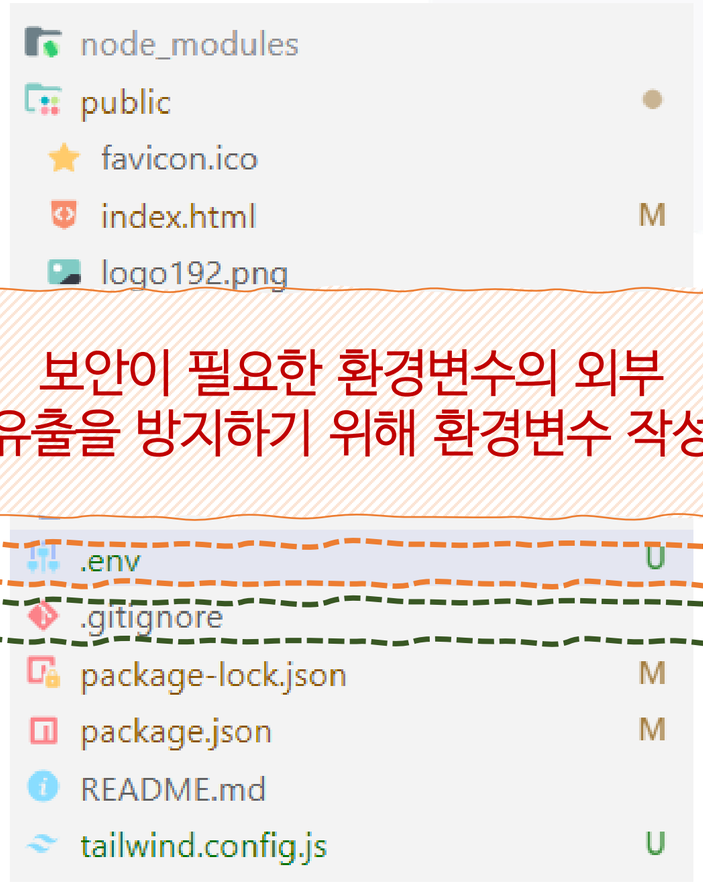
AI 활용 빅데이터분석
풀스택웹서비스 SW 개발자 양성과정

React

환경변수 설정



보안이 필요한 환경변수의 외부 유출을 방지하기 위해 환경변수 작성

.env 파일은 최상위 루트에 작성

환경변수명은 반드시 REACT_APP_으로 시작

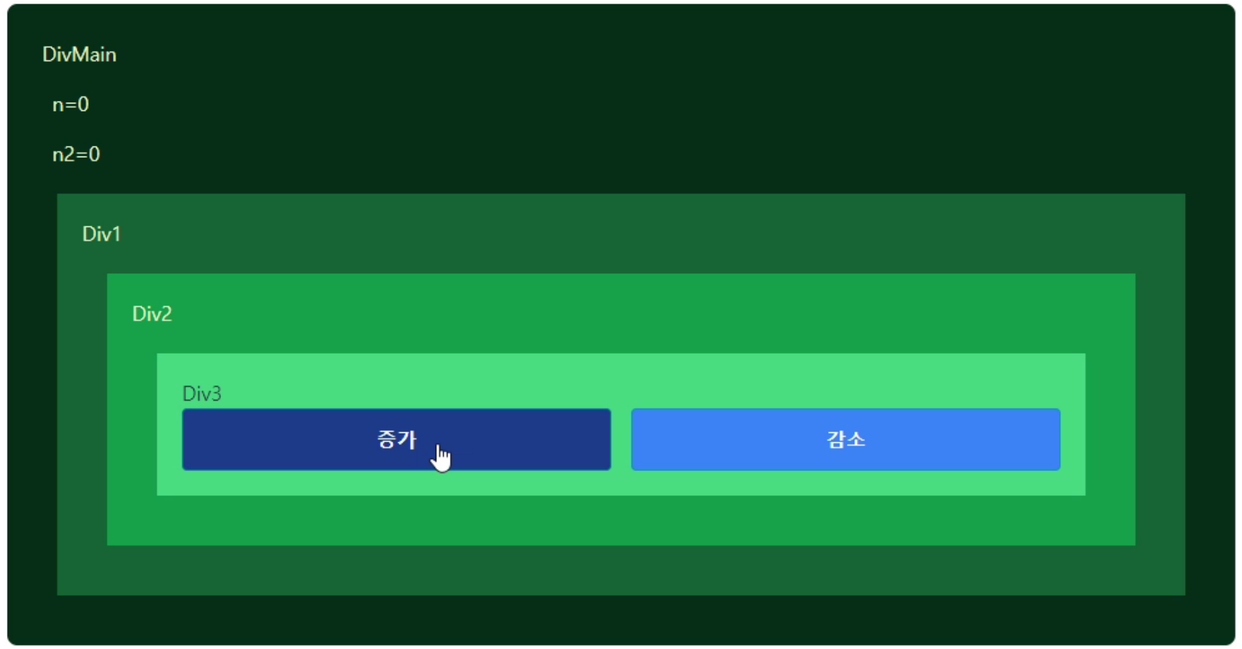
```
.env
1   REACT_APP_API_KEY = "8qw7g%2FC%2E
```

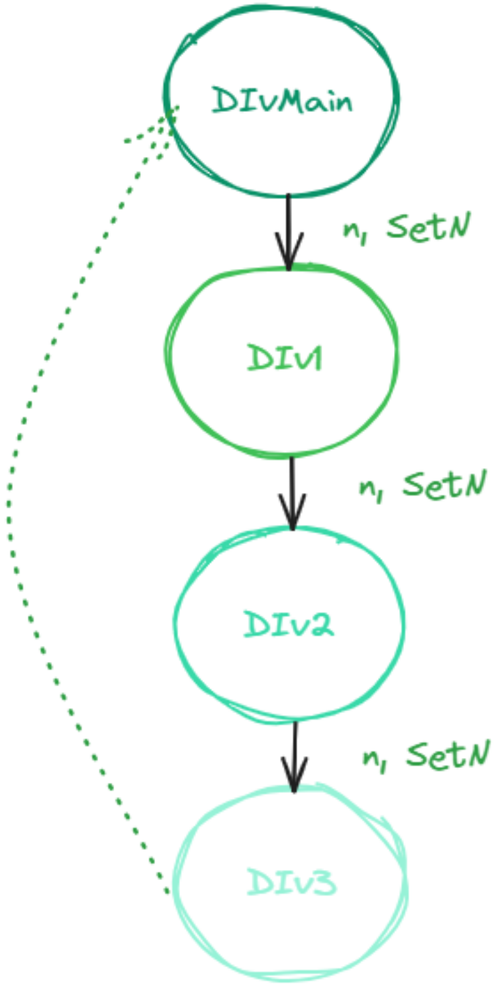
```
const apikey = process.env.REACT_APP_API_KEY ;
```

.env 파일이 올라가면 안 되기 때문에 gitIgnore에 .env를 꼭 추가

```
.gitignore
20  .env
```

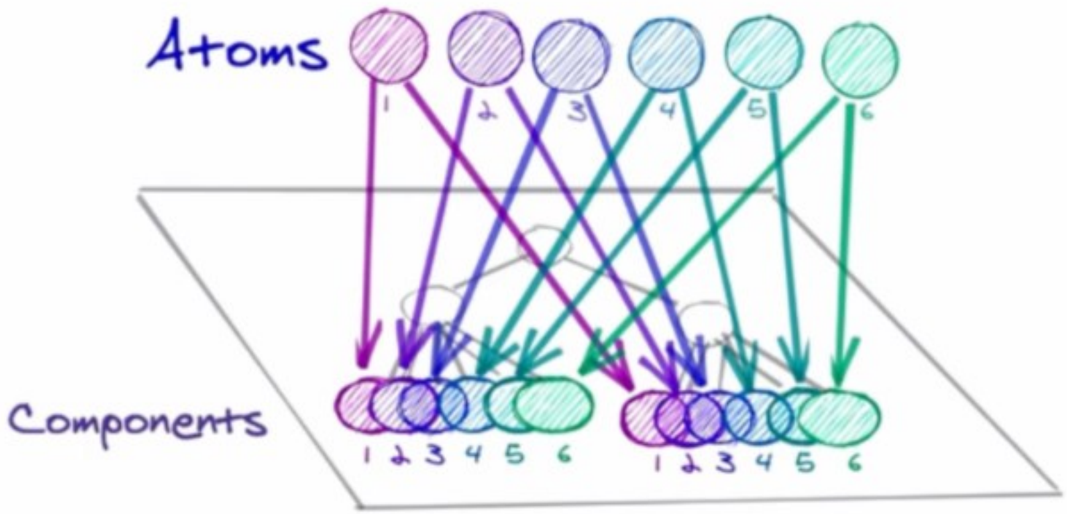
Probs를 활용한 상태 변경





리액트 Recoil

- React를 위한 상태 관리 라이브러리
  - Facebook에서 출시된 React 전용 상태관리 라이브러리
- Recoil 설치
  - npm install recoil
- RecoilRoot
  - 컴포넌트에서 Recoil state를 사용하기 위해서는 recoil 상태를 사용하고자 하는 컴포넌트의 부모에 RecoilRoot를 선언
  - RecoilRoot는 여러 개를 선언할 수도 있음



Atom

- 상태(state) 단위이며 업데이트와 구독이 가능
  - atom 값을 읽는 컴포넌트들은 암묵적으로 atom을 구독
  - atom에 어떤 변화가 있다면 그 atom을 구독하는 모든 컴포넌트가 리렌더링
- Atom 설정
  - atom()을 사용하여 key와 default값을 필수로 선언
  - key: 내부적으로 atom을 식별하는 데 사용되는 고유한 문자열
  - default: atom의 초깃값

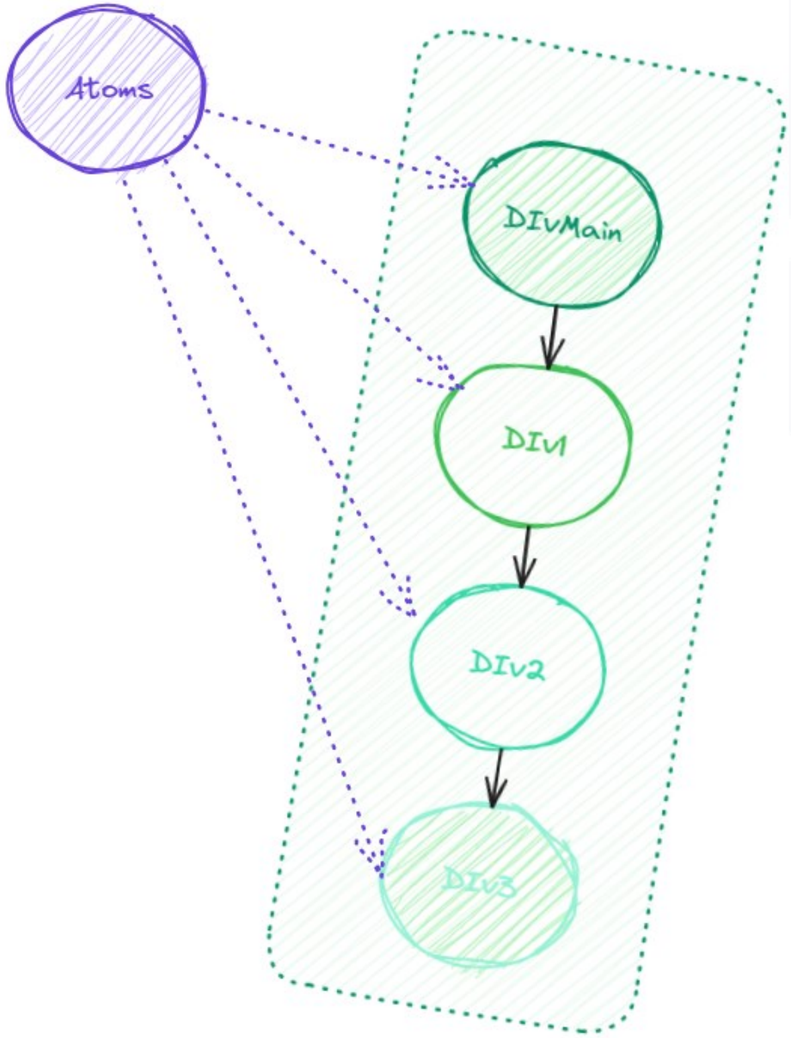
selector

- 전역 상태 값을 기반으로 어떤 계산을 통해 파생된 상태(derived state)를 반환하는 순수함수
  - get함수만 제공되면 Selector는 읽기만 가능한 RecoilValueReadOnly 객체를 반환
    - **get** 매개변수를 이용하여 **atom**이나 다른 **selector**를 참조
  - set 함수 또한 제공되며 (optional) Selector는 쓰기 가능한 RecoilState 객체를 반환

Atoms 및 Selectors 관련 hooks

- useRecoilState()
  - atom을 읽고, 쓰기 위해 사용. 컴포넌트는 atom을 구독함.
- useRecoilValue()
  - atom을 읽기만 할 때 사용. 컴포넌트는 atom을 구독함.
- useSetRecoilState()
  - atom을 쓰려고만 할 때 사용.
- useResetRecoilState()
  - atom을 default 값으로 초기화 할 때 사용

Recoil 상태관리




```jsx
import DivMain from "./122/DivMain" ;
import { RecoilRoot } from 'recoil'
function App() {
  return (
    <RecoilRoot>
      <DivMain />
    </RecoilRoot>
  );
}

export default App;
```

Recoil state를 사용하기 위해서는 **recoil** 상태를 사용하고자 하는 컴포넌트의 부모에 **RecoilRoot**를 선언

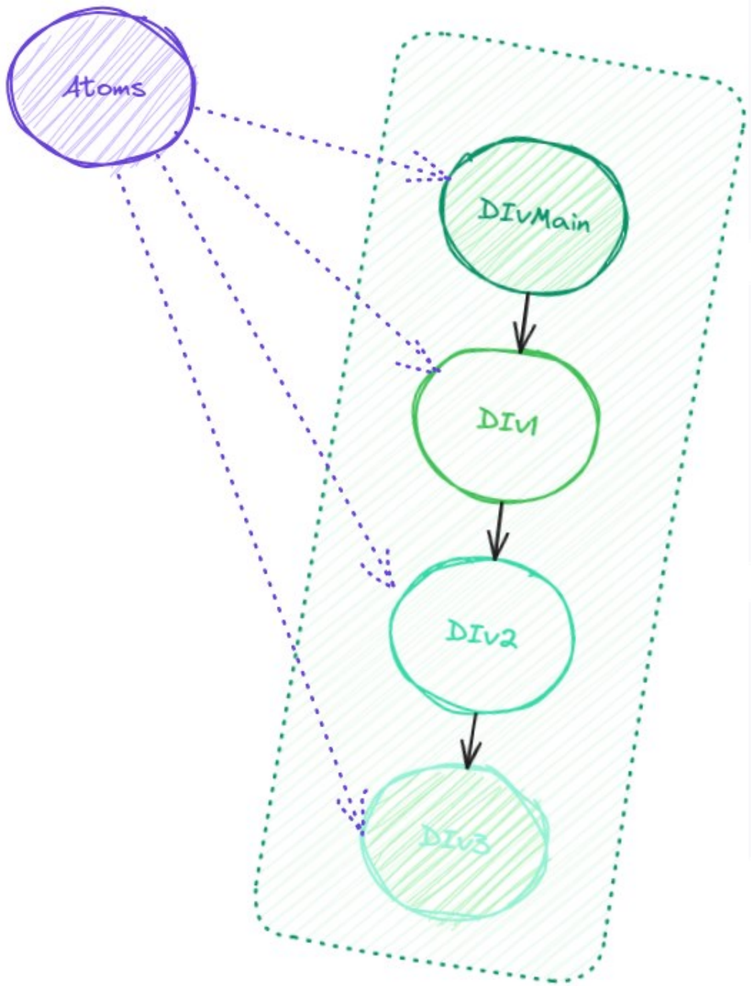
Recoil 상태관리

DivAtom.js

```
import { atom, selector } from "recoil";

export const DivAtom = atom({
  key : 'DivAtom',
  default : 0
});

export const DivAtom2 = selector({
  key : 'DivAtom2',
  get : ({get}) => {
      return get(DivAtom) * 2;
  }
});
```




```
import Div1 from './Div1'
import { DivAtom,DivAtom2 } from './DivAtom' ;
import { useRecoilValue } from 'recoil' ;
const DivMain = () => {
  const n = useRecoilValue(DivAtom) ;
  const n2 = useRecoilValue(DivAtom2) ;
  return (
    <main className='container'>
      <div className='bg-green-950 rounded-lg text-lime-100 p-5 m-5'>
        <h2 className='m-2 text-lime-100'>DivMain</h2>
        <h3 className='m-4 text-lime-100'>n={n}</h3>
        <h3 className='m-4 text-lime-100'>n2={n2}</h3>
        <Div1 />
      </div>
    </main>
  )
}

export default DivMain
```

```
import { DivAtom  } from './DivAtom' ;
import { useRecoilState } from "recoil";
const Div2 = () => {
    const [n, setN] = useRecoilState(DivAtom) ;

    const upN = () => {
      setN(n+1) ;
    }
    const downN = () => {
      setN(n-1) ;
    }
```

로컬 스토리지(Local Storage)

- 웹 브라우저에 데이터를 저장하는 데 사용되는 웹 스토리지 메커니즘 중 하나
  - 데이터를 키-값 쌍으로 저장
  - 웹 애플리케이션에서 이 데이터를 검색하거나 업데이트
  - 민감한 정보나 보안에 민감한 데이터를 저장하는 데에는 주의
- localStorage 객체
  - 저장하기 : localStorage.setItem('username', 'react');
  - 데이터 가져오기 : const username = localStorage.getItem('username');
  - 제거하기 : localStorage.removeItem('email');
  - 모두 제거하기 :localStorage.clear();

로컬 스토리지를 활용한 로그인

Login.js

```
const Login = () => {
  const [user, setUser] = useState(null) ;

  const handleLogin = (username) => {
    localStorage.setItem('user', username) ;
    setUser(username) ;
  }

  const handleLogout = () => {
    localStorage.removeItem('user') ;
    setUser(null) ;
  }

  useEffect(()=>{
    const lsUser = localStorage.getItem('user') ;
    if (lsUser) setUser(lsUser) ;
  }, []);

  return (
    <main className='container'>
      <div className='flex justify-center align-middle w-full h-full'>
      {user ? <Logout user={user} onLogout={handleLogout}/>
            : <LoginForm onLoign={handleLogin}/>}
      </div>
    </main>
  )
}
```

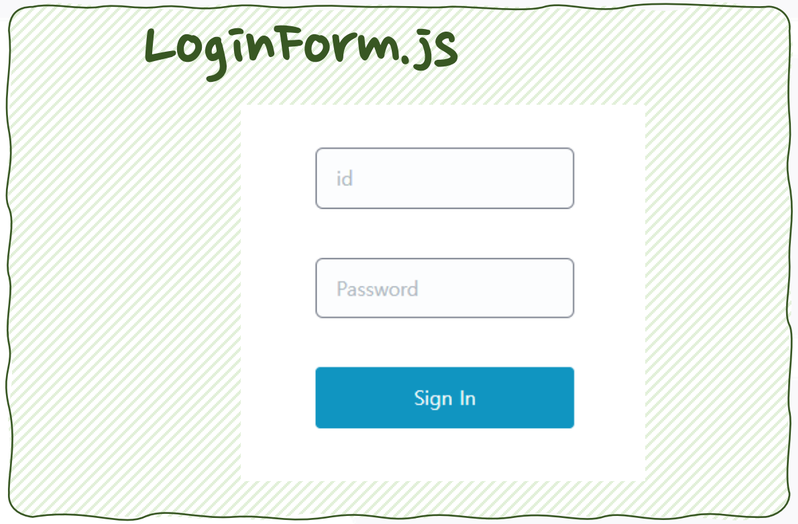
1. 상태변수 정의

2. 컴포넌트생성시 확인

3. 상태값에 따라 로그인과 로그아웃 결정

Loginout.js

LoginForm.js



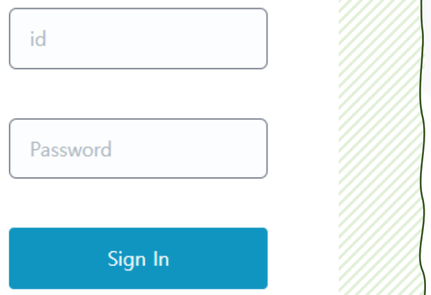
로컬 스토리지를 활용한 로그인

LoginOut.js


```jsx
const Logout = ({user, onLogout}) => {
  return (
    <section classname="py-20">
      <div classname="container">
        <div classname="flex flex-wrap -mx-4">
          <div classname="w-full px-4">
            <div classname="max-w-[525px] mx-auto text-
              <form>
              <div classname="w-full rounded-md border
                {user}님 반갑습니다.
              </div>
              <div classname="mb-10">
                <input
                  type="button"
                  value="Sign out"
                  classname="w-full rounded-md border
                  onClick={onLogout}
                />
              </div>
            </form>
          </div>
        </div>
      </div>
    </section>
  )
}
```

로컬 스토리지를 활용한 로그인

LoginForm.js

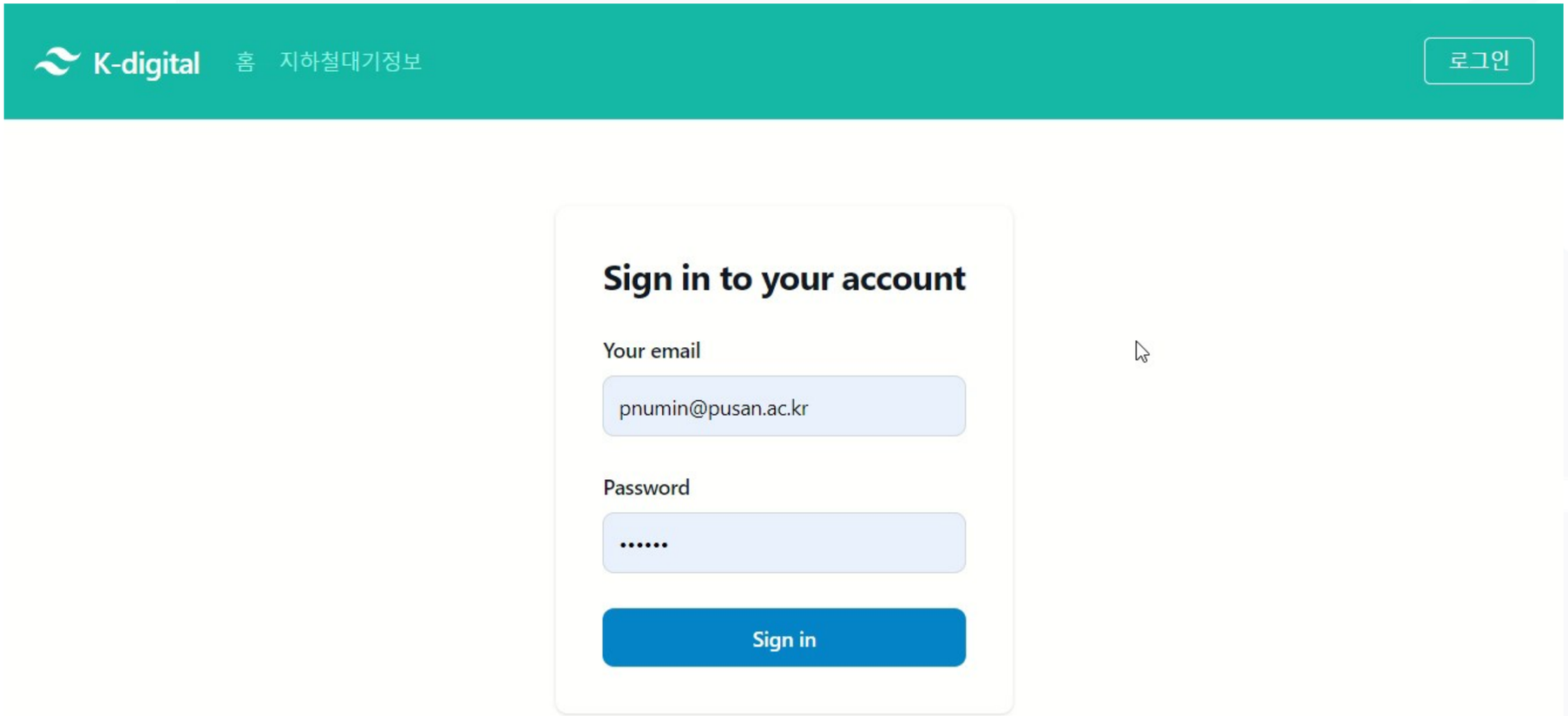



```jsx
const LoginForm = ({onLoign}) => {
  const [username, setUsername] = useState();
  const [pw, setPw] = useState();

  const handleLogin = () => {
    if (username === "user" & pw === 'pw1234') {
      onLoign(username);
    }
    else {
      alert("로그인 실패") ;
    }
  }
```

```jsx
return (
  <section class="py-20">
    <div class="container">
      <div class="flex flex-wrap -mx-4">
        <div class="w-full px-4">
          <div
            class="…
          >
            <form>
              <div class="mb-6">
                <input
                  type="text"…
                  onChange={(e) => setUsername(e.target.value)}
                />
              </div>
              <div class="mb-6">
                <input
                  type="password"…
                  onChange={(e) => setPw(e.target.value)}
                />
              </div>
              <div class="mb-10">
                <input
                  type="button"…
                  onClick={handleLogin}
                />
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </section>
```

해결문제


K-digital
홈
지하철대기정보
로그인
Sign in to your account
Your email
pnumin@pusan.ac.kr
Password
••••••
Sign in

해결문제

데이터명	부산광역시_실내공기질 실시간 측정 자료 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 | 활용신청	처리상태	승인
활용기간	2024-01-10 ~ 2026-01-10		

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	실내공기질 항목별 정보 조회	실내공기질 정보를 항목별 정보(날짜시간, 역이름, 장소, 초미세먼지, 미세먼지 등)를 제공	10000	확인
2	실내공기질 측정소별 정보 조회	실내공기질 정보를 측정소별 정보(날짜시간, 서면역1호선승강장, 사상역대합실, 수영역대합실 등)를 제공	10000	확인


```
▼"item": [
  ▼{
      "controlnumber": "2024040202",
      "arealndex": "201193",
      "office": "부산광역시",
      "site": "서면",
      "city": "1호선",
      "pm10": "48",
      "co2": "495",
      "co": "-",
      "no2": "0.044",
      "no": "0.005",
      "nox": "0.05",
      "o3": "-",
      "pm25": "33",
      "fad": "-"
    }
]
```