

2023 PORTFOLIO

화면 아래에 논리를 펼쳐주는
프론트엔드 엔지니어 조원주입니다

- 📞 온라인 비공개
- ✉ dnjswn9178@hanmail.net
- 🌐 <https://github.com/twoju>

“화면 아래에 논리를 펼쳐주는 **프론트엔드 엔지니어 조원주**입니다”

도움과 봉사의 가치를 알기에, 모든 사람이 쉽게 서비스를 접하고 효율적으로 사용하는 날을 만들기 위해 고민하고 노력합니다.

UX/UI를 전공하며 다양한 웹, 앱 서비스를 기획하고 디자인했습니다. 프로토타이핑에서 더 나아가 실 사용자와 상호작용할 수 있는 서비스를 직접 구현하고 싶어 개발을 시작하게 되었습니다.

다양한 데이터를 근거로 논리적인 추론 방식에 따라 경험을 디자인하는 UX/UI를 전공한 역량은 프론트엔드 엔지니어로서 효율적인 로직을 구현하는데 기여할 것입니다.

항상 고민하고 새로운 기술을 탐험하며 화면 아래에 최적화된 논리를 구현하는 프론트엔드 엔지니어로 함께 성장하겠습니다.



조원주

CHO WON JU

2000. 03. 01

청렴결백한 논리주의자 ISTJ

...

온라인 비공개

dnjswn9178@hanmail.net

<https://github.com/twoju>

학력

2019. 03 ~ 2023. 02

홍익대학교 디자인컨버전스학부 졸업

2016. 03 ~ 2019. 02

충남삼성고등학교 생명과학 전공 졸업

경험

2023. 01 ~ 현재

삼성 청년 SW 아카데미 9기

2023. 06

한국 디지털 콘텐츠 학회 논문 게제

한국사와 코딩을 연계한 초등학생 대상 교육용 앱 서비스

수상

2023. 08. 30

삼성 청년 SW 아카데미 공통 프로젝트 우수상

비대면 공부 습관 형성 웹 IoT 프로젝트, 도깨비봇

2021. 12. 14

ICT 이노베이션스퀘어 공모전 장려상

발달 장애인의 의사소통을 위한 모바일 앱, 악어톡

자격

2022. 09. 02

시각디자인기사

2012. 09. 13

정보기술자격(ITQ) A등급

SKILLS



- ES6 기반 프로젝트 구현 및 JS 기반 라이브러리 사용 가능
- 변수 타입 명시를 통한 타입 유효성 체크 가능
- 함수형 컴포넌트 기반 재사용성 높은 코드 작성 경험



- react-redux, recoil 을 활용한 데이터 상태관리 구현 가능
- react-pwa 로 웹앱 프로젝트 진행
- react hook 을 활용한 함수형 컴포넌트 제작 가능



- 독립적인 UI 컴포넌트 개발 경험
- 변수를 활용한 재사용성 높은 css 작성 가능

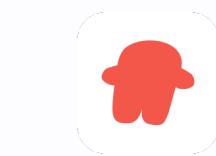


- axios 를 활용한 http 데이터 통신 구현 가능
- vuex와 vue routers 활용을 통한 소규모 프로젝트 진행



- 서버사이드렌더링 (SSR) 을 통한 페이지 라우팅 구현 가능
- pre-rendering 과 hydration 개념을 알고 있음

CONTENTS



Qookie

2023. 10 - 11 (6주)

간단한 일상생활을 통해 마음 건강을 챙길 수 있도록
chat GPT 프롬프트와 게이미피케이션을 접목한
프로그레시브 웹 앱 서비스, Qookie

p. 4 - 7



방글

2023. 08 - 10 (7주)

독립 출판물을 사랑하는 작가와 독자를 위한
블록체인 기반 전자 서점 플랫폼, 방글

p. 8 - 10



도깨비봇

2023. 07 - 08 (7주)

시간과 장소의 제약 없이 효율적으로 공부할 수 있도록
도와주는 공부 습관 형성 웹IoT 서비스, 도깨비봇

p. 11 - 13



Qookie

간단한 일상생활을 통해 마음 건강을 챙길 수 있도록 chat GPT 프롬프트와 게이미피케이션을 접목한 PWA 서비스, Qookie

창작 프로젝트

PWA

게이미피케이션

개발

BACKGROUND

진입 장벽이 낮은 마음 챙김 서비스는 없을까?

2021년 OECD 통계에 따르면 우리나라에서 우울증 증상을 경험한 사람은 약 40%입니다. 이 수치는 같은 기간 발표된 OECD 15개국 평균인 22%보다 높은 수치를 보입니다. 하지만 우울증과 불안장애 진단을 받은 사람 중 상담 등 정신건강서비스 이용 경험이 있는 비율은 13% 밖에 되지 않습니다. 이처럼 낮은 심리 건강 서비스 경험률은 해당 서비스를 향한 보편적인 거부감 혹은 무관심에 기인합니다. 보다 쉬운 마음 챙김 서비스를 받을 수 있도록 일상 생활에 캐릭터 게이미피케이션을 결합한 웹 앱 서비스 Qookie를 기획하게 되었습니다. 모바일 접근성을 높이기 위해 프로그래시브 웹 앱(PWA) 형태를 사용하였습니다.

GitHub 코드

기술 스택 및 구현



구현 내용

- React PWA 개발로 모바일 사용성을 고려한 개발
- TypeScript 를 활용한 타입 오류 및 가독성 관리
- state 관리 및 데이터 전달 활용을 위한 React-redux 사용

작업기간

2023. 10 ~ 2023. 11 (6주)

기여도

6인 팀 (프론트 3, 백2, 풀스택 1)

프론트엔드 개발 30%

프로그램

React, recoil, Typescript, 피그마
storybook, styled-component 등

기술 선택 이유

React : React 를 활용해서 PWA 개발이 가능함

recoil : get 과 set 을 활용한 부분 데이터를 변경하는 등의 활용이 쉽고, persist로 상태 저장이 용이함

TypeScript : 데이터의 타입을 지정하여 타입 오류를 방지함

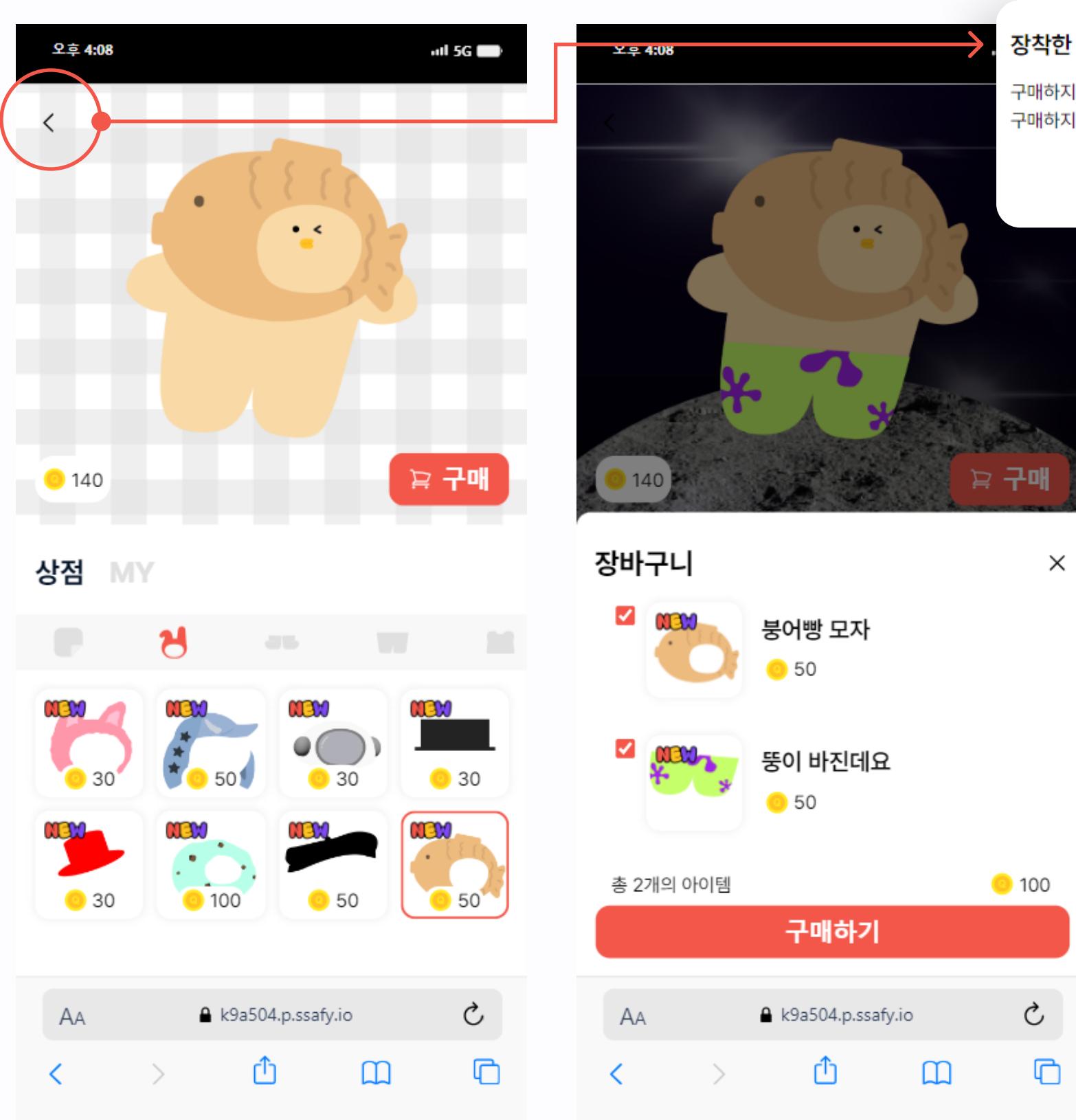
Styled-component : 템플릿 리터럴을 활용한 컴포넌트 다양화가 가능함





캐릭터 옷 입히기

상점과 옷장이 합쳐져있는 꾸미기 공간을 구현하기 위해서, 화면을 나가는 경우를 감지하여 myItemList에 있지 않는 아이템을 장착하고 있는 경우 팝업 알림창을 통해 아이템을 구매하지 않으면 저장되지 않음을 보여주었습니다.
만약 구매한 아이템만 있는 경우에는 알림창 없이 바로 저장되는 구조입니다. 이 때 뒤로 가는 상황을 감지하기 위해서 history를 활용하였습니다.



```
useEffect(() => {
  // 뒤로가기 감지
  const checkBackEvent = () => {
    if (checkItemToBuy().length > 0) {
      setIsExit(true);
      navigate('/store');
    } else {
      exitStoreHandler();
    }
  };

  const unCheckBackEvent = history.listen(({ action }) => {
    if (action === 'POP') {
      checkBackEvent();
    }
  });

  return unCheckBackEvent;
}, [selectedItemList, myItemList]);
```

```
const exitStoreHandler = () => {
  // 입은 옷 저장 api

  if (selectedItemList) {
    const acc: number[] = [1];
    selectedItemList.accessories.map(item) => acc.push(item.id));
  }

  const wearItemId: wearReqType = {
    hatId: selectedItemList.hat.id,
    topId: selectedItemList.top.id,
    bottomId: selectedItemList.bottom.id,
    shoeId: selectedItemList.shoe.id,
    backgroundId: selectedItemList.background.id,
    accessories: acc,
  };

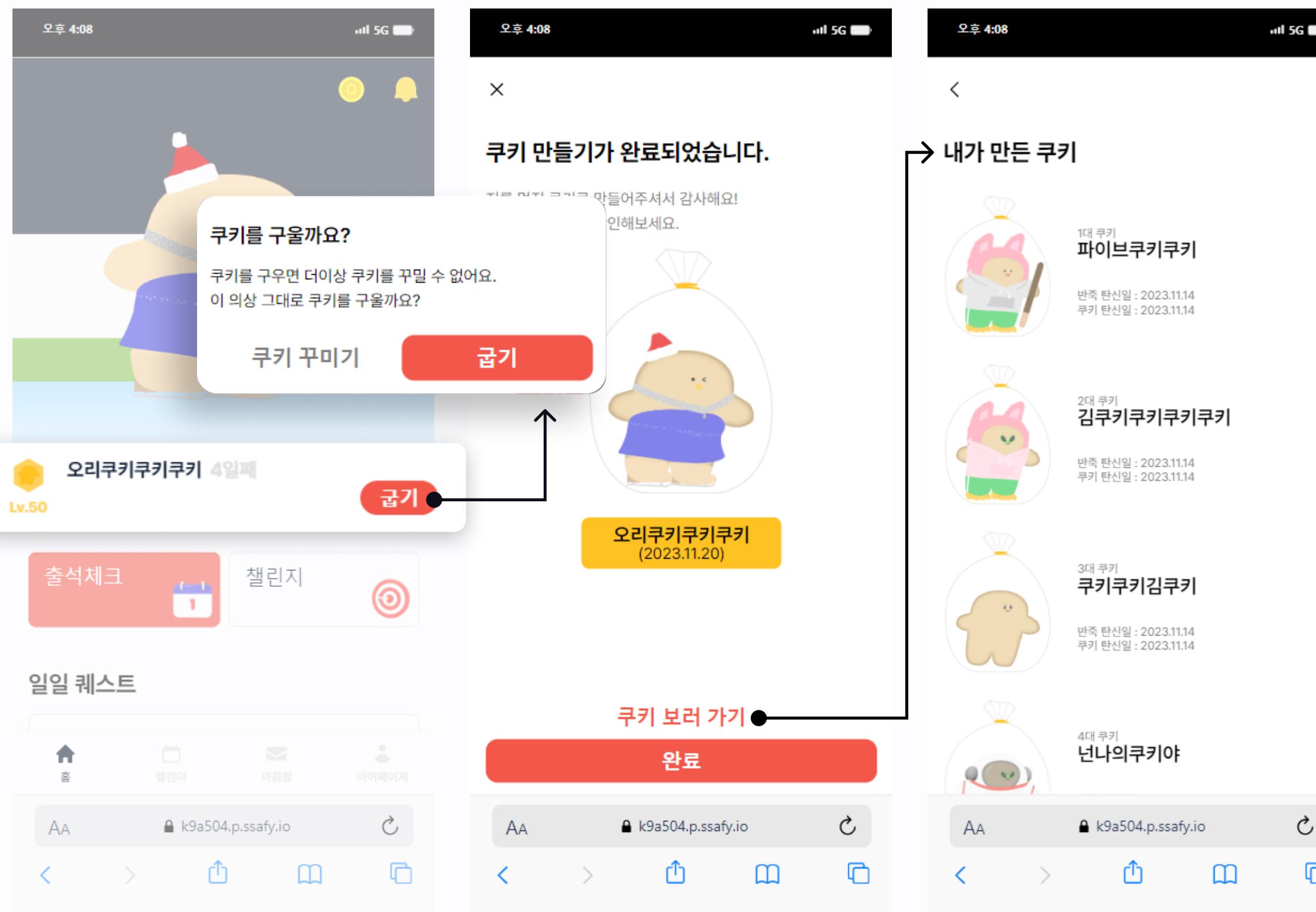
  itemApi.wearItemReq(wearItemId).then(() => {
    setQookie({ ...qookie, ...selectedItemList });
    navigate('/mypage');
  });
};
```



캐릭터 DOM 캡처

html2canvas 라이브러리를 활용하여 여러 이미지로 구성된 캐릭터를 하나의 png로 캡처하여 저장하는 방식을 구현했습니다.

AWS S3에서 받아온 이미지 url을 getProxyUrl api로 전송하여 CORS 에러를 해결하고, useRef로 하나의 DOM에 접근하여 bakePng라는 html2canvas를 활용한 util 함수를 통해 변환 후 다시 S3로 이미지를 업로드하는 과정을 구현했습니다.



```
const handleBakeClick = async () => {
  const res = await bakePng(divRef);
  if (res) {
    const imgFile = await converUrlToFile(res);
    qookieApi.bakeQookieReq(imgFile).then(() => {
      resetList();
      setIsBottomOpen((pre) => !pre);
    });
  }
};

const converUrlToFile = async (url: string) => {
  const res = await fetch(url);
  const data = await res.blob();
  const metaData = { type: `image/png` };
  return new File([data], `${props.name}.png`, metaData);
};

const getS3UrlHandler = async (url: string) => {
  return await qookieApi.getProxyUrl(url);
};
```



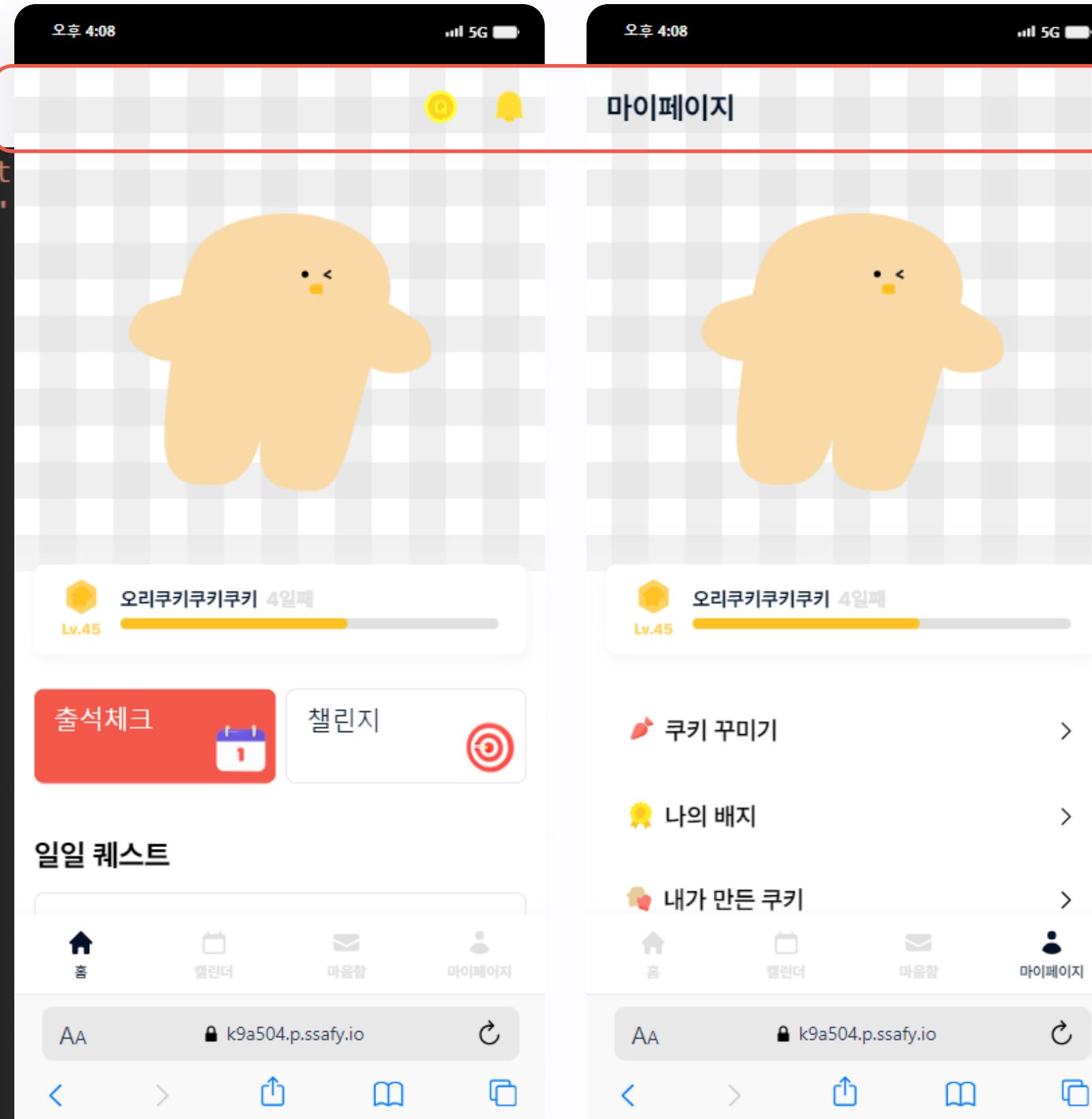
화면 구현 및 컴포넌트 구현

```
import { StoryObj, Meta } from '@storybook/react'
import { BrowserRouter } from 'react-router-dom'
import Header from './';

const meta: Meta<typeof Header> = {
  component: Header,
  decorators: [
    (Story) => (
      <BrowserRouter>
        | <Story />
      </BrowserRouter>
    ),
  ],
  title: 'Header',
};

export default meta;
type Story = StoryObj<typeof Header>;

const Template: Story = {
  argTypes: {
    page: {
      type: 'string',
      control: 'select',
      options: ['tab', 'home', 'other'],
      description: '페이지',
    },
  },
  render: (args) => <Header {...args} />,
};
```

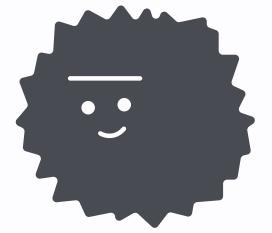


구현 내용

- react PWA로 홍 화면과 마이페이지 뷰 구현
- 아토믹 디자인을 접목하여 공통 컴포넌트 개발
- 쿠키 캐릭터의 상태를 보여주고 상점, 옷장을 통한 꾸미기 기능 구현
- 장바구니를 사용한 아이템 구매 기능 구현
- html2canvas를 활용한 DOM 캡처 구현
- axios를 활용한 api 통신 구현
- recoil을 통한 쿠키 상태 데이터 저장 및 수정
- Figma를 사용한 와이어 프레임 및 GUI 구현
- Cinema 4d를 활용한 3D 아이콘 디자인
- 어도비 일러스트레이터를 활용한 쿠키 캐릭터 아이템 제작

프로젝트 리뷰

- 프론트 인원이 많아서 Merge Request를 등록할 때 서로 리뷰어로 지정하고 코드리뷰를 진행하면서 바로 리팩토링을 진행 할 수 있었음
- CORS 에러나 많은 api response 수신 등 백엔드와 협력하여 문제를 해결하는 과정에서 팀과 함께하는 성장의 가치를 학습함
- 구조화된 아토믹 디자인 시스템을 적용하여 공통 컴포넌트와 페이지 컴포넌트를 분리하여 관리할 수 있어서 코드가 많아져도 쉽게 찾을 수 있음



방글

독립 출판물을 사랑하는 작가와 독자를 위한 블록체인 기반 전자 서점 플랫폼, 방글

[창작 프로젝트](#)
[블록체인](#)
[전자책 서점 플랫폼](#)
[개발](#)

BACKGROUND

독립 출판물을 더 가까이

[GitHub 코드](#)

독립출판물이란 대형 출판사를 통한 통상적인 책 제작 방식이 아닌 개인이나 소그룹이 기획, 편집, 인쇄, 제본해서 출판한 책들을 의미합니다. 기존 독립 출판물의 낮은 접근성과 적은 재고 문제를 전자책으로 해결하고, 전자책 플랫폼의 서비스 종료에 따른 구매 데이터 상실의 문제를 블록 체인 기술을 활용해 보완하여 기획한 전자 서점 플랫폼입니다.

PERSONA

독립 출판물 작가



🤔 취미로 글을 적었는데, 출판하기가 어려워요

“제가 쓴 글을 독자들에게 쉽게 보여주고 싶어요.

되도록 저렴한 가격에 말이죠. 출판사의 노예로 더 살진 싫어요 ... !”

독립 출판물 독자



😎 서점에 가는 건 COOL 하지 않잖아요?

“다른 사람이 추천하는 책은 이제 다 읽어본 것 같아서,

이제는 새로운 책들을 읽고 싶어요. 하지만 어디서 찾아야하는지

모르겠는걸요? 게다가 요즘 서점 찾기 힘들어요!”

작업기간

2023. 08 ~ 2023.10 (7주)

기여도

6인 팀 프로젝트 (프론트 2, 백 3, 풀스택 1)
(UX/UI 90%, 프론트엔드 40%)

프로그램

피그마, React, Next.js, Recoil,
storybook, SpringBoot 외

The image displays two screenshots of the Fanggeol mobile application. The left screenshot shows the main landing page with a dark theme, featuring a central lamp icon and the slogan '방에서 쓴 글이 모두의 블록 안으로'. Below the slogan is a text quote from a writer: '설명 길게 주저리 주저리 포인트는 감성을 담아서 열심히 적어보면 아주 좋을 것 같습니다. 저는 감성이 없어서 항상 이런 식으로 미루는 걸 좋아해요 ㅎㅎ'. A large green '시작하기' button is at the bottom. The right screenshot shows a detailed view of a book titled '90년생이 온다' by Fanggeol, published on 2023.09.06. The page includes the book's cover image, a brief description ('간단한, 병맛, 솔직함으로 기업의 흥망성쇠를 좌우하는'), and user interaction metrics like '구매하기' (10 면지), '대여하기' (1 면지), and a 4.0 rating from 3 reviews. The overall design is clean and modern, emphasizing a dark aesthetic.

기술 스택 및 선택 이유



NextJS : 서버사이드 렌더링으로 검색 페이지 구현에 적합하고, 스켈레톤 UI가 없어도 사용 가능

Storybook : 개별 컴포넌트 UI를 관리하고, 옵션을 통한 다양한 변화 확인이 용이

Styled-component : 변수를 활용한 css 구현이 쉽고, 코드 재사용이 용이함

Recoil : react hook 과 같이 쉬운 사용이 가능하고, 비동기 처리를 위한 미들웨어가 필요없음

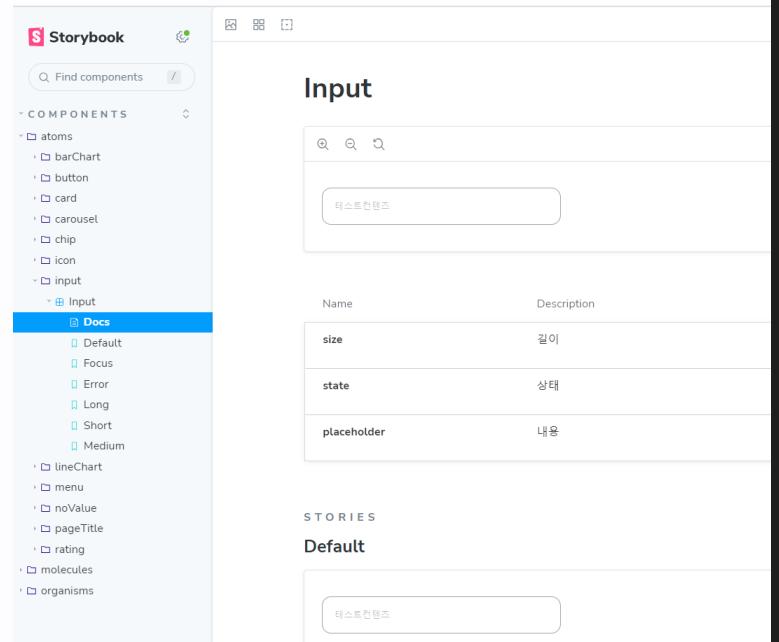
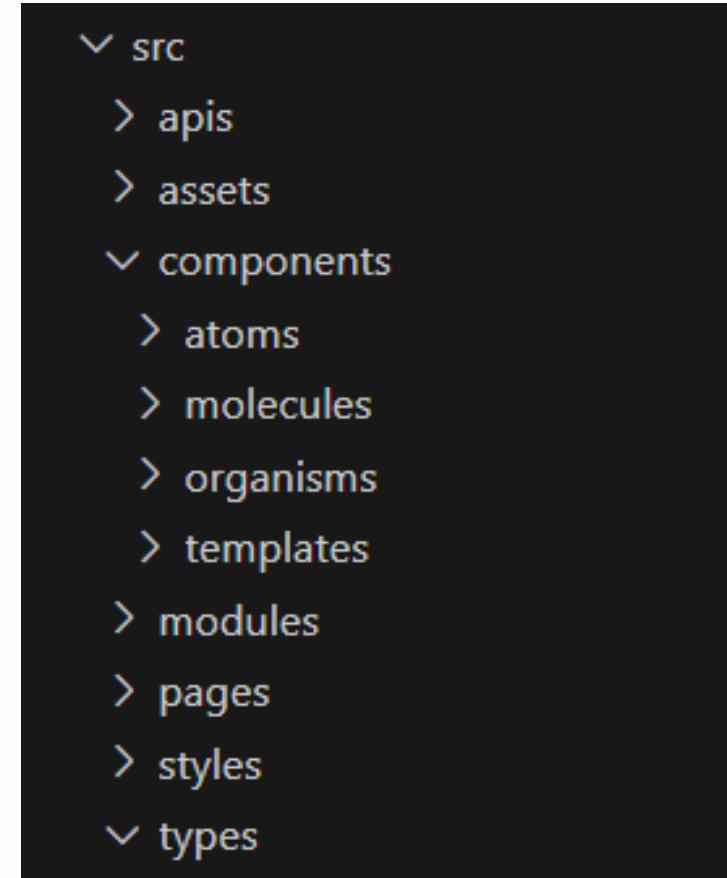
구현 내용

- React와 Next JS를 활용한 웹페이지 개발
- **Next JS**를 활용한 서버사이드 렌더링 구현
- TypeScript 를 활용한 타입 오류 및 가독성 관리
- state 관리 및 데이터 전달 활용을 위한 Recoil 사용
- Storybook 을 활용한 독립적인 UI 컴포넌트 개발
- 아토믹 디자인 방식으로 컴포넌트 재사용

아토믹 디자인과 스토리북

atom, molecule, organism, template, page 순으로 작은 컴포넌트를 재조합한 컴포넌트를 생성하며 불필요한 코드의 중복을 줄이고, 스토리북으로 독립적인 UI 컴포넌트를 개발했습니다.

아토믹 디자인과 스토리북 활용



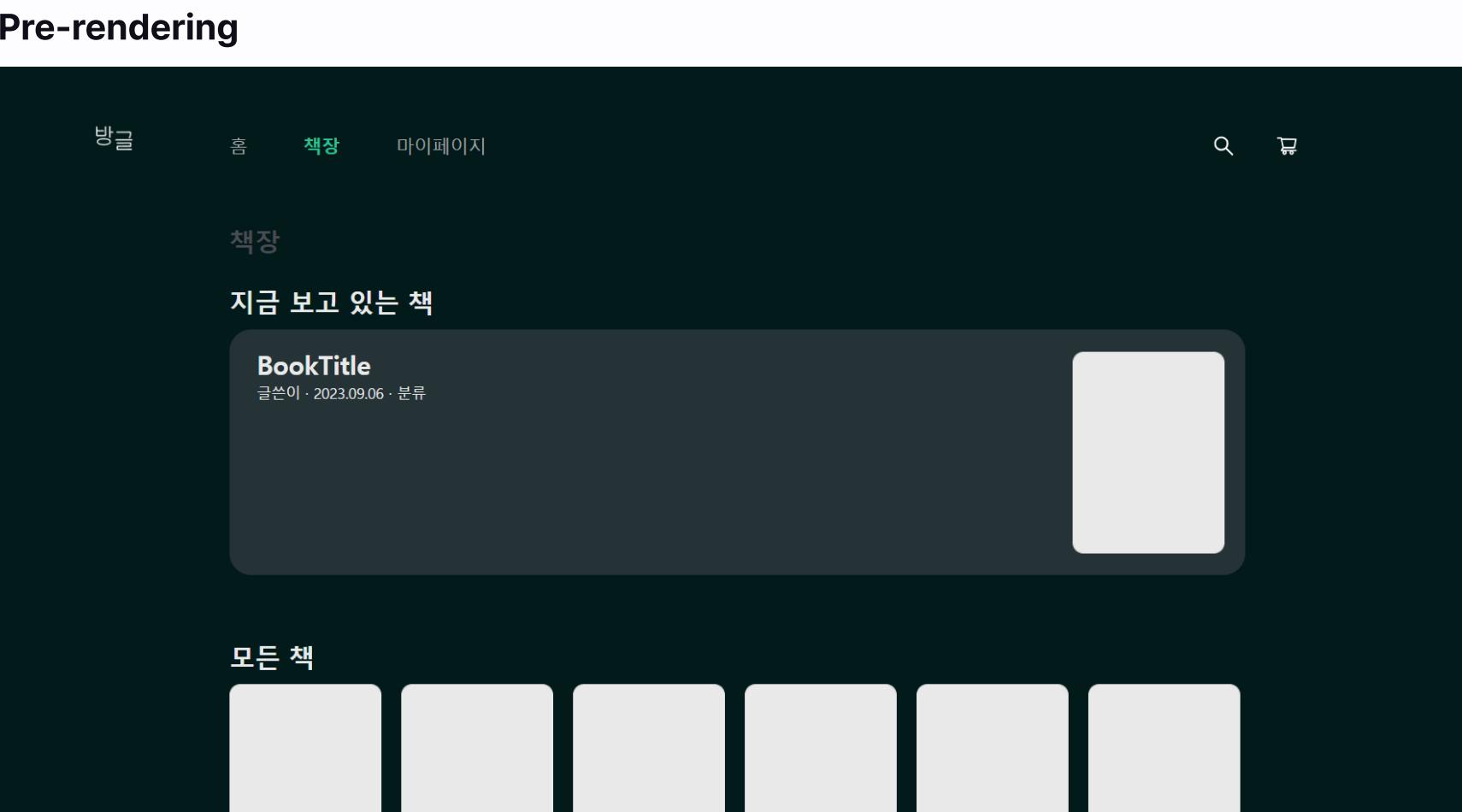
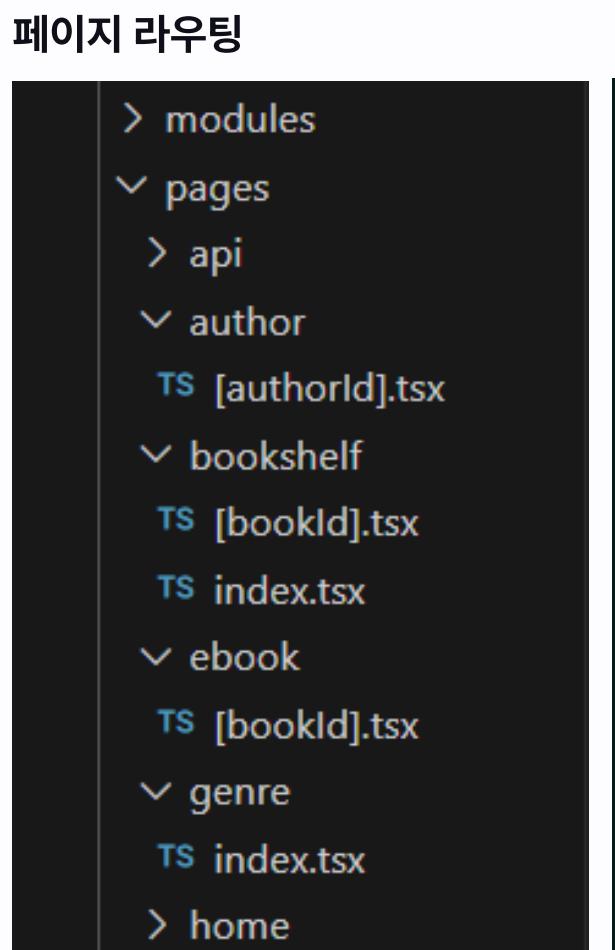
```

type Story = StoryObj<typeof Input>

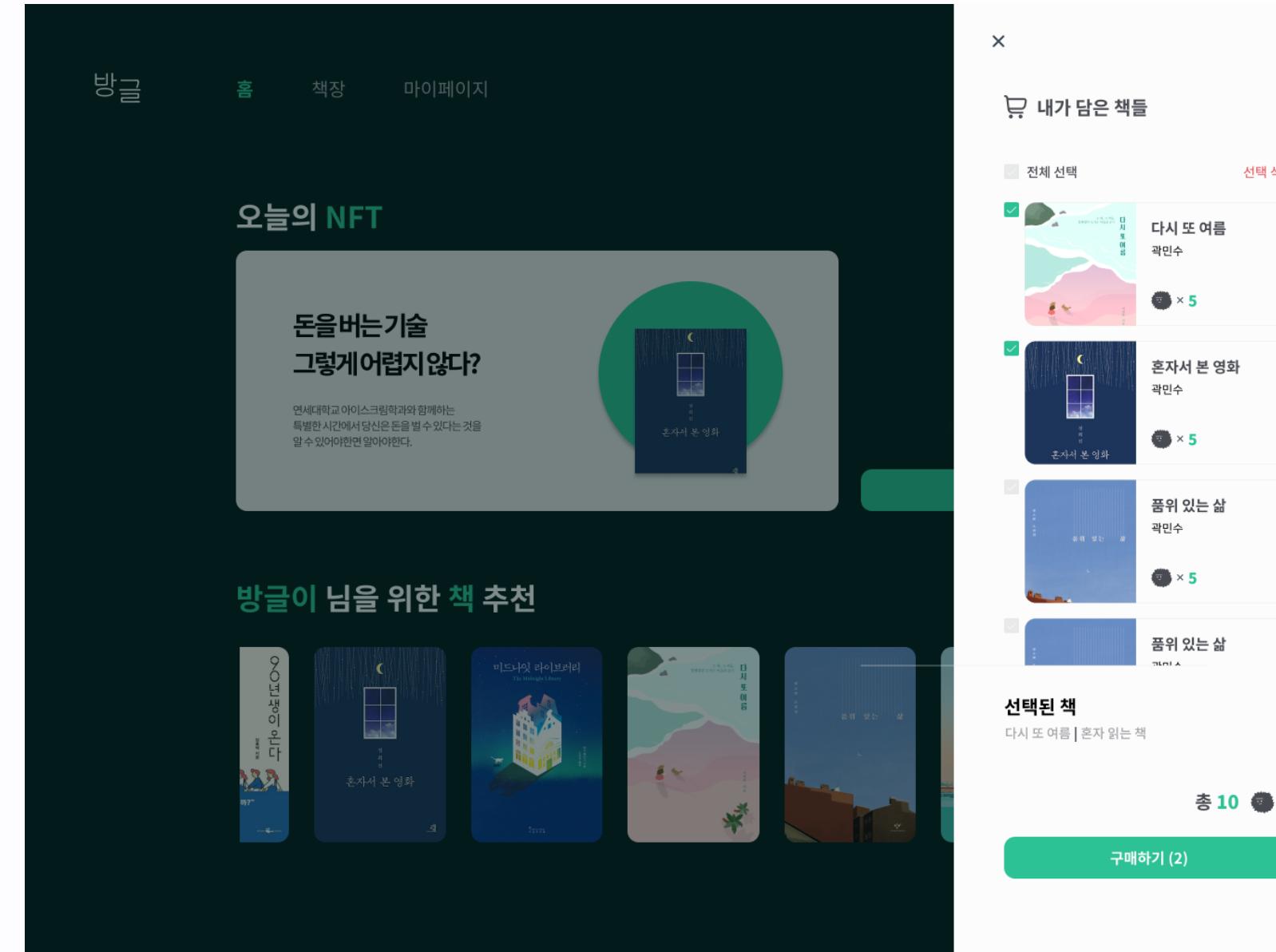
const Template: Story = {
  name: 'Default',
  argTypes: {
    size: {
      type: 'string',
      control: 'select',
      options: ['short', 'medium', 'default', 'long'],
      description: '길이',
    },
    state: {
      type: 'string',
      control: 'select',
      options: ['default', 'focus', 'error'],
      description: '상태',
    },
    placeholder: {
      type: 'string',
      description: '내용',
    },
  },
  render: (args) => <Input {...args} />,
}
  
```

Next JS 와 서버사이드 렌더링

Next JS를 사용하여, 별도의 라우팅 설정 없이 페이지를 제작하였으며, pre-rendering 방식에 따라서 스켈레톤 UI를 별도로 제작하지 않고도 적용했습니다. hydration 오류를 알고 수정하였습니다.



쿠키를 활용한 장바구니 구현



프로젝트 성과

- any 타입의 사용을 줄이고 Object 타입을 지정해줌으로서 오류를 줄임
- 아토믹 디자인을 활용해서 불필요한 코드 사용을 줄이고 재사용성을 높임
- 스토리북을 활용해서 컴포넌트를 문서화 함
- 쿠키를 활용한 장바구니 구현을 통해, 단권 바로 구매와 여러권 함께 구매 기능을 구현함
- recoil을 활용해서 get 과 set 을 통한 상태관리를 구현함

프로젝트 리뷰

- 짧은 기간의 토이 프로젝트여서, 블록체인의 단위인 먼지를 직접 충전하는 기능 구현을 하지 못함
- Next JS 의 이미지 최적화에도 불구하고 높은 이미지 퀄리티로 느리게 로딩 되는 경우가 있음
- 스토리북과 recoil 등 새로운 기술을 경험하는 기회가 됨
- atomic 디자인 시스템을 적용해서 컴포넌트화 하였지만, 페이지 내부에서만 쓰이는 컴포넌트와 공통 컴포넌트의 구분이 없어서 파일 구조화의 필요를 느낌



도깨비봇

시간과 장소의 제약 없이 효율적으로 공부할 수 있도록 도와주는 공부 습관 형성 웹IoT 서비스, 도깨비봇

창작 프로젝트

웹 IoT

학습 도움 콘텐츠

개발



2023 삼성 청년 SW 아카데미
공통 프로젝트 우수상

GitHub 코드

Web 기술 명세서

BACKGROUND

요즘 온라인 스터디가 대세라던데?

집에서 혼자 공부를 하다보면 집중력을 잃는 경우가 많아서, '카공족'이라는 단어가 생겨날 정도로 밖에서 공부하는 사람들을 많이 볼 수 있습니다. 요즘 1020 세대를 중심으로 스터디 카페나 **온라인 스터디 플랫폼** (열풍타, 구루미 등)이 유행하고 있습니다. 호손효과에 따르면 사람들은 사람은 누군가 자신에게 관심을 쏟을 때 평상시보다 훨씬 분발하고 노력한다는 이론에 따라 **공부를 도와주는 로봇**을 만들기로 결정했습니다.

IDEA

집중이 잘 안되는 집에서 공부를 도와주는 로봇이 있다면?

기술 스택 및 구현



구현 내용

- React 와 Scss 를 활용한 웹 어플리케이션 개발(SPA)
- TypeScript 를 활용한 타입 오류 및 가독성 관리
- state 관리 및 데이터 전달 활용을 위한 React-redux 사용

작업기간

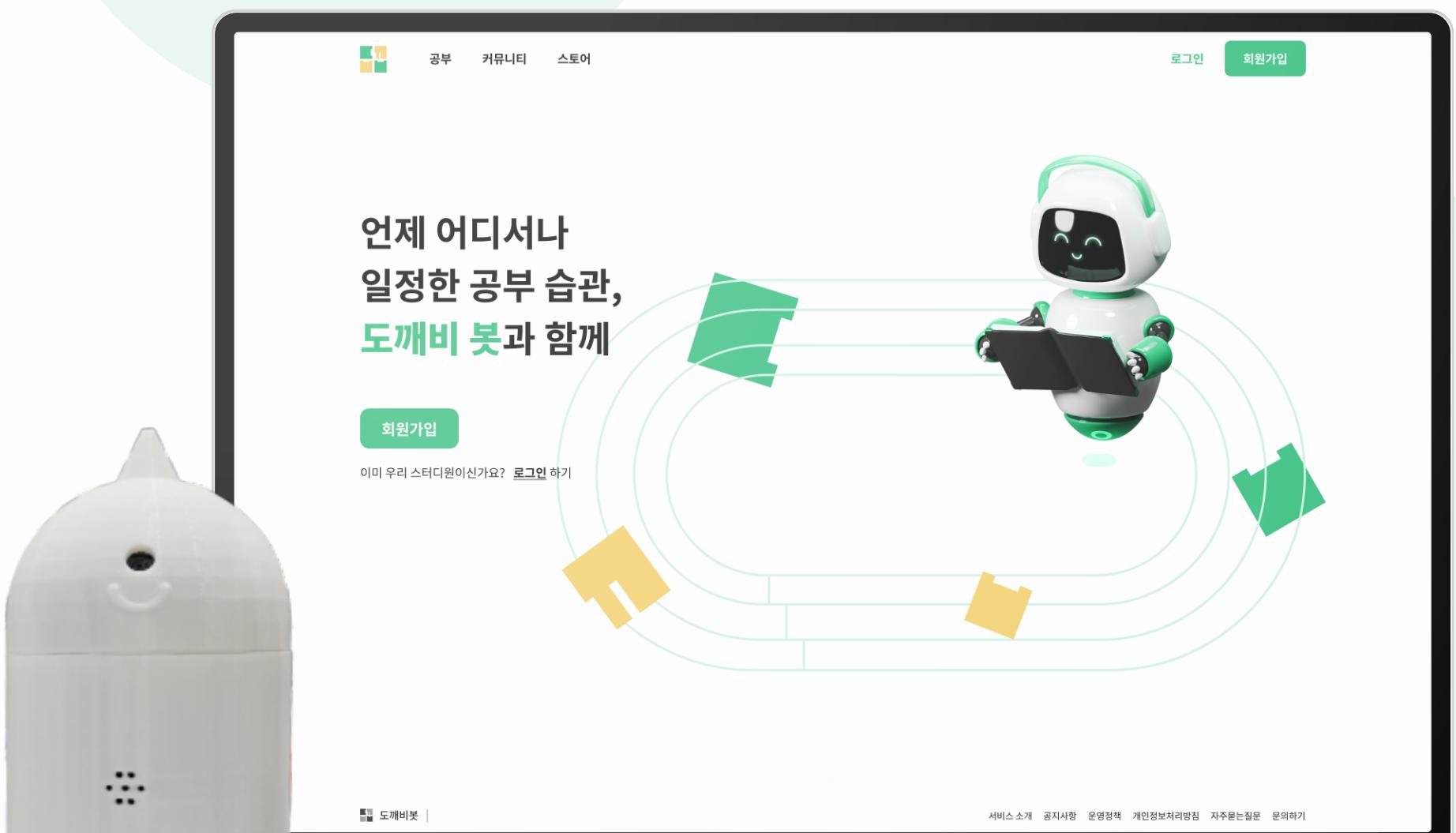
2023. 07 ~ 2023. 08 (7주)

기여도

6인 팀 (프론트 1, 백2, 임베디드3)
프론트엔드 개발 80%

프로그램

React, React-redux, 피그마
SCSS, TinkerCad 3d 등



기술 선택 이유

React : 컴포넌트 기반 라이브러리로 다양한 라이브러리나 프레임워크의 활용이 자유로움
react-redux : 상태 관리를 용이하게 할 수 있고, 과도한 prop 사용을 방지함

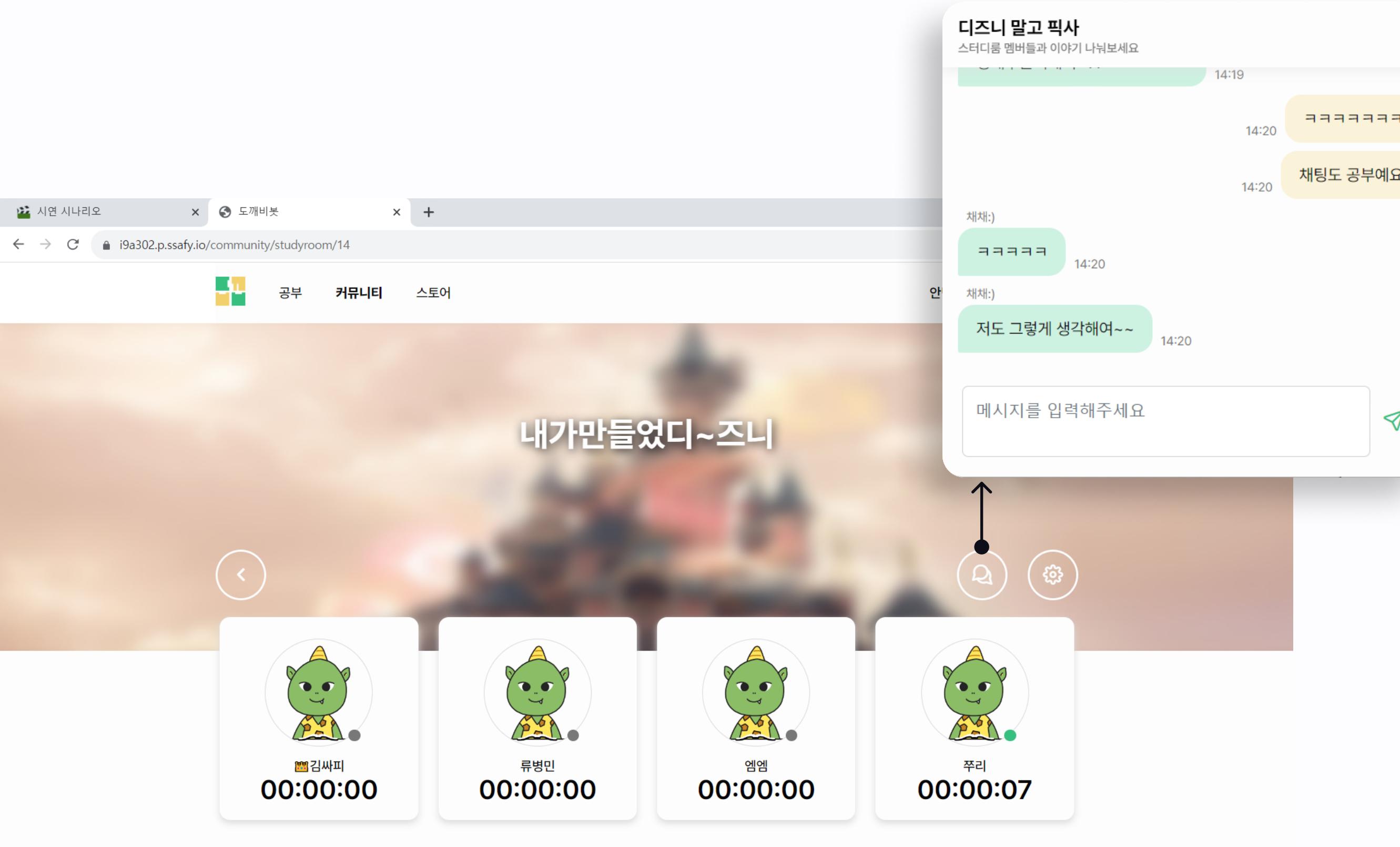
TypeScript : 데이터의 타입을 지정하여 타입 오류를 방지함

SCSS : css 기반으로 프로젝트 구조에 따라 작성하여 코드를 찾기 쉽고, 반복되는 속성을 유지보수하기 용이



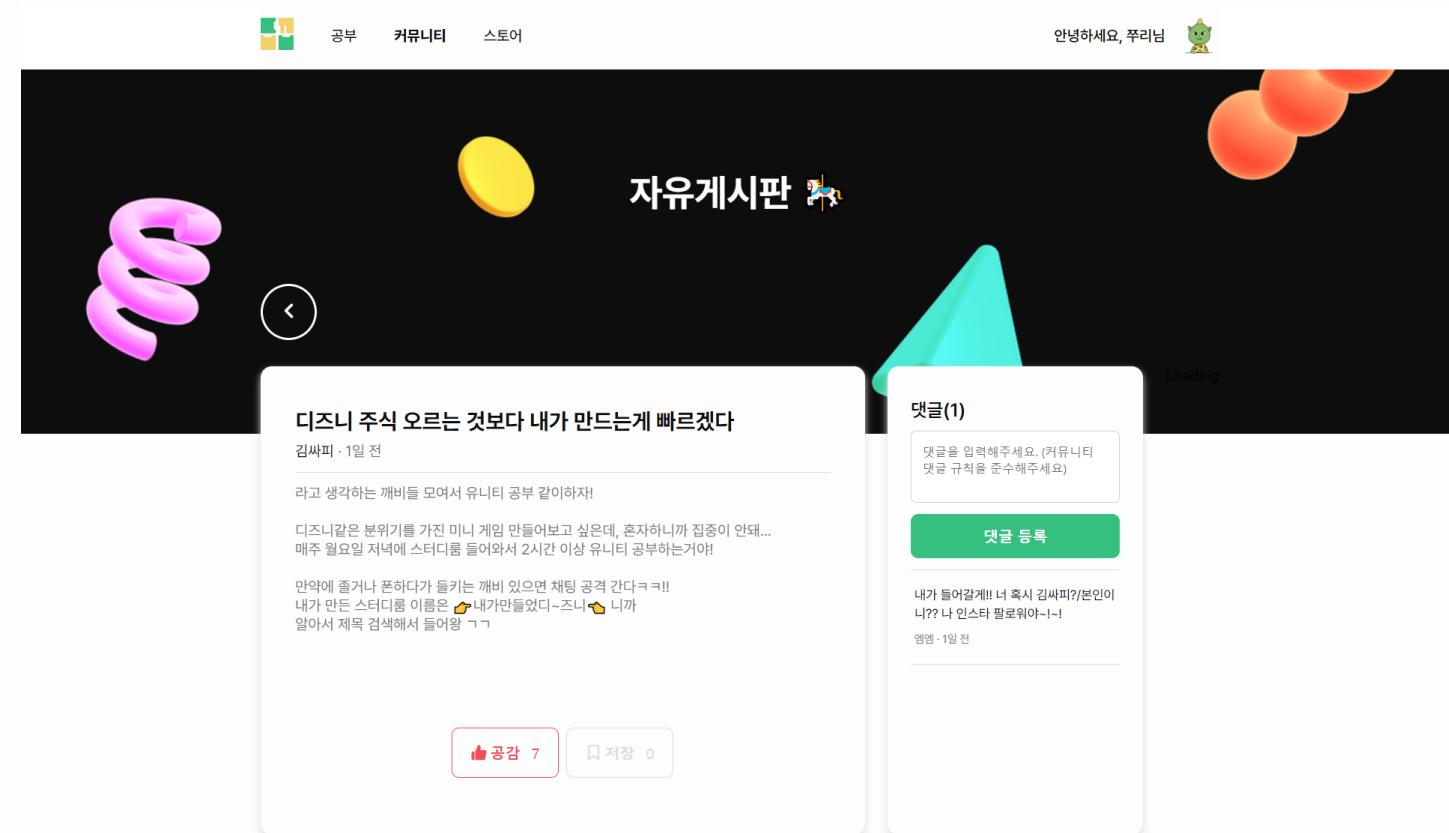
실시간 상태 공유 및 채팅 구현

Web socket과 useRef를 사용하여 실시간으로 다른 사용자가 어떤 상태로 공부를 하고 있는지 확인 할 수 있는 비대면 스터디룸을 구현했습니다. 또한 Stomp 를 통해 실시간 채팅을 구현했습니다.



REST API 와 CRUD 구현

REST API 를 기반으로 CRUD 를 구현하고 axios를 활용해서 http 통신을 구현했습니다.

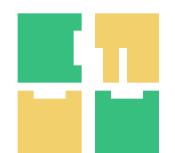


```
// 채팅방 메세지 조회
export const getChats = ({roomId}:{roomId:number}) => {
  return client().get(studyroomURL + `getChats/${roomId}`);
};

// 새로운 멤버 추가
export const updateStudyroomMember=({roomId}:{roomId:number})=>{
  return clientWithToken().post(studyroomURL + `enter/${roomId}/check`);
};

// 스터디룸 디폴트 이미지 등록
export const updateStudyRoomMemberTime = ({ roomId, time }: { roomId: number, time: number }) => {
  return clientWithToken().put(studyroomURL + `member/updateTime/${roomId}/${time}/check`);
};

// 스터디룸 삭제
export const deleteStudyroom=({roomId}:{roomId:number})=>{
  return clientWithToken().delete(studyroomURL + `delete/${roomId}/check`);
};
```



전체 Web View 구현

스터디룸 ■

우리는 고민했습니다.
언제 어디서나
집중하는 방법 없을까?

온라인으로 공부하고 있는 모습을 아이콘으로 실시간 공유

stomp를 활용하여 실시간 채팅 기능을 구현함

React.lazy를 react-router 라이브러리를 사용해서 라우트 기반 코드 분할 설정함

쭈리님을 위한 추천 스터디룸

“깨비야, 공부시작”
공부 습관 기록하고
분석받아보세요

전체 스터디룸

구현 내용

- Web-socket 을 활용하여 비대면으로 스터디룸에 입장한 상황 및 공부하고 있는 모습을 아이콘으로 실시간 공유
- stomp를 활용하여 실시간 채팅 기능을 구현함
- React.lazy를 react-router 라이브러리를 사용해서 라우트 기반 코드 분할 설정함

프로젝트 성과

- React 어플리케이션 구조를 설계함
- IoT 기계와 원활한 데이터 통신을 통해 그래프 및 분석을 제공함
- react 와 typescript 등 새로운 기술을 통해 프로젝트를 구현함
- web socket 등을 활용해서 실시간 기술을 구현함
- Rest API 를 사용해서 CRUD 작업을 진행함

프로젝트 리뷰

- TypeScript 사용 시, Object 타입을 지정할 때 any 타입을 많이 사용함
- 프론트엔드 담당이 혼자여서 페이지 레이아웃 및 GUI 구현에 힘을 많이 쓰느라 web socket이나 chart.js 같은 기술 적용을 깊게 공부하지 못한 아쉬움이 있음
- 다른 분야인 백엔드, 임베디드 팀원들과 JIRA, Notion 을 활용해서 원활하게 소통함

건강한 공부 습관을 찾고 있나요?

지금 시작하기

2023 PORTFOLIO

화면 아래에 논리를 펼쳐주는
프론트엔드 엔지니어 조원주입니다

감사합니다.



온라인 비공개



dnjswn9178@hanmail.net



<https://github.com/twoju>