INTRO

{ TRY } && { GROW }

안녕하세요 !

오늘의 **시도**로 **내일** 더 **성장**하는 **개발자**

황 연 주 입니다.

E-mail. hyjgood2000@gmail.com / **Phone.** 010-7193-3827

이수 교육.

2024.07 ~ 2025.06 삼성 청년 소프트웨어 아카데미(SSAFY) 수료 (비전공 JAVA반)

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

Links.

GitHub | 기술 블로그 | HELLO. MY NAME IS YEONJOO. 개인 사이트

INDEX

제가 쌓아온 기술과 **프로젝트** 경험을 소개합니다.

1

2

3

4

SKILL MAP

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

기술 수준
하 상

한눈에 보는 스킬맵

프레임워크 및 라이브러리

Next.js 중

- **CSR, SSR, SSG의 차이점**을 이해하고 적절히 활용할 수 있습니다.
- **API 라우트를 활용한 서버리스 함수 구현**이 가능합니다.

React 상

- **useState, useEffect등 다양한 Hooks를 사용**해 상태를 관리할 수 있습니다.
- **상태 관리 라이브러리(Zustand 등)**로 **복잡한 상태를 효율적으로 관리**할 수 있습니다.

Vue.js 중

- **Composition API**를 활용해 **재사용 가능한 로직을 구현**할 수 있습니다.
- Pinia를 활용한 **상태 관리**를 할 수 있습니다.

프로그래밍 언어

JavaScript 상

- **프로토타입 기반 객체지향 프로그래밍**을 이해하고 있습니다.
- **비동기 프로그래밍**(Promise, async/await)과 **클로저**와 **스코프**에 대해 이해하고 있습니다.

TypeScript 중상

- 유니온 타입, 인터섹션 타입 등의 **타입 연산자**와 **유틸리티, 고급 타입**에 대해 이해하고 있습니다.
- **함수형 프로그래밍**과 ES6+ 문법을 알고 있습니다.

HTML 상

- **시맨틱 태그를 활용**해 웹 접근성을 고려한 **마크업**을 작성합니다.
- **Keyframes**를 사용하여 슬라이드, 글리치 등 여러 **애니메이션 작업**을 할 수 있습니다.

CSS 상

- **Flexbox**와 **Grid**를 활용한 **레이아웃 구성**을 할 수 있습니다.
- SCSS 등 **CSS 전처리기 및 CSS-in-JS** 기법을 활용하여 동적인 스타일링을 구현할 수 있습니다.

협업툴

Git & GitHub 중

- Git을 사용해 **프로젝트**를 진행하며,팀원들과 **code-review**를 통해 **코드를 개선한 경험**이 있습니다.

JIRA 중

JIRA 기반 **스크럼** 방식으로 스프린트 계획 등 **체계적인 프로젝트 관리 경험**이 있습니다.

개발 및 UI/UX 설계 도구

Figma | Postman | VS Code | IntelliJ

PROJECTS

직접 기획 및 설계한 **UI/UX**와 **학습한 기술**을 **적용**하여 구현한 저의 **프로젝트**를 소개합니다.
UI/UX 설계는 **Figma**를 사용하였습니다.

 프로젝트 경진대회 **수상작**

PROJECT 1. VOPLE

한 줄 소개 : **중 · 고등학생**을 위한 **선거 유세 및 관리 플랫폼**
담당 업무 **: 프론트엔드**
기술 스택 및 설계
React + TypeScript / SPA / Atomic 패턴 / SCSS 모듈 분리

PROJECT 2. YOOHOO

한 줄 소개 : **투명하고, 신뢰가는 유기견 후원을 위한 기부 플랫폼**
담당 업무 **: 프론트엔드**
기술 스택 및 설계
Next + TypeScript / CSR·SSR / Atomic · FSD 병용 / SCSS 모듈 분리

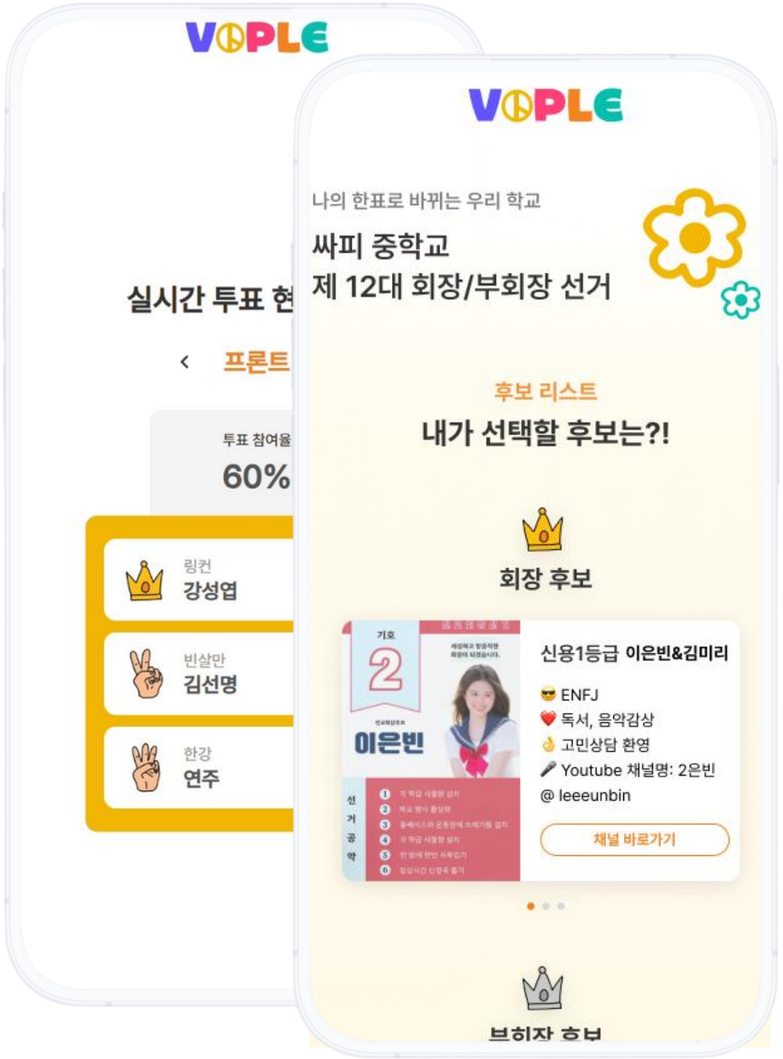
PROJECT 3. Qgen

한 줄 소개 : **AI RAG 기반 맞춤형 CBT 문제 생성 및 해설 서비스**
담당 업무 **: 프론트엔드**
기술 스택 및 설계
React + TypeScript / SPA / Atomic · FSD 병용 / SCSS 모듈 분리

PROJECT 4. JustOn

한 줄 소개 : **홈 트레이너를 위한 맞춤형 운동 추천 및 운동 습관 형성 서비스**
담당 업무 **: 백엔드 · 프론트엔드**
기술 스택 및 설계
Spring Boot / REST API / MVC / Vue.js / SPA

PROJECT 1

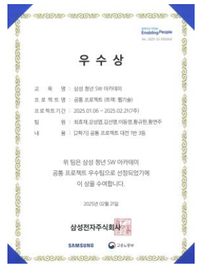


중 · 고등학생을 위한 **선거 관리 플랫폼**

VOPLE

기간 2025.01.05 ~ 2025.02.21 | 참여인원 6명

삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 **3위 수상작**



GitHub 바로가기

중고등학생을 위한 선거 관리 서비스로, **React**와 **TypeScript**를 활용하고, **Storybook**을 활용해 컴포넌트를 문서화하며 **Atomic Design Pattern**을 적용해 개발했습니다. **실시간 투표 현황**을 직관적으로 보여주는 **시각화 컴포넌트**와 **WebSocket**을 활용해 **라이브 선거 화면**을 **구현**했습니다.

담당 업무

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

주요 기능

실시간 채팅 기능 (선거 유세, 투표 결과)
실시간 선거 유세 라이브 스트리밍 구현
실시간 투표 결과 시각화

설계 방식

BackEnd REST API + WebSocket
FrontEnd SPA + Atomic 패턴 + SCSS 모듈 분리

주요 기술 스택

React · TypeScript · yarn · Storybook · Zustand · MSW · TanStack Query · Sass · Vite

트러블 슈팅 및 서비스 개선 사례

1) 라이브 선거 화면의 후보자와 유권자에 따른 화면 렌더링 분기 처리 문제 해결
2) WebSocket 중복 연결 오류 해결
3) 라이브 선거 화면의 실시간 채팅창의 탭 이동 시 초기화되는 현상 해결

{ 1 트러블 슈팅 및 서비스 개선 사례

라이브 선거 화면의 후보자와 유권자에 따른 화면 렌더링 분기 처리 문제 해결

발생한 문제

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

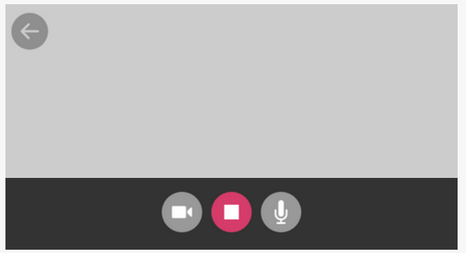
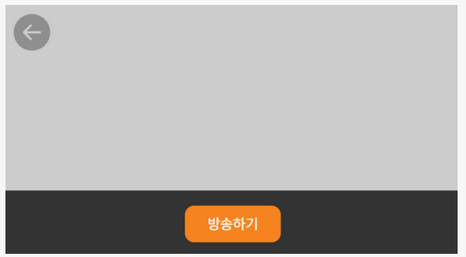
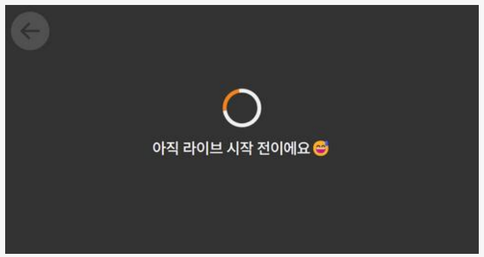
원인 분석 방법

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

파악한 원인

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

스트리밍 상태 및 사용자 권한별 화면

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

해결한 방법

1) 분산된 상태 통합 : 개별적이었던 isLoading, error, isCandidate와 같은 state 변수들을 **streamingStatus라는 단일 객체로 통합하여 상태 간 충돌을 방지**했습니다.

2) 컴포넌트 분리 : 조건부 렌더링이 더 독립적으로 이루어지도록 **StreamSender, StreamReceiver와 같은 별도 컴포넌트로 분리**했습니다.

3) 상태 전환 로직 단순화 : 명확한 분기 로직으로 **UI 전환을 개선**했습니다.

{ 2 트러블 슈팅 및 서비스 개선 사례

WebSocket 중복 연결 오류 해결

발생한 문제

1) 실시간 채팅 기능에서 **WebSocket 연결이 중복으로 생성**되어 메모리 누수 발생
2) **컴포넌트 언마운트 시에도 연결이 제대로 정리되지 않아** 백그라운드에서 계속 실행
3) **채팅 메시지가 중복으로 표시**되거나 누적되는 현상 발생

원인 분석 방법

1) 브라우저 개발자 도구의 Network 탭에서 **WebSocket 연결 상태 확인**
2) React DevTools에서 **컴포넌트 마운트/언마운트 상태 추적**
3) 메모리 사용량 모니터링으로 **누수 패턴 파악**

파악한 원인

1) useWebSocket 훅에서 useEffect의 의존성 배열에 sessionId, roomId, type이 포함되어 있어 **값이 변경될 때마다 새로운 연결이 생성** 되고, 컴포넌트가 리렌더링될 때마다 기존 구독이 제대로 해제되지 않음
2**) isMounted** ref를 사용했지만 **cleanup 함수에서 완전히 정리되지 않음**

해결한 방법

1) 중복 연결 방지 로직 추가 : 연결 전, 연결 상태를 확인해서 연결 시 return 되도록 해 중복 연결을 방지. 만약, 새로운 연결을 생성할 때는 기존에 존재하는 메세지 구독을 해제하도록 함. (구독 참조를 null 로 설정)

2) Cleanup 함수 로직 개선 : 컴포넌트 언마운트 시 실행되는 cleanup 함수에서 isMounted를 false로 설정하여 비동기 작업 중단 및 메시지 구독을 해제하고 참조를 null 로 설정되도록 함.

중복 입력되었던 채팅 메세지 화면

{ 3 트러블 슈팅 및 서비스 개선 사례

라이브 선거 화면의 실시간 채팅창의 탭 이동 시 초기화되는 현상 해결

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 입력**됨

원인 분석 방법

1) 브라우저 **개발자 도구를 활용**한 단계별 디버깅
2) WebSocket **이벤트 로깅 시스템 구축**
3) 컴포넌트 **마운트/언마운트 추적**

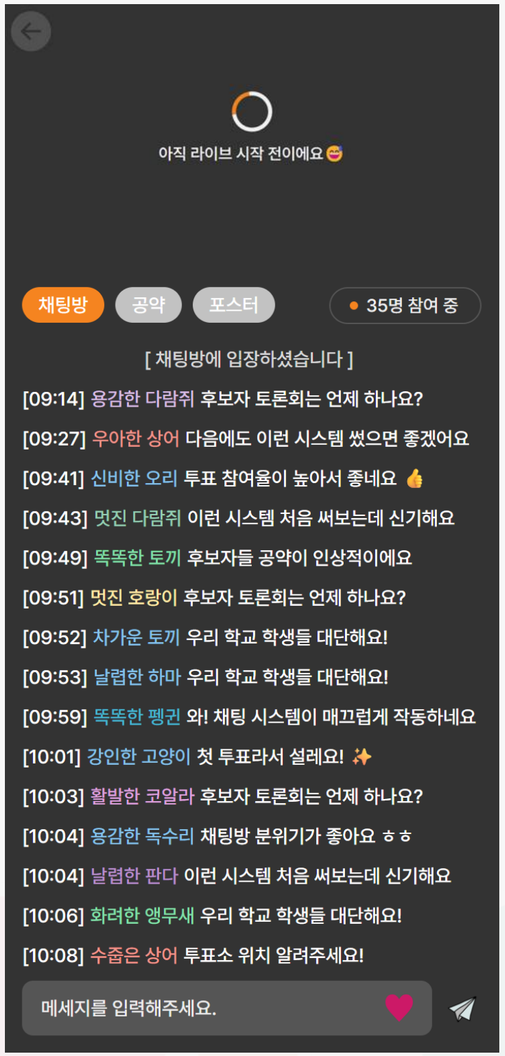
파악한 원인

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

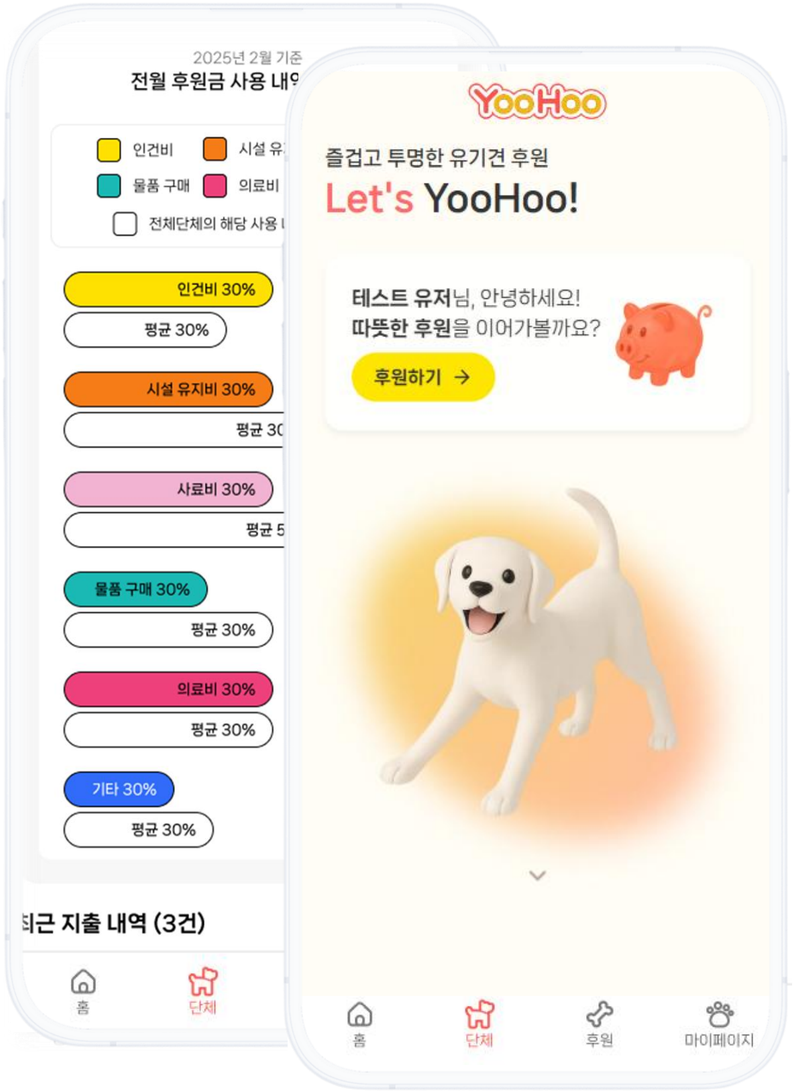
해결한 방법

1) ChatBoard에서 WebSocket 연결 분리 : WebSocket 연결을 UI 컴포넌트와 분리하여 ChatBoard에서 호출하여 사용
2) 컴포넌트 조건부 표시 방식 개선 : 컴포넌트를 언마운트하지 않고 CSS로 숨김/표시 처리
3) 메시지 배열 추가 방식 개선 : 배열 추가 방식을 메시지 렌더링 데이터와 통일하고 항상 배열 끝에 추가되도록 변경.

실시간 선거 유세 화면



PROJECT 2



즐겁고 투명한 유기견 후원

YooHoo

기간 2025.02.24 ~ 2025.04.11 | 참여인원 6명

삼성 청년 소프트웨어 아카데미 프로젝트 경진대회 **2위 수상작**

GitHub 바로가기

유기견 후원을 위한 플랫폼으로, 후원금 내역을 투명하게 확인하고 쉽게 후원할 수 있는 서비스입니다. **Next**와 **TypeScript**를 사용했으며, **Chart.js**로 **후원금 사용 내역 대시보드**를 개발했습니다.

담당 업무

프론트엔드 개발 팀장

UI/UX 디자인 및 설계

단체의 후원금 사용 내역 시각화

후원금 사용 내역 매출전표 API 연결 및 시각화

주요 기능

투명한 후원금 사용 내역 차트 제공

강아지 후원 및 응원 메시지 전달

후원금 사용 내역 증빙 자료 및 매출 전표 조회

설계 방식

BackEnd REST API + WebSocket

FrontEnd CSR/SSR + FSD 및 Atomic 패턴 + SCSS 모듈 분리

주요 기술 스택

Next · TypeScript · Sass · Storybook · Zustand · MSW · TanStack Query · Vite · yarn

트러블 슈팅 및 서비스 개선 사례

1) 후원금 사용 내역 Chart 조회 시 대시보드 렌더링 지연 문제 해결
2) 로그인 인증 권한에 따른 페이지 라우팅 오류 문제 해결
3) 테스트 유저 대상 자체 1,2차 QA 진행을 통한 서비스 개선

{ 1 트러블 슈팅 및 서비스 개선 사례

후원금 사용 내역 Chart 조회 시 대시보드 렌더링 지연 문제 해결

발생한 문제

후원금 사용 내역 **대시보드 렌더링 시 성능 저하** 문제 발생
대량의 후원 데이터가 있을 경우 **페이지 로딩 시간이 3초 이상 소요**됨

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**

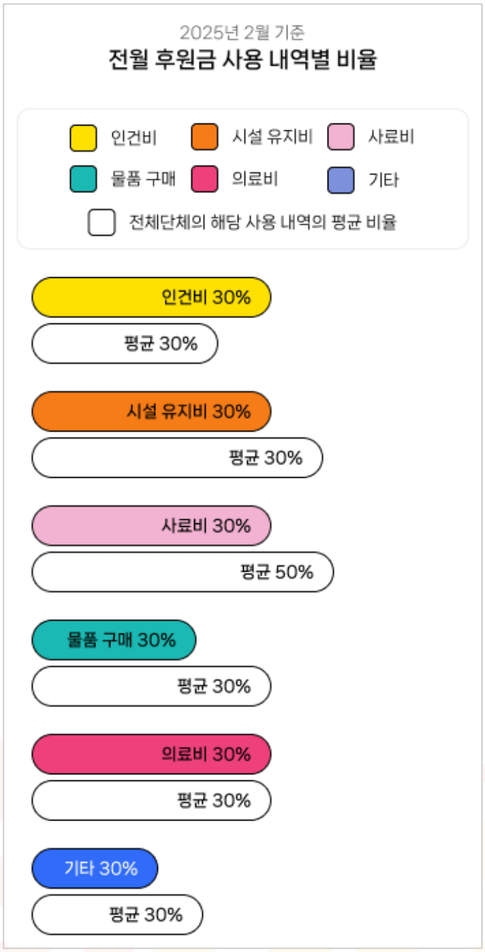
파악한 원인

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 호출하여 네트워크 부하 증가**

해결한 방법

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

긴 로딩 시간의 원인이 된 주요 차트 컴포넌트들



{ 2 트러블 슈팅 및 서비스 개선 사례

로그인 인증 권한에 따른 페이지 라우팅 오류 문제 해결

발생한 문제

유저 정보에 관한 **Zustand 상태 업데이트가 완료되기 전에 Data가 사용**되면서,
이전 상태의 데이터로 라우팅이 결정되고, **관리자/일반 사용자 권한**에 따라 **라우팅 분기 처리가 되지 않음**

원인 분석 방법

1) 브라우저 개발자 도구로 유저 정보 업데이트 상태 확인 : console.log로 인증 결과와 상태 변화 추적
2) 관리자/일반 사용자 분기 처리 검증 : isAdmin 값이 올바르게 반영되는지, 이전 상태가 사용되는지 확인
3) Zustand 상태 변화 시점 체크 : checkAuthStatus() 호출 후 상태가 실제로 언제 업데이트되는지 확인

파악한 원인

- **Zustand의 상태 업데이트가 비동기로 동작**하기 때문에 checkAuthStatus()가 반환된 직후에도 최신 상태가 반영되지 않을 가능성이 있다 판단함
- 그 결과, **이전 상태의 데이터(예: 이전 로그인 정보, isAdmin 값 등)로 라우팅이 결정**되어 관리자인데도 일반 사용자 페이지로 이동하거나, 반대로 잘못된 분기 처리가 발생함.
- 특히, checkAuthStatus()내부에서 상태를 갱신하지만, 외부에서 바로 사용하면 최신 값이 아닐 수 있음

해결한 방법

1) 상태 업데이트 완료 대기 로직 추가 : await new Promise((resolve) => setTimeout(resolve, 0)); 와 같이,
마이크로태스크 큐를 활용해 한 번의 이벤트 루프를 넘기고 그 후에 최신 상태를 사용하도록 개선함. 이로써,
Zustand의 상태가 완전히 반영된 후에 관리자/일반 사용자 권한에 따라 **올바른 페이지 라우팅이 진행**되도록 함

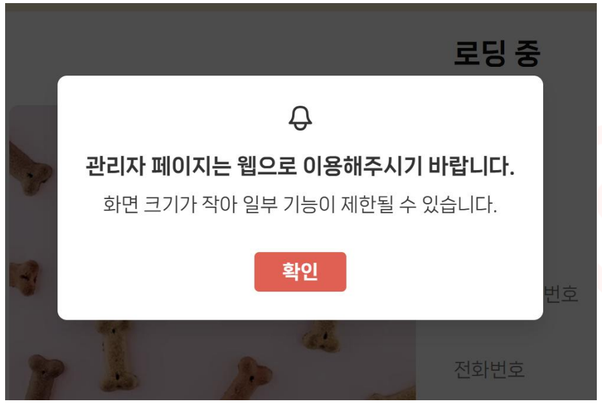
로그인 시, 권한별 화면 분기 처리

{ 3 트러블 슈팅 및 서비스 개선 사례
테스트 유저 대상 자체 1,2차 QA 진행을 통한 서비스 개선

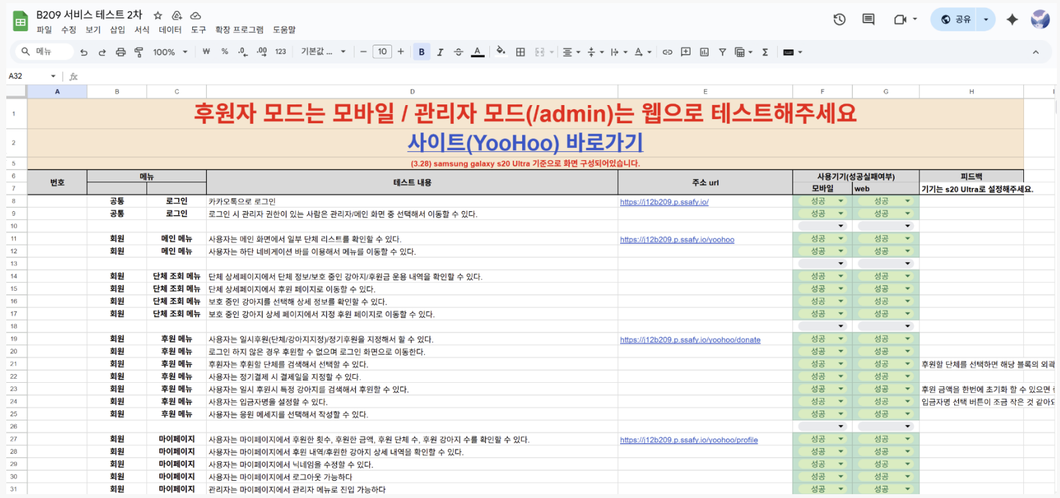
60명의 테스트 유저를 대상으로 자체 **1,2차 QA 진행**을 통해 피드백 받은 결과를 토대로 전반적인 **UI/UX를 개선**하였습니다.

주요 개선 사항

1) 기존에 따라 발생하는 **네비게이션 바 하단 1px 공백 개선**
2) 하단 **네비게이션 메뉴 클릭 범위 확대**
3) 후원 페이지에서 **후원 금액 입력란**에 **초기화 버튼 추가**
4) **입금자명 입력 칸 크기 증가**를 통한 입력 칸 선택 UX 개선
5) **아이폰 13 mini** 에서 단체 검색페이지의 **검색 버튼 치우침 현상 해결**
6) 모바일 환경에서 **관리자 페이지** 접근 시, **pc환경 이동 권고 안내 모달** 추가
7) 후원 완료 시, **빵빠레 효과 추가** (라이브러리' 로티' 활용)



QA 진행 시, 작성했던 문서들



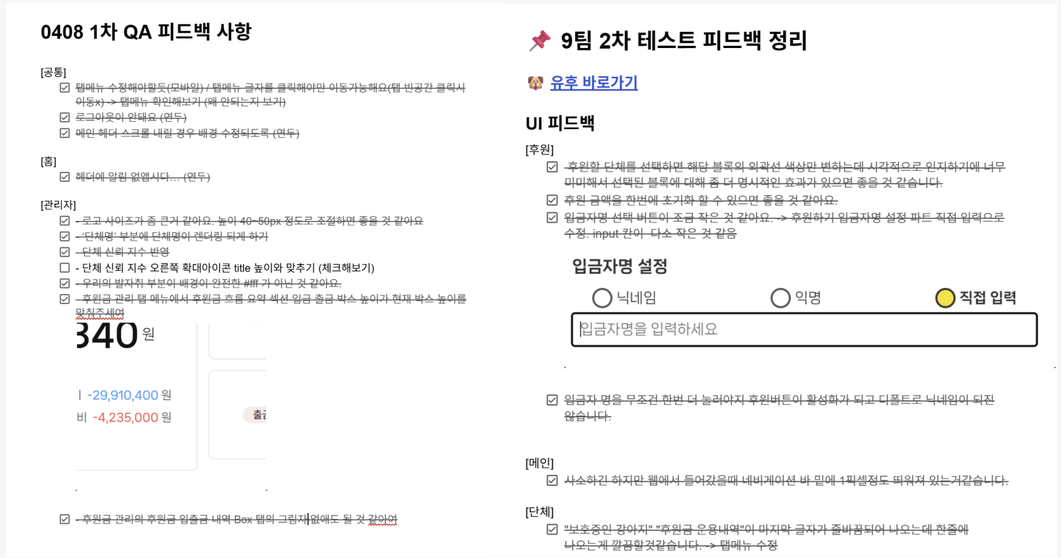
B209 서비스 테스트 2차

후원자 모드는 모바일 / 관리자 모드(/admin)는 웹으로 테스트해주세요

사이트(YooHoo) 바로가기

(3.28) samsung galaxy s20 Ultra 기준으로 화면 구성되어있습니다.

번호	메뉴		테스트 내용	주소 url	사용기기(성공실패여부)		피드백
					모바일	web	기기는 s20 Ultra로 설정해주세요.
	공통	로그인	카카오톡으로 로그인	https://i12b209.p.ssafy.io/	성공	성공	
	공통	로그인	로그인 시 관리자 권한이 있는 사람은 관리자/메인 화면 중 선택해서 이동할 수 있다.		성공	성공	
	회원	메인 메뉴	사용자는 메인 화면에서 일부 단체 리스트를 확인할 수 있다.	https://i12b209.p.ssafy.io/yoohoo	성공	성공	
	회원	메인 메뉴	사용자는 하단 네비게이션 바를 이용해서 메뉴를 이동할 수 있다.		성공	성공	
	회원	단체 조회 메뉴	단체 상세페이지에서 단체 정보/보호 중인 강아지/후원금 운용 내역을 확인할 수 있다.		성공	성공	
	회원	단체 조회 메뉴	단체 상세페이지에서 후원 페이지로 이동할 수 있다.		성공	성공	
	회원	단체 조회 메뉴	보호 중인 강아지를 선택해 상세 정보를 확인할 수 있다.		성공	성공	
	회원	단체 조회 메뉴	보호 중인 강아지 상세 페이지에서 지정 후원 페이지로 이동할 수 있다.		성공	성공	
	회원	후원 메뉴	사용자는 일시후원(단체/강아지지정)/정기후원을 지정해서 할 수 있다.	https://i12b209.p.ssafy.io/yoohoo/donate	성공	성공	
	회원	후원 메뉴	로그인 하지 않은 경우 후원할 수 없으며 로그인 화면으로 이동한다.		성공	성공	
	회원	후원 메뉴	후원자는 후원할 단체를 검색해서 선택할 수 있다.		성공	성공	후원할 단체를 선택하면 해당 블록의 외곽
	회원	후원 메뉴	사용자는 정기결제 시 결제일을 지정할 수 있다.		성공	성공	
	회원	후원 메뉴	사용자는 일시 후원시 특정 강아지를 검색해서 후원할 수 있다.		성공	성공	후원 금액을 한번에 초기화 할 수 있으면
	회원	후원 메뉴	사용자는 입금자명을 설정할 수 있다.		성공	성공	입금자명 선택 버튼이 조금 작은 것 같아
	회원	후원 메뉴	사용자는 응원 메세지를 선택해서 작성할 수 있다.		성공	성공	
	회원	마이페이지	사용자는 마이페이지에서 후원한 횟수, 후원한 금액, 후원 단체 수, 후원 강아지 수를 확인할 수 있다.	https://i12b209.p.ssafy.io/yoohoo/profile	성공	성공	
	회원	마이페이지	사용자는 마이페이지에서 후원 내역/후원한 강아지 상세 내역을 확인할 수 있다.		성공	성공	
	회원	마이페이지	사용자는 마이페이지에서 닉네임을 수정할 수 있다.		성공	성공	
	회원	마이페이지	사용자는 마이페이지에서 로그아웃 가능하다		성공	성공	
	회원	마이페이지	관리자는 마이페이지에서 관리자 메뉴로 진입 가능하다		성공	성공	



0408 1차 QA 피드백 사항

[공통]
- ~~탭메뉴 수정해야할듯(모바일) / 탭메뉴 글자를 클릭해야만 이동가능해요(탭 빈공간 클릭시 이동x) -> 탭메뉴 확인해보기 (왜 안되는지 보기)~~
- ~~로그아웃이 안돼요 (연두)~~
- ~~메인 헤더 스크롤 내릴 경우 배경 수정되도록 (연두)~~

[홈]
- ~~헤더에 알림 없앱시다... (연두)~~

[관리자]
- ~~로고 사이즈가 좀 큰거 같아요. 높이 40~50px 정도로 조절하면 좋을 것 같아요~~
- ~~단체명 부분에 단체명이 랜더링 되게 하기~~
- ~~단체 신뢰 지수 반영~~
- 단체 신뢰 지수 오른쪽 확대아이콘 title 높이와 맞추기 (체크해보기)
- ~~우리의 발자취 부분이 배경이 완전한 #fff 가 아닌 것 같아요.~~
- ~~후원금 관리 탭 메뉴에서 후원금 흐름 요약 섹션 입금 출금 박스 높이가 현재 박스 높이를 맞춰주세요~~

340 원

| -29,910,400 원

비 -4,235,000 원

- ~~후원금 관리의 후원금 입출금 내역 Box 탭의 그림자없애도 될 것 같아요~~

9팀 2차 테스트 피드백 정리

유후 바로가기

UI 피드백

[후원]
- ~~후원할 단체를 선택하면 해당 블록의 외곽선 색상만 변하는데 시각적으로 인지하기에 너무 미미해서 선택된 블록에 대해 좀 더 명시적인 효과가 있으면 좋을 것 같습니다.~~
- ~~후원 금액을 한번에 초기화 할 수 있으면 좋을 것 같아요.~~
- ~~입금자명 선택 버튼이 조금 작은 것 같아요. -> 후원하기 입금자명 설정 파트 직접 입력으로 수정 input 칸이 더 작은 것 같음~~

입금자명 설정

닉네임 익명 직접 입력

입금자명을 입력하세요

- ~~입금자 명을 무조건 한번 더 눌러야지 후원버튼이 활성화가 되고 디폴트로 닉네임이 되진 않습니다.~~

[메인]
- ~~사소하긴 하지만 웹에서 들어갔을때 네비게이션 바 밑에 1픽셀정도 띄워져 있는거같습니다.~~

[단체]
- ~~"보호중인 강아지" "후원금 운용내역"이 마지막 글자가 줄바꿈되어 나오는데 한줄에 나오는게 깔끔할것같습니다. -> 탭메뉴 수정~~

PROJECT 2

AI가 만드는 나만의 시험지
Qgen

기간 2025.04.14 ~ 2025.05.27 | 참여인원 6명

삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 **2위 수상작**

GitHub 바로가기

AI의 RAG를 기반으로 자격증 CBT 시험 대비를 위한 문제를 생성하는 서비스로 , **Atomic Design Pattern**과 **FSD 구조**를 병용해 개발했습니다. 도메인 주소(http://q-generator.com)를 적용해 **배포하여 실서비스화** 하였습니다.

담당 업무

팀장 및 프론트엔드 개발 팀장 / 데이터 처리

시험지/문제집 리스트 페이지 구현
자료 업로드, 문제 생성 페이지 구현
사용자 및 문제 생성 모달 구현
RAG 기반 데이터 청킹

주요 기능

문제집별 시험지 생성 및 관리 기능
다양한 유형의 사용자 자료 업로드 지원
사용자 업로드 자료와 유사한 문제 생성
PDF 시험지 생성 및 출력 기능
문제별 AI 해설 제공 및 풀이 이력 관리 및 복습

주요 기술 스택

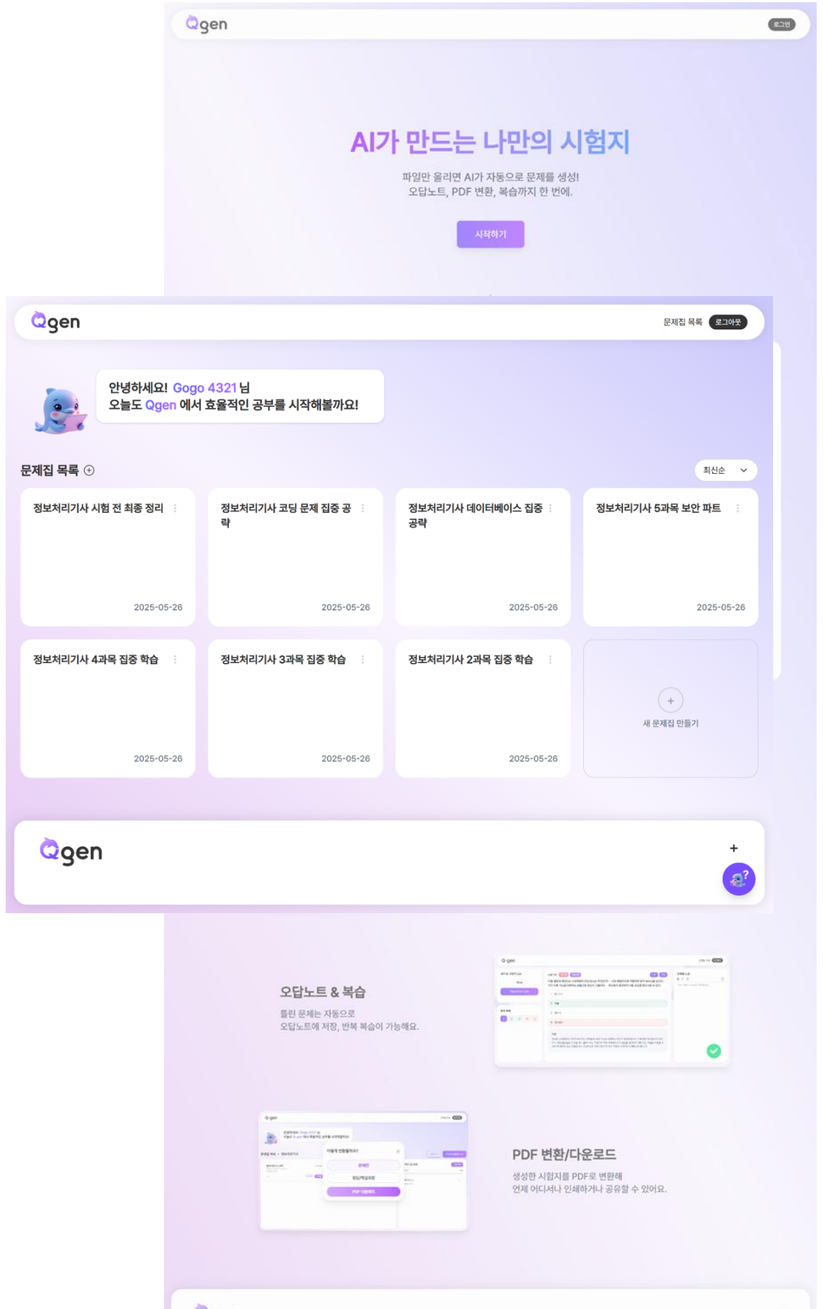
React, TypeScript, yarn
Zustand, Vite, Storybook
TanStack Query, Tailwind CSS

설계 방식

BackEnd REST API
FrontEnd SPA + FSD 및 Atomic 패턴 + SCSS 모듈 분리
AI REST API(FastAPI) + FAISS + 비동기 처리

트러블 슈팅 및 서비스 개선 사례

1) 텍스트 청킹 방법 개선
2) 가이드 안내 모달 간 display 상태 충돌 문제 해결
3) 사용자 피드백을 통한 서비스 및 UI/UX 개선



{ 1 트러블 슈팅 및 서비스 개선 사례

텍스트 청킹 방법 개선

발생한 문제

문제 생성을 위해 입력해 둔 **기반 문제 데이터를 문제별로 구분하지 못하고, 문제 번호(원숫자) 특수문자를 제대로 인식하지 못함.**

해결한 방법

1) 의미 단위 기반 청킹 구현 : separators=["\n\n", "\n", " "]: 의미 단위 구분자를 추가하고, '---' 구분자를 도입하여 문서 내 논리적 구분점으로 활용함.

2) 특수 문자 정규화 시스템 구축 : 원문자 매핑(❶→1, Ⓐ→A)을 통해 특수문자를 체계적으로 변환하고, 정규표현식을 통해 불필요한 특수 문자를 제거하였으며, 공백 정규화를 통해 연속된 공백이나 탭 문자를 정리하였습니다.

개선 효과

1) 검색 정확도 향상 : 개선 전 65%의 정확도를 보였던 반면에 개선 후 80%로 15% 향상된 정확도를 보여줌.

2) 임베딩 품질 개선 : 정규화된 텍스트로 정확한 벡터가 생성되면서, 특수 문자로 인한 문제 생성 에러 발생률이 10%에서 1%대로 감소함

개선 전 코드

```
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=512,
    chunk_overlap=50,
    separators=["\n", " "]
)

def build_index(indexId: str = "questions"):
    with open(text_path, "r", encoding="utf-8") as f:
        raw_text = f.read()

    # 단순 분할로 인한 문맥 손실
    chunks = text_splitter.split_text(raw_text)

    # 특수 문자 처리 없음
    vectors = model.encode(chunks, batch_size=8)
```

개선 후 코드

```
# 의미 단위 기반 청킹
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=256,  # 더 작은 청크로 정밀도 향상
    chunk_overlap=25,  # 적절한 오버랩
    separators=["\n\n", "\n", " "]  # 의미 단위 구분자 추가
)

def clean_text(text: str) -> str:
    # 특수 문자 정규화 (❶-❿, Ⓐ-Ⓩ, ⓐ-ⓩ)
    circled_1_to_10 = {
        '❶': '1', '❷': '2', '❸': '3', '❹': '4', '❺': '5',
        '❻': '6', '❼': '7', '❽': '8', '❾': '9', '❿': '10'
    }
    # ... 특수 문자 매핑 로직

    return re.sub(r"[^\w\s\n\-~!@#$%^&*()_+`=\[\]{};:'\",.<>/?\\|]", "", text)

def build_index(indexId: str = "questions"):
    # ✅ '---' 기준으로 의미 단위 청크 분리
    chunks = [chunk.strip() for chunk in cleaned.split('---') if chunk.strip()]

    vectors = model.encode(chunks, batch_size=16, show_progress_bar=True)
    # ... 나머지 코드
```

{ 2 트러블 슈팅 및 서비스 개선 사례

가이드 안내 모달 간 display 상태 충돌 문제 해결

발생한 문제

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

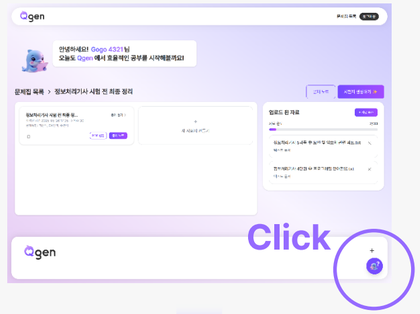
파악한 원인

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

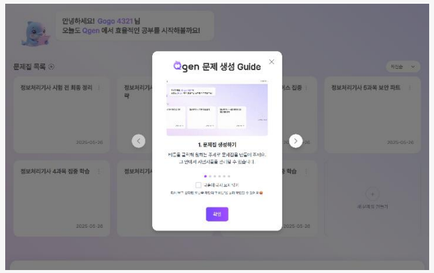
해결한 방법

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



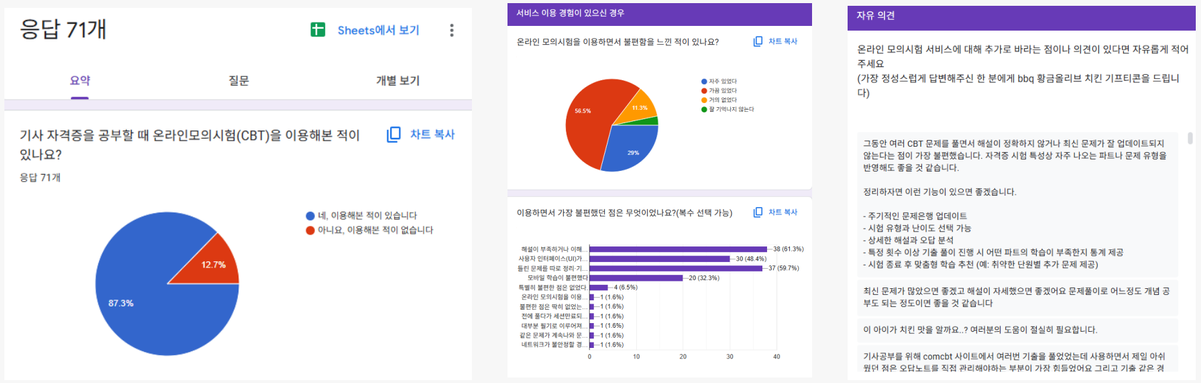
{ 3 트러블 슈팅 및 서비스 개선 사례
사용자 피드백을 통한 서비스 및 UI/UX 개선

60여명의 테스트 유저를 대상으로 사전 설문 및 배포 후 설문을 통해 피드백 받은 결과를 토대로 전반적인 **UI/UX를 개선**하였습니다.

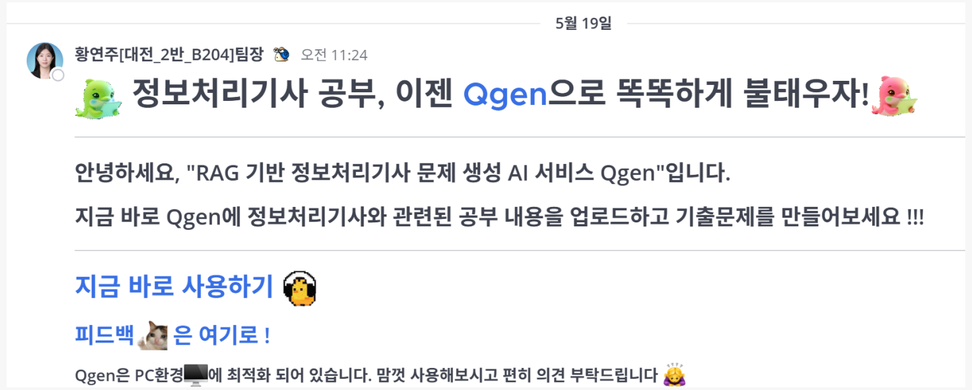
주요 개선 사항

1) 문제 생성 및 **서비스 이용 가이드 모달 추가**
2) **PDF 시험지 생성 기능** 구현
3) AI **기반 풍부한 문제 해설 기능** 제공
4) 오답 및 반복 학습을 위한 **문제 풀이 이력 기록 제공**
5) 문제 **답안 선택 시 보다 직관적인 색 변화** 추가
6) 사용자 자료 추가 시, **바로 파일 리스트가 추가되는 낙관 및 직관적인 애니메이션 구현**

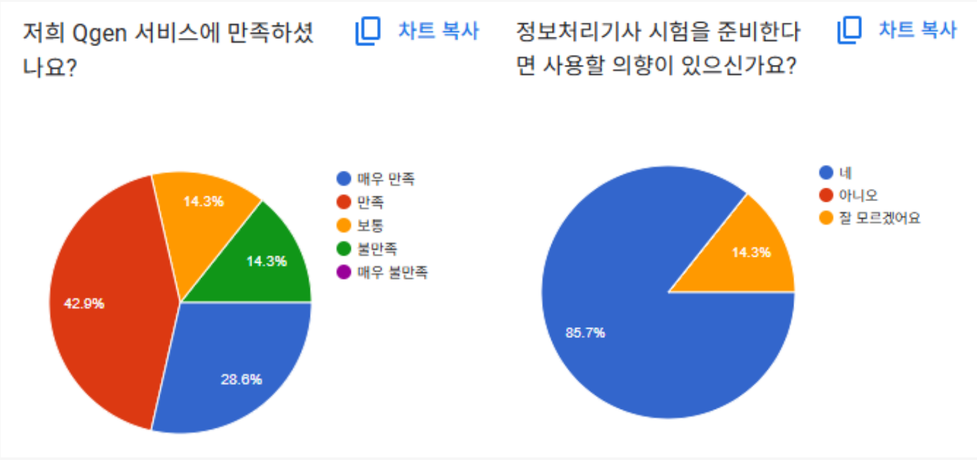
1차 설문 내용



당시 진행한 설문 공지



최종 설문 결과



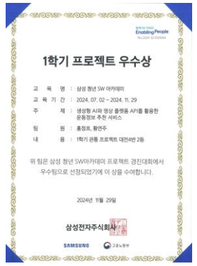
PROJECT 1

날 위한 홈 트레이닝 Mate

Just On

기간 2024.10.24 ~ 2024.11.25 | 참여인원 2명

삼성 청년 소프트웨어 아카데미
프로젝트 경진대회 **2위 수상작**



영상 포트폴리오

GitHub 바로가기

운동 습관 형성을 돕는 홈트레이닝 지원 서비스입니다. **YouTube API를 활용**해 사용자가 선호하는 운동 영상과 음악을 등록하고 관리할 수 있으며, 운동 기록을 분석해 **운동이 부족한 부위를 자동으로 추천**해주는 기능을 구현했습니다.

담당 업무

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

주요 기능

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

설계 방식

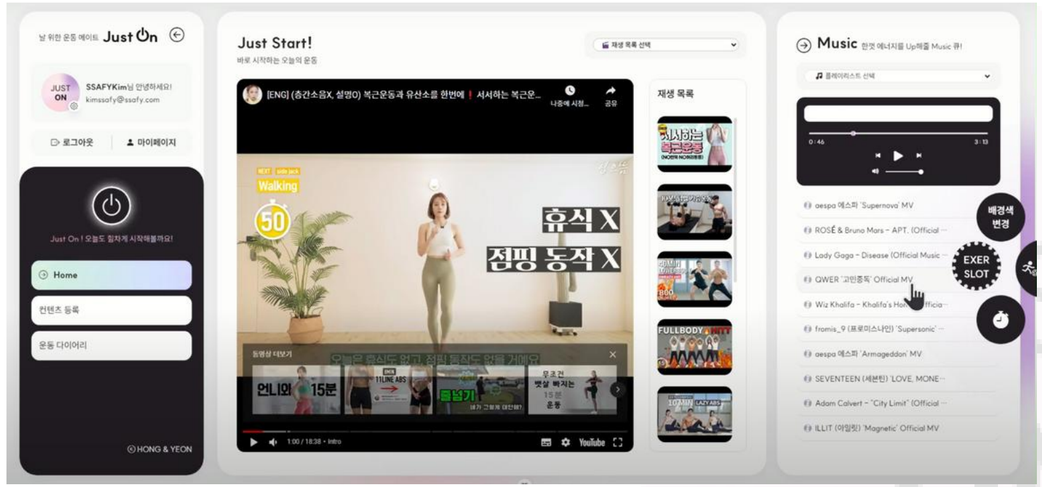
BackEnd REST API + MVC 패턴
FrontEnd SPA

주요 기술 스택

Spring Boot · MySQL · Pinia
MyBatis · Vue.js · Vite
JavaScript · Tailwind CSS

트러블 슈팅 및 서비스 개선 사례

1) 이미지 업로드 파일 저장 경로 배포 의존성 및 덮어쓰기 문제 해결
2) 페이지 간 상태 동기화 문제 및 운동 데이터 중복 입력 문제 해결



{ 1

트러블 슈팅 및 서비스 개선 사례

이미지 업로드 파일 저장 경로 배포 의존성 및 덮어쓰기 문제 해결

발생한 문제

사용자 프로필 파일 업로드 시스템에서 **파일 경로 하드코딩으로 인한 배포 환경 의존성 문제** 발생.
이미지 **동일한 파일명으로 업로드 시 기존 파일이 덮어씌워지는 문제** 발생

해결한 방법

1) 업로드 경로 설정 외부화 : application.properties에서 설정 관리 및 해당 경로 값을 UserServiceImpl에서 @Value 어노테이션으로 주입
2) UUID를 활용한 고유 파일명 생성 : 파일명 중복 방지를 위한 UUID 생성하고, 웹에서 접근 가능한 URL 경로로 저장

개선 효과

1) 배포 환경에 대한 독립성 확보 : 개발/운영 환경에서 각각 다른 설정 파일 사용 가능
2) 파일 관리 시스템 안정성 향상 : UUID를 통해 100% 고유한 파일명을 생성하여 덮어쓰기 문제 해결 및 데이터 무결성 보장.
3) 접근성 개선 : 웹 접근 경로를 표준화함으로써 /uploads/profile_images/ 의 웹 URL로 저장해 프론트엔드에서 직접 접근 가능하도록 함.

개선 전 코드

```java
@Override
public boolean uploadProfile(MultipartFile file, String userId) throws IOException {
    // 기존 파일 삭제
    int result1 = userDao.deleteUserProfile(userId);

    // 파일 저장 경로 생성 (하드코딩)
    String oriName = file.getOriginalFilename();
    String filePath = "C:/SSAFY/JustOn-00/JustOn/JustOn_backend/uploads/profile_images/" + oriName;

    // 파일 저장 (중복 파일명 처리 없음)
    File dest = new File(filePath);
    file.transferTo(dest);

    // DB 저장
    UserProfile userProfile = new UserProfile();
    userProfile.setFilePath(filePath);
    userProfile.setOriName(oriName);
    userProfile.setSystemName(oriName);
    userProfile.setUserId(userId);

    int result2 = userDao.insertUserProfile(userProfile);
    return result1 >= 1 && result2 >= 1;
}
```

개선 후 코드

```java
@Value("${file.upload-dir}")
private String uploadDir;

@Override
public boolean uploadProfile(MultipartFile file, String userId) throws IOException {
    // 기존 파일 삭제
    int result1 = userDao.deleteUserProfile(userId);

    // 파일 저장 경로 생성
    String oriName = file.getOriginalFilename();
    String systemName = UUID.randomUUID().toString() + "_" + oriName;
    String filePath = uploadDir + File.separator + systemName;

    // 파일 저장
    File dest = new File(filePath);
    dest.mkdirs();
    File f = new File(dest, systemName);
    file.transferTo(f);

    // DB 저장 (URL 경로로 설정)
    UserProfile userProfile = new UserProfile();
    userProfile.setFilePath("/uploads/profile_images/" + systemName);
    userProfile.setOriName(oriName);
    userProfile.setSystemName(systemName);
    userProfile.setUserId(userId);

    int result2 = userDao.insertUserProfile(userProfile);
    return result1 >= 1 && result2 >= 1;
}
```

{ 2 트러블 슈팅 및 서비스 개선 사례

페이지 간 상태 동기화 문제 및 운동 데이터 중복 입력 문제 해결

발생한 문제

운동 다이어리 쓰기/보기 페이지 간 **상태 동기화 문제** 발생
페이지 전환 시 이전에 입력했던 **운동 데이터가 유지되지 않거나 중복 저장되는 현상**

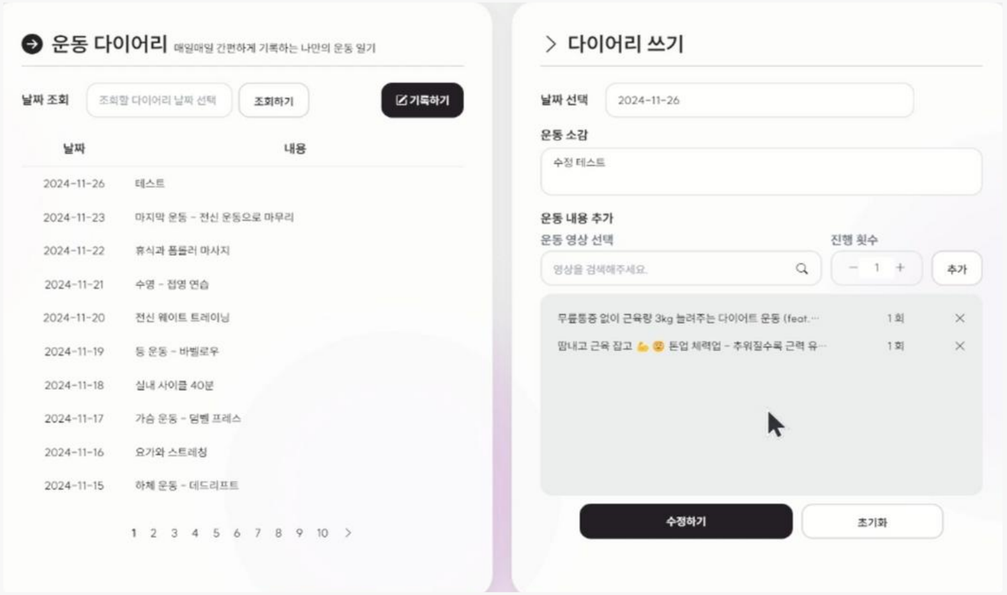
원인 분석 방법

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 전환**이 이루어져 데이터 불일치 발생
라우팅 시 각 페이지에서 개별적으로 API를 호출하여 데이터 중복 또는 불일치 문제 발생

오류가 발생했던 운동 다이어리 페이지



해결한 방법

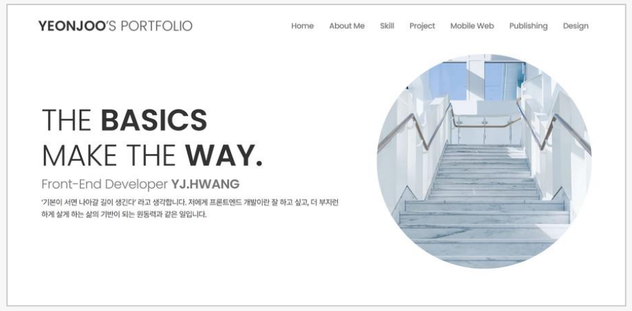
1) API 호출 로직 일원화 : API 호출 로직을 Pinia 액션으로 일원화하여 여러 컴포넌트에서 중복 호출되는 문제를 해결하고, 데이터 일관성을 유지했습니다.
2) 낙관적인 UI 업데이트 : 응답을 기다리지 않고 UI를 먼저 업데이트한 후, 서버 응답에 따라 상태를 조정하는 방식을 적용하여 사용자 경험을 개선했습니다.

UI·UX

편리하고 감각적인 서비스 개발을 위해 **UI·UX 설계에 대해서도 다양한 경험**을 하며 **기초부터** 쌓아왔습니다. 디자인 툴을 활용해 **직접 설계**하고 **퍼블리싱**한 **UI·UX 작업물**을 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

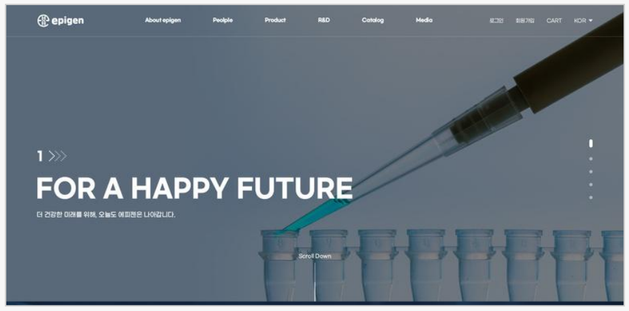
2 음성 품바 축제 사이트 리뉴얼

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

LINKS.

GitHub

기술 블로그

HELLO. MY NAME IS YEONJOO. 개인 사이트

EMAIL.

hyjgood2000@gmail.com