

오픈소스 프로젝트 보고서

2025.01.15

컴퓨터공학과	차민수	32217755
모바일시스템공학과	김태경	32211203
컴퓨터공학과	노현정	32227291

프로젝트 팀 : 전세계 도시별 날씨 정보 웹사이트

Github link : <https://github.com/Minsu4302/OSPJ>

보고서 목차

I. 프로젝트 개요	-2-
II. 사용 기술	-3-
III. 개발 과정	-4-
IV. 주요 기능	-6-
V. 결과물 설명	-9-
VI. 기대효과 및 결론	--
VII. 참고자료	--

I. 프로젝트 개요

1. 프로젝트 제목

: 날씨 정보 제공 웹사이트의 UI 개선 및 API 확장

2. 프로젝트 배경

1) 프로젝트를 시작하게 된 이유 :

핸드폰의 간편한 날씨 정보 제공과 같은 빠르고 직관적인 경험을 웹사이트에서도 구현하기 위해 해당 오픈소스를 발견하였고, 이를 기반으로 기능과 디자인을 개선하여 더욱 편리하고 풍부한 정보를 제공하는 날씨 웹사이트를 개발하고자 한다.

2) 현재 오픈소스의 문제점 :

기존의 오픈소스 기반 날씨 정보 제공 웹사이트는 영어로 정확한 도시명을 검색해야 위치 기반 날씨 정보를 얻을 수 있으며, 지역별 세부 날씨 데이터 제공에서 한계를 보이고 있다. 또한, 사용자 인터페이스(UI)가 복잡하고 확장성이 부족하여 사용자 경험(UX)이 제한적이다.

3) 개선 방향

사용자 경험 개선 :

- 자동완성 기능: 검색어를 입력할 때, 실시간 추천 검색어를 제공하여 검색 편의성을 높임.
- 다국어 검색 지원: 글로벌 사용자를 위한 검색어 번역 API 연동.

정보 제공 강화 :

- 날씨 데이터를 시각화(예: 그래프, 차트)하여 이해도를 높임.
- 깔끔하고 직관적인 시작 화면과 정보 정렬로 사용자 친화적인 UI를 제공.

3. 프로젝트 목표

최종 목표

: 기존의 오픈소스에서 사용자 편의성을 높이고, 검색 기능 개선과 데이터 시각화를 이용한 날씨 정보 제공 플랫폼 구축하기.

세부 목표 :

- 검색어 자동완성과 다국어 지원 기능을 통한 사용자 접근성 향상.
- 시각화 도구를 활용한 직관적이고 풍부한 날씨 데이터 제공.
- 시작 화면의 정돈된 UI와 반응형 웹 설계를 통한 사용자 경험 개선.
- 기존 API의 확장 및 세부 데이터 제공(대기질, 강수량 예보 등).

II. 사용 기술

1. 활용할 오픈소스 및 이유

오픈소스 이름 : Weather by pekkiriscim

오픈소스 링크 : <https://github.com/pekkiriscim/weather?tab=readme-ov-file>

1) 오픈소스의 기능 :

- 도시 검색을 통해 날씨 정보(온도, 풍속, 습도, 기압, 가시성 등)를 제공.
- 5일간의 시간대별 예보 데이터 출력.
- 모던하고 깔끔한 사용자 인터페이스(UI).

2) 오픈소스 선택 이유 :

- React.js를 기반으로 구현되어 있어 확장성이 높음.
- 날씨 데이터의 세부정보를 제공하는 기능이 포함되어있어 프로젝트의 시작점으로 적합.
- MIT License로 라이선스 문제가 없어 수정 및 배포가 가능.
- 기존 오픈소스는 기본적인 날씨 데이터 제공에 초점을 맞추고 있으나, 이를 기반으로 검색어 자동완성, 다국어 지원 등 기능을 추가하여 사용자 경험을 대폭 개선할 수 있음.

2. 기술 스택

1) 프론트엔드

- HTML, CSS, React, JSX: 사용자 인터페이스 설계 및 구현. React와 JSX를 활용하여 컴포넌트 기반의 구조적 UI 개발.
- React Icons: 날씨 아이콘과 시각적 요소를 React 컴포넌트로 제공하여 직관적인 UI 구성.

2) 백엔드

- Node.js: API 요청 처리 및 데이터 관리.
- JSX: React와의 통합을 통해 백엔드와 프론트엔드 간 데이터 연동.

3) 날씨 API

- OpenWeatherMap: 실시간 날씨 데이터 및 5일간의 시간대별 예보 제공.
- Weather API: 검색어 자동완성과 도시 이름 추천 기능 지원.

4) UI

- Budiarti: 날씨 시각화와 데이터 정렬을 위한 반응형 UI 디자인 시스템.
- React Icons: 간결하고 직관적인 아이콘으로 사용자 경험 강화.

5) 빌드 및 배포

- React: 컴포넌트 기반 프론트엔드 구현.
- Bootstrap: 반응형 스타일링 및 UI 요소 설계.
- Vercel: React 기반 프로젝트를 위한 빠르고 안정적인 배포 플랫폼.

III. 개발 과정

1. 개발 일정

일정	작업내용	담당자	작업 확인
~12.30	GitHub 레포지토리 생성 및 프로젝트 초기화 개발 환경 설정 (React 프로젝트 생성, Node.js 설치)	전 인원	12.31 완료
~01.01	활용할 오픈소스 선정 및 오픈소스 활용 전략 수립 UI/UX 설계 및 React 컴포넌트 구조 설계	김태경	01.02 완료
~01.03	프로젝트 제안서 작성 OpenWeatherMap API 연동 및 데이터 출력 구현	김태경, 차민수	01.04 완료
~01.08	Weather API 연동 및 검색어 자동완성 기능 구현 Google Translate API 연동 및 다국어 검색 지원 구현	차민수	01.08 완료
~01.11	디자인 시스템을 반영한 웹 UI 구현 데이터 시각화(Chart.js) 기능 구현	노현정	01.09 완료
~01.14	통합 테스트 및 프로젝트 보고서 작성	노현정, 김태경	01.12 완료

2. 역할 분담

이름	역할	주요 작업
차민수	깃허브 관리 프론트 & 백엔드 개발	깃허브 관리 총괄 React 컴포넌트 설계, API 연동 및 기능 구현
김태경	백엔드 개발 및 조정	오픈소스 활용 전략 수립 및 제안서, 보고서 작성 Node.js를 활용한 API 연동 및 데이터 처리
노현정	UI/UX 개발 및 시각화	디자인 시스템을 반영한 반응형 웹 UI 구현 Chart.js를 활용한 날씨 데이터 시각화 구현 및 발표

3. 개발 도구

1) Visual Studio Code (VS Code):

- React.js와 Node.js 코드를 작성하고 디버깅하기 위해 사용된 IDE(통합 개발 환경).
- 확장 프로그램을 통해 JavaScript, JSX, CSS 등의 코드 작성 효율성을 높임.

2) Postman:

- Weather API와 OpenWeatherMap API의 요청 및 응답 데이터를 테스트하고 확인하는데 사용.

3) Git & GitHub:

- Git: 코드 버전 관리와 협업을 위한 분산 버전 관리 시스템.
- GitHub: 프로젝트 레포지토리 호스팅 및 협업 플랫폼. Pull Request와 Issue 관리를 통해 팀 작업을 지원.

4) Vite:

- React.js 기반의 개발 환경 설정과 빌드 속도를 최적화하기 위한 프론트엔드 툴링 시스템.
- 빠른 개발 서버 제공 및 효율적인 번들링 수행.

5) Vercel:

- 최종 React 애플리케이션을 배포하기 위한 플랫폼.
- 자동으로 GitHub와 연동되어 배포 프로세스를 간소화.

4. 개발 환경 설정

1) 프로젝트 초기화: GitHub 레포지토리 클론

```
git clone https://github.com/Minsu4302/OpenSource-Project.git
cd OpenSource-Project
```

2) 의존성 설치: npm을 통한 패키지 설치

```
npm install
```

3) 환경 변수 설정: .env 파일 생성 및 API 키 설정

: 프로젝트 루트 디렉토리에 .env 파일을 생성하고, 다음과 같이 API 키를 설정합니다:

```
VITE_OWM_API=<OpenWeatherMap_API_KEY>
```

```
VITE_WA_API=<WeatherApi_API_KEY>
```

- <OpenWeatherMap_API_KEY>와 <WeatherApi_API_KEY>는 각각 OpenWeatherMap과 WeatherApi에서 발급받은 API 키로 대체해야 합니다.

4) 개발 서버 실행: Vite를 통한 개발 서버 시작 & 로컬 개발 서버 접속

```
npm run dev
```

- 브라우저에서 <http://localhost:3000>을 열어 애플리케이션을 확인합니다.

5) 빌드 및 배포: 프로덕션 빌드 생성 & Vercel을 통한 배포

```
npm run build
```

- Vercel 계정을 생성하고, GitHub 레포지토리를 연결하여 손쉽게 배포할 수 있습니다.

6) API 테스트: Postman을 활용한 API 테스트

: WeatherApi 및 OpenWeatherMap API의 엔드포인트를 테스트하여 올바른 응답을 받는지 확인합니다.

WeatherAPI 도시검색

GET https://api.weatherapi.com/v1/search.json?key=<WeatherApi_API_KEY>&q=Seoul

OpenWeatherMap 현재 날씨:

GET https://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=<OpenWeatherMap_API_KEY>

IV. 주요 기능

1. 검색어 자동완성

1) 구현 파일 : SearchBar.jsx

2) 기능 설명 :

: 사용자가 검색창에 도시명을 입력하면 입력값을 실시간으로 Weather API에 전달하여 해당 입력과 일치하거나 관련된 도시명을 추천 리스트로 반환한다. 추천 리스트는 드롭다운 형식으로 표시되며, 사용자는 리스트 중 하나를 선택할 수 있습니다.

3) 작동 과정 :

1. 사용자 입력 이벤트 감지: onChange를 통해 사용자의 입력값을 실시간으로 감지.
2. Weather API 호출: 입력된 텍스트를 기반으로 API를 호출하여 관련된 도시명을 검색.
3. 추천 리스트 표시: API에서 반환된 도시 리스트를 드롭다운으로 렌더링.
4. 도시 선택: 사용자가 도시를 선택하면 해당 도시명을 부모 컴포넌트(App.jsx)로 전달하여 다음 단계로 처리.

4) 사용자 경험 강화 :

- 실시간 추천으로 검색 시간을 단축하고 정확도를 높임.
- 오타 방지 및 편리한 검색 제공.

3. 실시간 날씨 정보 제공

1) 구현 파일 : WeatherCard.jsx

2) 기능 설명 :

- 검색한 도시의 현재 날씨 데이터를 OpenWeatherMap API를 통해 가져와 화면에 표시.
- 주요 기상 정보(온도, 풍속, 습도, 기압 등)를 카드 형태로 구성하여 한눈에 볼 수 있음.

3) 작동 과정 :

1. API 호출: 사용자가 도시를 선택하면 OpenWeatherMap API에 요청을 보내 현재 날씨 데이터를 가져옴.
2. 데이터 처리: 반환된 JSON 데이터를 가공하여 React 상태(State)에 저장.
3. UI 렌더링: React를 사용하여 카드 형태로 데이터를 렌더링하여 온도, 풍속, 습도 등의 데이터를 사용자에게 시각적으로 제공.

4) 기능의 특징 :

- 실시간으로 최신 데이터를 불러와 사용자에게 제공.
- 직관적인 UI를 통해 중요한 정보를 한눈에 확인 가능.

4. 날씨 데이터 시각화

1) 구현 파일 : ForecastChart.jsx

2) 기능 설명 :

- 선택한 도시의 5일 예보 데이터를 시간대별로 시각화하여 사용자에게 제공.
- 온도 변화, 강수량, 풍속 등 주요 데이터를 그래프나 차트 형태로 표시.

3) 작동 과정 :

1. API 호출: OpenWeatherMap API에서 5일 예보 데이터를 요청.
2. 데이터 가공: API에서 반환된 데이터(날짜별 온도, 강수량 등)를 필요한 형태로 가공.
3. Chart.js를 활용한 그래프 생성: React와 Chart.js를 연동하여 라인 그래프(Line Chart)와 막대 그래프(Bar Chart)로 데이터 시각화.
4. UI 렌더링: 시각화된 데이터를 ForecastChart 컴포넌트에 출력하여 사용자에게 제공.

4) 기능의 특징 :

- 데이터의 시각적 표현으로 복잡한 정보를 쉽게 이해할 수 있음.
- 시간대별 데이터를 직관적으로 파악 가능.

5. 다국어 검색 지원

1) 구현 파일 : ForecastChart.jsx

2) 기능 설명 :

- 선택한 도시의 5일 예보 데이터를 시간대별로 시각화하여 사용자에게 제공.
- 온도 변화, 강수량, 풍속 등 주요 데이터를 그래프나 차트 형태로 표시.

3) 작동 과정 :

1. API 호출: OpenWeatherMap API에서 5일 예보 데이터를 요청.
2. 데이터 가공: API에서 반환된 데이터(날짜별 온도, 강수량 등)를 필요한 형태로 가공.
3. Chart.js를 활용한 그래프 생성: React와 Chart.js를 연동하여 라인 그래프(Line Chart)와 막대 그래프(Bar Chart)로 데이터 시각화.
4. UI 렌더링: 시각화된 데이터를 ForecastChart 컴포넌트에 출력하여 사용자에게 제공.

4) 기능의 특징 :

- 데이터의 시각적 표현으로 복잡한 정보를 쉽게 이해할 수 있음.
- 시간대별 데이터를 직관적으로 파악 가능.

6. 반응형 UI

1) 구현 파일 : styles/App.css
styles/components.cs

2) 기능 설명 :

- CSS 미디어 쿼리와 Bootstrap을 사용하여 모든 기기(모바일, 태블릿, 데스크톱)에서 최적화된 UI 제공.

3) 작동 과정 :

1. CSS 미디어쿼리 적용: 화면크기에 따라 UI 요소(글자크기, 카드크기 등)를 동적으로 변경.
2. Bootstrap 그리드 시스템 활용: 다양한 화면 크기에서 레이아웃이 자동 조정되도록 설계.
3. 테스트 & 최적화: 브라우저 개발자도구를 사용하여 다양한 디바이스환경에서 UI를 테스트.

4) 기능의 특징 :

- 모든 디바이스에서 일관된 사용자 경험 제공.
- 모바일에서도 손쉬운 조작과 접근성 보장.

7. 프로젝트 관리 및 배포

1) 기능 설명 :

- GitHub를 통해 프로젝트를 체계적인 관리와, Vercel을 이용해 완성된 웹사이트를 배포.

2) 작동 과정 :

1. GitHub 관리:
 - Git 브랜치를 활용하여 팀원 간 작업 분담 및 병합 관리.
 - 코드 리뷰 및 PR(Pull Request)을 통해 품질 유지.
2. Vercel 배포:
 - CI/CD(Continuous Integration/Continuous Deployment)를 통해 변경 사항을 실시간으로 업데이트.

4) 기능의 특징 :

- 코드 리뷰 및 Pull Request를 통한 품질 관리.
- 실시간으로 변경 사항을 배포하여 테스트 환경 제공.

V. 결과물 설명

1. 프로젝트 링크

: <https://github.com/Minsu4302/OpenSource-Project>

- 프로젝트의 소스 코드가 저장된 GitHub 레포지토리.
- 프로젝트의 구조, 코딩 방식, 사용된 기술 스택 등을 직접 확인 가능.
- README 파일에서 프로젝트의 설치, 실행 방법, 주요 기능 설명 제공.

2. 특징 및 주요 결과

- 1) 검색어 자동완성: 입력 중에도 실시간 추천 도시를 제공.
- 2) 다국어 검색지원: 글로벌 사용자 경험을 위한 다국어 검색 지원.
- 3) 날씨 데이터 시각화: 온도와 강수량 데이터를 직관적으로 이해할 수 있는 그래프 표시.
- 4) 반응형 디자인: 모바일, 태블릿, 데스크톱 등 다양한 화면 크기에 최적화.

3. 프로젝트 파일 구조

```
OpenSource-Project/
├── public/
│   ├── favicon.ico           # 브라우저 탭에 표시되는 아이콘
│   └── index.html            # 애플리케이션의 루트 HTML 파일
├── src/
│   ├── components/          # React 컴포넌트 디렉토리
│   │   ├── SearchBar.jsx    # 도시 검색 및 자동완성 컴포넌트
│   │   ├── WeatherCard.jsx  # 현재 날씨 정보 표시 컴포넌트
│   │   └── ForecastChart.jsx # 날씨 데이터 시각화 컴포넌트
│   ├── styles/              # 스타일 관련 디렉토리
│   │   ├── App.css          # 전체적인 스타일 설정
│   │   └── components.css    # 개별 컴포넌트 스타일 정의
│   ├── App.jsx              # 메인 애플리케이션 컴포넌트
│   ├── main.jsx              # React DOM 진입점
│   └── vite-env.d.ts         # Vite와 TypeScript 환경 설정 파일
├── .env                      # 환경 변수 파일(API 키 관리)
├── package.json              # 프로젝트 의존성과 스크립트 정의
└── vite.config.js            # Vite 빌드 및 개발 설정
```

4. 작동 방식 :

1) 검색 및 데이터 흐름:

1. 사용자가 도시명을 입력하면 자동완성 추천 목록이 표시됩니다.
2. 검색에 따라 OpenWeatherMap API에서 해당 도시의 날씨 데이터를 가져옵니다.
3. 가져온 데이터는 현재 날씨와 5일 예보로 나뉘어 화면에 출력됩니다.
4. 데이터는 시각화(그래프) 및 카드 형태로 사용자에게 표시됩니다.
5. 사용자가 언어 설정을 변경하면 다국어로 UI와 데이터가 변환됩니다.

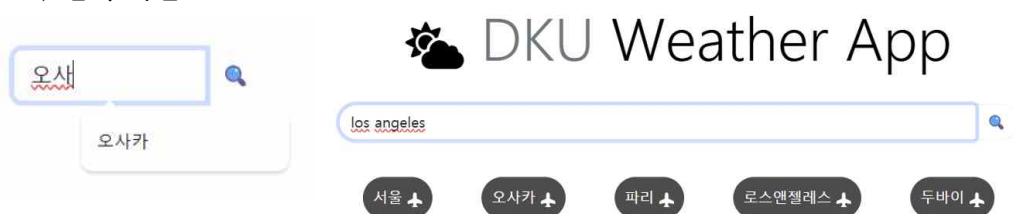
5. 테스트 및 검증 (with 스크린샷)

1) 메인 화면



: 검색창과 초기 상태의 UI를 보여주는 화면.

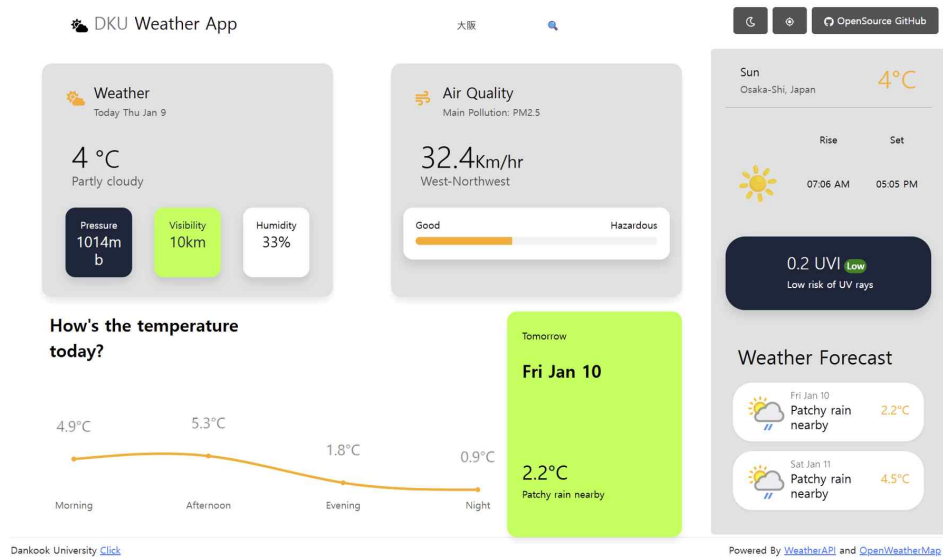
2) 검색 화면



: 도시명을 입력 후, 검색어 자동완성 기능이 작동하는 화면.

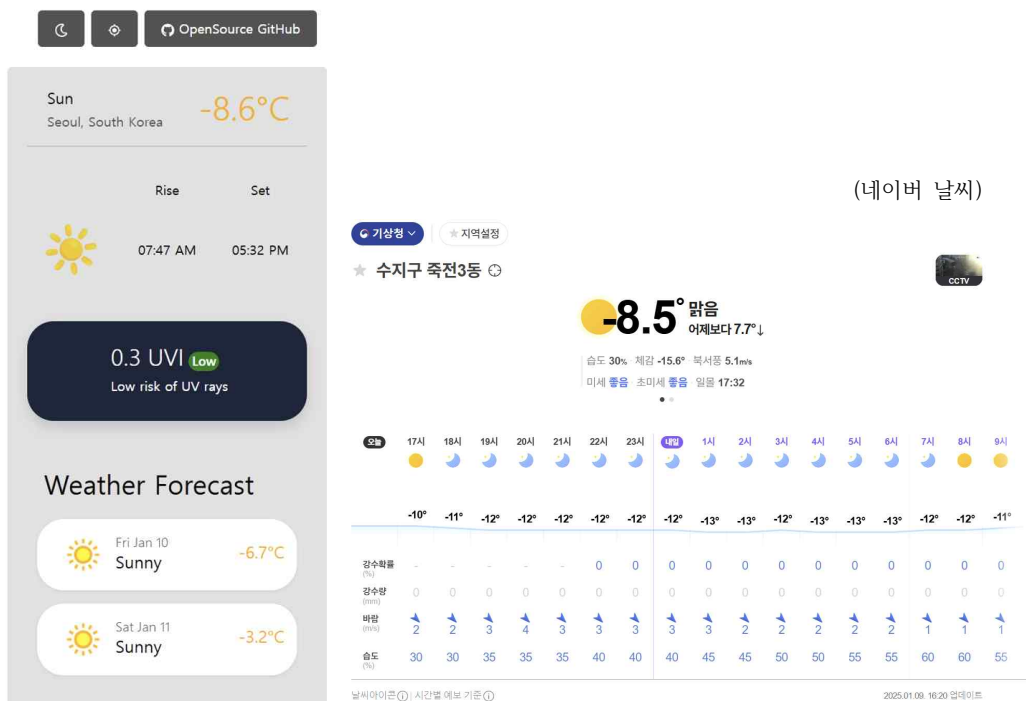
: 검색어가 다른 언어(예: 한국어, 일본어 등)로 번역되는 것을 보여줌.

3) 날씨 데이터 출력 화면



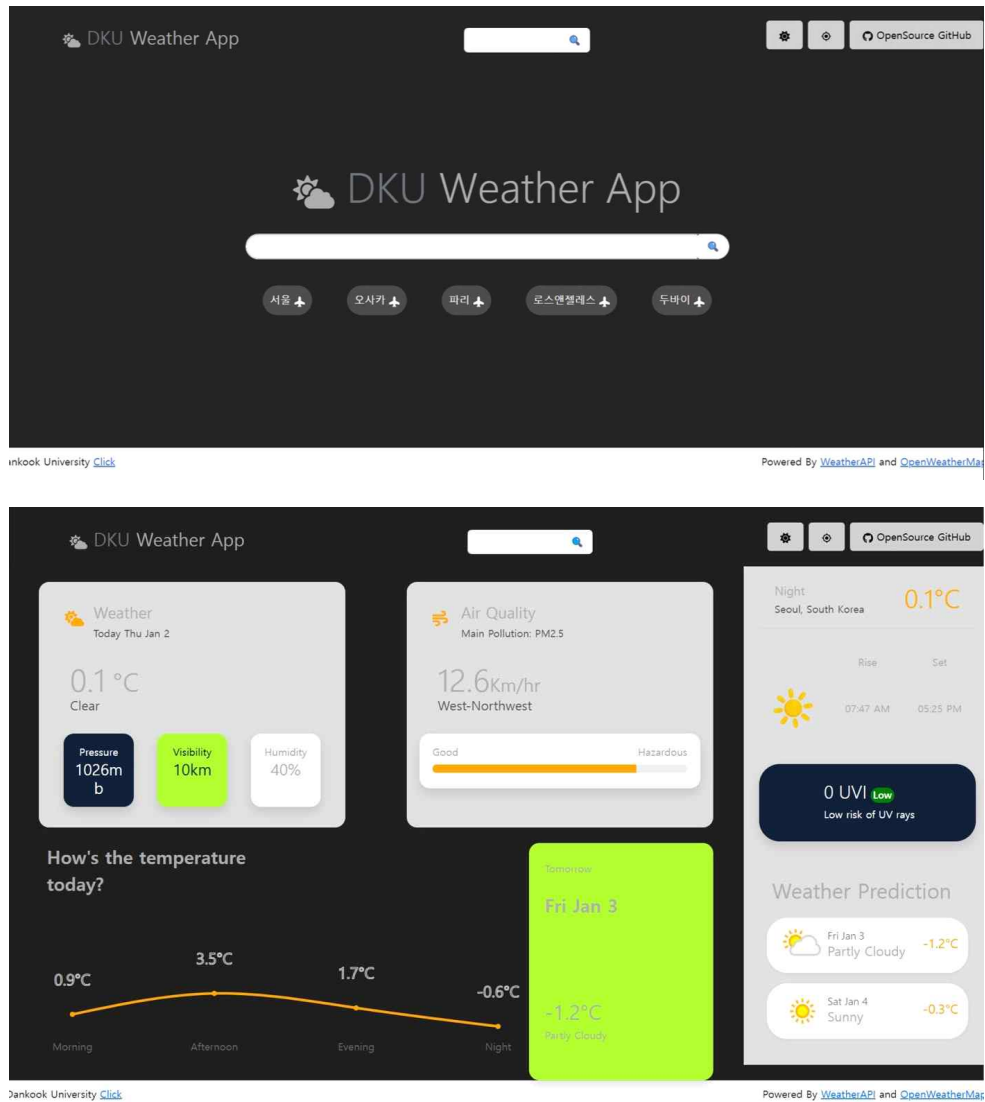
- : 선택된 도시의 현재 날씨 데이터(온도, 습도, 풍속 등)를 표시.
- : 5일 예보 데이터를 그래프로 시각화한 화면.

5) 현재 위치로 검색한 화면



- : 현위치(단국대 인근 날씨 정보 수집구역)로 날씨 정보 연동

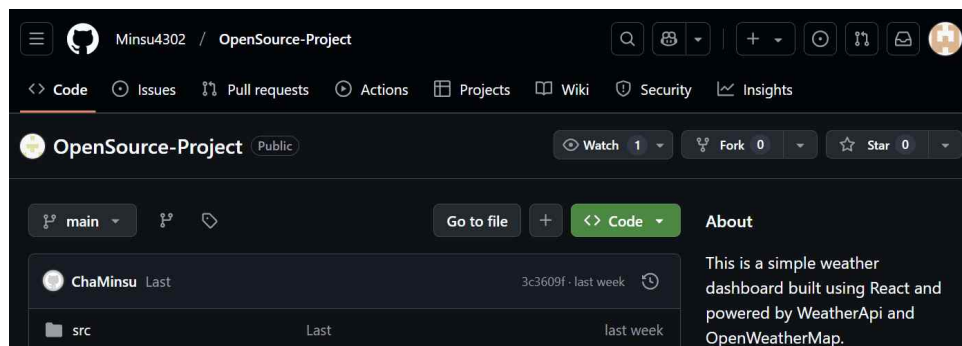
6) 다크 모드 변경



7) 프로젝트 제작팀 링크



github 아이콘(우측 버튼) 클릭



: Github 프로젝트 Link로 이동

VI. 기대효과 및 결론

1. 프로젝트를 통해 기대할 수 있는 효과와 결과

1) 문제 해결:

기존의 오픈소스는 영어로 정확한 도시명을 입력해야만 날씨 정보를 제공하며, 검색 및 UI에서 사용자 경험(UX)에 제한이 있었습니다. 본 프로젝트는 이를 개선하기 위해 아래의 기능들을 추가하였습니다.

- 검색어 자동완성 기능: 사용자가 도시명을 입력할 때 실시간 추천 도시를 제공하여 검색 편의성을 대폭 향상.
- 다국어 검색지원: Google Translate API를 활용해 다양한 언어로 검색하여도 날씨 정보를 제공하여 글로벌 사용자가 손쉽게 이용 가능.
- 시각화 도구: 온도, 강수량 등 주요 데이터를 그래프로 시각화하여 정보를 직관적으로 이해할 수 있도록 지원.
- UI 개선: 반응형 디자인과 깔끔한 레이아웃을 통해 접근성과 사용자 경험을 강화.

2) 결과물의 가치:

- 사용자 친화적이고 직관적인 날씨 대시보드 구현.
- 오픈소스를 활용하여 개발된 독창적인 GitHub 레포지토리 생성.

2. 기술적/비즈니스적 측면에서의 잠재적 가치

1) 기술적 확장성:

글로벌 플랫폼으로의 발전 가능성:

- 다국어 지원을 통해 다양한 사용자층 확보.
- 향후 교통, 이벤트 등 추가 API와의 연동을 통해 기능 확장 가능.

유지보수 및 기능 추가의 용이성:

- React와 Node.js를 기반으로 설계되어 코드 확장이 쉽고 유지보수에 강점이 있음.

2) 비즈니스적 가치:

다양한 산업 활용 가능성:

- 시각적 데이터 표현 및 지역별 세부 데이터는 관광 산업, 대기질 관리, 농업 등 다양한 분야에서 응용 가능.

사용자 경험 강화:

- 자동완성, 다국어 지원, 데이터 시각화는 사용자의 접근성과 만족도를 높이며, 상업적 활용도 가능.

3. 대학생으로서 배울 수 있는 점

1) 기술 습득:

- React.js, Node.js, Chart.js 등의 최신 웹 기술을 활용하여 프론트엔드와 백엔드의 통합적 개발 능력을 습득.
- OpenWeatherMap API, Weather API와 같은 실시간 데이터 연동 기술 학습.
- Vercel과 Bootstrap을 활용한 반응형 웹 개발 경험.

2) 협업 경험:

- Git과 GitHub를 통해 코드 관리와 협업을 진행하여 팀 프로젝트 경험.
- 코드 리뷰, Pull Request, Issue 관리 등을 통해 소통 능력과 협업 능력을 강화.
- 역할 분담과 팀원 간 피드백을 통해 효과적인 문제 해결 경험.

4. 향후 기대 효과

1) 사용자 만족도 향상:

- 검색어 자동완성, 시각화된 데이터 제공, 다국어 지원 등을 통해 글로벌 사용자에게도 접근성과 만족도를 제공.

2) 학습 및 적용의 용이성:

- MIT 라이선스의 오픈소스를 기반으로 개발했기 때문에, 다른 학생, 개발자 또는 연구자들이 학습 자료로 활용 가능.

3) 장기적 유지보수 및 발전 가능성:

- 설계된 구조는 유지보수와 새로운 데이터 및 기능 추가가 용이하며, 다양한 분야로 확장이 가능.

5. 결론

이번 프로젝트는 기존 오픈소스를 기반으로 사용자 편의성과 접근성을 개선하고, 확장 가능한 날씨 정보 제공 플랫폼을 구축하는 것을 목표로 하였습니다. React.js와 Node.js를 활용해 검색어 자동완성, 다국어 지원, 데이터 시각화 기능을 구현하였으며, OpenWeatherMap API, Weather API 등 다양한 데이터를 연동하여 사용자 경험을 강화했습니다.

프로젝트 결과, 사용자는 직관적이고 풍부한 날씨 정보를 쉽게 탐색할 수 있는 플랫폼을 이용할 수 있게 되었으며, 팀원들은 최신 개발 기술과 협업 경험을 통해 실질적인 성장을 이루었습니다.

향후 이 플랫폼은 교통 정보, 대기질 데이터 등 추가 API 연동을 통해 더욱 확장 가능하며, 다양한 산업 및 응용 분야에서 활용될 수 있는 잠재력을 가지고 있습니다.

VII. 참고 자료

- * 활용할 오픈소스 github link

URL : <https://github.com/pekkiriscim/weather?tab=readme-ov-file>

- * 날씨 정보 웹사이트 구현 코드 정보

URL : <https://weather-pekkiriscim.vercel.app/>

- * 날씨 정보 웹사이트 layout 및 디자인 정보

URL : <https://dribbble.com/shots/16833006-Weather-App-Dashboard-Design>

- * 프론트 엔드 CSS, JSX, HTML 정보

URL : <https://ko.legacy.reactjs.org/tutorial/tutorial.html>