Portfolio

신 예 진

CONTACT

shinyejin0212@gmail.com

010-8685-3403

https://github.com/shyjnnn

https://shyjnnn-dev.vercel.app

Profile

사람 친화적인
PM 신예진입니다.

신예진 / Yejin Shin

shinyejin0212@gmail.com

010-8685-3403

https://github.com/shyjnnn

https://shyjnnn-dev.vercel.app

CAREER

2023.12 ~ 코드잇 스프린트 디비전 교육 PM

EDUCATION

2019.03~2023.08 동국대학교 정보통신공학과

총 학점 : 4.06/4.5

전공 평점 : 4.22/4.5

EXPERIENCE

2023.03 ~ 2023.09 코드잇 프론트엔드 부트캠프 0기

2020.03~2022.02 멋쟁이사자처럼 9-10기

2020.03~2022.02 동국대학교 참사람봉사단 14-16기

AWARD

2022.06 한국통신학회 하계 종합 학술 발표회
5G기반의 ICT 융합 서비스
아이디어 경진대회 우수상

2021.12 동국대학교 ICIP & 캡스톤 디자인 장려상

SKILL

Soft Skill

추진력 / 설득력 / 매니징 / 멀티태스킹 역량

Front-end

javascript, typescript, React.js, Next.js, Storybook, Tanstack query, Recoil

Back-end

Django, AWS

Cooperation Tool

Figma, Jira, Confluence, Notion, Slack

Project Index

01

나 사용 설명서 제작 플랫폼 '나자신'

· 기능 소개
· 협업 및 개발 방식
· 프로젝트 결과

02

아르바이트 구인 구직 플랫폼 '더줄게'

· 기능 소개
· 협업 및 개발 방식
· 프로젝트 결과

03

축제 정보 제공 플랫폼 '레츠끼릿'

· 기능 소개
· 협업 및 개발 방식
· 프로젝트 결과

2023. 08.03 - 08.31 (4주)

나 사용 설명서 제작 플랫폼 나자신

메타인지를 위해 스스로 나 사용 설명서 입력 및 제작하고
타인에게 평가를 받는 웹 서비스입니다.

팀원 프론트(4), 백(2), 디자인(1)

기술 스택

typescript, Next.js, Storybook, Tanstack query, Recoil, SCSS, Framer, Husky, lint-staged

Demo Link GitHub

Demo Link GitHub

기능 #1 나 사용 설명서 만들기

주요 기능

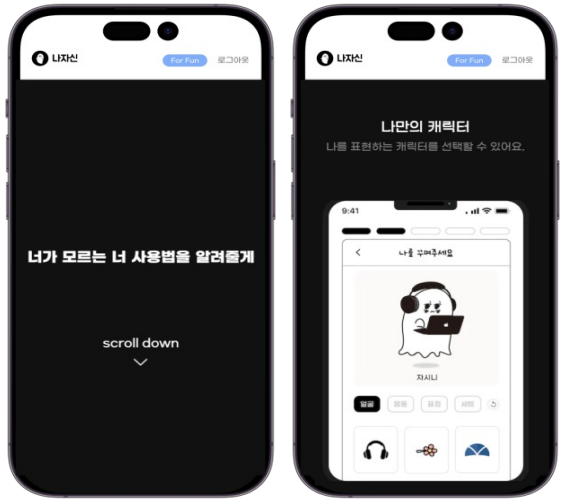
- 최초 회원가입 시, 유저는 나 사용 설명서 작성을 할 수 있습니다.
- 공유받은 링크로 타인이 나에 대한 설명을 작성할 수 있습니다.

기능 #2 사용 설명서 공유하기

주요 기능

- 마이페이지에서 공유 링크를 생성할 수 있습니다.
- 공유 링크로 접속한 사람은 로그인 없이 해당 유저에 대한 설명서를 작성할 수 있습니다.

기능 #3 서비스 이용 설명서



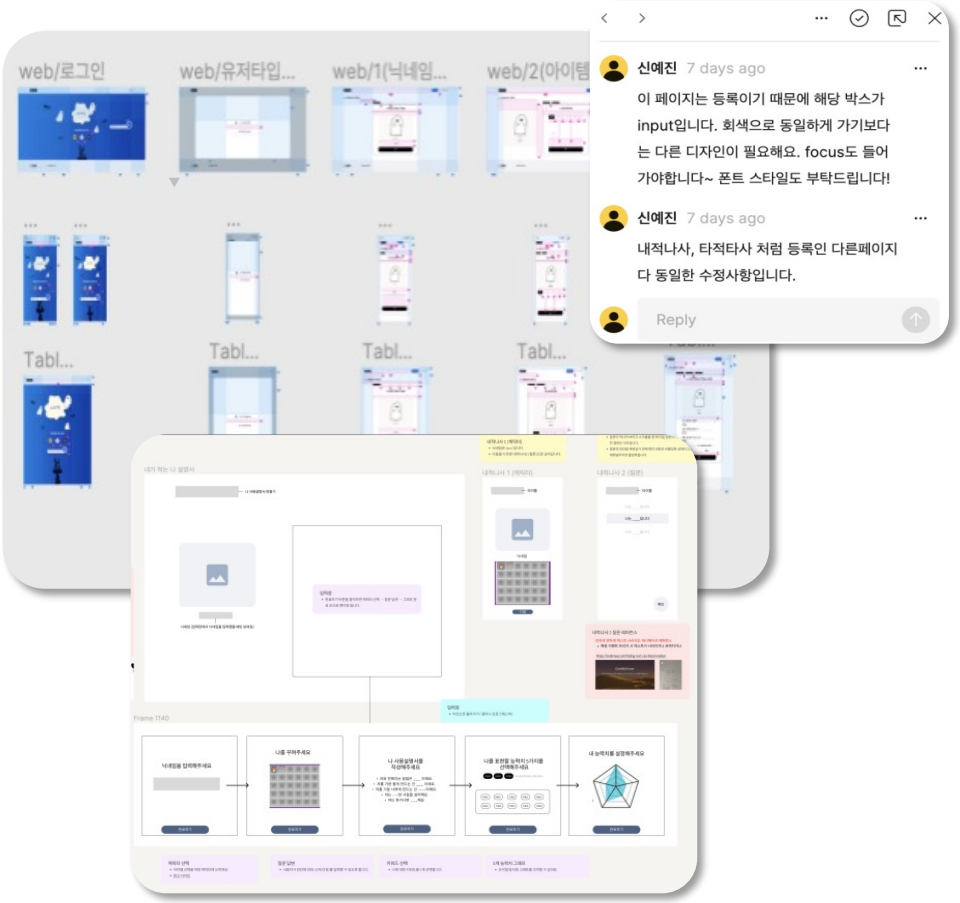
주요 기능

- 서비스에 대한 사용 설명을 볼 수 있습니다.
- 인터렉티브하게 스크롤에 대한 서비스 플로우가 전개됩니다.

기획 및 디자인

Figma를 통한 디자이너와의 비동기 소통

와이어프레임 전달 후 구체화



디자인 패턴

Compound 디자인 패턴 사용

```
<component>
  <component.sub1
/><component.sub2
/><component.sub3
/>...
</component>
```

장점

- 의존성을 분리한 독립적인 컴포넌트를 만들 수 있습니다.
- 잦은 변경에 대응 가능합니다.

단점

- 불필요한 컴포넌트가 많아질 수 있습니다.
- 컴포넌트를 독립적으로 만드는 건 쉽지 않습니다.

결론

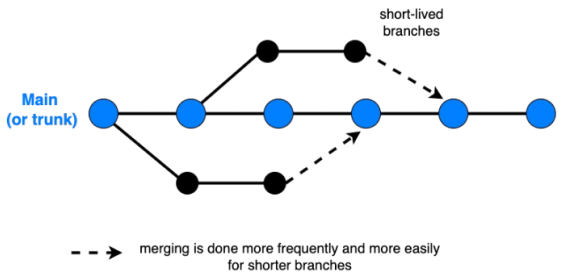
- 기획, 디자인, 개발 전 과정의 Agile한 개발을 위해 잦은 변경에 용이한 Comound 패턴으로 컴포넌트 설계 및 개발했습니다.

02 협업 및 개발 방식: 담당 역할

Demo Link GitHub

브랜치 전략

Trunk-based development

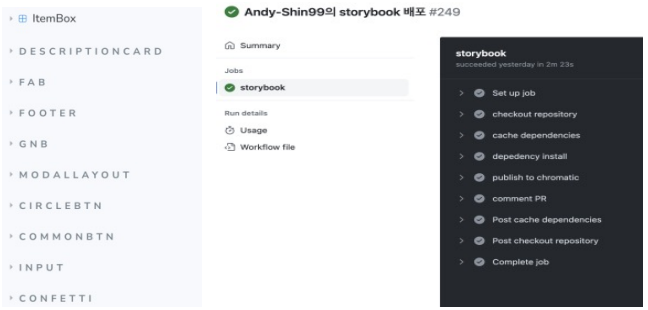


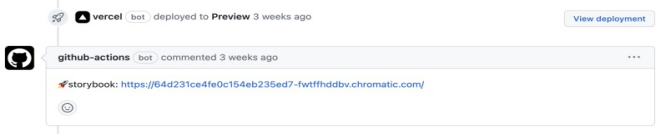
항상 배포가능한 main 브랜치를 두고, 짧은 수명을 가진 feature 브랜치를 운용

Squash merge로 여러개의 커밋을 하나의 커밋으로 합친 후 merge

Storybook

스토리북을 활용한 컴포넌트 문서화





- 범용적으로 사용되는 공통 컴포넌트의 다양한 쓰임세를 스토리북으로 문서화
- chromatic과 github action을 활용한 스토리북 자동 배포
- 매 pr 마다 스토리북도 자동으로 배포해 협업 방식 개선

Mock API

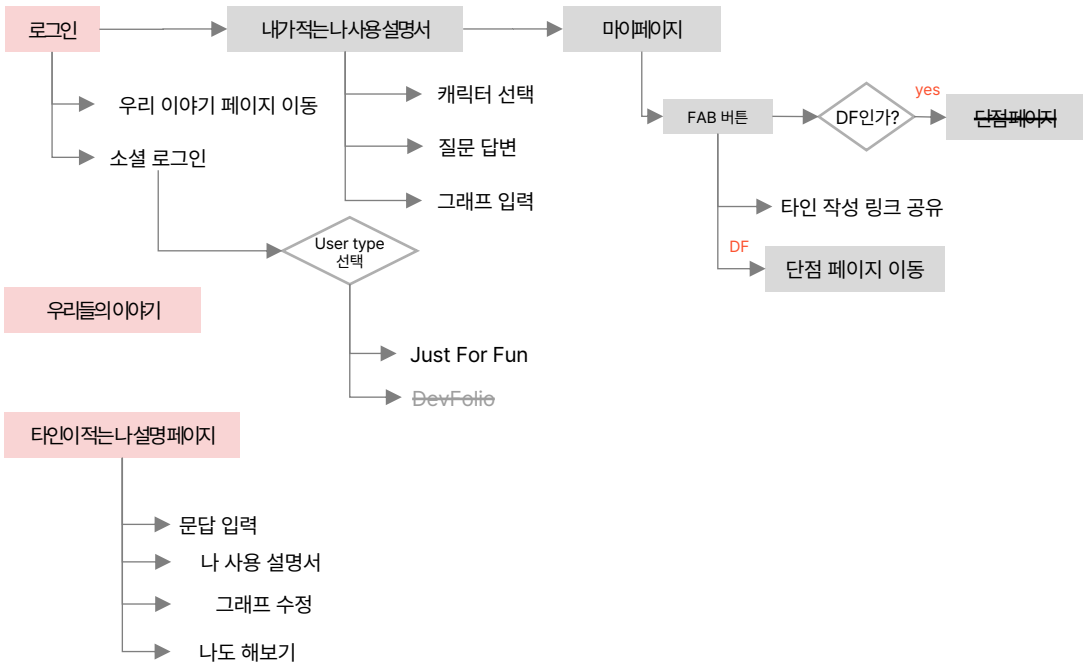
Next.js api로 Mock API 구축

Next.js로 Mock API를 만들어
백엔드 api 개발 전 프론트엔드 개발 시 활용

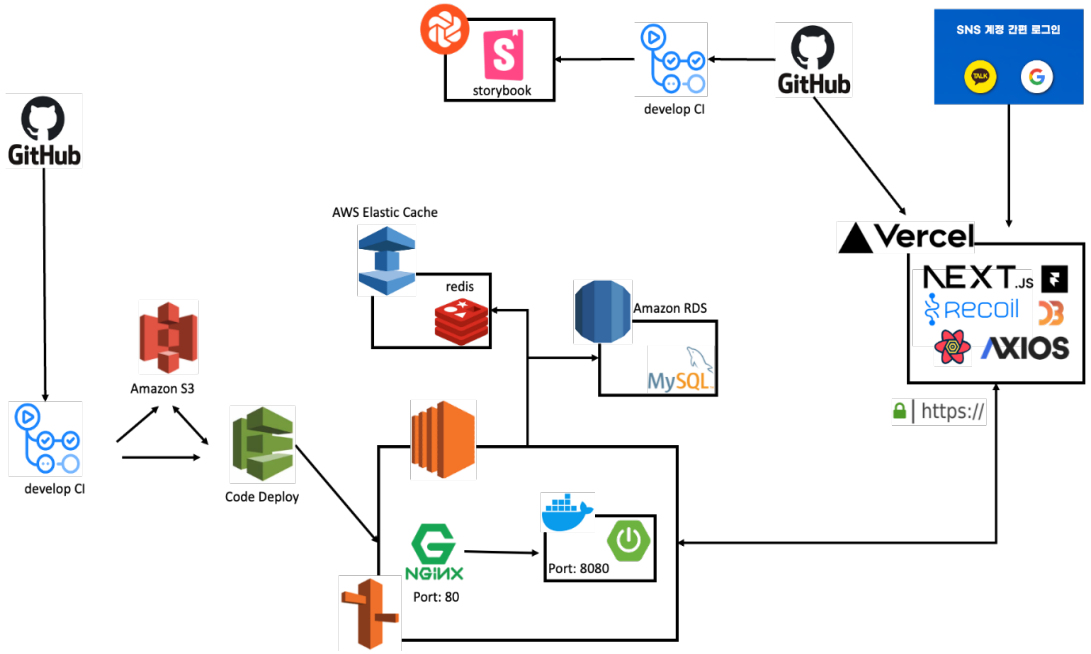
03 프로젝트 결과: 담당 역할

Demo Link GitHub

유저 플로우



서비스 아키텍처



Demo Link GitHub

useFunnel hook 제작

필요성

- 크기가 한정된 모바일에서 사용자 입력 흐름 제어해 사용자 경험 개선 필요
- 기존 slash 라이브러리의 useFunnel hook이 내부적으로 next 12버전에 의존

기능 구현

- State에 따라 렌더링
- 브라우저 뒤로가기 시, 마치 페이지 이동처럼 이전 Step으로 가기

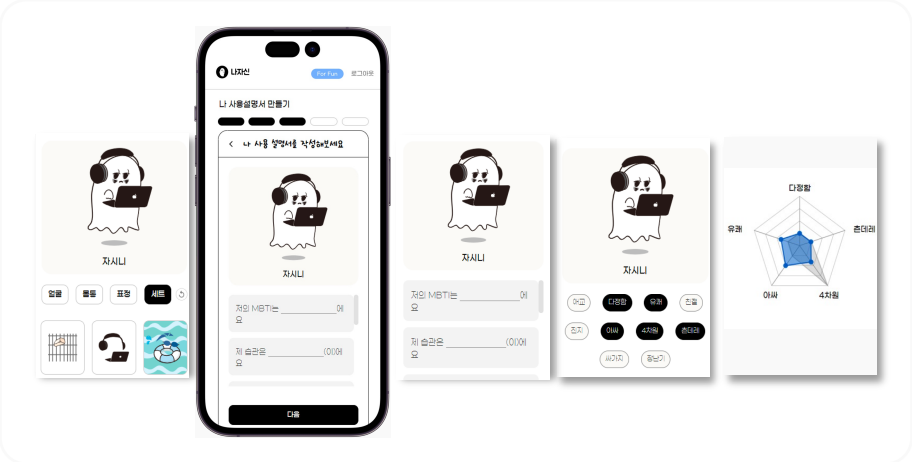
결과

useMemo를 사용해 step이 변경되었을 때만 컴포넌트 새로 생성해 렌더링 최적화

```
// Funnel 컴포넌트와 함께 Step 컴포넌트를 반환하는 객체를 생성
const FunnelElement = useMemo(
  () =>
    Object.assign(
      (props: Omit<IFunnelProps<T>, 'step'>) => (
        <Funnel step={step} {...props} />
      ),
      { Step: (props: IStepProps<T>) => <Step<T> {...props} /> },
    ),
  [step],
)
```

React-Query를 사용해 API 호출 최적화

등록하기 흐름에 따라 5회 이상의 API 호출이 요구됩니다.



- 하나의 Endpoint에서 전체 입력 데이터를 받을 수 있도록 설계를 변경하였습니다.
- React-Query의 캐싱을 사용해 API 호출을 5회에서 1회로 감소시켰습니다.
- 데이터 가져오기를 위한 API 호출을 한번으로 합쳐 State가 비대해지는 것을 방지하면서 API 호출 횟수도 감소시켰습니다.

2023. 06.22 - 07.21 (4주)

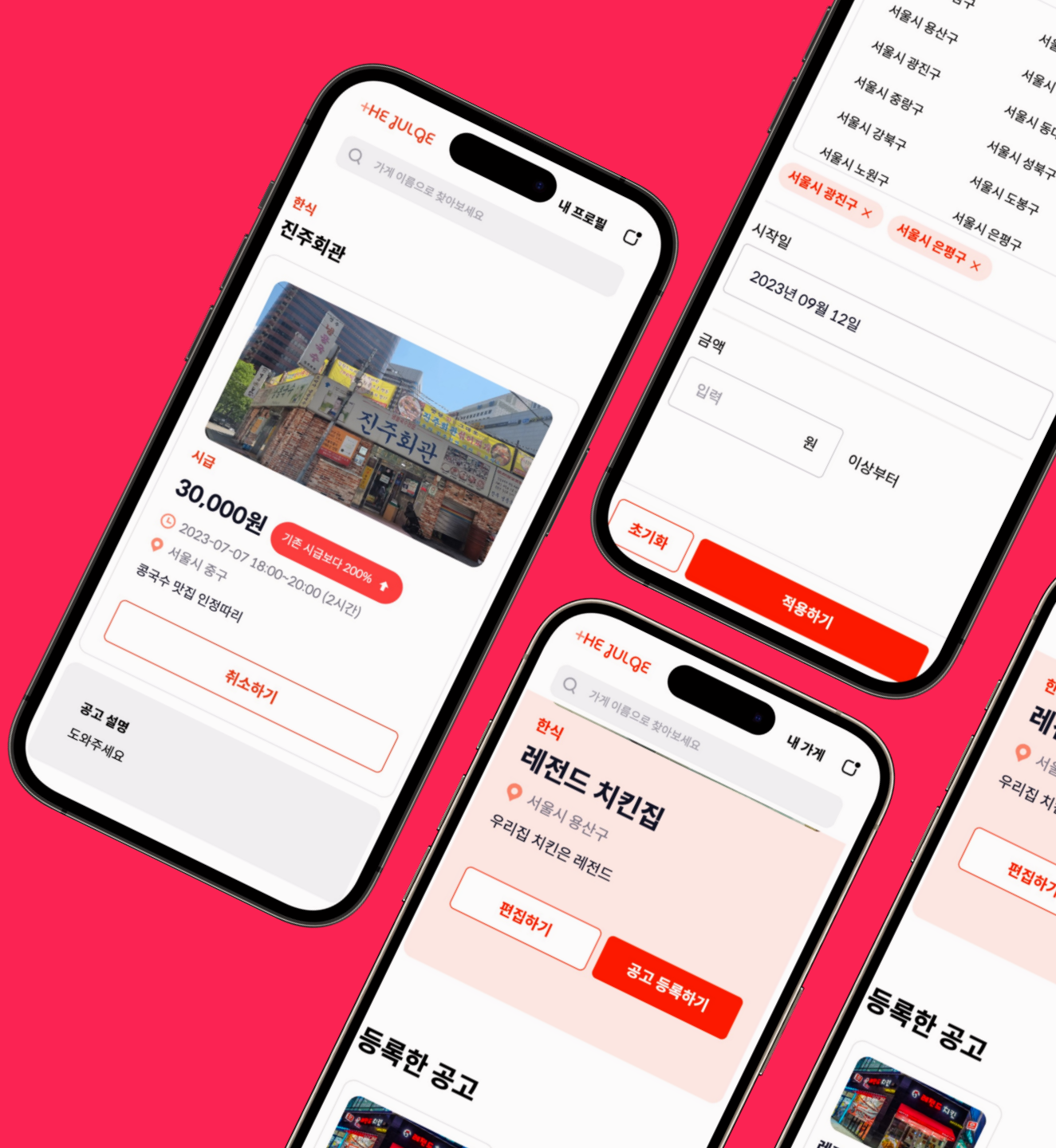
아르바이트 구인 구직 플랫폼 더줄게

급하게 일손이 필요한 자리에 더 많은 시급을 제공해서
아르바이트생을 구할 수 있는 서비스

팀원 프론트(4), 백(2)

기술 스택 Next.js, typescript, SCSS, Husky, lint-staged

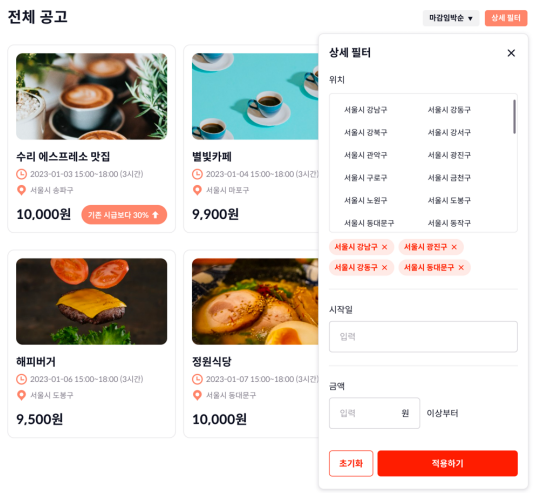
Demo Link GitHub



01 기능 소개

Demo Link GitHub

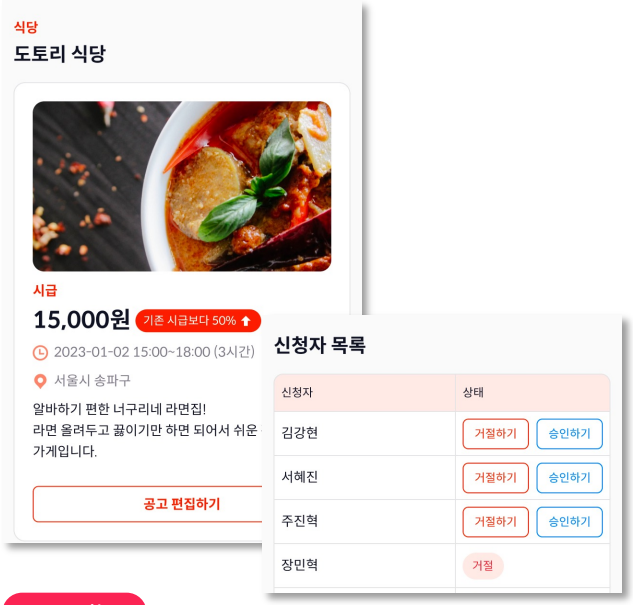
기능 #1 전체 공고 검색 및 필터링



주요 기능

- 홈페이지에서 전체 공고 목록을 볼 수 있습니다.
- 공고 검색 페이지에서는 키워드를 통해 특정 공고를 검색할 수 있습니다.
- 비회원, 회원에 상관없이 이용 가능합니다.

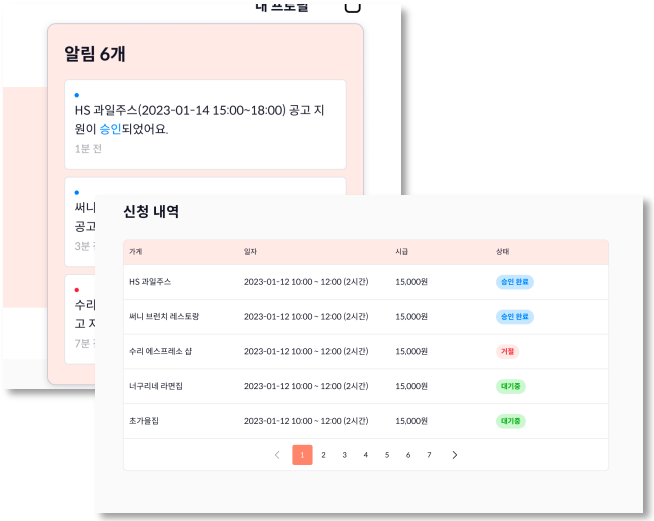
기능 #2 사장님 가게 및 공고 등록



주요 기능

- 사장님은 가게 페이지를 생성하고, 기본 공고를 등록할 수 있습니다.
- 기존의 시급보다 급여를 올리거나 낮춰서 추가 공고를 등록할 수 있습니다.
- 공고 별 신청자 목록을 관리할 수 있습니다.

기능 #3 구직자 신청하기



주요 기능

- 공고 확인 후 공고 상세 페이지에서 신청 가능합니다.
- 내 프로필 상세 페이지에서 신청한 공고 내역을 한눈에 확인 가능합니다.
- 신청한 공고에 거절, 승인 처리가 되면 알림이 옵니다.
- 최근 본 공고 목록을 확인할 수 있습니다

Demo Link GitHub

폴조 구조

NX Framework의 Library Types 구조에서 착안

4가지 폴더 타입으로 관심사 분리

Feature

- 일명 Smart UI
- 데이터 소스에 대한 액세스 권한을 포함
- 기능 라이브러리
- 다른 타입이 조합되는 곳

UI

- 일명 Dumb UI
- 프레젠테이션 포험
- feature를 가질 수 없습니다.

Data-access

- 백엔드 시스템과 상호작용하기 위한 코드
- 상태 관리와 관련된 모든 코드

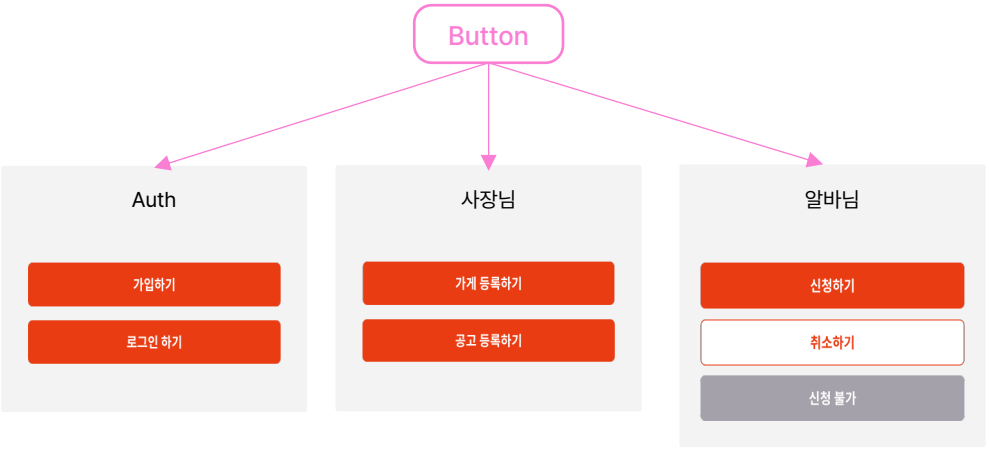
Util

- 다른 타입 or 라이브러리에서 사용되는 저수준 유틸리티가 포함

컴포넌트 설계

Polymorphic한 컴포넌트

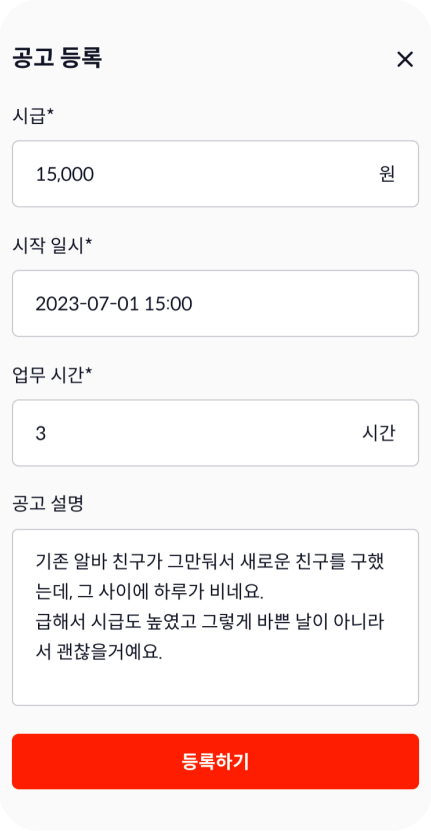
ForwardRef와 Props를 적극 활용해 범용적인 공통 컴포넌트 제작



03 프로젝트 결과 : 담당 역할

Demo Link GitHub

AS-IS

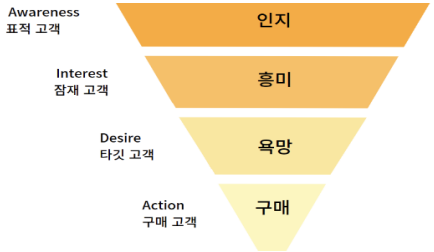


문제점

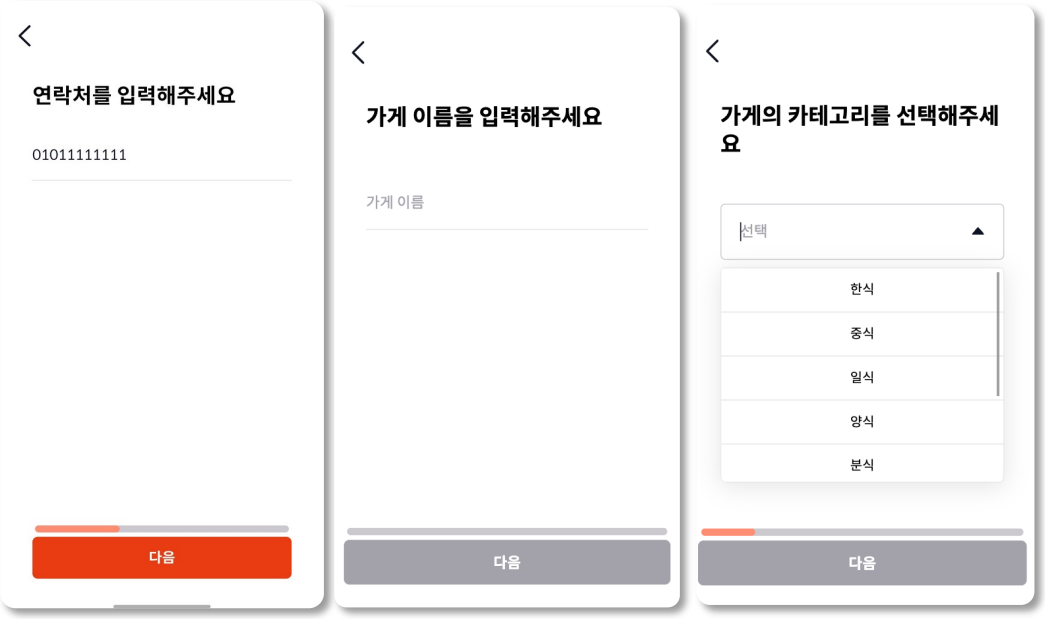
- 모바일 사이즈에서 입력 시, 모달 내 스크롤이 생깁니다.
- 사용자 경험 개선을 위해 입력 흐름 관리가 필요합니다

해결 방법

- 퍼널로 유저가 가입할 때까지의 과정을 한번에 처리하지 않고 단계를 나눠 유저 이탈을 막았습니다.



TO-BE



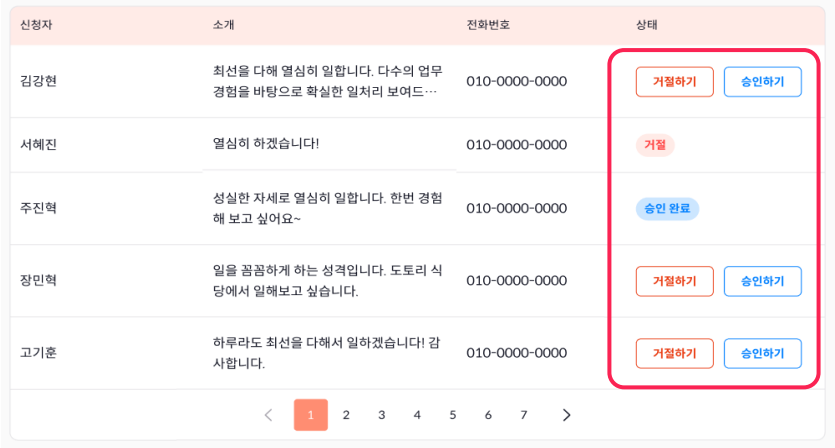
Funnel Pattern을 사용해 입력 단계를 가지도록 해 모달 내 스크롤을 방지했습니다.

Demo Link GitHub

서버 컴포넌트

Zero 번들 사이즈를 위한 서버 컴포넌트 활용

서버/클라이언트 컴포넌트 로직 분리



신청자	소개	전화번호	상태
김강현	최선을 다해 열심히 일합니다. 다수의 업무 경험을 바탕으로 확실한 일처리 보여드…	010-0000-0000	거절하기 승인하기
서혜진	열심히 하겠습니다!	010-0000-0000	거절
주진혁	성실한 자세로 열심히 일합니다. 한번 경험해 보고 싶어요~	010-0000-0000	승인 완료
장민혁	일을 꼼꼼하게 하는 성격입니다. 도토리 식당에서 일해보고 싶습니다.	010-0000-0000	거절하기 승인하기
고기훈	하루라도 최선을 다해서 일하겠습니다! 감사합니다.	010-0000-0000	거절하기 승인하기

< 1 2 3 4 5 6 7 >

인터랙션이 필요한 버튼 칸은 클라이언트 컴포넌트로, 나머지 정적 데이터는 서버 컴포넌트로 분리

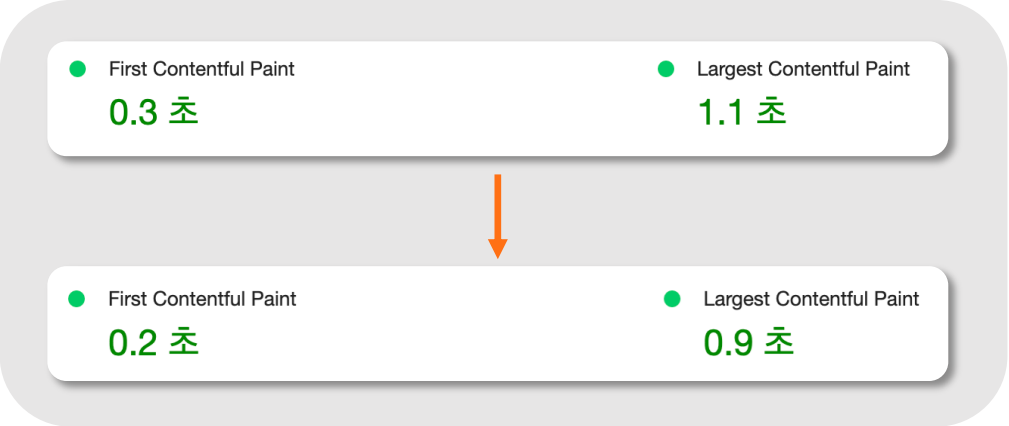
번들 사이즈 5KB 감소

```
+ First Load JS shared by all          80.7 kB
```

↓

```
+ First Load JS shared by all          75.7 kB
```

FCP & LCP 1ms ~ 2ms 개선



2022. 08 - 09. (4주)

축제 정보 제공 플랫폼
레츠끼릿

교내 가을 축제를 위한 커뮤니티 및 축제 정보를 제공하는
모바일 웹 서비스

팀원 프론트(4), 백(4), 기획 및 디자인(3)

기술 스택 React.js, axios, docker, AWS

Demo Link GitHub

01 기능 소개

Demo Link GitHub

기능 #1 축제 공연 정보

주요 기능

- 요일별 타임테이블로 공연 정보를 제공합니다.
- 총학생회의 공지사항을 볼 수 있습니다.

기능 #2 부스 정보 제공

주요 기능

- 교내 건물 위치를 기반으로 부스 위치를 확인할 수 있습니다.
- 부스별 판매 메뉴 등 정보를 제공하며, 좋아요 및 방명록 등 커뮤니티 기능을 제공합니다.

기능 #3 커뮤니티 기능

주요 기능

- 부스별 좋아요 및 방명록 기능을 제공합니다.
- 로그인 없이도 좋아요 기능을 사용할 수 있으며, 브라우저 별 쿠키로 동일 사용자를 식별합니다.
- 인기 Top10 부스 목록을 확인할 수 있습니다.

프론트엔드를 중심으로 기획 및 디자인, 백엔드 일정 매니징

프론트엔드 ⟷ 백엔드

API

Entity	Method	URL	Aa 설명	완료여부
Notification	GET	/api/notifications	공지 리스트 2	☑
Notification	GET	/api/notifications?filter={enum}	공지 리스트(필터)	☑
Notification	POST	/api/notifications	공지 생성(관리자용)	☑
Notification	GET	/api/notifications/{id}	공지 디테일	☑
Notification	PUT	/api/notifications/{id}	공지 수정(관리자용)	☑
Notification	DELETE	/api/notifications/{id}	공지 삭제(관리자용)	☑
Booth	GET	/api/booths/top5	부스 좋아요 top5	☑
Booth	GET	/api/booths?filter={menu, title, location}	부스 검색 리스트	☑
Booth	POST	/api/booths	부스 생성(관리자용)	☑
Booth	GET	/api/booths/{id}	부스 디테일	☑
Booth	PUT	/api/booths/{id}	부스 수정(관리자용)	☑
Booth	DELETE	/api/booths/{id}	부스 삭제(관리자용)	☑
Booth	GET	/api/booths?day={day}&location={location}	부스 날짜 위치 필터 2	☑
Comment	GET	/api/booths/{id}/comments	부스 방명록 리스트	☑
Comment	POST	/api/booths/{id}/comments	부스 방명록 생성	☑
Comment	DELETE	/api/comments/{comment_id}	부스 방명록 삭제(유저용)	☑
Comment	DELETE	/api/comments/{comment_id}/force	부스 방명록 삭제(관리자용)	☑
Menu	GET	/api/booths/{id}/menus	부스 메뉴 리스트	☑
Menu	POST	/api/booths/{id}/menus	부스 메뉴 생성(관리자용)	☑
Menu	PUT	/api/menus/{id}	부스 메뉴 수정(관리자용)	☑
Menu	DELETE	/api/menus/{id}	부스 메뉴 삭제(관리자용)	☑
Like	POST	/api/booths/{booth_id}/likes	부스 좋아요	☑
Like	DELETE	/api/booths/{booth_id}/likes	부스 좋아요 삭제	☑

프론트엔드 ⟷ 기획 및 디자인

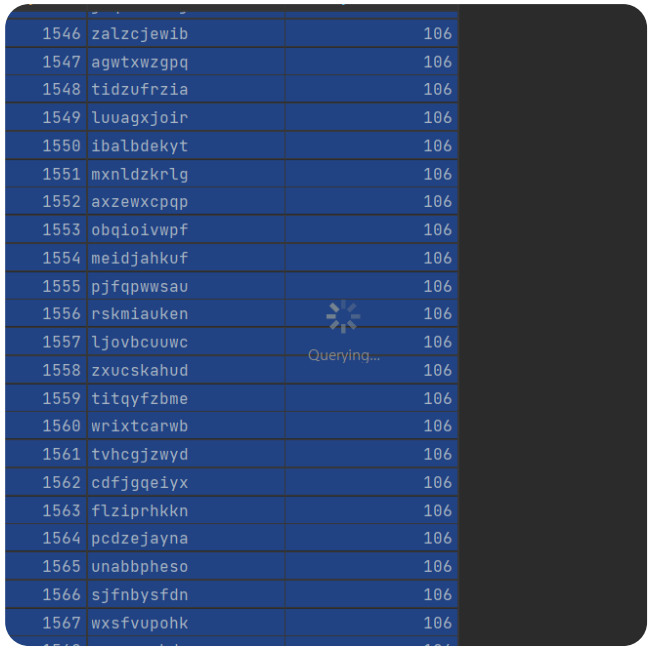
- 기획안을 기반으로 API 설계 및 명세를 했습니다.
  - 미리 협의한 API 명세를 기반으로 Restful하게 개발을 진행했습니다.
- 총학생회 및 축제준비위원회와 비동기 소통하며 2주 스프린트 주기의 Agile 방법론으로 개발 진행했습니다.
  - 욕설 필터링, 좋아요 기능 등을 추가 개발했습니다.

로그인없는 좋아요 기능

쿠키 특성을 활용해 같은 브라우저 접속 시, 동일한 사용자로 판단되도록 구현

기존의 문제점

브라우저 하나 당 좋아요 개수를 제한하지 않았더니
운영 중, 10초 내 5만개의 좋아요가 생성되었습니다.



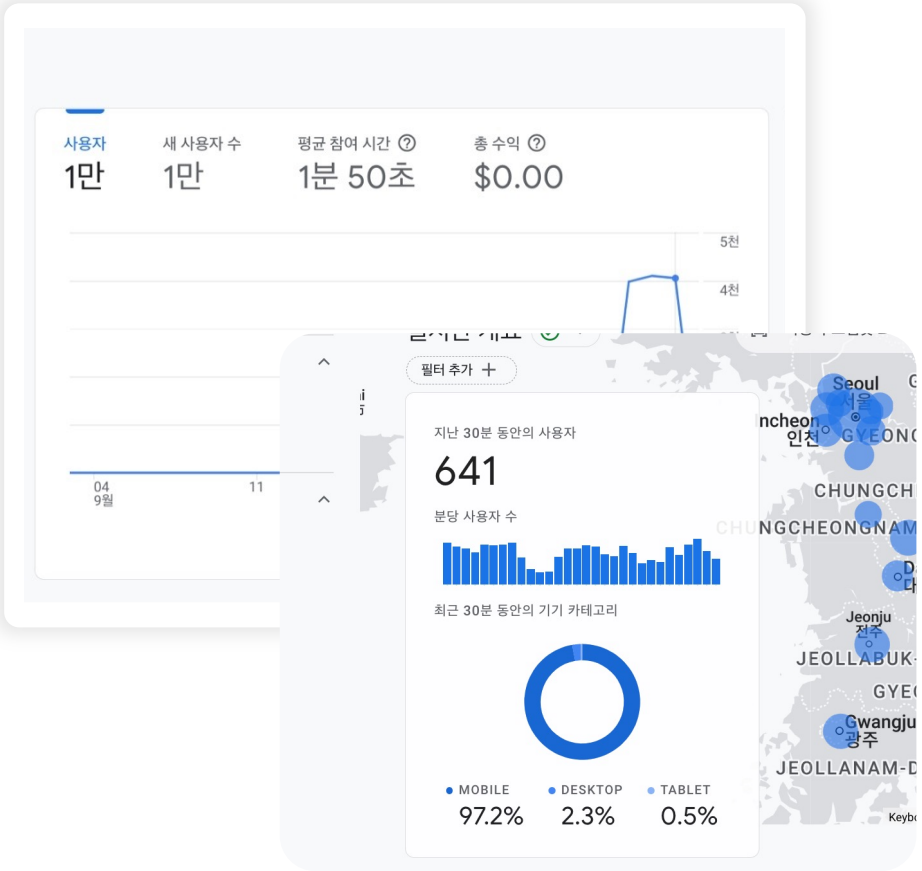
백엔드에서 쿠키 생성하고, 프론트엔드에서 쿠키 http-only 설정을 해 중복 생성을 방지했습니다.

```java
@PostMapping("/{id}/likes")
public LikesResponseDto likeCreate(@PathVariable Long id, HttpServletRequest request
    Optional<Cookie> boothCookie = likesService.findBoothCookie(request, id);
    if(boothCookie.isPresent()){
        throw new IllegalArgumentException("이미 쿠키 있음");
    }
    LikesResponseDto likes = likesService.create(id);
    Cookie keyCokkie = new Cookie(id.toString(), likes.getCookieKey());
    keyCokkie.setMaxAge(7*60*60*24);
    keyCokkie.setPath("/");
    response.addCookie(keyCokkie);
    return likes;
}
```

백엔드 서버 코드

3일 간, 약 1.2만명의 교내 사용자 확보

그러나 약 3천여명의 사용자 이탈이 있었음 추정됩니다.



30분 간 이용자 수

원인

약 2시간의 운영 중단

- 30분 당 500여명의 신규 이용자가 들어오는 것으로 예상돼 운영 중단된 2시간 동안 약 3천 여명의 이용자 이탈이 예상됩니다.
- 과도한 렌더링으로 인한 서버 과부하가 원인이었습니다.
- 코드 리뷰의 부재로 빠른 복구에 실패했습니다.

해결 방법

- **적절한 로드 밸런싱 및 오토 스케일링 설정**
  - 서버 부하를 분산시키고 트래픽 증가에 유연하게 대응하기 위해 확장성을 고려한 설정을 해 해결할 수 있습니다.
- **CDN을 통한 자원 최적화, 캐싱**
  - CDN을 활용해 정적 자원을 최적화 하고, 데이터를 적절히 캐시해 서버 부하를 줄일 수 있을 것으로 예상됩니다.

감사합니다

CONTACT

shinyejin0212@gmail.com

010-8685-3403

https://github.com/shyjnnn

https://shyjnnn-dev.vercel.app