Air Quality Monitoring and Visualization using Node-RED, InfluxDB, and Grafana

Kandanoor Deepika Sahi
56800
Westsächsische Hochschule Zwickau
Zwickau, Germany
Deepika.Kandanoor.odz@fh-zwickau.de

Saijaya Chilekampalli
50731
Westsächsische Hochschule Zwickau
Zwickau, Germany
Saijaya.Chilekampalli.nht@fh-zwickau.de

Srikar Ravula Reddy
56831
Westsächsische Hochschule Zwickau
Zwickau, Germany
Srikar.Ravula.oa0@fh-zwickau.de

ABSTRACT
This project focuses on building an Internet of Things (IoT) system to monitor air quality in five German cities: Berlin, Munich, Dresden, Essen, and Zwickau. Our main goal is to show air quality data clearly and effectively, which is crucial for public health and environmental decisions.

We used a three-part open-source system: Node-RED to collect and prepare the data, InfluxDB to store this data efficiently over time, and Grafana to create detailed visual dashboards. We simulated hourly air quality data, including particles (PM10, PM2.5), carbon monoxide, carbon dioxide, dust, methane, and UV index, using Node-RED. This data was then saved in InfluxDB. Grafana then connected to InfluxDB to display this information on a comprehensive dashboard. This dashboard includes maps, time-series graphs, bar charts, heatmaps, gauges, pie charts, and histograms, each designed to show specific pollutants.

Our system successfully demonstrated how air quality data flows from collection to dynamic visualization. The Grafana dashboard effectively shows hourly changes, daily averages, and comparisons between the five cities for all tracked pollutants. We gained insights into peak pollution times, average carbon dioxide levels, dust ranges, and comparative UV indices. Using a simple color code (green for good, yellow for moderate, red for unhealthy) provides immediate understanding of air quality. This project proves that our open-source setup is a practical solution for environmental monitoring and understanding real-time data. It provides a clear picture of urban air quality, helping support informed decisions.

Keywords: IoT, Air Quality, Node-RED, InfluxDB, Grafana, Data Visualization, Environmental Monitoring, Time-Series Database.

1 INTRODUCTION
The air we breathe is vital for our health and the environment. As cities grow and industries develop, knowing what's in our air becomes more important than ever. This project set out to create an Internet of Things (IoT) system that monitors and visualizes key air quality information in real-time. By using free and open-source tools, we aimed to build a reliable and easy-to-use system that gives immediate insights into the air conditions in specific German cities.

Throughout this report, we'll answer questions like:

- How can we gather and process real-time air quality data efficiently?
- Which open-source tools are best for building such an IoT monitoring system?
- How can we turn complex environmental data into clear and useful visuals for everyone?
- What can we learn by tracking pollutants like particulate matter, carbon gases, dust, methane, and UV radiation across different German cities?

This project is a first step toward helping communities and officials use data to protect our environment and improve public health.

2 OBJECTIVES
Our main goals for this IoT Air Quality Monitoring and Visualization project were:

1. **Set up Data Collection:** Create a system in Node-RED to either fetch real-time or simulate air quality data for Berlin, Munich, Dresden, Essen, and Zwickau.
2. **Process and Store Data:** Design Node-RED workflows to clean and format air quality data, then store it efficiently in InfluxDB, a database built for time-stamped information.
3. **Build a Dashboard:** Create an interactive dashboard in Grafana to display various air quality details (PM10, PM2.5, carbon monoxide, carbon dioxide, dust, methane, UV index) using clear and diverse visuals.
4. **Show Real-Time and Forecast Data:** Display hourly air quality data and provide a 2-days forecast to help assess environmental conditions quickly.
5. **Enable City Comparisons:** Make it easy to compare air quality metrics across the five selected German cities.
6. **Add Status Indicators:** Use a simple color-coding system (green, yellow, red) to show whether air quality is good, moderate, or unhealthy.

3 METHODOLOGY
For this project, we chose a popular set of open-source tools to build our IoT system. This approach offers flexibility, saves costs, and provides powerful features for monitoring the environment in real-time.

- **Node-RED:** We picked Node-RED because it's an easy-to-use platform that lets you connect devices and services using a visual, "flow-based" programming style. It was perfect for collecting data from APIs and transforming it into the right format.

- **InfluxDB:** This database is specially designed for time-series data, meaning it's great at handling large amounts of data that come with a timestamp, like sensor readings. Its efficiency in storing and querying this data is key for our real-time visualizations.
- **Grafana:** We used Grafana for our dashboard because it's a powerful tool for creating interactive displays. It offers many types of charts and graphs and works seamlessly with InfluxDB, helping us turn complex data into clear, actionable visuals.

While our Node-RED setup could fetch live data, we also simulated some hourly data using random values. This allowed us to develop and test the system continuously without constantly relying on the live API. This way, we ensured our data pipeline worked well under various conditions.
In short, Node-RED either fetched or created hourly air quality data for each city, transformed it, sent it to InfluxDB, and then Grafana pulled this data from InfluxDB to create our dynamic and insightful dashboards.

4 DATA COLLECTION & DATA PROCESSING

To retrieve weather data for five different cities in Germany, a workflow was initially designed for a single city using Node-RED (see Figure 1). This workflow was then replicated and customized for each of the five cities. The Node-RED workflow is designed to retrieve air quality data from the Open Meteo API [1], process the data,vand store it in an InfluxDB database. Below is a detailed explanation of the process and the workflow structure, which consists of the following nodes arranged in a sequential manner:

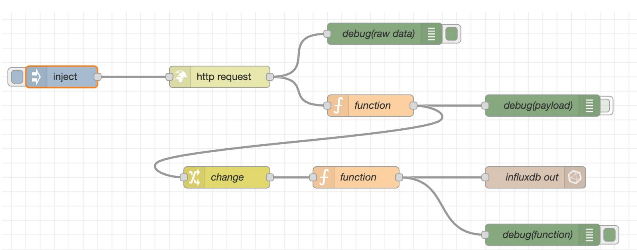

Figure 1: Node-RED Workflow - Fetch Live Air Quality Data of a City

- Inject Node: which is used to obtain data at 1-hour intervals.
- Http Request Node: To retrieve the data, a URL must be included
in this node. The URL is structured as follows:

```
https://air-quality-api.open-meteo.com/v1/air-quality?latitude={lati}&longitude={long}{parms}
```

where {LAT}, {LON}, {PARAMS}, should be replaced with the respective details of interests.
The geographical coordinates of the selected cities are presented in Table 1.

City	Latitude	Longitude	Key Charecteristics
Berlin	52.5244	13.4105	Generally good
Munich	48.1374	11.5755	Usually good
Dresden	51.0509	13.7383	Variable, often good
Essen	51.4566	7.0123	Industrial influence
Zwickau	50.7272	12.4884	Generally good

Table 1: Latitude, Longitude, and Key Characteristics of Selected Cities

- Function Process Data Node: Used to parse and format the API response (e.g. Dust, Corbondioxide, UV Index, etc.).
- Change Node: Maps the msg.payload.timestamp to global.timestamp
with deep copying enabled, ensuring accurate and consistent times-
tamps for time-series storage in InfluxDB.
- Function Prepare Data for Influx Node: Formats the parsed and modified data into the specific structure required by InfluxDB.

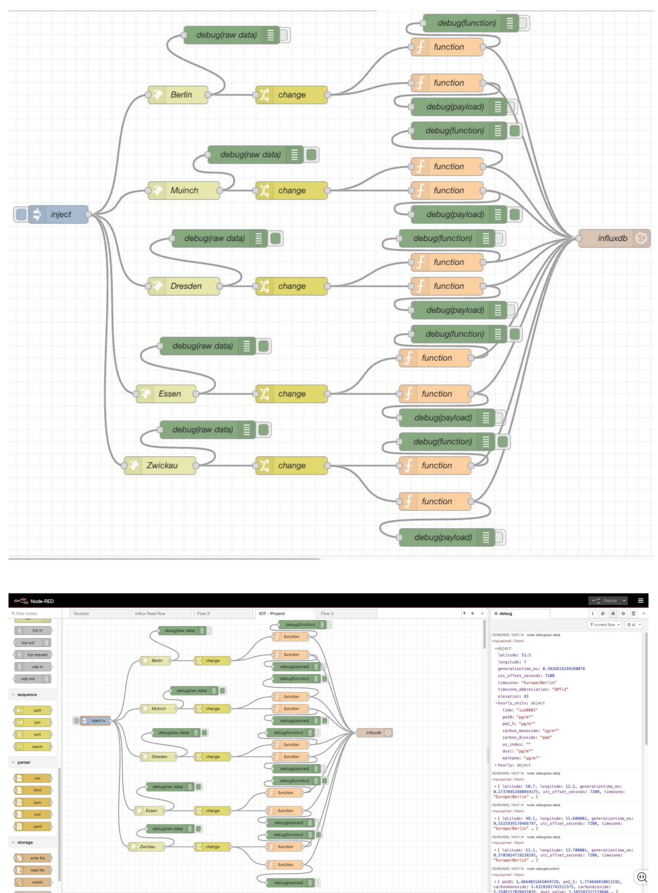

Figure 2: Node-RED Workflow - Fetch Data of 5 Cities

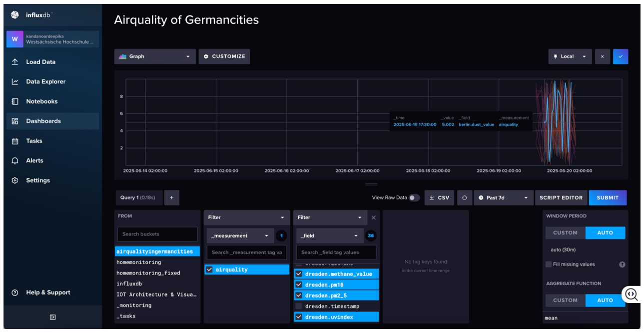

Figure 3: Storing Data in InfluxDB

Through the Influx Out node, we send the data to InfluxDB, where it is stored. As shown in the (Figure 3), the data is organized inside its respective bucket in InfluxDB. We have all the previous mentioned weather indicators stored for each city.

Finally, to achieve better visualization results on the dashboard, we decided to transfer the data to Grafana and proceed with the design phase. To connect InfluxDB with Grafana, Grafana was con-figured to use InfluxDB as a data source by specifying the InfluxDB URL, database name, and authentication details in the data source settings. Once the connection was established, dashboards and visualizations were created to display the data stored in InfluxDB.

5 DASHBOARD DESIGN

5.1 Introduction

This section provides an overview of designs used for elements of dashboards that address the purpose of selection of designs of the visualization of data. The objective of the dashboards is to present Air quality data of a certain time in an understandable man-ner allowing for easy interpretation and quick analysis. Another purpose of better dashboard design is to allow the user to derive results from the trends and patterns of data for whatever reason. The design process includes the selection of appropriate layouts, charts, colors, shapes, etc. depending on the type of data presented. The dashboard contextualizes the weather factors with these de-signs. The continuation of this section will include the designs that we used in the context of the project and the reasoning behind it.

5.2 Dashboards

Below are a few examples of different types of visualization charts and maps used in our dashboard. Each element has a reason for selection based on readability and the type of data being presented.

Geographical Map of 5 Cities

Figure 4: Geographical Map of Cities

This panel shows where our monitoring stations are. Each of the five German cities (Berlin, Munich, Dresden, Essen, Zwickau) is marked with a red dot. If you click on a dot, you can see the exact longitude and latitude coordinates of that city.

PM10 & PM2.5 of All Cities

Here, we have detailed graphs showing the hourly levels of PM10 and PM2.5 particles for all five cities. The measurements are in micrograms per cubic meter (µg/m3). Each city has its own color, making it simple to compare how these particles change hour by hour over a 2-day forecast. These graphs quickly show the highest and lowest concentrations for each city. We also have separate, more detailed graphs for PM2.5 and PM10 for each individual city.

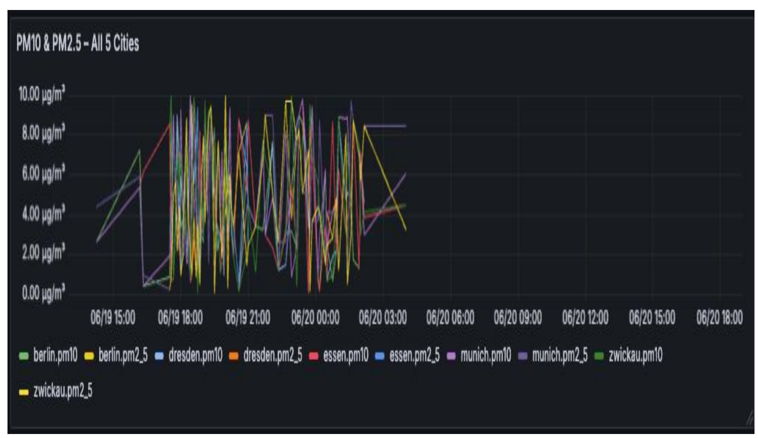

Figure 5: PM10 and PM2.5 Time-Series Graphs 5 Cities

Carbon Dioxide of All 5 Cities

We use a bar chart to display the average carbon dioxide (CO2) levels for each of the five cities. Each bar represents a city, with colors showing variations in CO2 . The x-axis shows different dates, highlighting the highest average values, and the y-axis is in micrograms per cubic meter (µg/m3).

Formula for Average CO_2 :

Average CO_2 =N1 i=1∑N $CO_{2,i}$

where N is the number of data points over a given period, and $CO_{2,i}$ is the carbon dioxide concentration at point i.

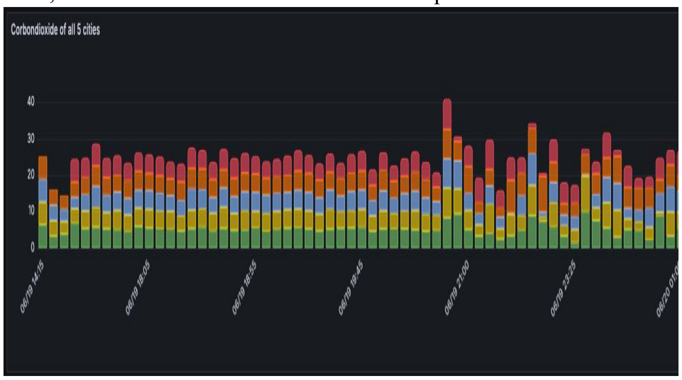

Figure 6: Carbon Dioxide Bar Chart 5 Cities

Dust of All 5 Cities

This panel uses a heatmap to show dust concentrations across the five cities. It effectively displays dust values ranging from 0.201µg/m3 to 9.71µg/m3. The x-axis shows hourly differences, and the y-axis maps the color intensity to each city. Hotter colors mean higher dust levels, giving a clear visual of how dust varies throughout the day and in different locations.

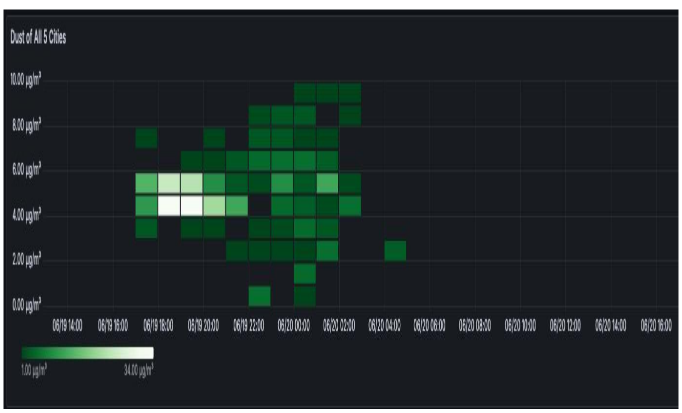

Figure 7: Dust Heatmap of 5 Cities

Carbon Monoxide of All 5 Cities

Carbon monoxide (CO) levels are shown using a "Stat" graph. This graph displays current or summarized CO values for all cities and includes threshold lines to quickly see if levels are safe. Maximum values are shown in micrograms per cubic meter (µg/m3). This visual is great for a quick check of the highest carbon monoxide concentrations.

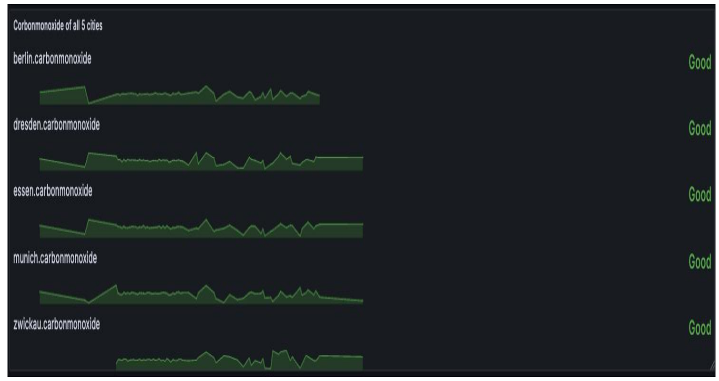

Figure 8: Carbon Monoxide Stat Graph 5 Cities

PM10 of 5 Cities

Individual gauge meters are used to show PM10 levels for each city. These gauges are excellent for showing a single measurement against set limits. For instance, Berlin shows 4.6µg/m3, Dresden 4.5µg/m3, Essen 4.5µg/m3, Munich 6.1µg/m3, and Zwickau 4.5µg/m3. The gauges visually represent these values, often with color zones (good, moderate, unhealthy) in the background.

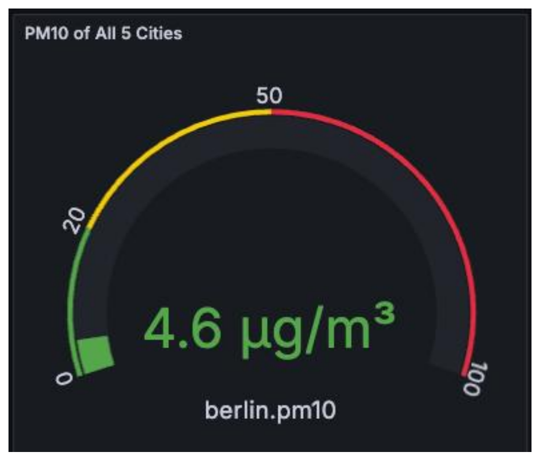

Figure 9: PM10 Gauge Meters 5 Cities

UV-Index of All 5 Cities

The UV Index for all five cities is shown in a pie chart. This chart clearly displays the proportion of UV index values across the cities based on a set threshold. Berlin accounts for 35%, Munich 29%, Dresden 12%, Essen 12%, and Zwickau 12% of the total UV index. This panel quickly highlights which cities have higher UV radiation.

Formula for Percentage of UV Index:

City UV Index Percentage=∑All Cities UV Index ValuesCity UV Index Value ×100%

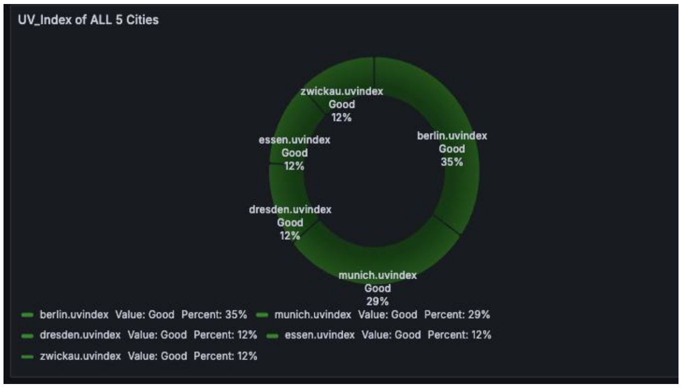

Figure 10: UV Index Pie Chart 5 Cities

Methane of All 5 Cities

A histogram is used to visualize methane concentrations across the five cities. Each city is shown with a different color within the histogram bars, allowing for quick visual comparison of methane levels. The graph also indicates a threshold value, with units in micrograms per cubic meter (µg/m3), helping understand the amount and spread of methane.

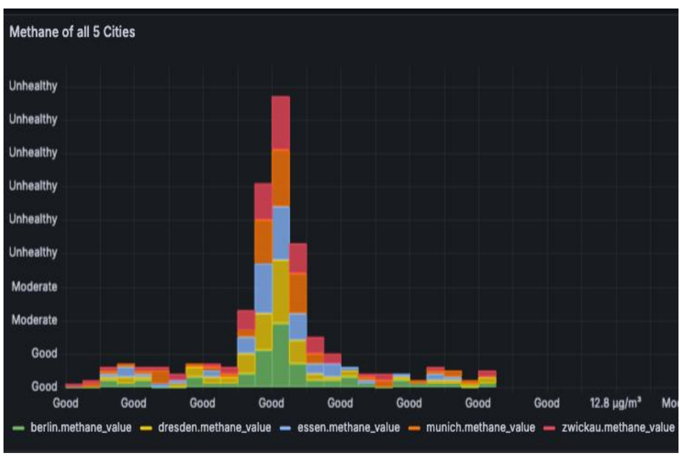

Figure 11: Methane Histogram 5 Cities

Complete Dashboard and Threshold Values

The final dashboard is a complete and well-organized display of hourly air quality reports for the five German cities. By combining various visuals of time series, heatmaps, gauges, pie charts, and stat graphs. It offers diverse insights into the data. We strategically chose each visualization type to best present the characteristics of each pollutant.

A key feature is the color-coded system based on air quality thresholds:

- Green: Means air quality is Good.
- Yellow: Means air quality is Moderate.
- Red: Means air quality is Unhealthy.

Figure 12: Thershold of value mappings

This system provides an immediate visual signal about the air quality status, making complex data easy for anyone to understand. Each city keeps the same color throughout the dashboard to avoid confusion.

5.3 Provisions of The Journey

Several additional dashboards were created during the project, each designed to explore different aspects of the air data with unique visualization techniques within the cities. These dashboards provided insights into specific trends, such as Hourly variations, dust, Corbondioxide, UV Index across different time frames. While not all of them were included in the report, they played a valuable role in refining our approach, testing different visualization styles, and ensuring that the final dashboard effectively captured the most relevant air patterns.

6 RESULTS

Our project successfully created a working IoT system for air quality monitoring, complete with a comprehensive Grafana dashboard. Here are the main results:

1. **Verified Data Flow:** We successfully demonstrated that air quality data could be captured (or simulated), processed in Node-RED, and reliably stored in InfluxDB, confirming the entire data pipeline worked.
2. **Real-Time View:** The Grafana dashboard provided an almost real-time view of air quality parameters across the five German cities, with hourly updates showing changing conditions.
3. **Thorough Air Quality Assessment:** The dashboard effectively displayed all the chosen air quality parameters:
   - **PM10 & PM2.5:** Time-series graphs clearly showed hourly trends and allowed easy comparison of particulate matter levels between all five cities over two days.
   - **Carbon Dioxide:** Bar charts provided insights into average carbon dioxide levels, helping to identify cities with higher or lower concentrations.
   - **Dust:** The heatmap uniquely visualized dust concentration patterns, highlighting periods and cities with higher dust.
   - **Carbon Monoxide:** "Stat" graphs showed current and maximum recorded levels, important for immediate safety checks.

- **PM10:** Individual city gauges offered a quick, color-coded assessment of PM10 levels against healthy limits.
- **UV Index:** The pie chart provided a clear visual comparison of UV radiation exposure across the cities.
- **Methane:** The histogram showed the distribution of methane concentrations, indicating typical levels and any significant increases.

4. **Actionable Information:** The use of color-coded thresholds (green for good, yellow for moderate, red for unhealthy) made the dashboard easy to understand and use, even for non-technical users.
5. **Proof of Scalability:** The modular design in Node-RED and InfluxDB's time-series capabilities showed that the system could easily be expanded to include more cities or different sensors in the future.

The project successfully met all its goals, proving that our chosen open-source tools are effective for building a practical and insightful air quality monitoring system.

6.1 Geographical Influences on Air Quality

While this project focused on visualizing air quality, it's worth noting how the geography of the five German cities naturally affects their air quality. Understanding these influences can help explain the data we see.

- **City Size and Industry:** Larger cities like Berlin, Munich, and Essen, with more people and industry, naturally have more traffic and emissions. This generally leads to higher levels of pollutants like PM10, PM2.5, carbon monoxide, and carbon dioxide compared to smaller cities like Zwickau. Our simulated data would reflect these general patterns.
- **Land Features and Wind:** Local land features, like valleys, can trap pollutants near the ground. Wind patterns are also crucial: they can either blow away pollution, improving air quality, or bring it in from other areas, making it worse. For example, if prevailing winds come from industrial zones, nearby cities might see higher pollution.
- **Seasonal Changes:** Air quality changes with the seasons. In winter, more heating can lead to higher particle and carbon gas levels. In summer, higher temperatures and sunlight can increase UV index values. Our 2-day forecast helps us anticipate these short-term weather-related changes.
- **Green Spaces:** Cities with more parks and forests can help clean the air by absorbing carbon dioxide and trapping particles. Differences in green areas among the five cities could subtly affect their overall air quality.
- **Traffic:** Vehicle traffic is a major source of PM, CO, and CO_2 . Areas with heavy traffic will show higher concentrations of these pollutants. Our geographical map helps visually connect pollutant levels to their urban settings and consider the impact of local traffic.

The geographical map panel in Grafana, though simple, is a fundamental part of the dashboard. It helps users link the displayed pollution data to specific city locations and think about the local factors that might be influencing the air quality. Adding weather data in the future could provide even deeper insights.

7 CONCLUSION

This project successfully built a complete and effective IoT system for monitoring and visualizing air quality data in five key German cities. By carefully combining Node-RED for data handling, InfluxDB for efficient storage, and Grafana for clear and interactive visuals, we created a strong solution for environmental insights.
The detailed Grafana dashboard, with its variety of charts and graphs—including maps, time-series, bar charts, heatmaps, gauges, pie charts, and histograms—successfully transforms complex raw data into easy-to-understand information. The clear color-coded system (green, yellow, red) for air quality ensures that anyone can quickly see the current environmental status.
While we used some simulated data during development to ensure the system always ran smoothly, the core design is ready to connect with real-time API feeds or live sensors. This project showcases a powerful, open-source approach to environmental monitoring, offering a scalable and affordable way to assess public health and inform policies. The ability to track hourly changes, compare cities, and understand pollution trends helps us better understand urban air quality, paving the way for healthier and more sustainable communities.

APPENDIX: FULL DASHBOARD

See Figure 13 on the next page

8 REFERENCES

[1] Open-Meteo. Open-meteo: Free weather api. https://open-meteo.com, 2025.
Accessed: February 2025.
[2] Max Wertheimer. Laws of Organization in Perceptual Forms, volume 4. Springer,
1923.
[3] Arshad Khan. Visual Analytics for Dashboards: A Step-by-Step Guide to Principles
and Practical Techniques. Apress Media, LLC, Tracy, CA, USA, 2024.
[4] Patrick Zippenfenig. Open-meteo.com weather api, 2023.
[5] Bell B. Berrisford P. Biavati G. Horányi A. Muñoz Sabater J. Nicolas J. Peubey C.
Radu R. Rozum I. Schepers D. Simmons A. Soci C. Dee D. Thépaut J-N. Hersbach,
H. Era5 hourly data on single levels from 1940 to present, 2023.
[6] Grafana Labs. Grafana Documentation, 2025. Available at: https://grafana.com/
docs/ (Accessed: February 2025).

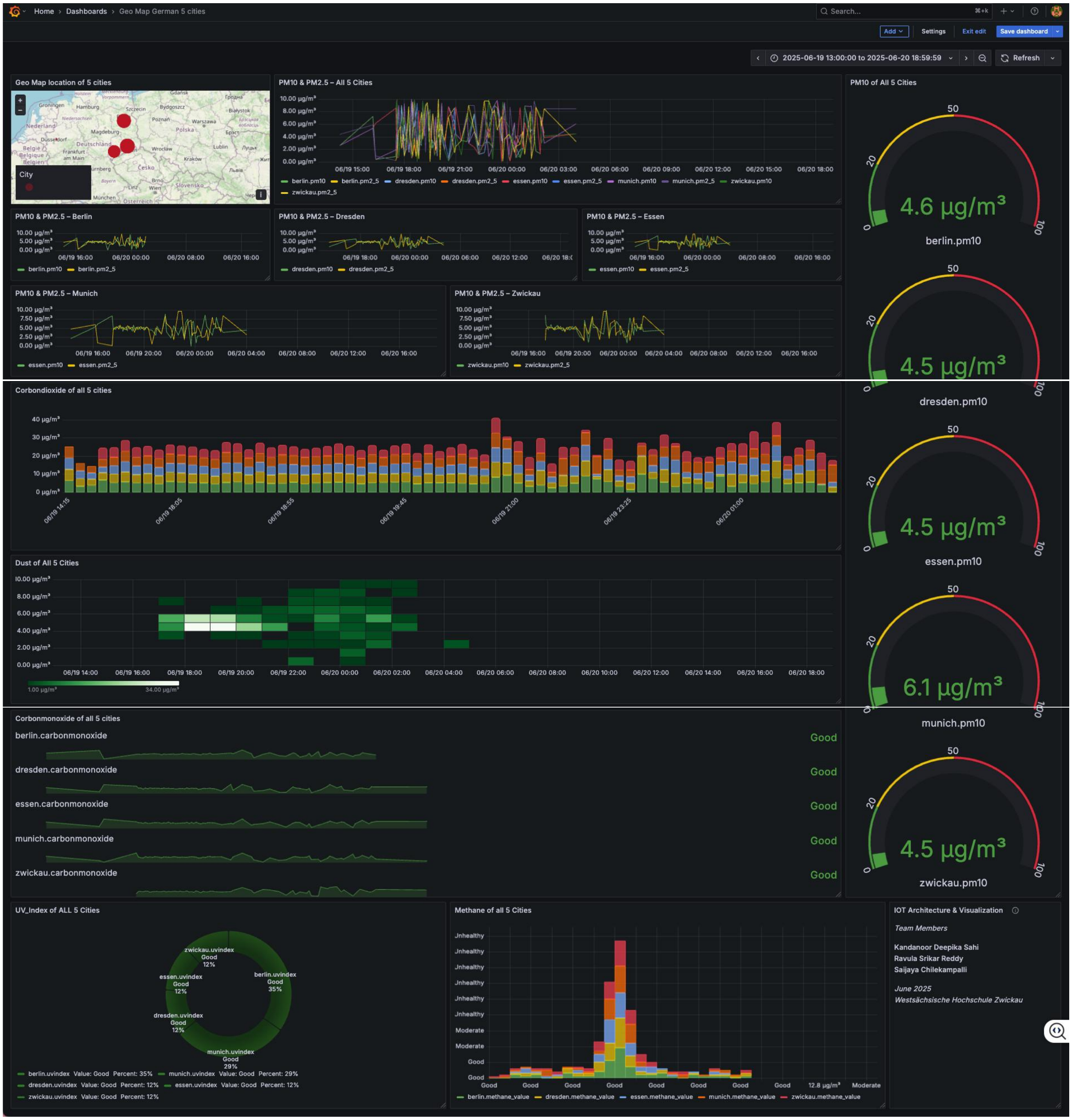

Figure 13: Complete Dashboard