**Weather Dashboard**

**Project Report**

Web Programming

2024/12/09

202355535

Andrea Mena Soriano

1. **Objective**

The objective of this project was to develop a responsive web application to monitor and display real-time air quality data, including temperature, humidity, PM2.5, PM10, NOx, NH3, CO2, SO2, and VOC levels. This data is provided in easy-to-understand charts and comparisons, in order to raise awareness about air pollution in a user-friendly way.

Tools & Technologies Used:

* **HTML**: For structuring the webpage’s content and layout.
* **CSS**: To style the webpage, ensuring a visually appealing and responsive design.
* **JavaScript**: For adding interactivity with the user, data handling, and rendering dynamic content.
* **OpenWeatherAPI:** Used to fetch real-time weather data and forecasts.
* **Chart.js**: For creating visually attractive charts to display the air quality data.
* **Json Location Lists**: To store and access location-based data for weather monitoring from all around the world.
* **SheetJS**: For reading and processing Excel files to integrate data into the webpage.

1. **Webpage Design**

Main Layout:

Captura de pantalla de un celular

Descripción generada automáticamente

**Header:**

Texto

Descripción generada automáticamente

**Navigation Bar:**

Provides three main options: the Home section, Charts and About Us.

Texto

Descripción generada automáticamente

**Main menu:** Home

The home page displays the weather dashboard according to the user’s current location, displaying real-time weather data such as temperature, humidity, feels like temperature, pressure, visibility, wind speed, sunrise and sunset, 5-day forecast, air quality levels, specific air pollutant data, date and time.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

**Main menu:** Charts

The charts menu allows the user to choose between three types of charts to visualize different air quality data with a visually appealing design:

* **Line Chart**: Trends for temperature and humidity.
* **Bar Chart**: Pollutant levels (PM2.5, PM10, NOx, NH3, CO2, SO2, VOC)
* **Doughnut Chart**: Distribution of pollutants.

Respectively, the line chart has a dropdown bar to choose a date range between four options: one day, one week, one month or one year; and the bar and doughnut charts also have three different options to choose from: all pollutants, particulate matter (PM2.5, PM10) or gaseous pollutants (Nox, NH3, CO2, SO2).

The specific values of the data are only shown once hovering over the chart, ensuring user interactivity.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Gráfico, Gráfico de proyección solar

Descripción generada automáticamente

**Main menu:** About

The "About Us" section provides an overview of the website's mission to raise awareness about air pollution with real-time air quality data. It highlights the importance of monitoring air quality for health, vulnerable populations, and the environment, supported by visuals for better understanding. It also educates users on key pollutants like PM2.5, NOx, and SO2, and includes an engaging animated video to further explore air pollution sources.

Interfaz de usuario gráfica

Descripción generada automáticamente

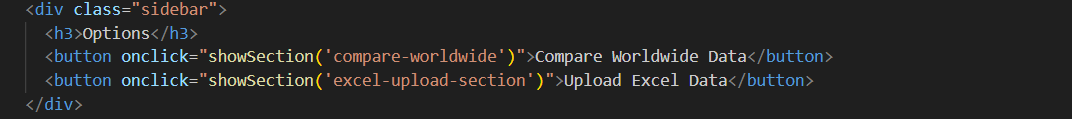
**Sidebar:**

The sidebar provides two options:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* **Compare Worldwide Data:** to compare and visualize the differences between the air quality data of different cities around the world, the user can choose between four different charts: bar, line, radar and pie chart.
* **Upload Excel Data:** the user can directly upload their Excel file that contains air pollutant data to visualize it with a pie chart and a polar area chart.



Compare Worldwide Data:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

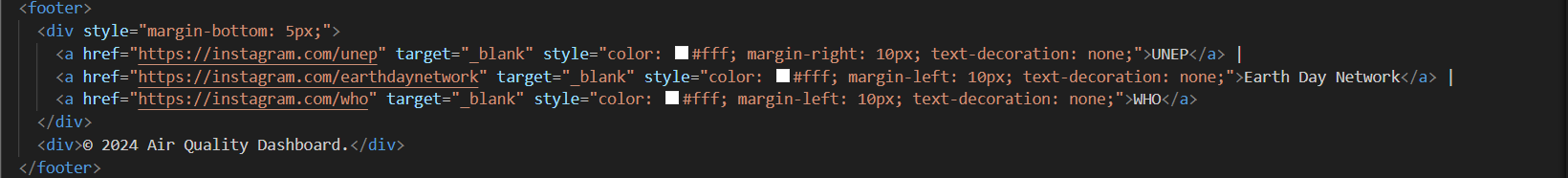
Upload Excel Data:

Gráfico

Descripción generada automáticamente

**Footer:**

The footer provides various links to the Instagram of three different accounts that are related to air pollution and climate change so the user can explore related information and learn more about it.





1. **Data Handling & Real-Time Updates**

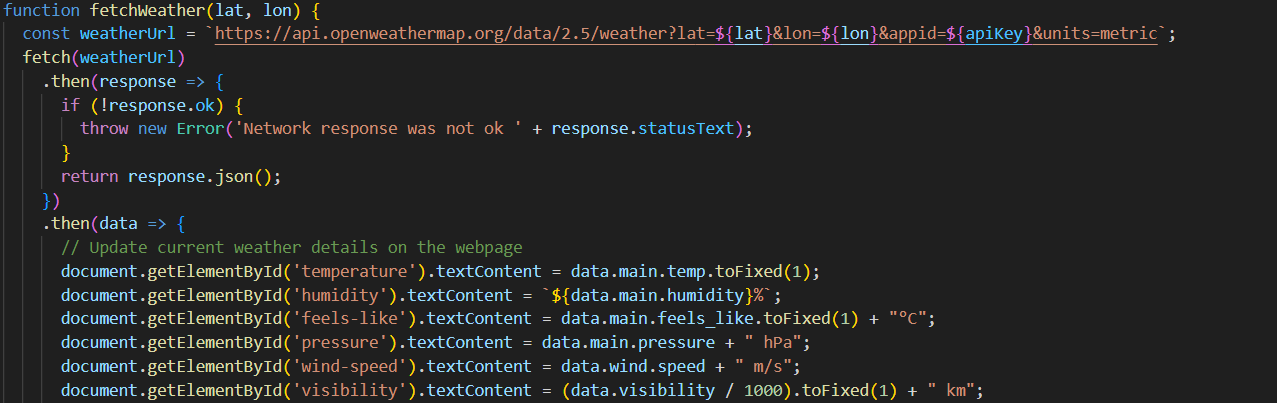
Main Weather Dashboard:

I integrated geolocation to get the user’s current location when first loading the webpage. This ensures that the data displayed in the weather dashboard is local.

Texto

Descripción generada automáticamente

The real-time air quality data dynamically updates through API calls that take the latitude and longitude values of the current location.



Charts:

By using Chart.js, the charts dynamically update to visualize air quality data and respond to user interactions such as filter changes.

Chart Update Functions:

Texto

Descripción generada automáticamente

Dynamic Rendering of Charts:

Texto

Descripción generada automáticamente

Compare Worldwide Data:

This comparison feature allows users to compare air quality data between two cities, using input validation and API requests for accurate and reliable data. It integrates multiple datasets for comparative analysis.

City Selection and Validation:

Texto

Descripción generada automáticamente

Fetching Data for City Comparison:

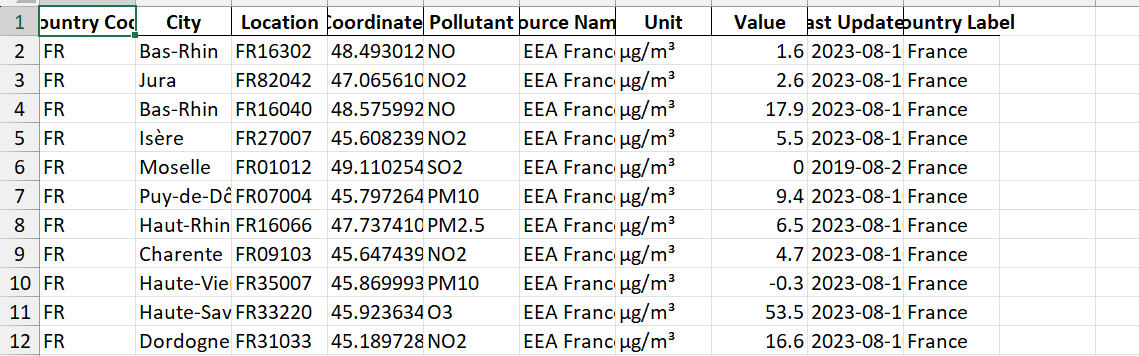
Texto

Descripción generada automáticamente

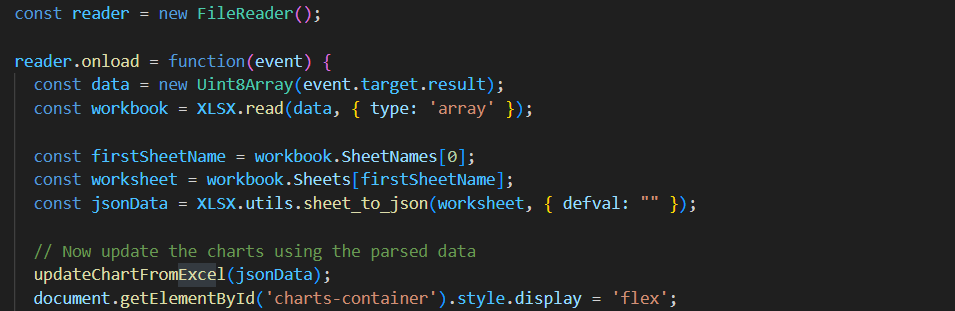
Process Excel File Data:

This feature allows the user to upload an Excel file for further analysis. The uploaded file is processed to extract and display the average of relevant air pollutant data. SheetJS was used to process, parse the Excel file and integrate the data into the webpage.

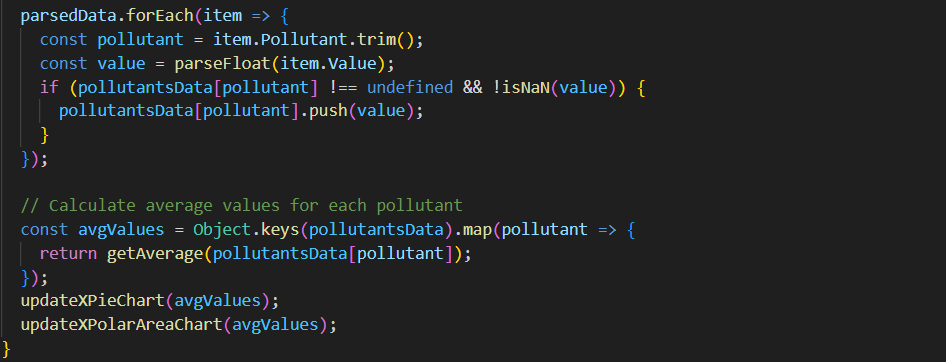
Air Pollutant Data Excel Example :



Excel File Parsing:



Processing Parsed Data:



1. **Challenges & Successes**

Real-time updates:

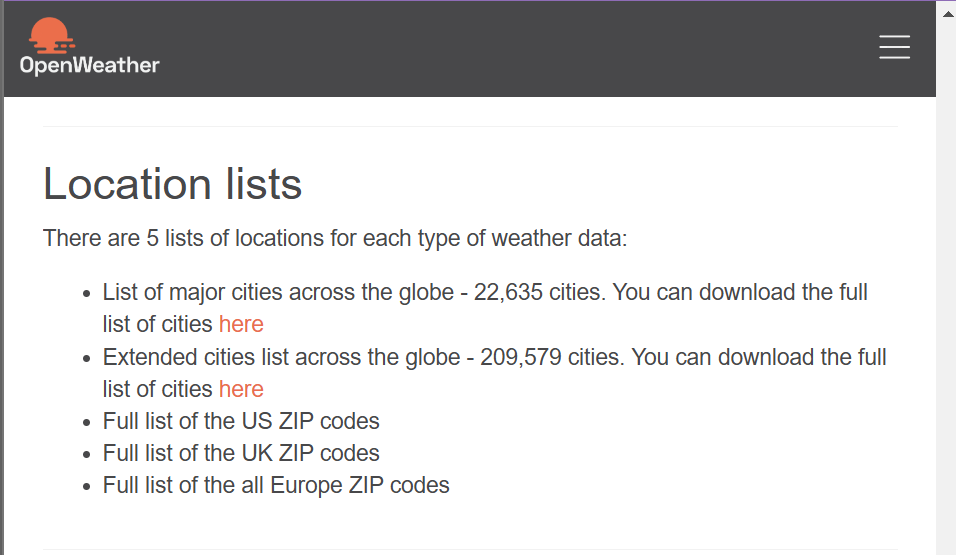
When I first created my webpage, the weather data displayed only corresponded to my current location, Busan. However, after implementing Geolocation in my JavaScript code to obtain the user’s location, the real-time updates were not showing on the weather dashboard anymore. After doing some research, I realized I needed to add ‘Live Server’ because the problem was that the dynamic content handling needed a local server to function properly.

Interfaz de usuario gráfica, Texto

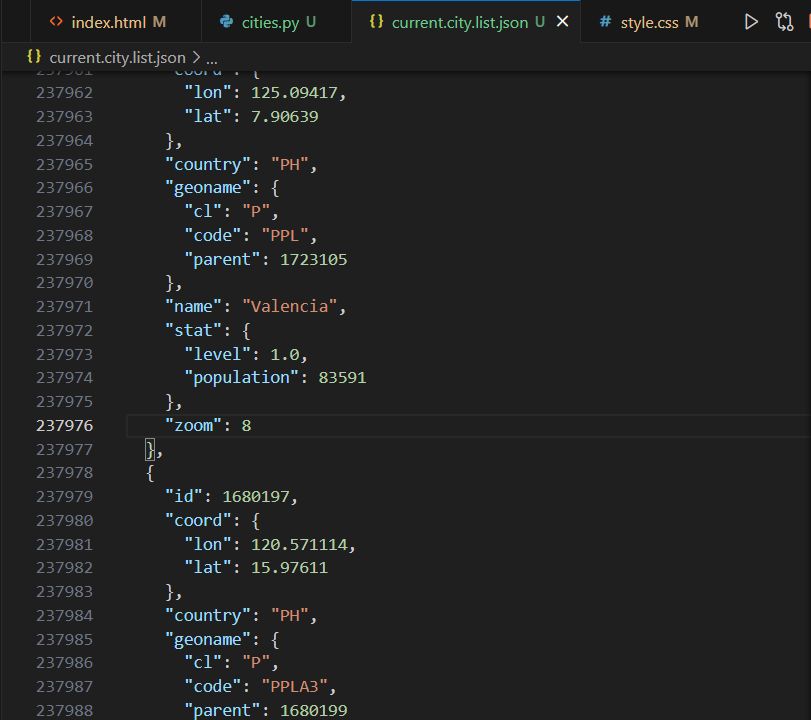
Descripción generada automáticamente

Compare Worldwide Data:

For this feature I wanted to implement data from cites from all around the world to be able to compare the air pollutant data of any two cities in the world. I found Json Cities Lists from OpenWeatherAPI that offered free weather data from more than 200,000 cities.



However, this Json list was very extensive: it contained many extra data about each city such as population data that I did not need for this project, and it also was not in the format that I wanted.



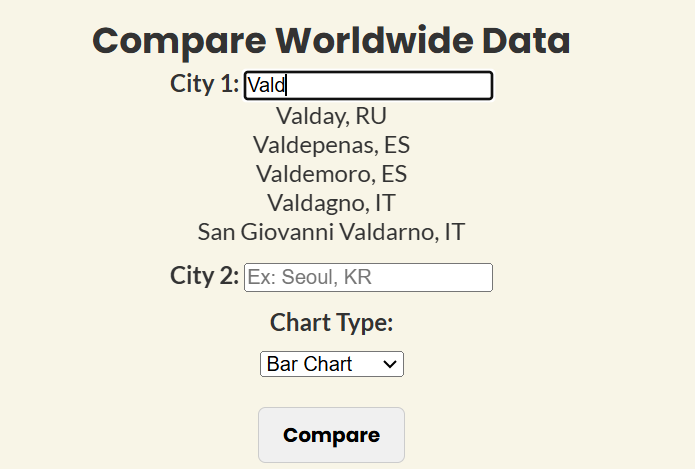
Therefore, using Python I simplified the cities list to get only the information I needed from each city: ID, name, country, latitude, longitude.

Texto

Descripción generada automáticamente

This was the final Json list that I integrated in my JavaScript code:

Texto

Descripción generada automáticamente

Finally, in the form where the user inputs the cities, I used the data I had simplified to show suggestions of city names from all over the world.

CSS Styling:

There were many times when, although I was getting the correct data output that I wanted on my webpage, it was not visually appealing or one specific feature was not aligned to my liking. Therefore, I also spent a lot of time editing the CSS of my webpage because I wanted the webpage layout and the charts to be clear and concise but at the same time aesthetically pleasing.

Texto

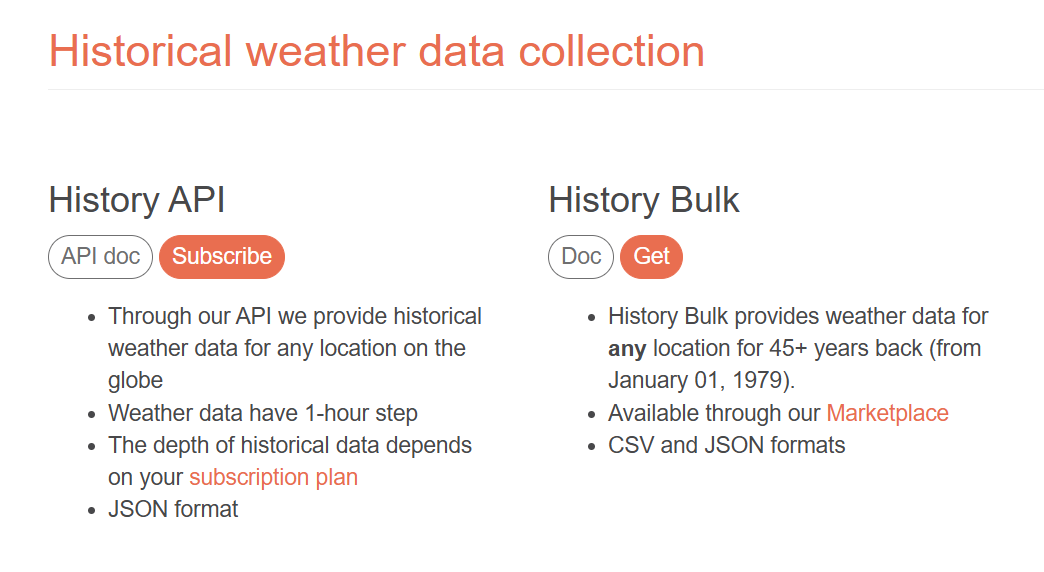
Descripción generada automáticamenteTexto

Descripción generada automáticamente

1. **Feature Failures**

Historical Data Comparison:

Initially, I was going to have two different options to compare data on the sidebar: Compare Worldwide Data and Compare Historical Data. For the historical data, I wanted to visualize the comparison of the current air pollutants data of one region, with the data of one, five or ten years ago. I thought that representing this comparison with charts would be a good way to realize the changes in air pollution from the past until now. However, most of the APIs that I found only had free weather data as old as two weeks earlier. In the case of OpenWeatherAPI, the historical weather data was not free either.



Therefore, I gave up on this feature and decided to add the Excel file upload option that I had not initially planned.

Unit Toggle:

I initially also wanted to add a Unit Toggle option between ºC/ºF and ppm/µg/m³, as they are the most frequently represented units in the air quality charts. However, I tried to add this feature after I had most of my code ready and I realized that I should have integrated it in the beginning of my project because it would take me a lot of effort to go back and edit most of the chart code to make the unit toggle work. I also came to the conclusion that this would not be a very useful feature for my webpage, so I decided not to add it in the end.

Real-time CO2 Data:

Finally, I also attempted to integrate real-time CO2 data and show future predictions of CO2 levels using machine learning based on historical data. I had established a structure for fetching and visualizing the data, but I was unfortunately unable to implement it due to Cross-Origin Resource Sharing (CORS) policy restrictions, which blocked direct requests from my front end to the API.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Despite exploring potential solutions, such as setting up a proxy server to bypass the CORS issue, I was unable to fully resolve the problem within the project due date. As a result, I was unable to incorporate live data into the project as planned.

1. **Conclusion**

This project aimed to develop a responsive web application for monitoring air quality data, meeting requirements such as user interface design, data visualization with charts, real-time updates, and web accessibility. While several aspects were successfully implemented, including a visually appealing layout, dynamic charts for data visualization, and simulated data integration, there were also some challenges that remained unresolved.

Despite these setbacks, I can say that I have learned a lot throughout the process. This project deepened my understanding of web development concepts, including front-end design, working with APIs, and implementing dynamic data visualization Overall, this project has been a valuable experience that helped me improve my problem-solving skills and gain a better understanding of building interactive and functional web applications.

**Webpage Demo Link**: <https://andreamna.github.io/air_quality/>