

포트폴리오 명세서

권 동 환

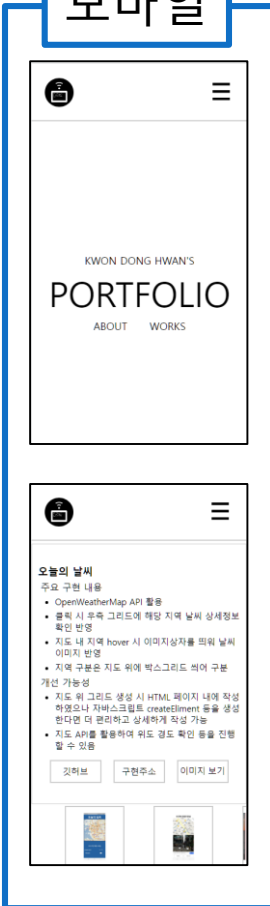
2023-02-22

1. 포트폴리오 홈페이지 디자인 설계

데스크탑

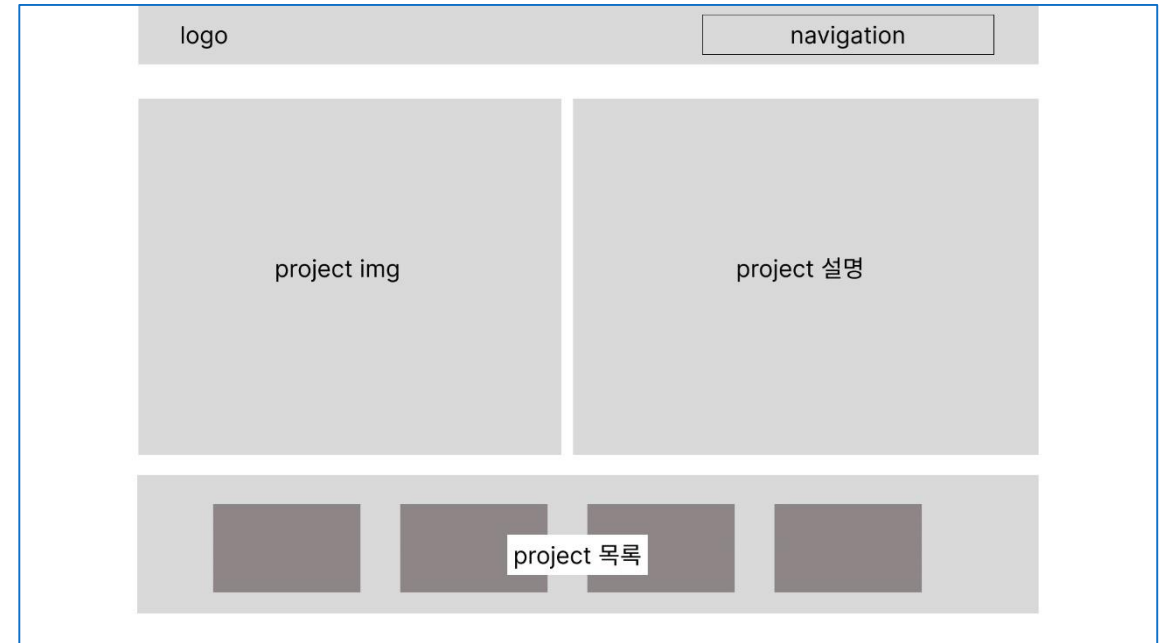


모바일



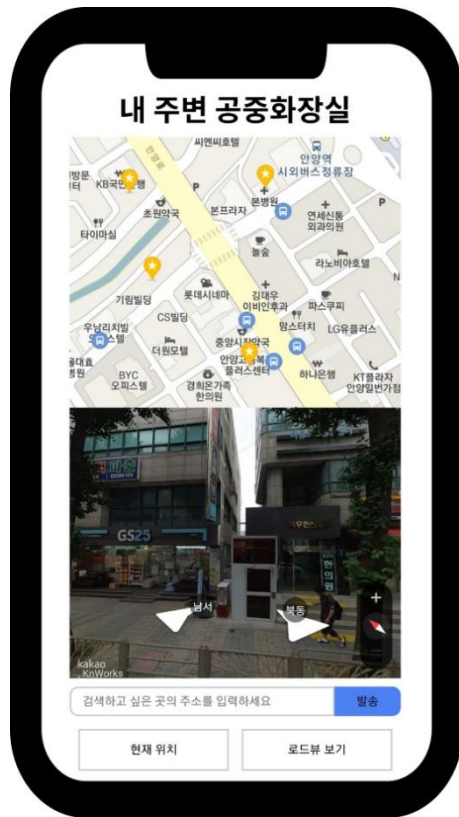
깃 허브

홈페이지



2-1. 프로젝트 소개 : 공중화장실 검색 앱

공중화장실 검색 앱



공중화장실 검색 : [깃허브](#) [바로가기](#)

주요 구현 내용

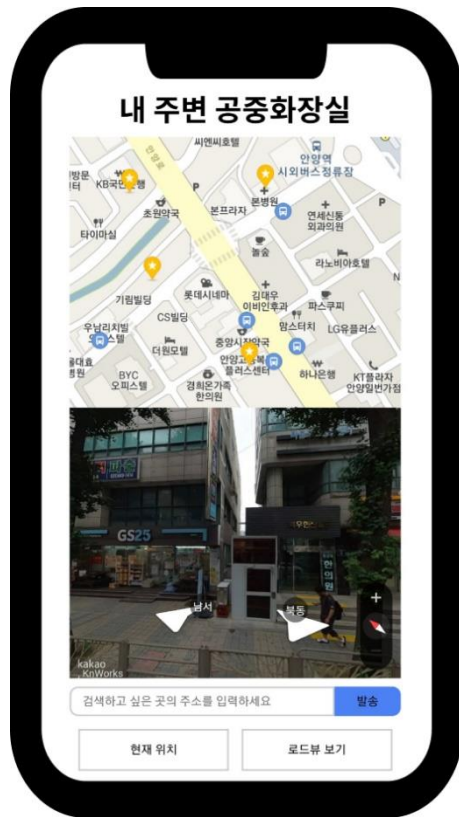
- 공공데이터 활용 : 경기도 공중화장실 위치 데이터
- 카카오 MAP API 활용

개선 가능성

- 추가적인 자료를 수집, 전국 기준으로 진행
- 지역 확인 시 주소가 아닌 키워드로 검색 가능토록 방법 모색

2-1. 프로젝트 소개 : 공중화장실 검색 앱

공중화장실 검색 앱



공중화장실 검색 : 프로젝트 정보

```
{
  "name": "toilet_react",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-kakao-maps-sdk": "^1.1.6",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
```

```
"extends": [
  "react-app",
  "react-app/jest"
],
"browserslist": {
  "production": [
    ">0.2%",
    "not dead",
    "not op_mini all"
  ],
  "development": [
    "last 1 chrome version",
    "last 1 firefox version",
    "last 1 safari version"
  ]
}]}
```

2-1. 프로젝트 소개 : 공중화장실 검색 앱

공중화장실 검색 앱



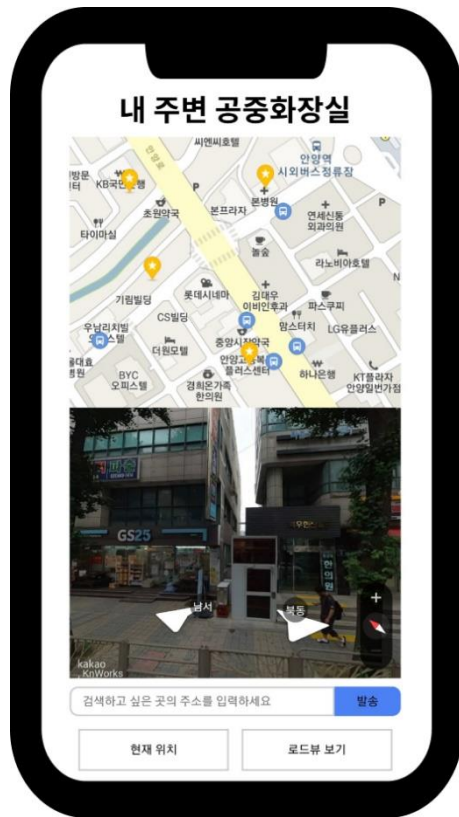
공중화장실 검색 : 주요 문제 해결

```
41 + let arr = [];  
42 + kyunggi2.map((e) => {  
43 +   if (e.REFINE_LOTNO_ADDR.indexOf(region) > -1) {  
44 +     arr.push(e);  
45 +   }  
46 + });  
47 + setMarkers(arr);  
48 + }, [region]);
```

- 지역 이동 등을 진행할 때(지역 검색, 현재위치 이동) 로딩이 오래 걸리는 문제 발생
- 화장실 위치 정보(json 파일) 로드 시 해당 지역 주변의 화장실 위치 정보만 불러오도록 조정

2-1. 프로젝트 소개 : 공중화장실 검색 앱

공중화장실 검색 앱



공중화장실 검색 : 배포

- 실행 어플리케이션 Package
 - Expo를 활용 WebView 방식으로 패키징

```
{
  "name": "toilets_around_me",
  "version": "1.0.0",
  "main": "node_modules/expo/AppEntry.js",
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web"
  },
  "dependencies": {
    "expo": "~47.0.12",
    "expo-permissions": "^14.0.0",
    "expo-status-bar": "~1.4.2",
    "react": "18.1.0",
    "react-native": "0.70.5",
    "react-native-webview": "11.23.1",
    "expo-location": "~15.0.1"
  },
}
```

```
"dependencies": {
  "expo": "~47.0.12",
  "expo-permissions": "^14.0.0",
  "expo-status-bar": "~1.4.2",
  "react": "18.1.0",
  "react-native": "0.70.5",
  "react-native-webview": "11.23.1",
  "expo-location": "~15.0.1"
},
"devDependencies": {
  "@babel/core": "^7.12.9"
},
"private": true
```

2-1. 프로젝트 소개 : 공중화장실 검색 앱

공중화장실 검색 앱



공중화장실 검색 : 배포 문제 해결

```
"android": {
  "adaptiveIcon": {
    "foregroundImage": "./assets/adaptive-icon.png",
    "backgroundColor": "#FFFFFF"
  },
  "package": "com.kwondh.toilets_arounds_me",
  "permissions": ["ACCESS_FINE_LOCATION"]
},
"web": {
  "favicon": "./assets/favicon.png"
},

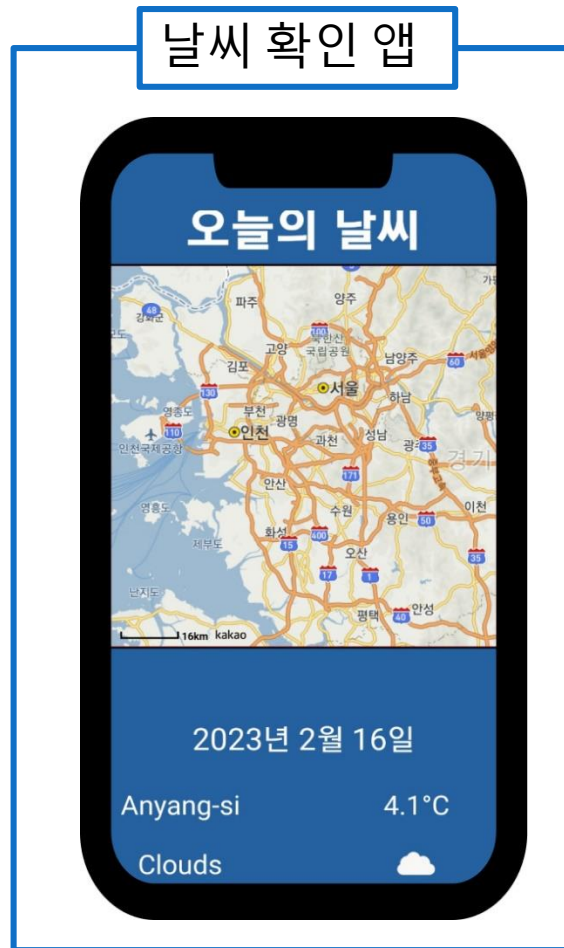
```

```
import { StatusBar } from "expo-status-bar";
import Constants from "expo-constants";
import { Platform, StyleSheet, Text, View } from "react-native";
import { WebView } from "react-native-webview";
import * as Location from "expo-location";
// import { PermissionsAndroid } from "react-native";

export default function App() {
  return (
    <>
    <StatusBar
      barstyle={"dark-content"}
      backgroundColor={"#ccc"}
      translucent={false}
      hidden={false}
    />
  )
}
```

- 위치정보 확인 불가 문제 발생
 - "android" – "permissions"에서 "ACCESS_FINE_LOCATION" 추가
- 핸드폰의 상태표시줄과 어플리케이션의 겹침문제 발생
 - Status Bar 태그 추가 및 설정 진행

2-2. 프로젝트 소개 : 날씨 확인 앱



오늘의 날씨 : 깃허브

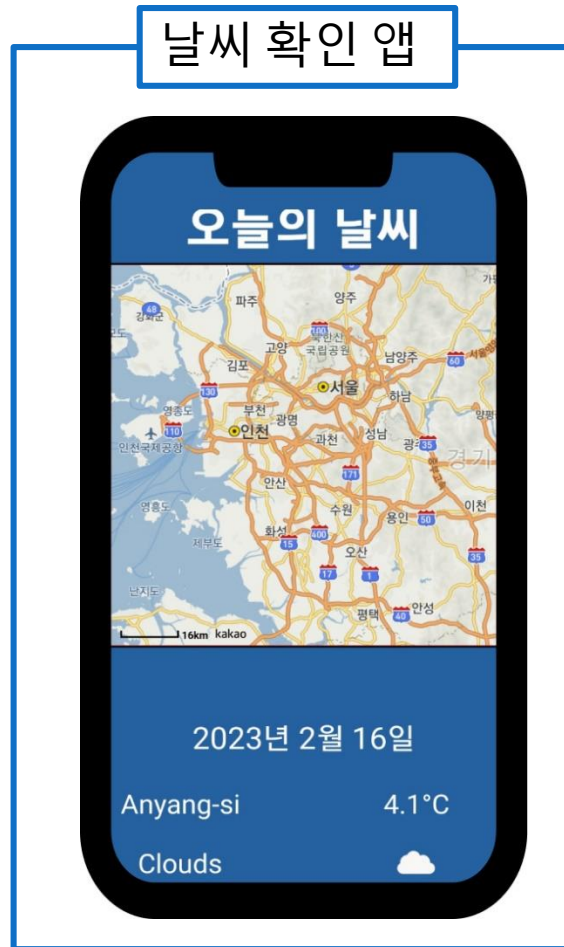
주요 구현 내용

- 접속 시 상단 지도를 통해 현재 위치 표시
- 하단 정보 창에 현재 온도와 날씨 정보 표시
- 지도 내 특정 위치 클릭(탭) 시 해당 지역의 날씨 확인

개선 가능성

- native의 기능들을 활용할 수 있도록 방향 설계 필요
- 지도 클릭 외에 지역 선택 혹은 검색할 수 있는 기능 반영

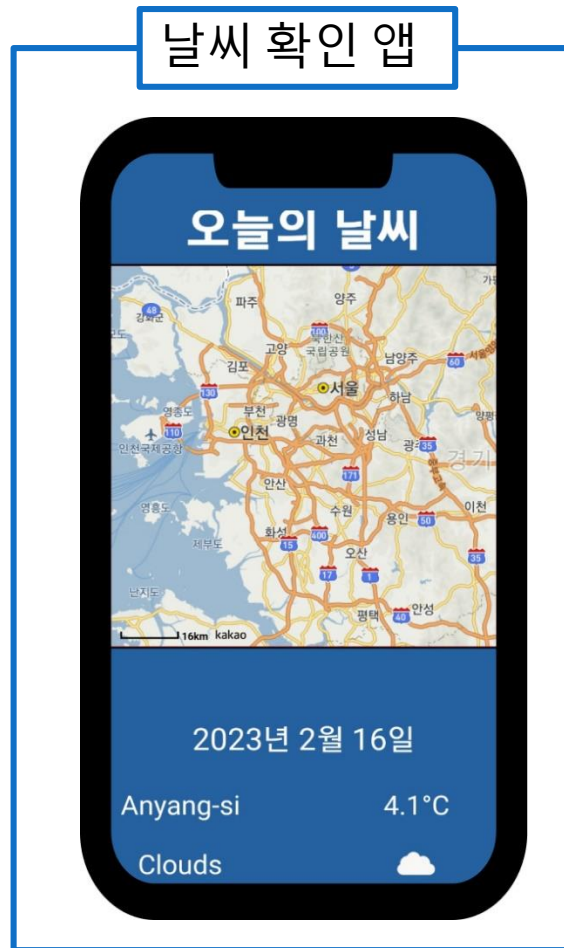
2-2. 프로젝트 소개 : 날씨 확인 앱



오늘의 날씨 : 프로젝트 정보

```
{  
  "name": "weather_app",  
  "version": "1.0.0",  
  "main": "node_modules/expo/AppEntry.js",  
  "scripts": {  
    "start": "expo start",  
    "android": "expo start --android",  
    "ios": "expo start --ios",  
    "web": "expo start --web"  
  },  
  "dependencies": {  
    "expo": "~47.0.12",  
    "expo-status-bar": "~1.4.2",  
    "react": "18.1.0",  
    "react-native": "0.70.5",  
    "react-native-webview": "11.23.1"  
  },  
  "devDependencies": {  
    "@babel/core": "^7.12.9"  
  },  
  "private": true  
}
```

2-2. 프로젝트 소개 : 날씨 확인 앱

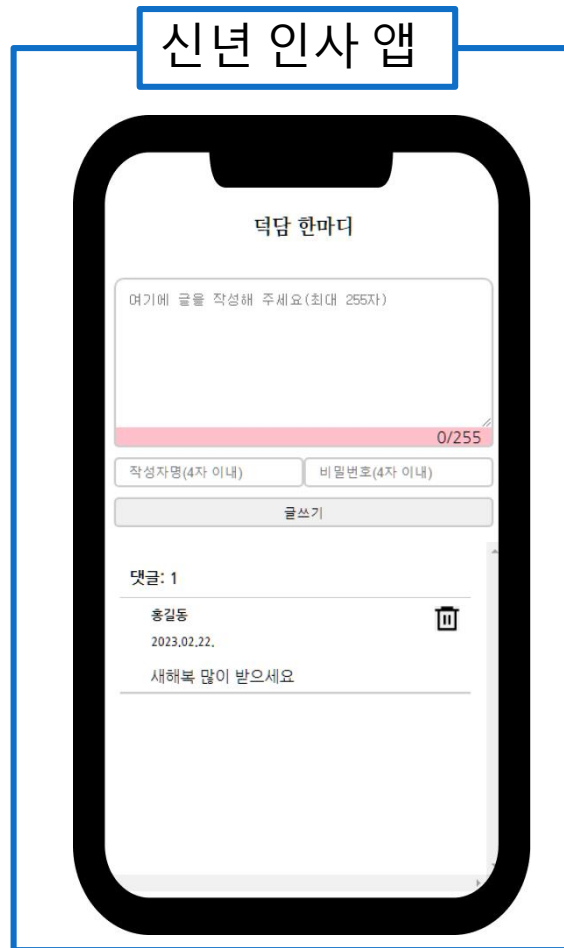


오늘의 날씨 : 주요 문제 해결

```
<WebView
  source={{
    uri: "https://web-kakao-422t0241bonwoan.gks12.cloudtype.app/",
  }}
  onMessage={(e) => {
    console.log(e.nativeEvent.data);
    let geoE1 = [];
    geoE1 = e.nativeEvent.data.split(",");
    console.log(geoE1);
    setLatLng([geoE1[0], geoE1[1]]);
  }}
/>
```

- React-Native를 활용함에 따라 카카오 지도를 불러오는데 어려움이 존재함
- WebView활용, onMessage 속성을 활용하여 통신을 통해 문제 해결

2-3. 프로젝트 소개 : 신년 인사 앱



신년 인사 앱 : [깃허브](#) [바로가기](#)

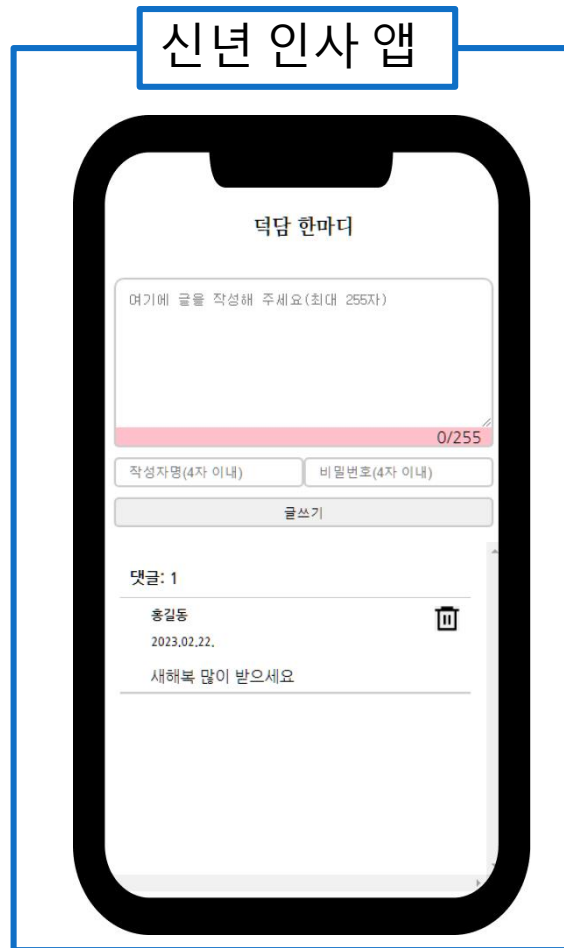
주요 구현 내용

- 글자 제한 255자, 작성자 명, 비밀번호 입력
- 글쓰기 및 삭제 시 본인 확인(비밀번호 확인)

개선 가능성

- 비밀번호 암호화 기능 추가 및 DB파일 구성을 통한 보안 정책 향상 필요

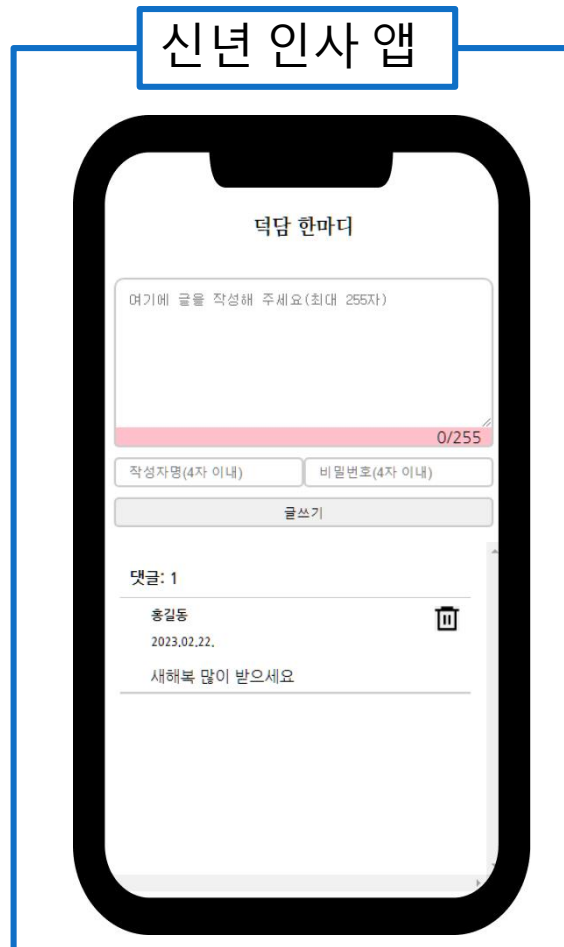
2-3. 프로젝트 소개 : 신년 인사 앱



신년 인사 앱 : 프로젝트 정보

```
{  
  "dependencies": {  
    "ejs": "^3.1.8",  
    "express": "^4.18.2",  
    "moment": "^2.29.4"  
  }  
}
```

2-3. 프로젝트 소개 : 신년 인사 앱



신년 인사 앱 : 댓글 삭제 시 비밀번호 입력 방법

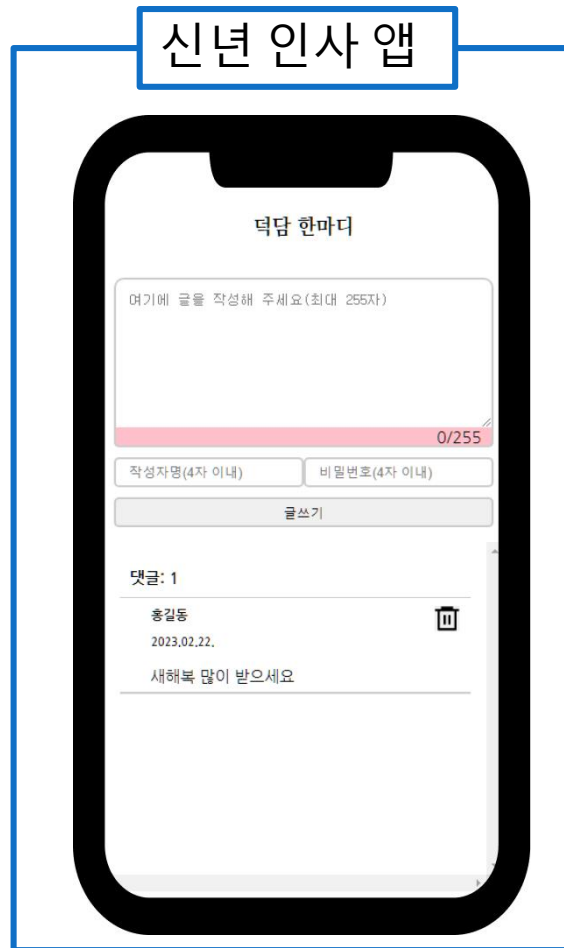
```
// 자바스크립트에서 form(hidden) 생성 후 비밀번호 제출

// visualEl.forEach(e=>{
  // e.addEventListener('click',i=>{
    //   let num = i.target.id;
    //   let repwd=prompt('비밀번호,');
    //   console.log(repwd);
    //   let fo = document.createElement('form');
    //   let inp = document.createElement('input');
    //   inp.setAttribute('type', 'hidden');
    //   inp.setAttribute('name', 'repwd');
    //   inp.setAttribute('value', repwd);
    //   fo.appendChild(inp);
    //   fo.setAttribute('method', 'post');
    //   fo.setAttribute('action', `/content/${num}`);
    //   document.body.appendChild(fo);
    //   fo.submit();

    // })
// })
```

- 삭제버튼 클릭 시 prompt 창 생성, 비밀번호를 입력
- 입력된 비밀번호를 서버로 발송(hidden form)
- 서버에서 비밀번호 일치 여부 확인 후 결과 발송

2-3. 프로젝트 소개 : 신년 인사 앱



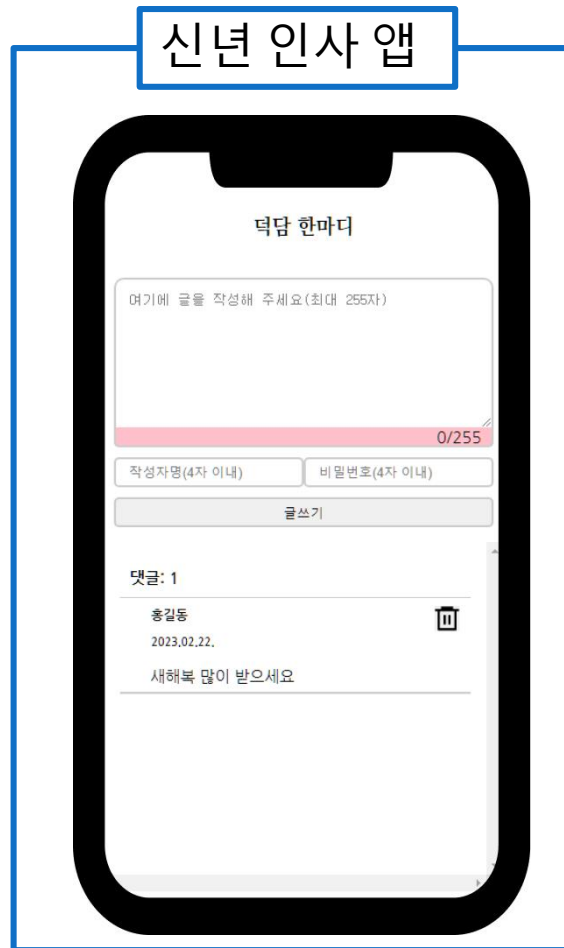
신년 인사 앱 : 댓글 삭제 시 비밀번호 입력 방법

//fetch로 비밀번호 확인 후 자바스크립트에서 구현

```
// visualEl.forEach(e=>{
//   e.addEventListener('click',i=>{
//     let num = i.target.id;
//     let repwd=prompt('비밀번호',);
//     fetch(`/remove/${num}`, {method : 'post'}).then((res)=>res.json()).then((data)=>{
//       if(data.pwds == repwd) {
//         formElArr[num].submit();
//       }else {
//         alert('비밀번호가 틀립니다.');
```

- 삭제 버튼 클릭 시 prompt창 생성, 비밀번호 입력
- 비밀번호를 post방식으로 요청(fetch 활용)
- 입력된 비밀번호와 서버에서 받아온 비밀번호를 비교하여 확인

2-3. 프로젝트 소개 : 신년 인사 앱



신년 인사 앱 : 댓글 삭제 시 비밀번호 입력 방법



- submit 버튼 외에 별도의 버튼을 만들어 놓고 비밀번호란과 submit버튼 숨김
- 삭제 버튼 클릭 시 submit버튼과 비밀번호란을 보임

2-4. 프로젝트 소개 : 팀 프로젝트1

팀 프로젝트 1



팀 프로젝트 1 : [깃허브](#) [바로가기](#)

주요 구현 내용

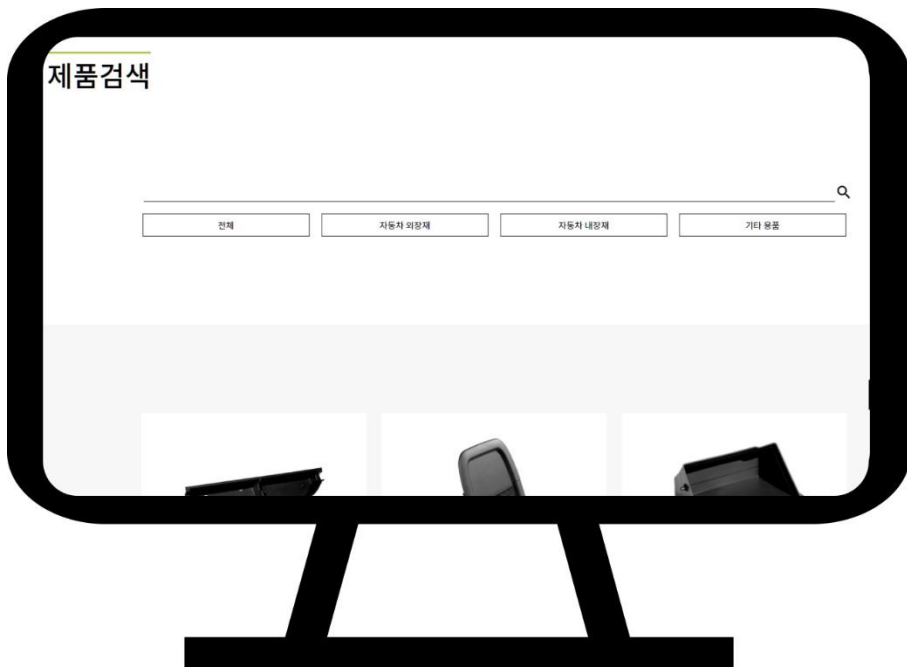
- 연구정보, 제품정보 페이지 작성
- HTML, CSS, JAVASCRIPT 활용 제품정보
제품정보 페이지 내 제품검색 및
정렬 기능 구현

개선 가능성

- Json파일 등을 활용 제품 목록을 외부에 두고 불러오는 형식이 더욱 효과적일 것

2-4. 프로젝트 소개 : 팀 프로젝트1

팀 프로젝트 1



팀 프로젝트 1: 제품 검색

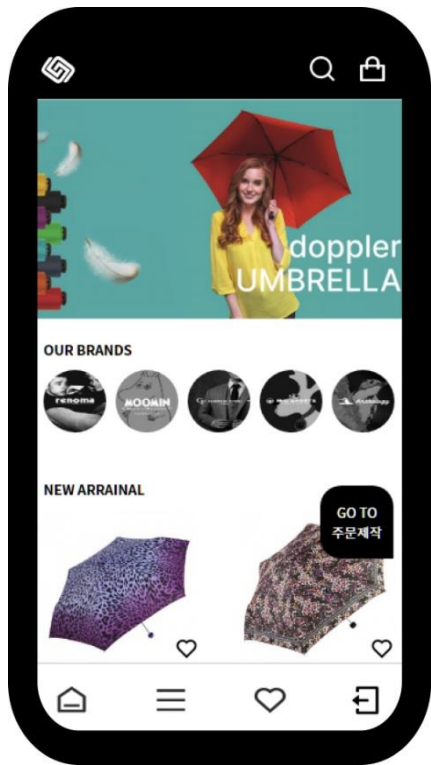
```
//검색
let itemArr = [];
for (i = 0; i < 23; i++) {
  itemArr.push($(".viewBox ul li").eq(i).text().toLowerCase());
}
console.log(itemArr[0]);
$(".search").on("click", search);
function search() {
  let result = itemArr.indexOf($(".textbox").val().toLowerCase());
  console.log(result);

  $(".viewBox ul").css({
    top: "0",
  });
  $(".viewBox .container ul").css("height", "1360px");
  $(".viewBox ul li").css({
    display: "none",
  });
  $(".viewBox ul li").eq(result).css({
    display: "block",
  });
  $(".viewBox ul .all").css({
    display: "block",
  });
  $(".dial .numbers p").css("display", "none");
  $(".dial .numbers p").eq(0).css("display", "block");
}
});
```

- 제품명들을 배열로 만듦
- Input태그에 검색하고자 하는 제품명 입력 및 Search버튼 클릭
- Input 값을 배열 위치에서 찾아 해당 제품만 출력

2-5. 프로젝트 소개 : 팀 프로젝트2

팀 프로젝트2



팀 프로젝트2 : 깃허브 바로가기

주요 구현 내용

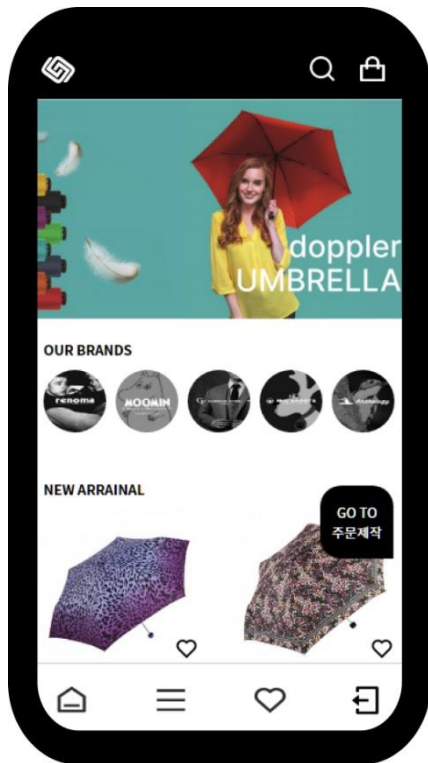
- sqlite를 활용한 데이터베이스 구축
- PRODUCT 페이지 작성 및 기능 구현
- 찜목록, 장바구니 임시 저장 기능 구현

개선 가능성

- 로그인관련 토큰, 웹세션 등을 활용하여 보안성을 강화할 수 있을 것
- 데이터베이스 설계 시 회원 개인정보와 회원의 관심 상품 목록을 이원화하여 관리하는 것이 효율성을 높일 수 있을 것

2-5. 프로젝트 소개 : 팀 프로젝트2

팀 프로젝트2



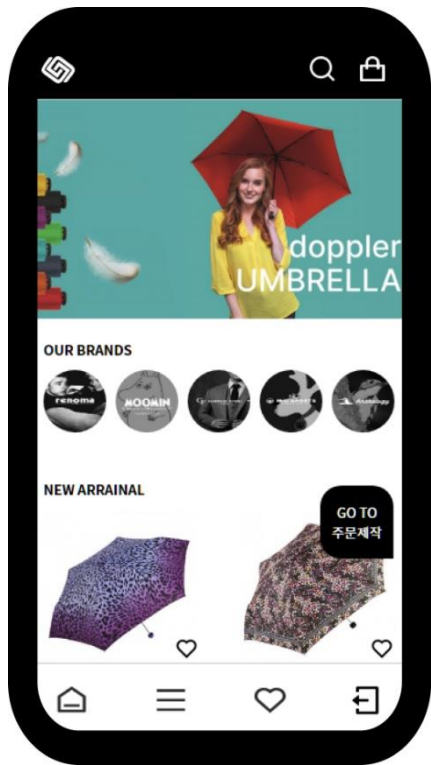
팀 프로젝트2 : 프로젝트 정보

```
{  
  "name": "ejs",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo W\"Error: no t  
est specifiedW\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",
```

```
  "dependencies": {  
    "cookie-parser": "^1.4.6",  
    "ejs": "^3.1.8",  
    "express": "^4.18.2",  
    "sequelize": "^6.28.0",  
    "sqlite3": "^5.1.4"  
  }  
}
```

2-5. 프로젝트 소개 : 팀 프로젝트2

팀 프로젝트2



팀 프로젝트2 : 주요 구현 사항

- ejs 템플릿을 활용한 SSR 구현
- local storage를 활용하여 제품 찜목록 구현
- session storage를 활용하여 로그인 전 장바구니 담기 구현
- 로그인 후 session storage의 내용을 db에 저장하고, 로그인 후 장바구니 담기는 데이터베이스에 저장
- 로그인 상태는 쿠키 발행을 통해 관리 진행

감사합니다.
