**Final Project Report**

[Real-time Air Quality Dashboard]

로고, 상징, 폰트, 등록 상표이(가) 표시된 사진

자동 생성된 설명

Web Application Programming

Professor: Aich Satyabrata

Major: Mathematics

Name: Ahn Yuri

Student ID: 202211124

Date: 2024.12.11 (Wed)

1. Introduction

* Project Overview

; I created a Real-time Air Quality Dashboard to help users easily check air quality data and weather. The dashboard displays real-time information like temperature, humidity, and pollutant levels (PM2.5, PM10, NOx, NH3, etc.). I used HTML, CSS, and JavaScript to design a clean and easy-to-use interface. Additionally, I added charts using Chart.js to make the data more visual and understandable.

My main goal was to create a clear and intuitive UI that allows users to view multiple data points at a glance. While making changes or adding features, the code became more complex, leading to frequent errors. To address this, I prioritized ensuring the overall dashboard worked smoothly.

* GitHub Link: <https://github.com/dubuuri/Web_Application.git>
* Demo: https://dubuuri.github.io/Web\_Application/Project
* Dashboard Overview

1. Home page  
   The page is divided into three main parts: Sidebar, Main-Content, and Footer.

텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

|  |  |  |
| --- | --- | --- |
| Sidebar | 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Current-Location: Gets the user's current location and displays the weather. |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Navigation bar:  - Home: Main dashboard  - Visualization: Charts for data  - About page: Information about the  data sources used in the dashboard |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Date & Time: Feature updated time every second and allow users to see the current time and today’s date in real-time. |
| Main  Content | 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Weather Message Container: Displays a message based on the weather conditions in user's location. |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | - City and Date Information: Displays the name of the current city and the date.  - Sunrise and Sunset Information: Visually represents sunrise and sunset times with icons.  - Air Quality Summary: Shows fine dust (PM10) and ultrafine dust (PM2.5) levels using colors to indicate the status. |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Weather Details: Displays the current temperature, weather icon, and weather condition. Provides detailed data on feels-like temperature, precipitation, wind speed, wind direction, humidity, cloudiness, and atmospheric pressure. |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | Air Quality Data: Displays various airpollutant measurements, including PM2.5, PM10, CO, NO, …. |
| Footer | 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | |
| Includes copyright information with the developer's name and a link to the data source, OpenWeatherMap API. | |

1. Visualization Page  
   On this page, you can view various charts (line, bar, and donut) visualizing the data.

텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| Buttons | 텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  자동 생성된 설명  Users can visualize data through charts by selecting the type or data. |
| Visualize  Charts | 텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  자동 생성된 설명  Temperature data for the next 5 days is displayed. Hover effects display detailed temperature and time information. This is same for the humidity chart. |
| 텍스트, 소프트웨어, 컴퓨터 아이콘, 운영 체제이(가) 표시된 사진  자동 생성된 설명  Current air quality data is displayed. Since the data includes gas and particle, I made it into two charts. This is same to the other bar charts. |
| 텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진  자동 생성된 설명  Air quality data is visualized as a donut chart to show the proportions of each element. |

1. About Page  
   This page includes information about the data sources used in the dashboard.

텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |
| --- | --- | --- |
| Data  Descriptions | 텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  자동 생성된 설명  Provides an overview of the data used in the dashboard  and explains each air quality component and its impact on human health. | |
| Developer  Information | 텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  자동 생성된 설명 | You can check information about the developer, including name, GitHub repository link, email address, and location. |

1. Software Tools & API used

|  |  |  |
| --- | --- | --- |
| Chart.js | 텍스트, 소프트웨어, 웹 페이지, 컴퓨터 아이콘이(가) 표시된 사진  자동 생성된 설명 | Used to create responsive charts for visualizing data |
| Font Awesome | 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진  자동 생성된 설명 | To provides icons to improve the design and make the data visually intuitive |
| Google Fonts (Poppins) | 텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진  자동 생성된 설명 | I used Poppins for better User Interface |
| HTML  CSS  JavaScript | IT 기본 지식] 웹 개발의 기본! HTML,CSS,Javascript | 인천일보아카데미 | HTML was used to structure the entire content, CSS to style the layout for UI, and JavaScript to add interactivity to the dashboard. |
| OpenWeatherMap  API | 텍스트, 웹사이트, 웹 페이지, 소프트웨어이(가) 표시된 사진  자동 생성된 설명 | Current Weather API was used to fetch and display real-time weather data. |
| 텍스트, 웹사이트, 웹 페이지, 소프트웨어이(가) 표시된 사진  자동 생성된 설명 | 5 Day / 3 Hour Forecast API was used to show temperature and humidity over 5 days. |
| 텍스트, 웹사이트, 웹 페이지, 소프트웨어이(가) 표시된 사진  자동 생성된 설명 | Air Pollution API was used to display real-time air quality data. |

1. Responsiveness

I moved the sidebar to the top and made the entire dashboard scrollable for easier navigation.

텍스트, 소프트웨어, 웹 페이지, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 웹 페이지, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 텍스트, 소프트웨어, 스크린샷, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

1. What I did !

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명  텍스트, 소프트웨어, 운영 체제, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

-> In donut chart, by default, only the data with higher proportions are shown on the chart. When a label on the right is hidden, the corresponding data disappears from the chart, and the remaining data proportions are recalculated and displayed. If the label is made visible again, the data will reappear on the chart.

텍스트, 소프트웨어, 컴퓨터 아이콘, 스크린샷이(가) 표시된 사진

자동 생성된 설명 텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

-> Whenever I check fine dust levels on Naver, I found the way the text color changed based on the values looks very intuitive. Inspired by this, I implemented a similar approach by creating separate boxes for the most critical air quality data, PM10 and PM2.5. The values and status change color dynamically based on the measurements.

텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

-> When the user clicks the Current-Location button, it fetches weather data based on the user's location. Although I managed to make it work, the biggest challenge was resolving conflicts between this function and another one that triggered on page load using DOM content. Debugging this issue was the hardest part. I also wanted to allow users to select a location and display weather data for that area, but I couldn't implement that feature yet.

1. What I tried …
2. API

I tried to use different APIs besides the OpenWeather API that I had been using before.

* BreezoMeter

I tried to use the BreezoMeter API to get air quality data, including VOC and CO2 levels. To use this API, I needed to connect it to Google Maps, which required adding a payment method to my Google account. However, even after trying different cards, I couldn’t complete the payment setup, so I wasn’t able to use the API.

* RapidAPI

After failing with BreezoMeter, I tried again to find a way to get VOC and CO2 data. Even though there were various API platforms and plenty of weather data options, there were very few APIs for air quality data. Other than OpenWeather, which I was already using, and BreezoMeter, which I couldn’t access, I couldn’t find a suitable API to retrieve the data.

* KaKao Map API

I added a "Current Location" button to display the weather at the user's current location. However, while testing, I noticed that at school, the current location showed as Busan, but at home, it showed as Gijang, a farther area in Busan. I suspected this was due to an error in OpenWeather's longitude and latitude input. To solve this, I tried using the Kakao Map API to get the user's exact location. However, the Kakao API was too complicated to connect, and I couldn't use it successfully.

1. JS

* Current-Location

; Through the "Current Location" button in the sidebar, I aimed to retrieve the user's location and display the weather for that location. While I successfully fetched the weather data and updated all relevant information, the button only worked properly if pressed a few moments after the page had fully loaded. Sometimes it works well, but sometimes not and I couldn’t find the reason why.

1. Conclusion

When I first started this project, I thought I could implement more features for the dashboard. I wanted to allow users to select a location on a map and display the weather and air quality data for that area. However, I realized that writing the code was much more complex than I thought. Even small overlaps in the logic caused immediate errors, making it a very delicate process.

But I feel that my understanding of HTML, CSS, and JavaScript has improved. Now, when I see UI/UX designs in everyday life, I start to notice areas that could be improved. Although there were many challenges and regrets, I’m proud that I persevered and completed the project.