Air Quality Monitoring dashboard

Computer Science Engineering

Web Application Programming

202255635 Yerbol Janbubu

2024.12.09

**Objective**

Create 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. The application will utilize HTML, CSS, and JavaScript

**Enhancements and Additions**

**1.**Displays today's temperature alongside comparisons of other locations, visualized in an area graph.

**2.** Pollutant levels are displayed in a semi-doughnut graph, indicating air quality levels through intuitive color-coded measures.

**3.** Shows the user's location using a map for better geographical context.

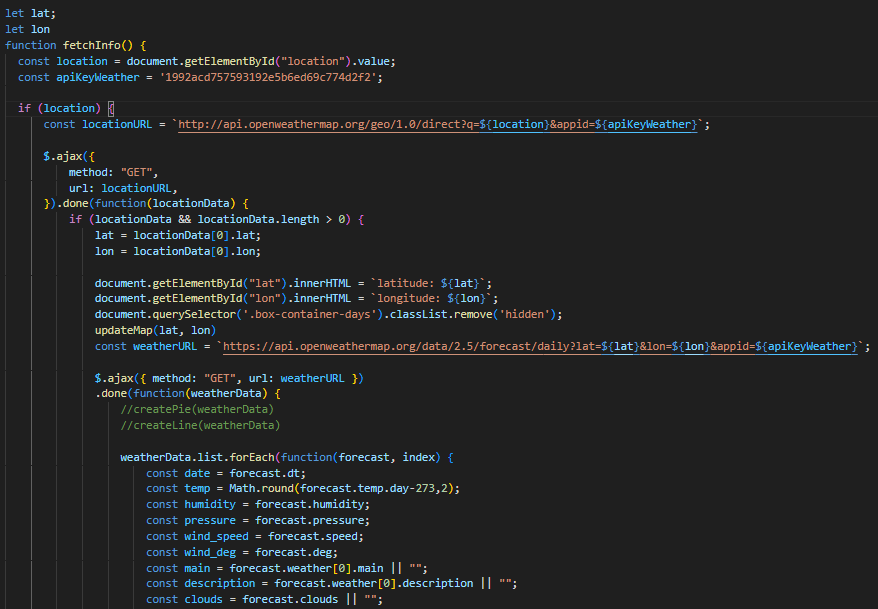
**4.** Reorganized sections into multiple pages navigable via a navbar, replacing the original single-page scrolling design.

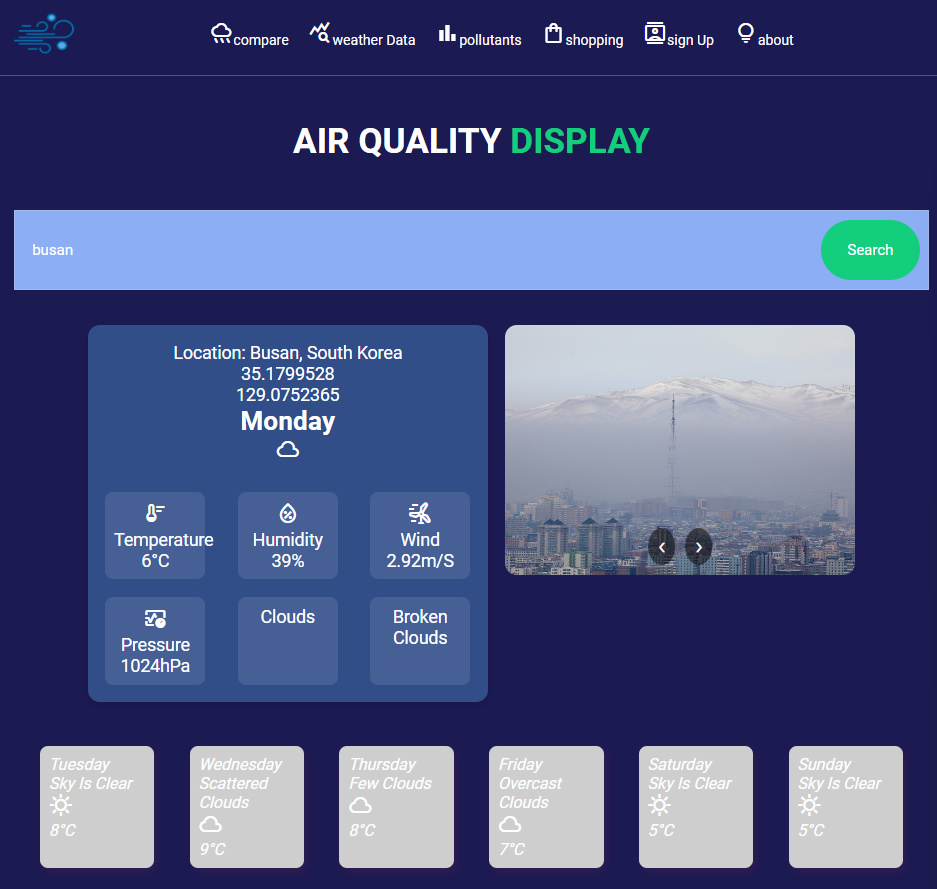
**5.** Introduced a signup section with form validation using jQuery and a products section.

**How it works**

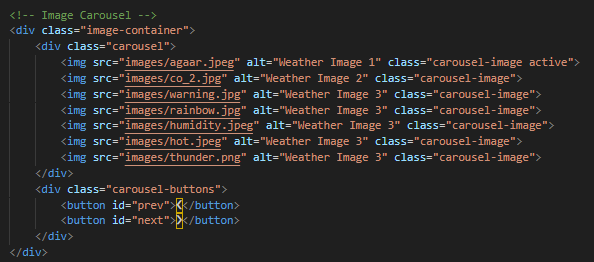
1. **Weather Data and Forecast**

* Based on user input, fetches and displays upcoming weekly weather forecasts, including temperature, descriptions, humidity, and weather icons.



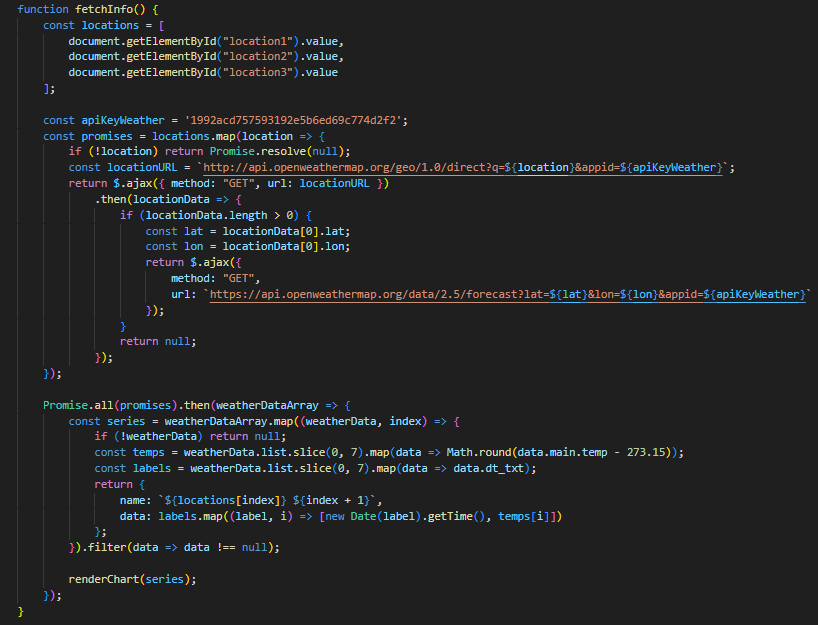


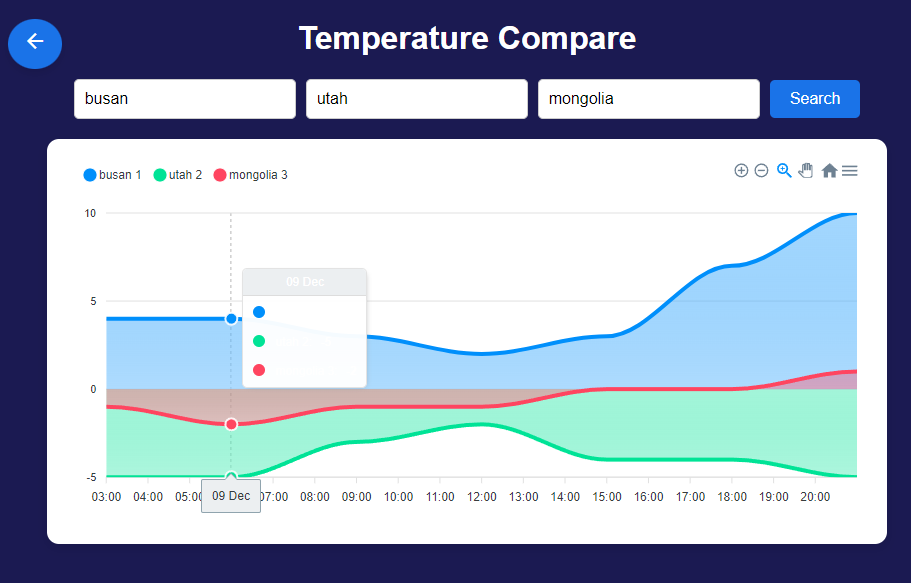
* A carousel showcases related weather images.

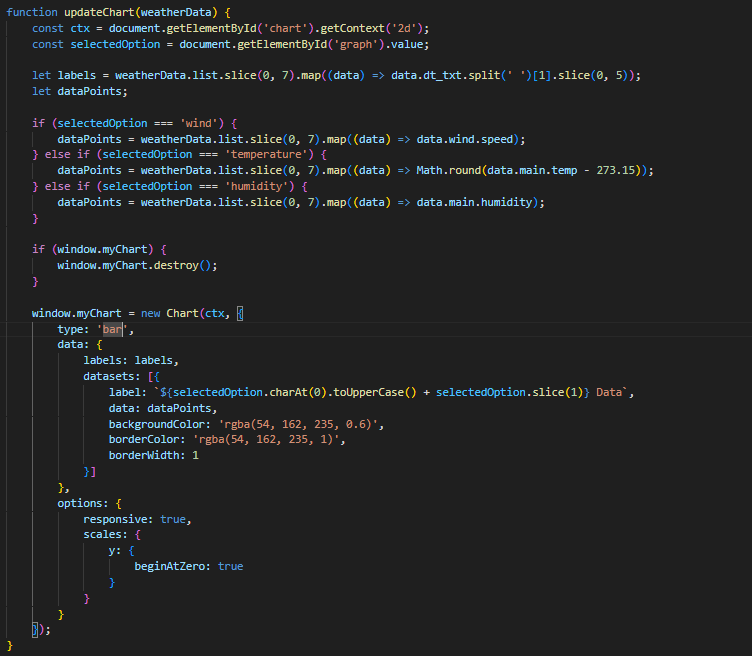


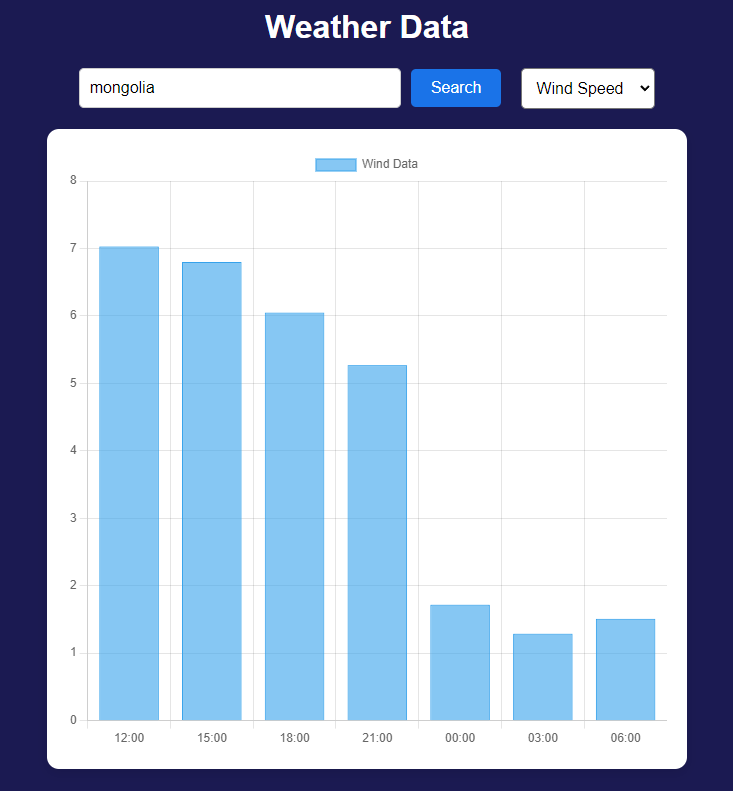
1. **Forecast Comparison**

* Enables comparison of weather data for up to three countries, presented in an area graph.



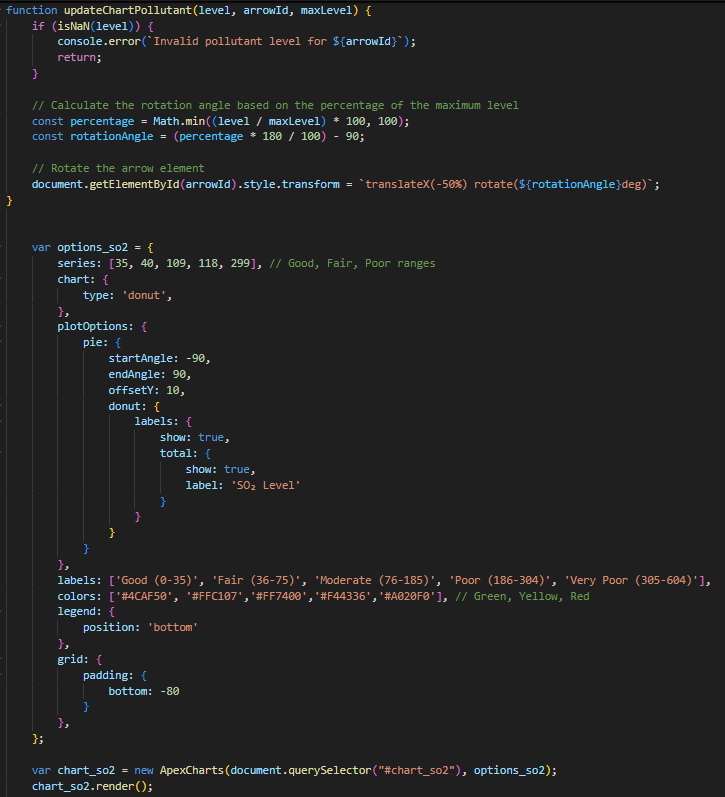


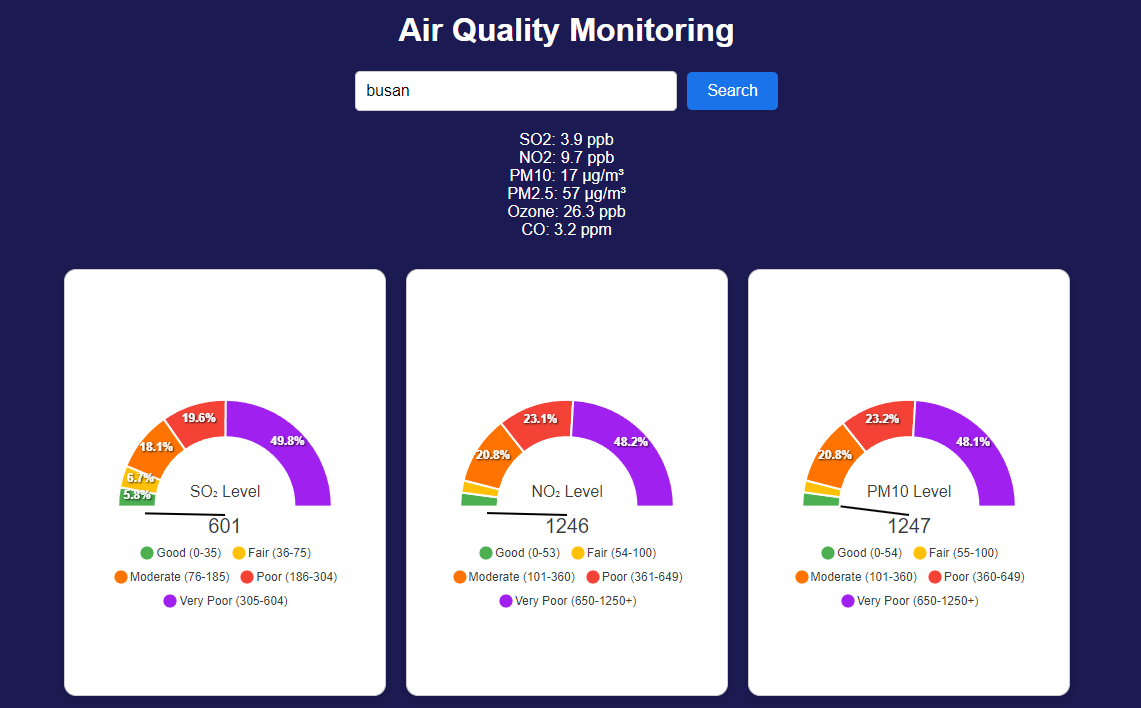
* Additional bar and line graphs show temperature, humidity, and wind speed comparisons.



1. **Air Quality Monitoring**

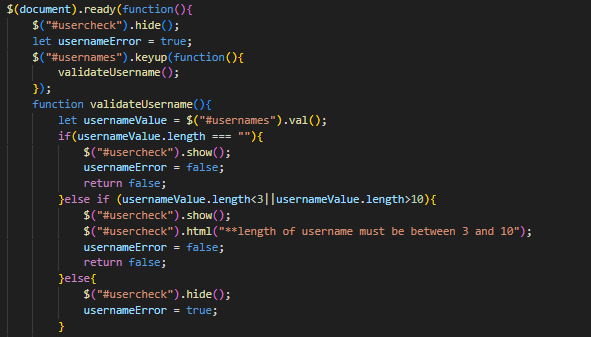
* Displays pollutant levels using a semi-doughnut (speedometer-style) graph.
* Hazard levels are color-coded, and an arrow marks the specific location's pollutant level for clarity.

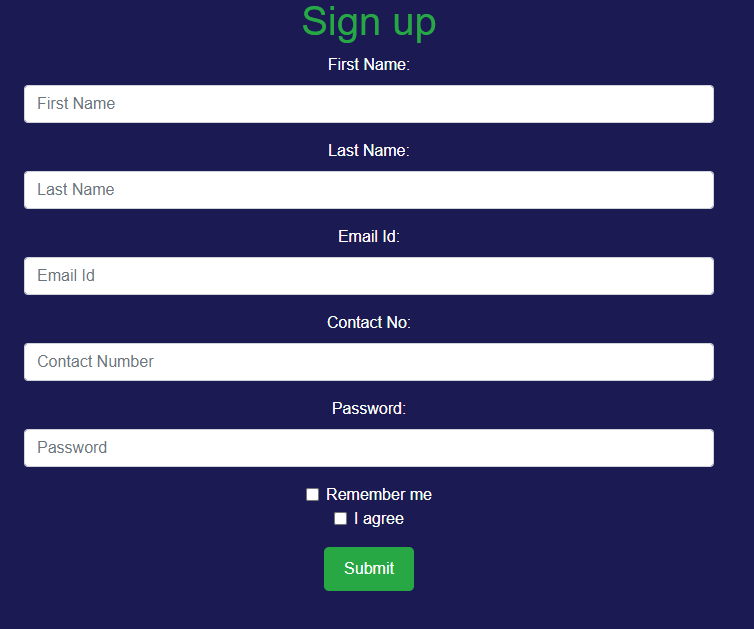




1. **Signup and Product Features**

* Includes a signup form with Bootstrap and jQuery-based validation.





* The products section highlights additional offerings



**Technical Implementation**

* **Libraries and APIs:**
  + Graphs are created using ApexCharts and Chart.js.
  + Google API icons are used for UI styling.
  + Leaflet.js integrates the map for location visualization.
* **Data Integration:**
  + Utilizes the OpenWeatherMap API to fetch weather and pollutant data.
  + Supports location input in any language or character format to determine coordinates and weather details.
  + The weekday format of the forecast was simplified using custom functions.
* **User Experience:**
  + Temperature and pollutant data are visualized intuitively for quick understanding.
  + Semi-doughnut graphs provide an effective visual indicator of air quality levels.
  + Interactive carousels and sectioned navigation enhance user engagement.

**Challenges**

* **Initial Design Complexity:**
  + The single-page design was difficult to navigate, leading to a restructured multi-page format with a navbar.
* **Less Polluted Area Data:**
  + Finding an API to identify the least polluted areas proved challenging. Kakao Map API was explored but found overly complex.
* **Data Segmentation:**
  + Weekly and hourly data integration in one section was challenging, so these were separated into distinct views.

**Conclusion**

This project was a valuable learning experience in web development, including:

* API integration and managing responses to display key data effectively.
* Creating user-friendly designs with intuitive navigation and visualizations.
* Leveraging GitHub for version control and project documentation.

Through this process, I gained a deeper understanding of front-end development and improved my ability to design engaging and functional web applications. I am motivated to continue learning and undertake more personal projects to enhance my skills.

https://janbubu247.github.io/web-project/index.html