|  |
| --- |
| Pusan National University |
| Air Quality Monitoring Dashboard |
| WEB APPLICATION PROGRAMMING – 062 |

|  |
| --- |
| JISAN KIM  201821181 |

1. **Software Tool**

Visual Studio Code (IDE)

1. **APIs**
2. Openweathermap API: to get air quality data
3. Weather API: to get weather, temperature, humidity forecast
4. Opencagedata API: to get city name of user’s location and change the location
5. OpenLayers API: to provide a visual map
6. **What I tried, and success & failure**
7. Implementing a feature to get user’s location

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Figure Getting user’s location

With this code, the application gets user’s permission to take their location. The latitude & longitude of the location are used as parameter for functions.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Figure Simulation of getting location permission

1. Returning the city name from the location

A computer screen with colorful text

Description automatically generated

Figure Getting city name

Opencagedata API returns the city name from latitude & longitude of user’s location.

A blue sign with white text

Description automatically generated

Figure Header with ‘Very Good’ air quality



Figure Header with ‘Bad’ air quality

With different locations, the program shows different city name on the header.

1. Being able to change location via typing the name of city or coordinates

A screen shot of a computer code

Description automatically generated

Figure Text input form for location change

Text input form transfers typing to the dedicated function in javascript file. Users can type the name or coordinate of the location.

A screen shot of a computer code

Description automatically generated

Figure Opencagedata API to return coordinate from the user input

If the user typed the name of their location, Opencagedata API returns the coordinate of the location, and it will be transferred to several functions.

1. Being able to change location via clicking any point on the map



Figure <div> for map



Figure OpenLayers API

There is a map on the webpage. It was created by OpenLayers API.

A screen shot of a computer code

Description automatically generated

Figure Event listener for clicking on the map

The user can change their location by clicking on the map, and all the information will be updated.

1. **Side bar to navigate to a dedicated point of scroll among three sections**

A screenshot of a phone

Description automatically generated

Figure Side bar

A screen shot of a computer screen

Description automatically generated

Figure Side bar in index.html, tooltip style in styles.css

The side bar navigates to a dedicated point of the scroll. Each option has its own icon taken from Bootstrap, Font awesome. If the user hover on the icons, it shows a tooltip that introduces where the option will navigate to. This feature was implemented on CSS file.

1. **Showing line chart for 7-day temperature & humidity forecast**

**For charts, Chart.js was utilized.**

**A graph showing the temperature and humidity

Description automatically generated**

Figure Line chart for 7-Day Temp & Humidity Forecast

A screen shot of a computer program

Description automatically generated

Figure Line chart implementation

A screenshot of a computer program

Description automatically generated

Figure Data update for the line chart

If the webpage is fully loaded, all the charts update their data.

1. **Cards containing all the data, with animation in hovering**

**A screenshot of a weather forecast

Description automatically generated**

Figure Cards for weather forecast, the middle one showing hovering effect

A screenshot of a computer

Description automatically generated

Figure Card implementation

All the numerical values, charts were contained in cards. By hovering mouse on them, the shadow changing animation shows up.

1. **Showing upcomming 7-day weather forecast, while getting icons from Weather API**

**A screen shot of a computer screen

Description automatically generated**

Figure Weather forecast implementation

The javascript code in Fig. 18 transfers series of codes to the html file. It contains numerical value of temperature and humidity from Weather API, weather information in icons of Weather API. The output was illustrated in Fig. 16.

1. **Showing numerical value of 7 different pollutants returned from Openweathermap API while changing the color of the cards**

A screen shot of a computer screen

Description automatically generated

Figure Getting pollutants information for Openweathermap API

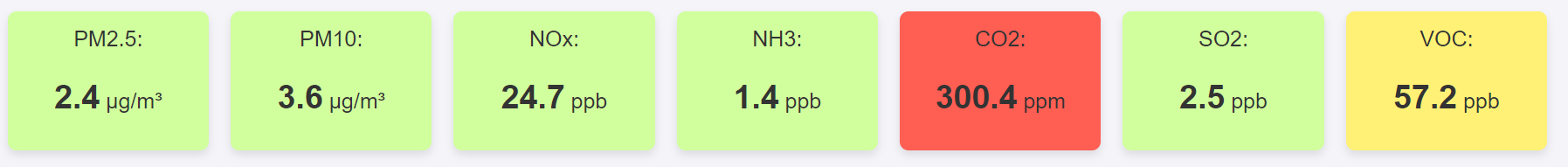


Figure Real-time pollutants level in card format

Openweathermap API provides information about pollutants. Numerical values of pollutants are shown in the card formats (Fig. 20). And the color of the cards change under the standard from U.S. Environmental Protection Agency (EPA).

1. Getting air quality with four standards: Very Good, Good, Normal, Bad

On the header (Fig. 4 & 5), the application shows the air quality status. The status is calculated with the concentration of PM 2.5 under the standard of Korean Ministry of Environment (K.M.O.E).

**A screen shot of a computer program

Description automatically generated**

Figure Air quality indication system under the standard of KMOE

1. **Bar chart containing real-time pollutant level**

**It was implemented in the same manner as the line chart.**

**A screenshot of a graph

Description automatically generated**

Figure Bar chart for real-time pollutant level

1. **Halfed doughnut chart containing real-time pollutant distribution**

**It was implemented in the same manner as line chart. For design, halfed-doughnut was chosen.**

**A graph of pollutant distribution

Description automatically generated**

Figure Halfed-doughnut chart for real-time pollutant distribution

1. **Showing Youtube video explaining “What’s in the air you breathe?”**

**A screen shot of a video

Description automatically generated**

Figure Youtube video explaining pollutants

Youtube video was inserted into the about section of the webpage.



Figure Youtube video insertion in html file

1. **Links for used API and Libraries**

**A screenshot of a computer

Description automatically generated**

Figure Links for used API, libraries

The about section of the webpage also contains the list of used APIs, Libraries.

**A computer screen shot of text

Description automatically generated**

Figure Implementation for links