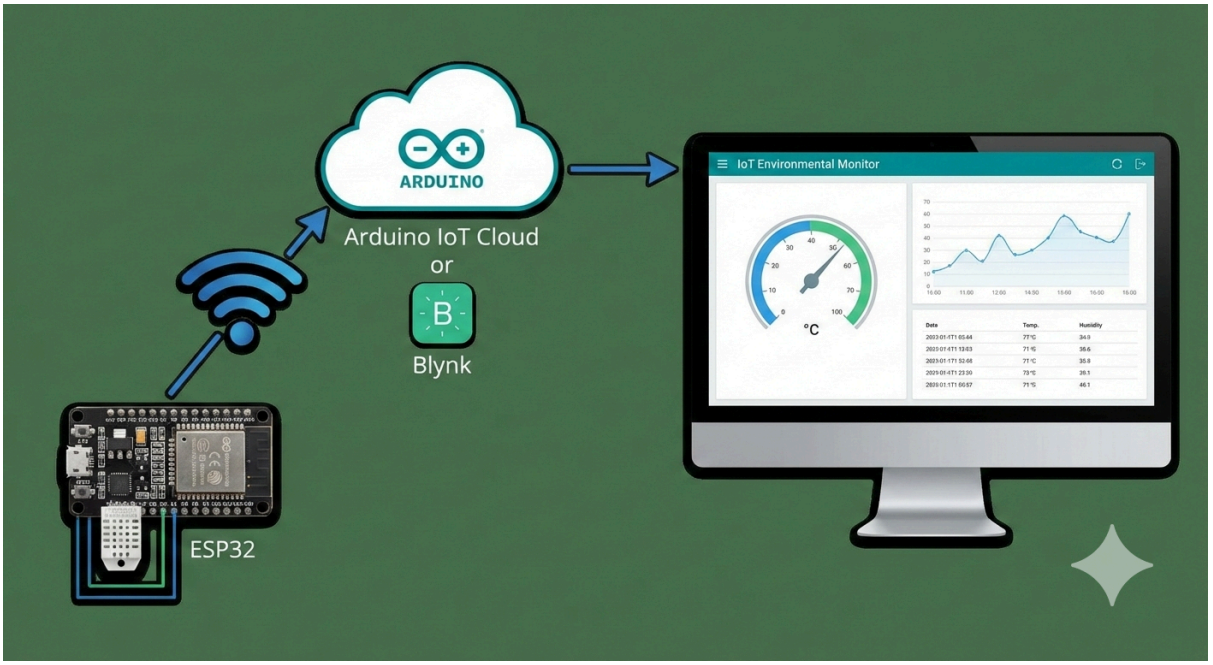
Task 3: IoT Data Monitoring Dashboard

An IoT dashboard serves as the user interface layer of your system. It translates the raw, continuous stream of data generated by microcontrollers and sensors into a readable, actionable format. To build this, you can utilize cloud service platforms such as Arduino IoT Cloud, Blynk, or a custom web application connected to Google Firebase.



Meeting the Key Requirements

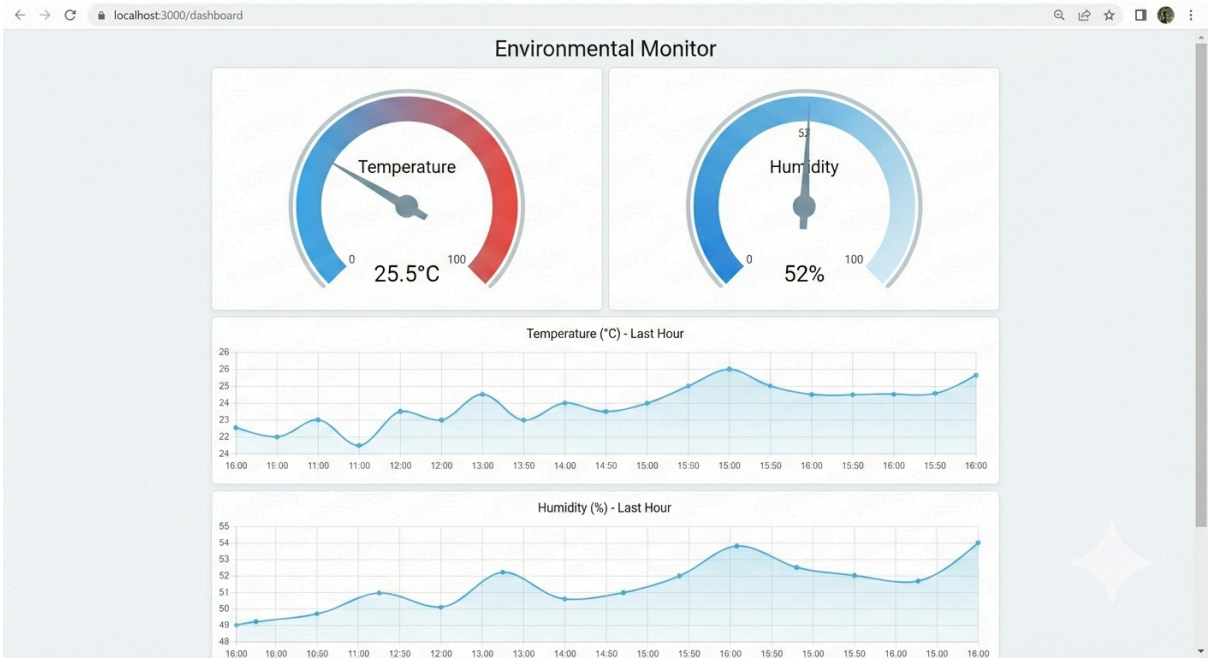
1. Show Sensor Readings Visually

Raw data logs (like the CSV output generated in Task 2) are difficult for humans to read quickly. A dashboard replaces these text logs with intuitive UI components called widgets.
Gauges & Dials: These are perfect for displaying the absolute current state of the environment. You would map your simulated temperature and humidity variables to individual gauge widgets, setting color gradients (e.g., red for high temperatures, blue for cool) to provide immediate visual context.

2. Use Charts or Tables

While gauges show the present moment, understanding environmental behavior requires historical context.
Time-Series Line Charts: This is the most effective way to visualize IoT data. The X-axis represents the time the data was logged, and the Y-axis plots the sensor values. Plotting temperature and humidity on a real-time line graph allows you to identify trends, spikes, or drops over the last hour or day. * Data Tables: For precise record-keeping, a dynamic table can be placed below the charts to display the raw incoming data (Timestamp | Temperature | Humidity), allowing users to export the logs for deeper analysis.

The following image is a screenshot of a well-designed IoT dashboard that incorporates both gauges for real-time readings and line charts for historical data analysis.



3. Update Data Periodically

A static dashboard is useless for monitoring. The interface must update to reflect the real-time conditions of your system.

Push vs. Pull: In modern IoT platforms, this is often handled automatically using WebSocket connections or the MQTT protocol. When the microcontroller publishes a new simulated reading to the cloud, the cloud instantly "pushes" that update to the dashboard, seamlessly animating the chart and adjusting the gauge without the user needing to refresh the page.