

Mission

To design and deploy a web-based data visualization dashboard that monitors community water quality in real-time. In underdeveloped regions, contamination often goes unnoticed between the treatment source and the point of use. While physical sensors collect raw environmental parameters, our mission is to translate that complex data into accessible, actionable intelligence. By integrating an automated "Early Warning" decision matrix, the dashboard provides immediate visual alerts, empowering local citizens to catch issues early, maintain their infrastructure, and prevent waterborne illness.



Figure 1. From Left to Right: GitHub, Supabase, HTML, CSS, Javascript.

System Architecture

The system architecture was designed to bridge the gap between field-deployed hardware and accessible cloud infrastructure. Utilizing Python-based uplink scripts running on Raspberry Pi microcontrollers, raw sensor data is securely transmitted to a Supabase PostgreSQL database. To account for varying internet reliability in developing communities, we conceptualized a scalable Multi-Tier Networking model. This allows the system to adapt its data distribution—from full interactive web applications in high-reliability zones to lightweight static interfaces in areas with minimal cellular reach, allowing for constant availability of real-time data. For extreme cases of minimal connectivity, the architecture even supports localized visualization directly at the edge node. This fallback guarantees that communities are never left blind to sudden contamination events during network outages.

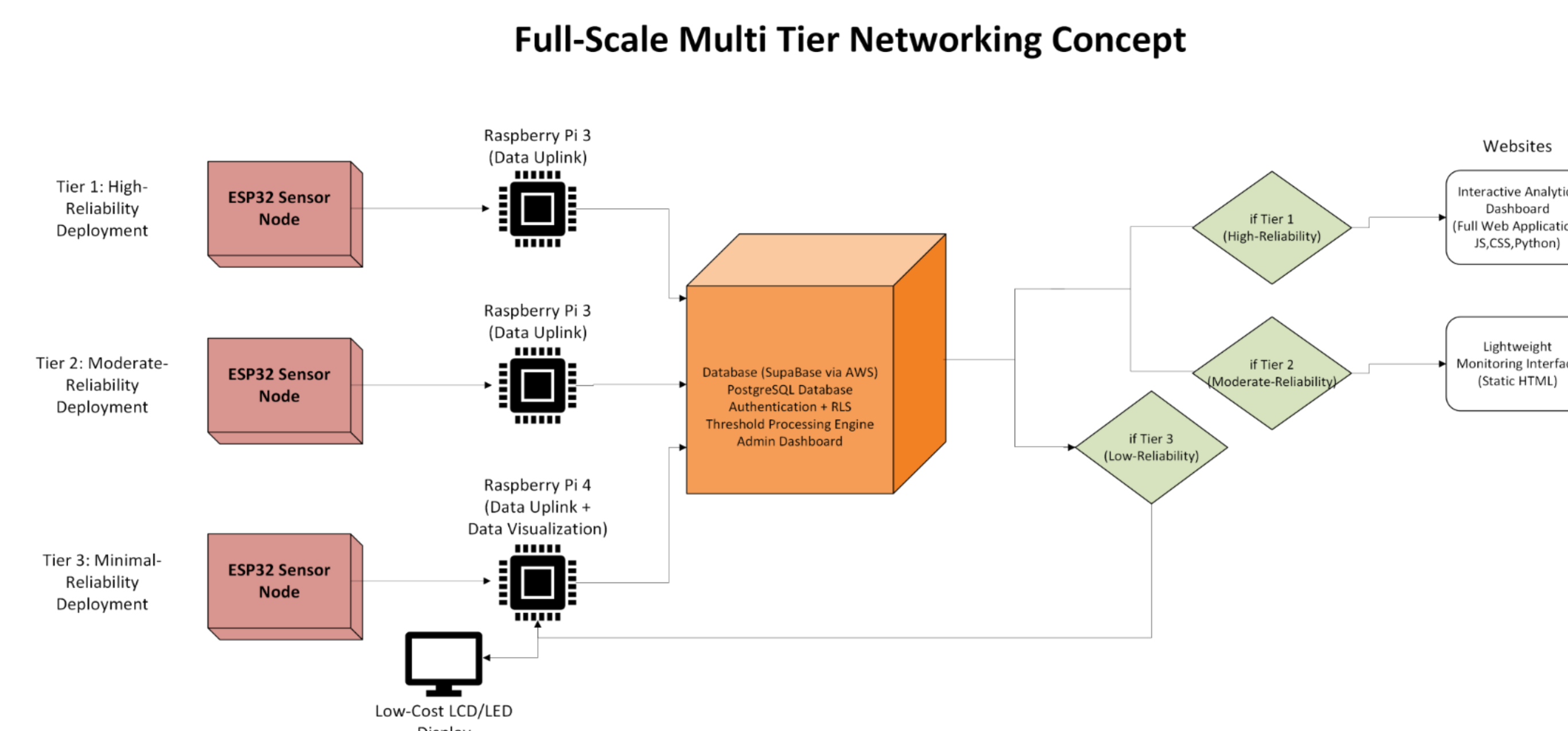


Figure 2. Full-Scale Multi-Tier Networking Concept.

Dashboard Development

Throughout the semester, the project evolved from a basic raw-data viewer into a comprehensive decision-support tool. The primary deliverable is a feature-complete dashboard prototype capable of visualizing live trends for critical parameters like pH, Turbidity, Temperature, and TDS. To eliminate guesswork, the system utilizes an automated safety threshold logic engine. This matrix instantly categorizes water quality as Safe (Green), Warning (Yellow), or Critical (Red), reducing the knowledge required for local monitoring.

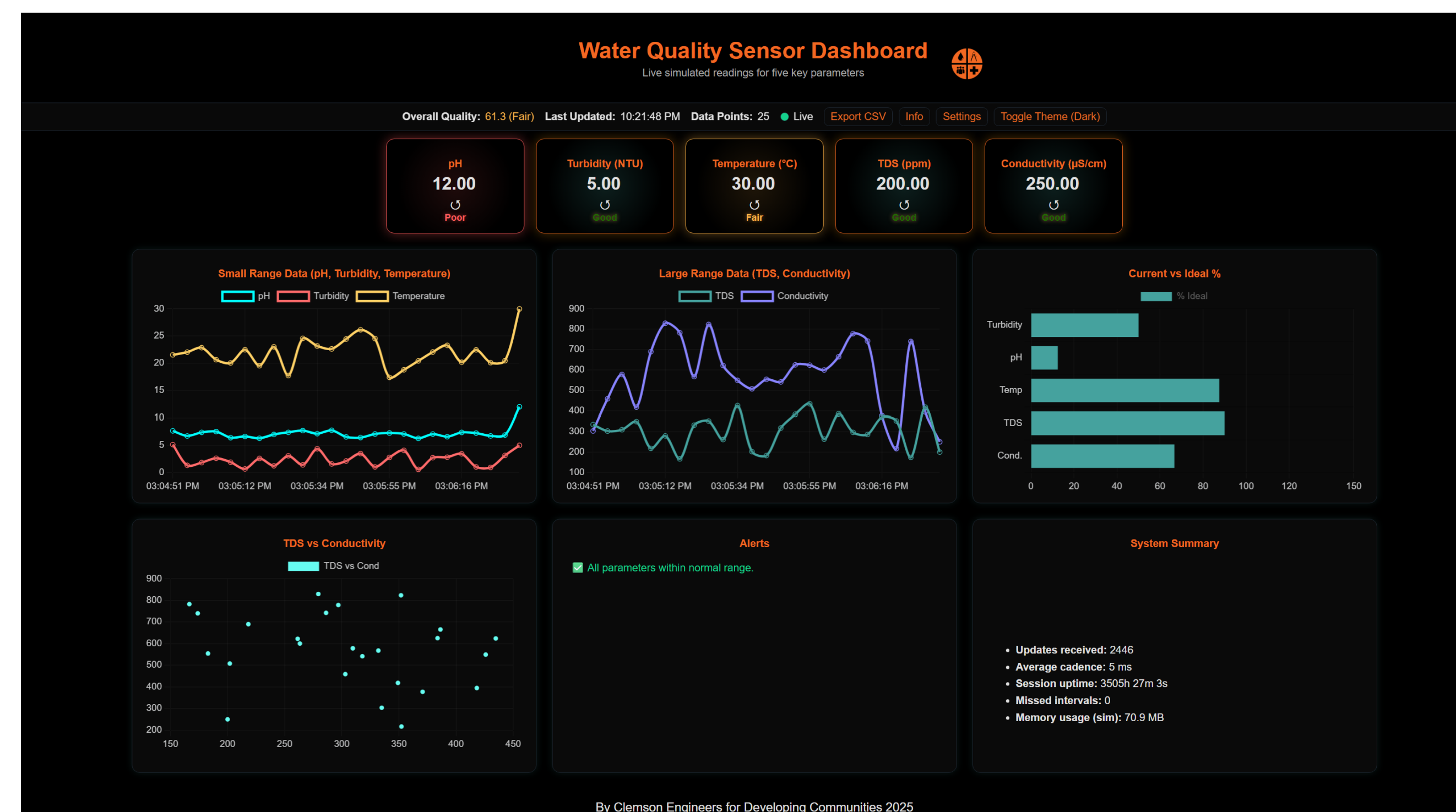


Figure 3. Analytics Dashboard Prototype (Single-Node View).

Automated Decision Matrix

To transition from raw sensor data to actionable intelligence, the dashboard integrates a robust multi-parameter decision matrix. Rather than evaluating metrics in isolation, the system correlates simultaneous changes across variables—such as pH, Turbidity, TDS, and Flow, to diagnose specific infrastructure failure modes. For example, a concurrent drop in ORP alongside a rise in Turbidity automatically triggers a 'Critical' flag, warning operators of a potential organic intrusion.

Actionable Patterns (ordered by severity)								
Case	Action Flag	Interpretation	ORP	pH	Temp	Turbidity	EC/TDS	Flow
C1	Critical	Intrusion/organic load consuming disinfectant; solids rise while ORP drops.	Decrease	No change	No change	Increase	No change	No change
C2	Critical	Ionic/source contamination likely; EC/TDS rises while ORP drops.	Decrease	No change	No change	No change	Increase	No change
C3	Critical	Low-flow vulnerability (stationary/pressure loss) with ORP drop.	Decrease	No change	No change	Increase or No change	No change	Decrease
C4	Critical	Warm water plus turbidity rise with ORP drop (rapid decay plus demand).	Decrease	No change	Increase	Increase	No change	No change
W1	Warning	Higher pH reduces	Decrease	Increase	No change	No change	No change	No change

Figure 4. "Early Warning" Decision Matrix (Critical Warnings).

Hardware Data Uplink

To support a continuous data pipeline from the physical water sensor to the database, a localized python script was developed for the nodes. The script was designed to be as simple as possible to run directly on the Pi without any complicated setup. It takes the readings gained from the sensor-like pH, Turbidity, and Temperature, and uploads them straight to the cloud, allowing for real time access. This constant, interval connection is what allows the dashboard to update in an instant and allow for the decision matrix to take full affect the moment that the readings are off.

```
Starting Uploader...
[21:08:29] Uploaded -> pH: 6.768401996 | Temp: 23.89247154°C
[21:08:33] Uploaded -> pH: 6.875927584 | Temp: 19.69458669°C
[21:08:37] Uploaded -> pH: 7.398917442 | Temp: 24.1740148°C
[21:08:42] Uploaded -> pH: 7.344148106 | Temp: 25.38909465°C
[21:08:46] Uploaded -> pH: 7.091639362 | Temp: 23.03358726°C
[21:08:50] Uploaded -> pH: 7.146930953 | Temp: 26.69198953°C
[21:08:54] Uploaded -> pH: 7.611065958 | Temp: 20.065527°C
[21:08:58] Uploaded -> pH: 6.863371444 | Temp: 18.88836324°C
[21:09:02] Uploaded -> pH: 7.318838952 | Temp: 17.55319938°C
```

Figure 5. Python Script Data Upload With 4 Second Interval.

Moving Forward

Moving forward, the Data Visualization team will integrate the web platform with the physical ESP32 sensor nodes developed by the hardware team for full-system live validation. Future iterations will focus on expanding the user interface to include scalable multi-node navigation, allowing operators to monitor several community standpipes simultaneously. Additionally, data export reporting tools and manual entry forms (for offline tests like Lead) will be finalized. This ensures the dashboard serves as a complete, centralized hub for long-term water quality tracking and infrastructure funding requests.

Conclusion

This project establishes a scalable, web-based foundation for remote water quality monitoring that bridges physical hardware sensors with an automated decision matrix. While live field integration remains ongoing, current dashboard prototypes demonstrate a clear path toward providing local communities with access to real time data visualization.

Acknowledgements

The Data Visualization Project Group would like to thank the Water Quality Sensor Project Group, and CEDC Faculty Director, David Vaughn.