날씨 조회 앱 리메이크

WEATHER NOW

PRESENTATION FOR SEMINAR

1 프로젝트 소개

- 프로젝트 소개

2 웹 구조 및 설계

- 웹 구조
- 주요 컴포넌트 구성

3 API 연동 방식

- 사용 API 소개
- 데이터 처리 과정

4 주요 코드 설명

- 프론트엔드/백엔드

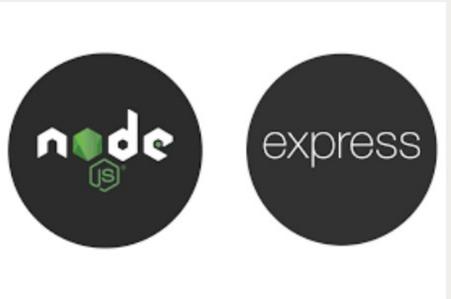
5 결론 및 향후 계획

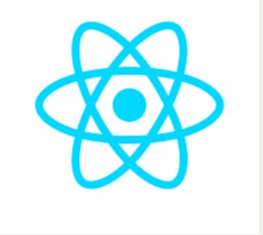
- 개발 성과
- 향후 개선 계획

프로젝트 소개

1학기와 뭐가 달라졌고 얼마나 성장했는지가 궁금해졌다
그래서 무엇을 할지 생각하다가 전과 얼마나 성장했는지
알기위해 같은 주제를 이번에 특강, 방과후 수업을 들으며
배운 언어들을 활용해서 다시 한 번 만들어보기로 했다

사용한 언어 & 기술





- 프론트엔드: React
- 백엔드: Node.js, Express
- API: OpenWeather

PRESENTATION FOR SEMINAR

웹 구조 및 설계

데이터 흐름도

- 사용자 요청 → 서버
- 서버 → 날씨 API
- 데이터 가공
- 클라이언트 전송
- 화면 렌더링

주요 컴포넌트 구성

- 로그인 / 회원가입
- API 연동
- 데이터 처리

API 연동 방식

사용한 날씨 API

- OpenWeather API
ㄴ>Current Weather Data

API 호출 방법

- GET 방식 요청
- 요청 파라미터 구성
- 응답 데이터 처리
-

데이터 가공 및 처리

- JSON 데이터 가공
- 필요 정보 추출

프론트엔드 코드 소개

검색

- useState
- Component props

응답

- useState
- fetch

회원가입

- useState
- useRef
- fetch

로그인

- useState
- useRef
- useEffect
- fetch

```jsx
export function Search() {
  const [city, setCity] = useState();
  const [cit, scit] = useState();
  return (
    <>
      <p className='intro' id='h'>🌤️ 오늘의 날씨를 가장 먼저, 가장 정확하게</p>
      <p className='intro'>📍 필요한 정보만 깔끔하게 확인하세요</p>
      <p className='intro'>👕 덤으로 날씨에 맞는 옷차림과 준비물까지</p>
      <p className='intro'>✨ 하루 준비, 이 웹 하나면 충분합니다</p>
      <div className='search'>
        <input type="text" placeholder='도시를 입력하세요' className='searchi' onChange={(e) => scit(e.target.value)}/>
        <button className='searchb' onClick={() => setCity(cit)}>조회</button>
      </div>
      {city && <Response city={city} />}

    </>
  )
}
```

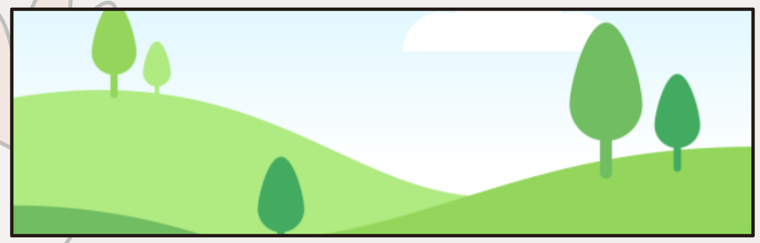
```
function Response(props) {
  const [weat, setweat] = useState([]);
  useEffect(() => {
    fetch(`http://localhost:3000/weather/${props.city}`)
    .then(res =>
      res.json()
    )
    .then(res => setweat(Object.entries(res).map(([k, v]) => (<p>{k}: {v}</p>))))
  }, [weat])
  return (
    <div className='box'>
      {weat}
    </div>
  )
}
```

```
export function SignUp() {
    const focus = useRef();
    const [id, sid] = useState();
    const [ps1, sps1] = useState();
    const [ps2, sps2] = useState();
    const [city, scity] = useState();
    function req() {
      fetch(`http://127.0.0.1:3000/signup/${id}/${ps1}/${city}`)
      alert('저장됨')
    }
    useEffect(() => {
        focus.current.focus();
    }, []);
  return (
    <div className='box'>
      <h2>회원 가입</h2>
      <input ref={focus} type="text" placeholder='아이디를 입력하세요' onChange={(e) => sid(e.target.value)}/>
      <input type="text" placeholder='비밀번호를 입력하세요' onChange={(e) => sps1(e.target.value)}/>
      <input type="text" placeholder='비밀번호를 다시 입력하세요' onChange={(e) => sps2(e.target.value)}/>
      <input type="text" placeholder='사는 도시' onChange={(e) => scity(e.target.value)}/>
      <button onClick={() => {ps1 === ps2 ? req() : alert('비밀번호가 틀렸습니다.')}}>확인</button>
    </div>
  )
}
```

```jsx
export function Login() {
    const focus = useRef();
    const [id, sid] = useState();
    const [ps1, sps1] = useState();
    const [state, cstate] = useState(0);
    const [need, sneed] = useState([]);
    function req() {
      fetch(`http://127.0.0.1:3000/login/${id}/${ps1}`).then(res =>
        res.json()
      ).then(resp => {alert(resp.res)})
      cstate(1)
      fetch(`http://localhost:3000/city/${id}`)
      .then(res=>res.json())
      .then(res => fetch(`http://localhost:3000/weather/${res.res}`))
      .then(res => res.json())
      .then(res => sneed([res.이름, res.날씨, res.최저온도, res.최고온도, res.풍속]))

    }
    useEffect(() => {
        focus.current.focus();
    }, []);
  return (
    <div className='box'>
      {state === 0 ?
      <>
        <h2>로그인</h2>
        <input  ref={focus} type="text" placeholder='아이디를 입력하세요' onChange={(e) => sid(e.target.value)}/>
        <input type="text" placeholder='비밀번호를 입력하세요' onChange={(e) => sps1(e.target.value)}/>
        <button onClick={() => req()}>확인</button>
      </>
      :
      <>
        <h2>안녕하세요 {id}님</h2>
        <p>오늘 {id}님이 거주하시는 {need[0]}의 날씨는 {need[1]}이고</p>
        <p>기온은 최저 {need[2]}도에서 최고 {need[3]}도 사이 일 것으로 보입니다</p>
        {(need[2]+need[3])/2<=14 ? <><p>기온이 따듯하니 겉옷은 필요 없을 듯 합니다 그리고</p>식사로는 조금씩 더워지므로 수분이 많은 음식
        {(need[4] >= 5.5) ? <p>오늘의 풍속은 {need[4]}m/s로 바람이 강하니 외출할때 바람막이를 입는 것도 좋을 것 같습니다</p> : <p></p>}

      </>
      }
    </div>
  )
}
```

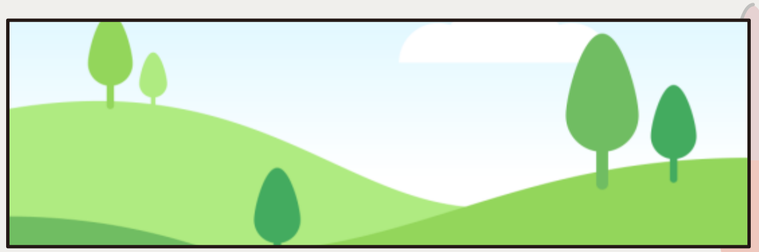
백엔드 코드 설명

API 연동 구현

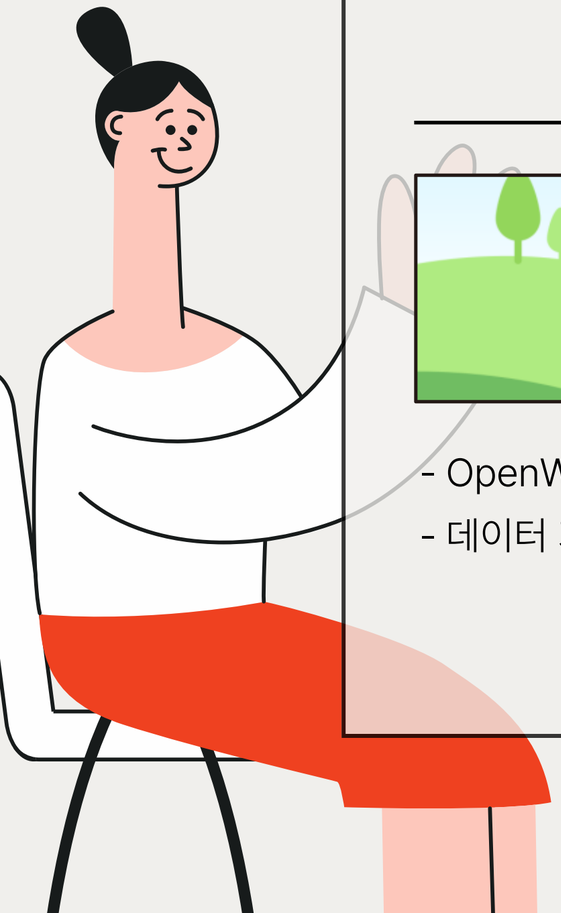


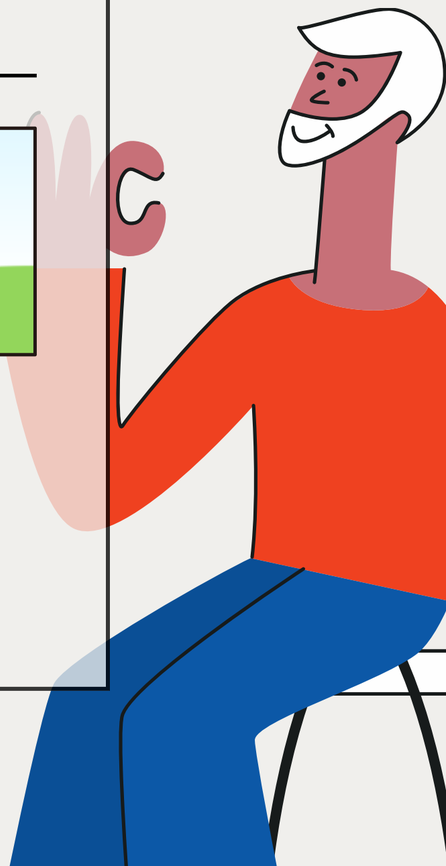
- OpenWeather API 연동
- 데이터 가공 및 처리

회원 정보 처리



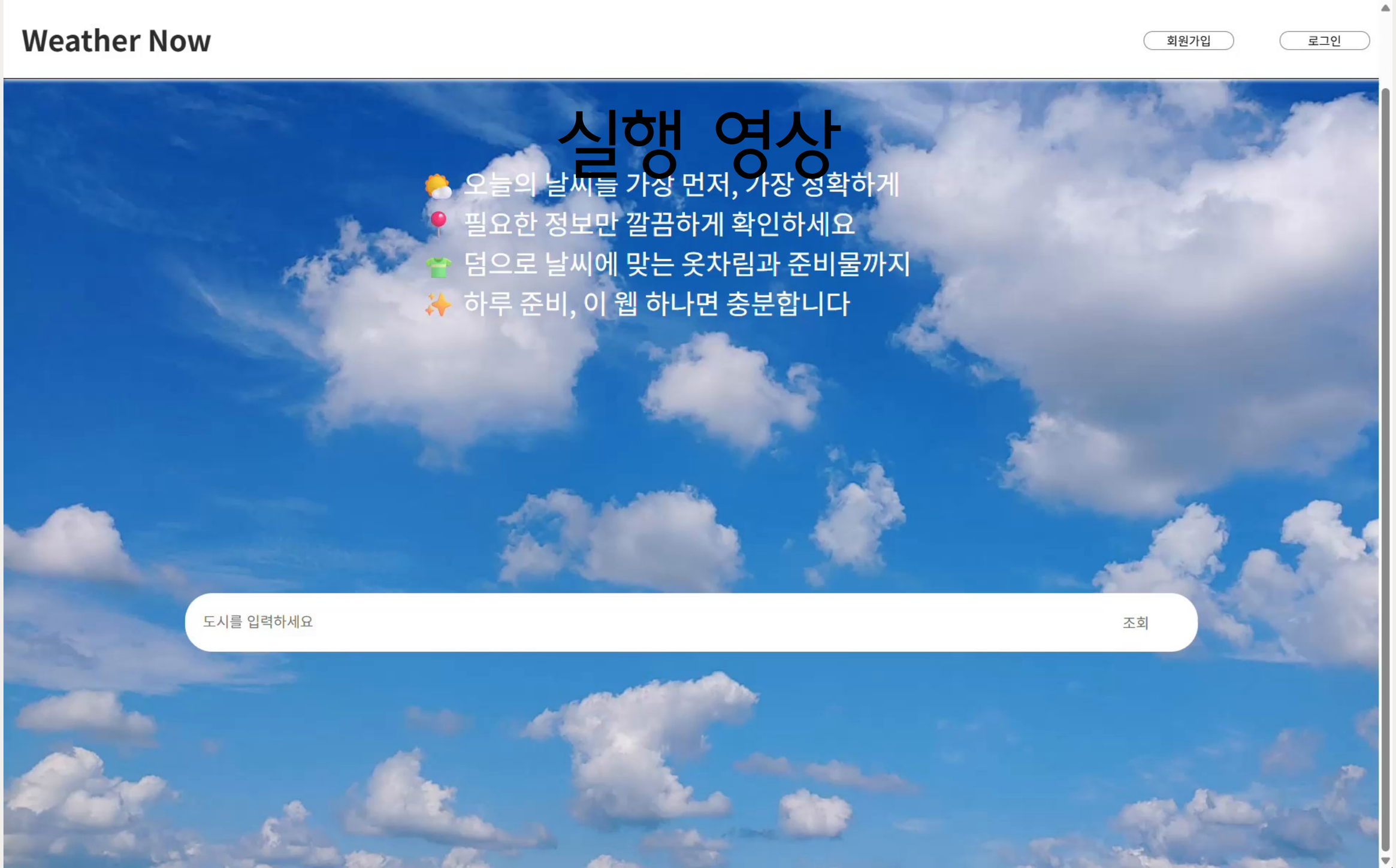
- JSON 데이터 파싱
- JSON 파일로 저장
- GET방식으로 정보 전달

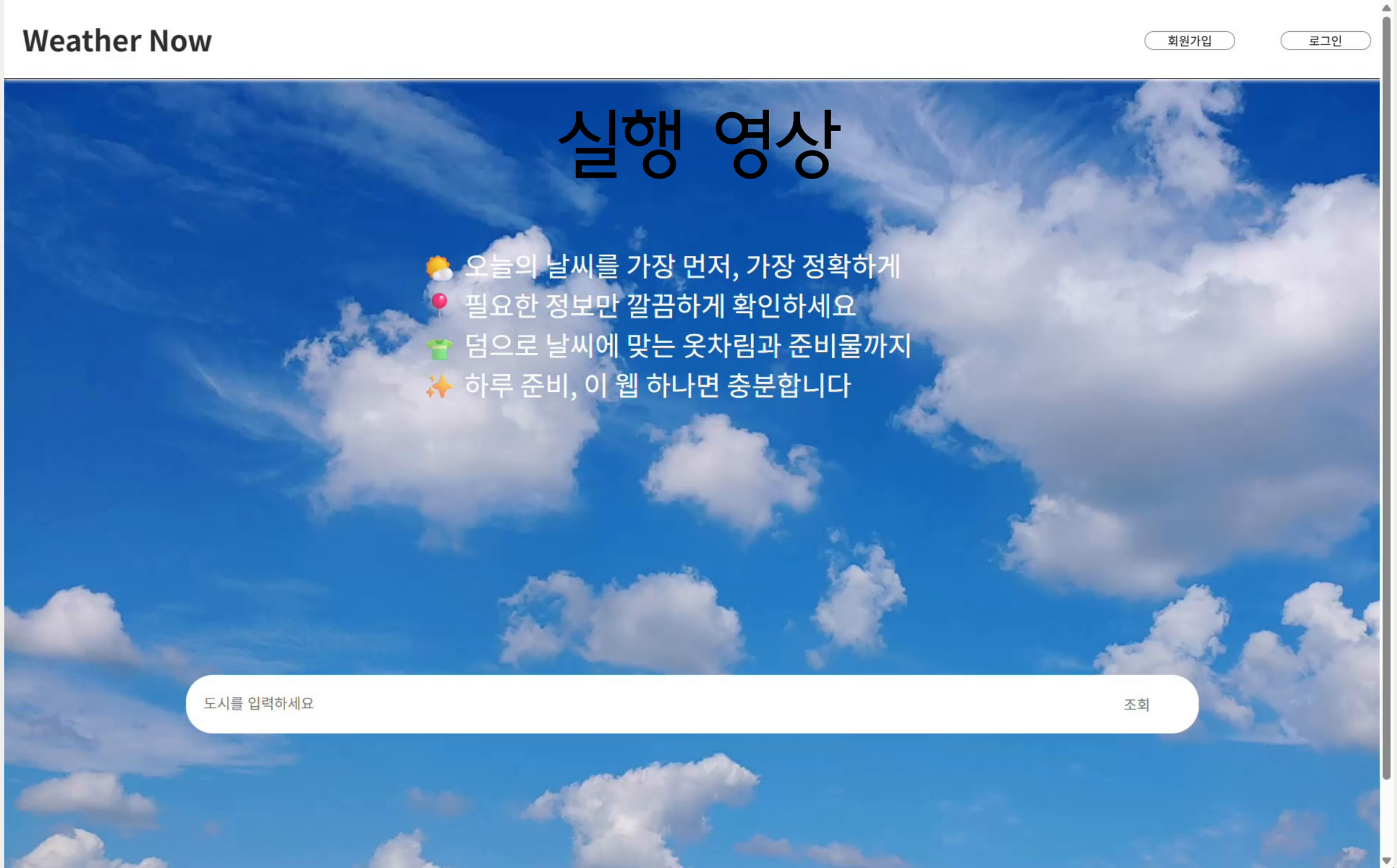




```
async function getWeather(city) {
  const result = await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${api_key}&units=metric`);
  const data = await result.json();
  const needed = {
    이름: data.name,
    날씨: data.weather[0].main,
    세부날씨: data.weather[0].description,
    온도: data.main.temp,
    체감온도: data.main.feels_like,
    최저온도: data.main.temp_min,
    최고온도: data.main.temp_max,
    기압: data.main.pressure,
    습도: data.main.humidity,
    풍속: data.wind.speed,
    풍향: data.wind.deg,
    구름량: data.clouds.all,
  }
  return needed;
}

module.exports = getWeather;
```

```
const fs = require('fs');

let rawData = fs.readFileSync('info.json');
let jsonData = JSON.parse(rawData);

function signup(id, passwd, city) {
    jsonData[id] = passwd;
    jsonData[id+'city'] = city;

    fs.writeFileSync('info.json', JSON.stringify(jsonData, null, 2));
}

function login(id, passwd) {
    if (jsonData[id] === passwd) {
        return true
    }
    else {
        return false
    }
}

function city(id) {
    return jsonData[id+'city'];
}

module.exports = { login, signup, city };
```


Weather Now
회원가입
로그인
실행 영상
오늘의 날씨를 가장 먼저, 가장 정확하게
필요한 정보만 깔끔하게 확인하세요
덤으로 날씨에 맞는 옷차림과 준비물까지
하루 준비, 이 웹 하나면 충분합니다
도시를 입력하세요
조회


Weather Now
회원가입
로그인
실행 영상
🌤️ 오늘의 날씨를 가장 먼저, 가장 정확하게
🎈 필요한 정보만 깔끔하게 확인하세요
👕 덤으로 날씨에 맞는 옷차림과 준비물까지
✨ 하루 준비, 이 웹 하나면 충분합니다
도시를 입력하세요
조회

결론 및 향후 계획

React, JS, OpenWeather API등을 사용하여

날씨 조회앱을 완성했습니다 하지만

비동기 연결을 이번에 처음 시도하여 이부분에 상당한 시간을 사용했으며

then이 계속 반복되어 코드를 해석하는데 어려움이 생기기도 하였습니다

또한 사용자의 거주 정보에 따라 여러 정보를 알려주는 시스템 역시

미흡한 부분이 많습니다. 만들다보니 이렇게 새로운 기술을 이용해보는 것보다

한가지 언어를 열심히 하는게 좋을 것 같다는 생각이 들었습니다.

향후에는 ai를 활용하여 번역기능을 추가하고

거주지역 정보 시스템도 완성하고 싶습니다.