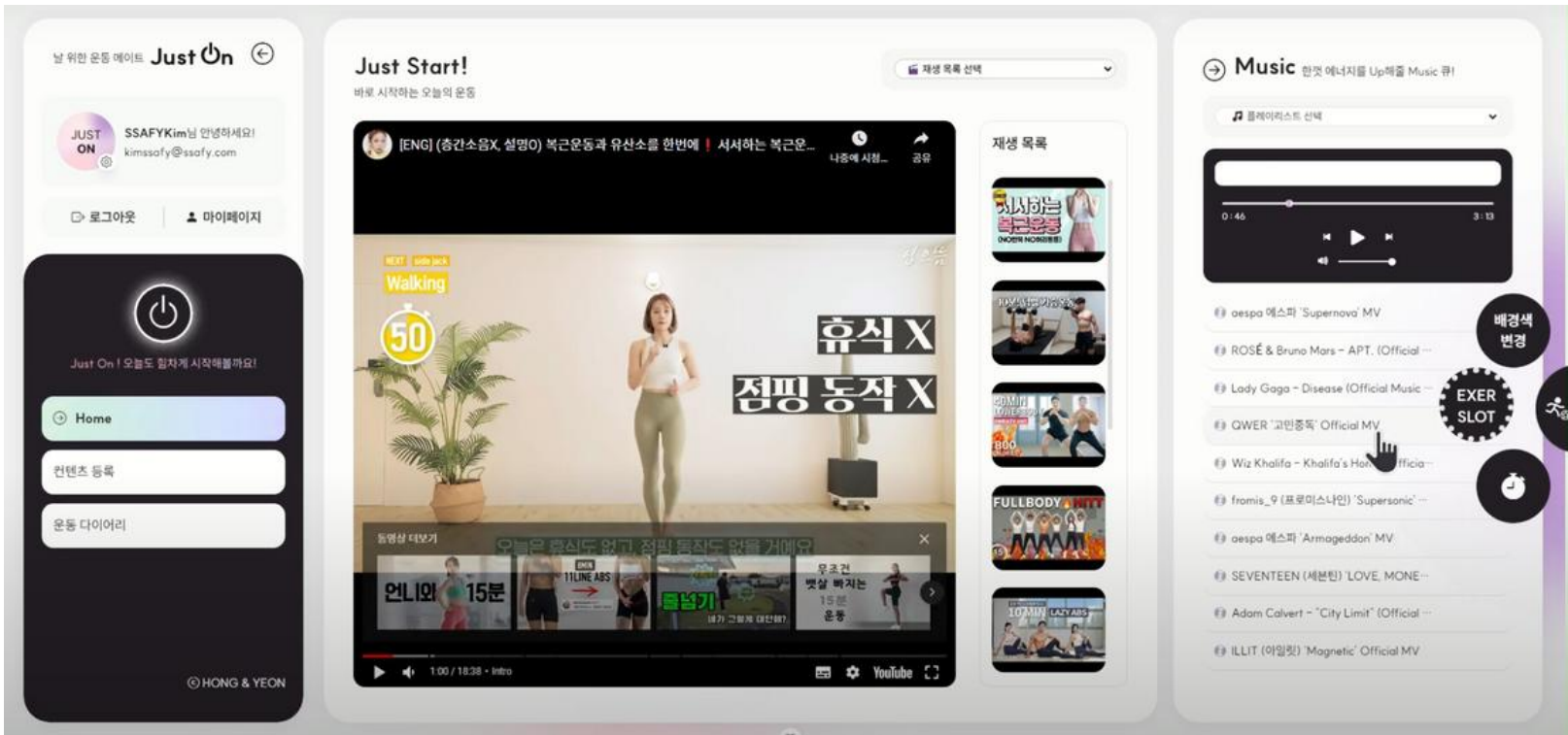
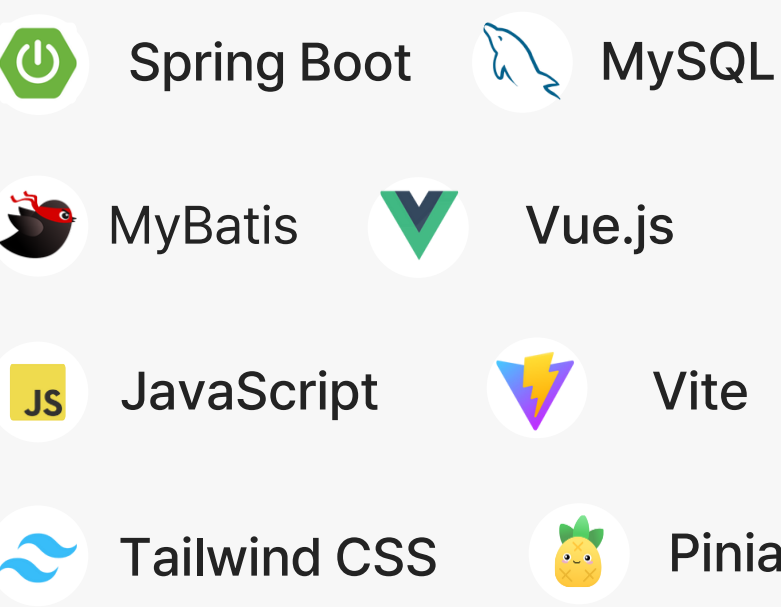
주요 기능

사용자 데이터 기반 운동 추천
실시간 운동 기록 관리
운동 영상 및 음악 관리
UI 커스텀 기능 (너비 조정 및 배경 변경)

트러블 슈팅 사례

운동 다이어리 보기 및 쓰기 페이지 간 상태 동기화 문제 및 운동 데이터가 유지 되지 않고 중복 입력되는 현상 해결

주요 기술 스택



Just On 트러블 슈팅 사례

발생한 문제

운동 다이어리 쓰기/보기 페이지 간 상태 동기화 문제 발생

페이지 전환 시 이전에 입력했던 운동 데이터가 유지되지 않거나 중복 저장되는 현상

원인 분석 방법

- 1) Vue Devtools를 활용한 컴포넌트 라이프사이클 및 상태 변화 추적
- 2) 네트워크 요청 모니터링을 통한 API 호출 시점 분석
- 3) 브라우저 저장소(localStorage) 데이터 검증

파악한 원인

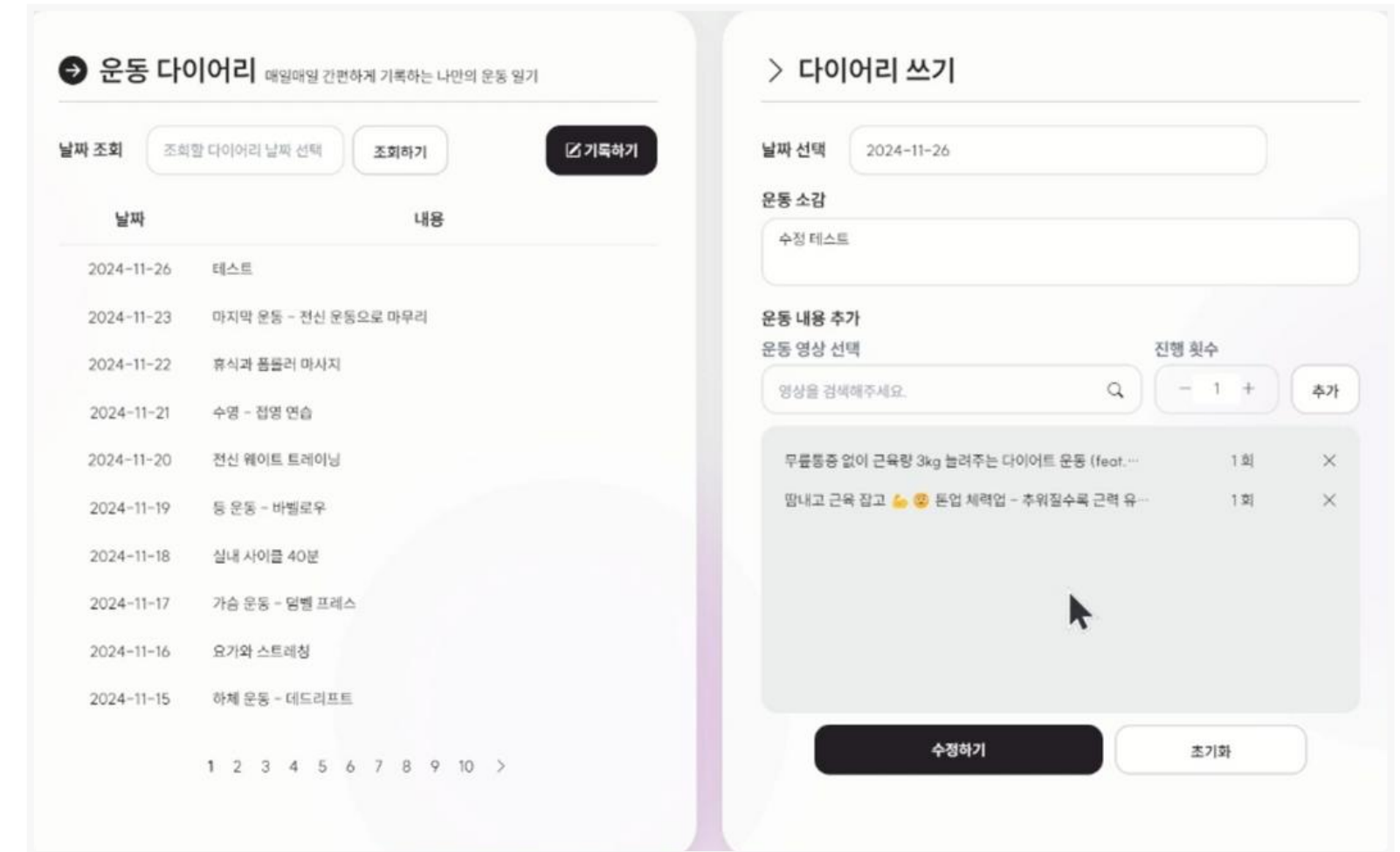
페이지 전환 시 **Pinia** 스토어의 상태 초기화로 인한 데이터 손실 발생

운동 다이어리 저장 시 비동기 처리가 완료되기 전에 페이지 전환이 이루어져 데이터 불일치 발생

라우팅 시 각 페이지에서 개별적으로 API를 호출하여 데이터 중복 또는 불일치 문제 발생

해결한 방법

- 1) **API 호출 로직 일원화** : API 호출 로직을 Pinia 액션으로 일원화하여 여러 컴포넌트에서 중복 호출되는 문제를 해결하고, 데이터 일관성을 유지했습니다.
- 2) **낙관적인 UI 업데이트** : 응답을 기다리지 않고 UI를 먼저 업데이트한 후, 서버 응답에 따라 상태를 조정하는 방식을 적용하여 사용자 경험을 개선했습니다.



UI·UX

편리하고 감각적인 서비스 개발을 위해 UI·UX 설계에 대해서도 다양한 경험을 쌓아왔습니다. 디자인 툴을 직접 활용하며 작업하고 퍼블리싱한 저의 UI·UX 작업물을 소개합니다.

1 개인 UI/UX 포트폴리오

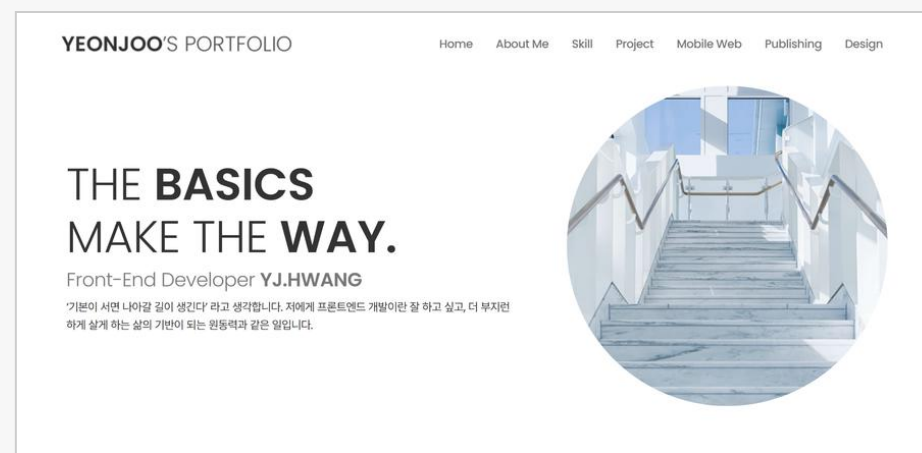


작업기간 2주

주요 특징 반응형 웹 · HTML 5.0 · CSS 3.0

· JavaScript · Swiper · Figma 활용

한줄 설명 개인 웹 포트폴리오 사이트이며,
Featherlight 라이브러리를 활용해 외부 사이트
이동을 최소화하였습니다.



바로가기

2 음성 품바 축제 사이트 리뉴얼



작업기간 5주

주요 특징 반응형 웹 · HTML 5.0 · CSS 3.0

· JavaScript · Gsap · PhotoShop · Illustrator

한줄 설명 기존 음성 품바 축제 사이트를 리뉴얼한
작업물 입니다. Gsap 라이브러리를 활용하여
페이지에 동적인 느낌을 부여하였습니다.



바로가기

3 에피젠 사이트 리뉴얼

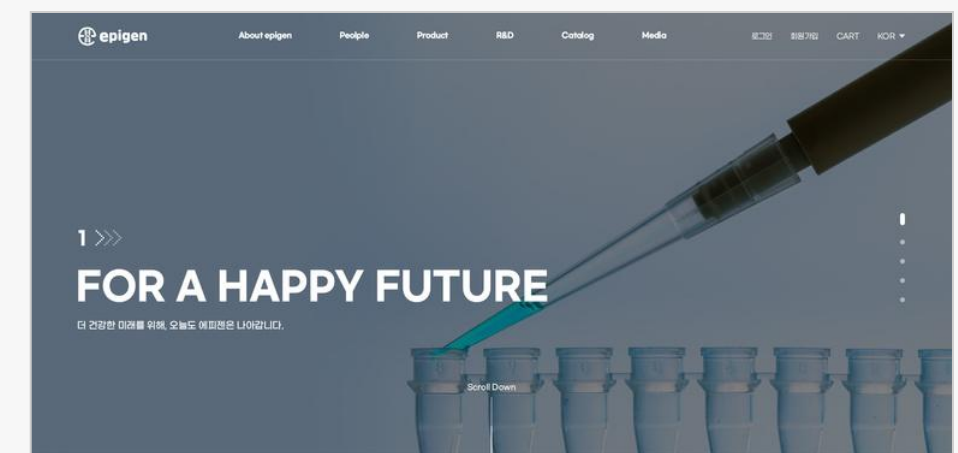


작업기간 4주

주요 특징 반응형 웹 · HTML 5.0 · CSS 3.0

· JavaScript · PhotoShop · Illustrator

한줄 설명 기존 에피젠 사이트를 리뉴얼한
작업물로 회사의 제품을 편리하게 확인할 수
있도록 설계하였습니다.



바로가기

CONTACT 

PHONE.

010-7193-3827

EMAIL.

hyjgood2000@gmail.com

