Student ID: 2111869

Name: Lê Chánh Nguyên

Class: L01

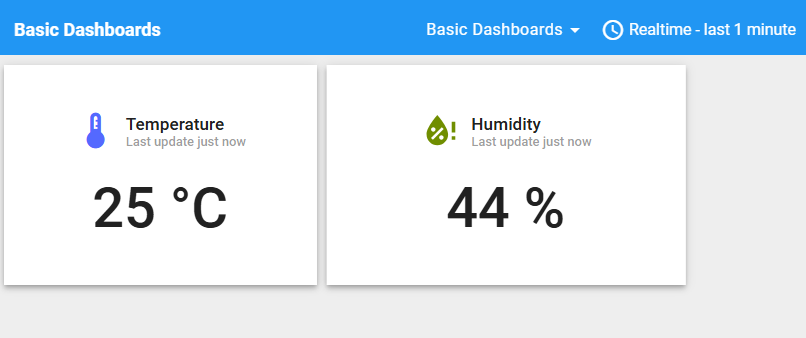
IoT Lab 1: Temperature & Humidity Sensors

# Overview

Temperature & Humidity sensors solution suitable for multiple applications. In this lab, you are supposed to get an interactive dashboard with the ability to observe real-time data.

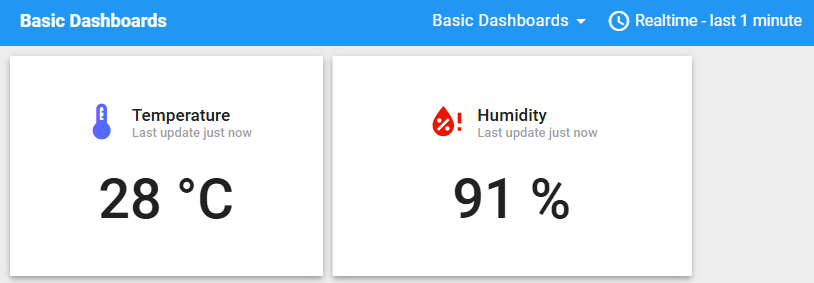
# Requirements

Design and implement an IoT dashboard that displays temperature and humidity in real time.



In terminal:





In terminal:



# Hardware:

* ESP32 or ESP32 S3 Development board.
* DHT20 or DHT11 temperature and humidity sensors.
* Connectors.

# Technical Risks:

* Hardware connection issues.
* Inaccurate data.
* The dashboard failing to update in real-time.

# Test Plan

### Test Cases:

* Connection test.
* Dashboard Display Test.

### Approach

* Monitor the temperature value displayed on the dashboard. Then, attempt to change the temperature reading by using your breath or finger. Observe any changes in the displayed value.

# Questions

* List some technical specifications of your temperature and humidity monitoring IoT solution.
* **DHT11 Measurement Range:** Temperature 0-50°C (±2°C), Humidity 20-80% (±5%).
* **DHT20 Measurement Range:** Temperature -40-80°C (±0.5°C), Humidity 0-100% (±3%).
* **Data Update Frequency:** 1 second per update (configurable).
* **Power Supply:** 3.3V or 5V (depending on the sensor and ESP32).
* **Communication:** I2C protocol (DHT20) or 1-Wire (DHT11).
* Write instructions and notable points when deploying the solutions.
* **Instructions**:
  + Connect the DHT sensor to the ESP32 (DHT11/DHT20 has Data, VCC, GND pins; check the datasheet for correct wiring).
  + Upload sample code (e.g., reading DHT data) to the ESP32 using an IDE like Arduino IDE.
  + Set up a dashboard (e.g., using Blynk, ThingSpeak, or a custom web interface).
  + Verify Wi-Fi connectivity to send data from the ESP32 to the dashboard.
* **Notable Points:**
  + Ensure a stable power supply to avoid signal interference.
  + Check the sensor’s sampling interval (DHT11 should not be read continuously under 1 second).
  + Handle errors if the sensor fails to respond (timeout, reconnect).

# Instructions:

* Do a quick test of the development board and development environment by running “hello world” example.
* Run a DHT20 example to read sensor data and print out to the serial monitor to make sure your hardware is working as expected.

# Resources

* Official ESP32 documentation (Espressif website).
* DHT11/DHT20 datasheets.