GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

HOST INSTITUTION

xxxxx

COIMBATORE – 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

Name	Roll No
Name 1	1
Name 2	2
Name 3	3
Name 4	4
Name 5	5

Table Of Contents

Abstract

This project presents the design and implementation of a smart temperature-based fan control system using an ESP32 microcontroller, a DHT11 temperature and humidity sensor, and a relay module. The system is designed to automatically regulate the operation of a fan based on ambient temperature readings. When the temperature exceeds a predefined threshold, the ESP32 activates the relay to turn on the fan. If the temperature drops below the threshold, the fan is turned off.

Additionally, the system integrates with the ThingzMate IoT cloud platform, enabling remote monitoring of the fan's status. This allows users to track the system's operation in real-time from any location. The project demonstrates the practical application of IoT technology in home automation, offering enhanced convenience and energy efficiency.

The primary goal of this project is to automate the operation of a fan based on the ambient room temperature. The DHT11 sensor continuously monitors the room's temperature, and when the temperature exceeds a predefined threshold, the ESP32 triggers the relay to turn on the fan. Conversely, when the temperature drops below the threshold, the relay is deactivated, turning off the fan.

Introduction

The integration of IoT technology with environmental sensors is revolutionizing how we monitor and control home appliances. This project focuses on developing a smart temperature-based fan control system using an ESP32 microcontroller, a DHT11 temperature and humidity sensor, and a relay module. The ESP32, a powerful microcontroller with built-in Wi-Fi capabilities, allows for seamless connectivity and control over cloud platforms like ThingzMate.

The primary goal of this project is to automate the operation of a fan based on the ambient room temperature. The DHT11 sensor continuously monitors the room's temperature, and when the temperature exceeds a predefined threshold, the ESP32 triggers the relay to turn on the fan. Conversely, when the temperature drops below the threshold, the relay is deactivated, turning off the fan.

In addition to local control, the project leverages ThingzMate, an IoT cloud platform, to remotely monitor the system's status. The ESP32 uploads the fan's operating status (on or off) to the cloud, enabling users to track the system's performance in real-time from anywhere using a mobile device or web interface.

This smart automation not only enhances convenience but also contributes to energy efficiency by ensuring that the fan operates only when necessary. The project is a practical example of how IoT can be applied to create smart home solutions that are both functional and user-friendly.

Hardware and Software Requirements

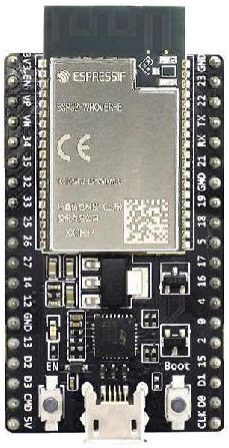
Hardware Requirements

1. ESP32 Microcontroller
2. DHT_11 SENSOR
3. Relay Module
4. USB Cable
5. Jumper Wires

Software Requirements

1. Wokwi Simulator
2. Arduino IDE
3. Thingzmate Cloud

ESP32 Microcontroller



The ESP32 microcontroller is a powerful and versatile chip used in this project to control the four-way traffic light system. It features integrated Wi-Fi and Bluetooth capabilities, allowing for seamless communication with the ThingzMate Cloud for remote monitoring and control. With its dual-core processor and multiple GPIO pins, the ESP32 efficiently handles the timing and sequencing of the traffic lights, making it ideal for real-time IoT applications.

DHT 11 SENSOR

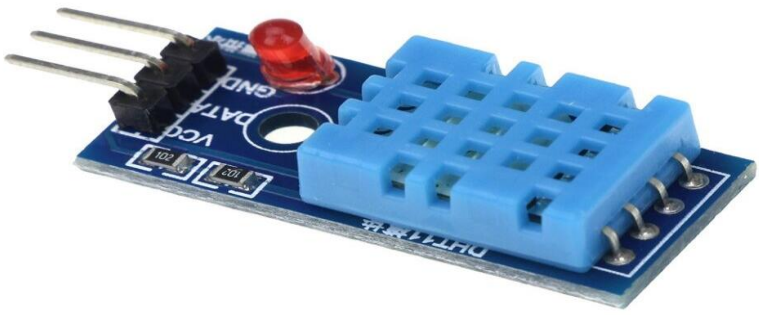
The DHT11 is a widely used temperature and humidity sensor that provides reliable measurements at a low cost. It combines a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a digital signal on the data pin. The sensor is commonly used in various applications such as weather monitoring, home automation systems, and environmental monitoring.

Key Features:

- **Temperature Range**: The DHT11 can measure temperatures from 0°C to 50°C with an accuracy of ±2°C.
- **Humidity Range**: It can measure humidity from 20% to 90% RH with an accuracy of ±5% RH.
- **Digital Output**: The DHT11 provides a digital output, making it easy to interface with microcontrollers like the ESP32.
- **Low Power Consumption**: The sensor operates at a voltage of 3.3V to 5V and consumes very little power, making it suitable for battery-powered applications.
- **Slow Sampling Rate**: The DHT11 has a sampling rate of 1 Hz (one reading per second), which is adequate for most environmental monitoring tasks but may not be suitable for real-time applications requiring fast response.

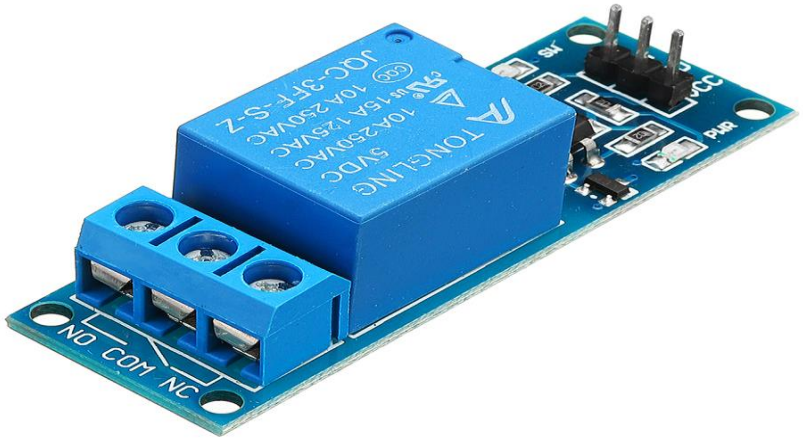
Applications:

- **Weather Stations**: The DHT11 is often used in weather stations to monitor temperature and humidity levels.
- **Home Automation**: It is used to control HVAC (Heating, Ventilation, and Air Conditioning) systems based on environmental conditions.
- **Greenhouses**: The sensor helps in maintaining optimal conditions for plant growth by monitoring and controlling temperature and humidity.



RELAY MODULE

A relay module is an electronic device used to control high-current or high-voltage loads with a low-current control signal, such as that from a microcontroller or other low-power digital device. It acts as an electrically operated switch that can turn on or off a larger load (like a motor, lamp, or appliance) based on a signal received from a microcontroller or other control system.



USB-Cable

The USB cable is a critical tool in this project, used to connect the ESP32 microcontroller to a computer for power supply, programming, and debugging. It enables the transfer of code and data between the development environment and the microcontroller, facilitating the upload of firmware and real-time communication during the development process. The USB connection also allows for serial monitoring, providing valuable insights into the system's performance and behavior.

Jumper Wires

Jumper wires are essential in this project, used to connect the ESP32 microcontroller to the components on the breadboard. These wires provide a flexible and reliable way to link the microcontroller's GPIO pins to the LEDs, resistors, and other circuit elements, enabling proper signal and power flow. Their ease of use allows for quick modifications and testing during the prototyping stage.

Wokwi Simulator

Wokwi Simulator is a powerful online tool used in this project to simulate the four-way traffic light controller system before physical implementation. It allows for the virtual testing of the ESP32 microcontroller, LEDs, and other components, enabling real-time visualization and debugging of the circuit and code. By using Wokwi, developers can ensure the system's functionality and make necessary adjustments without needing the actual hardware setup.

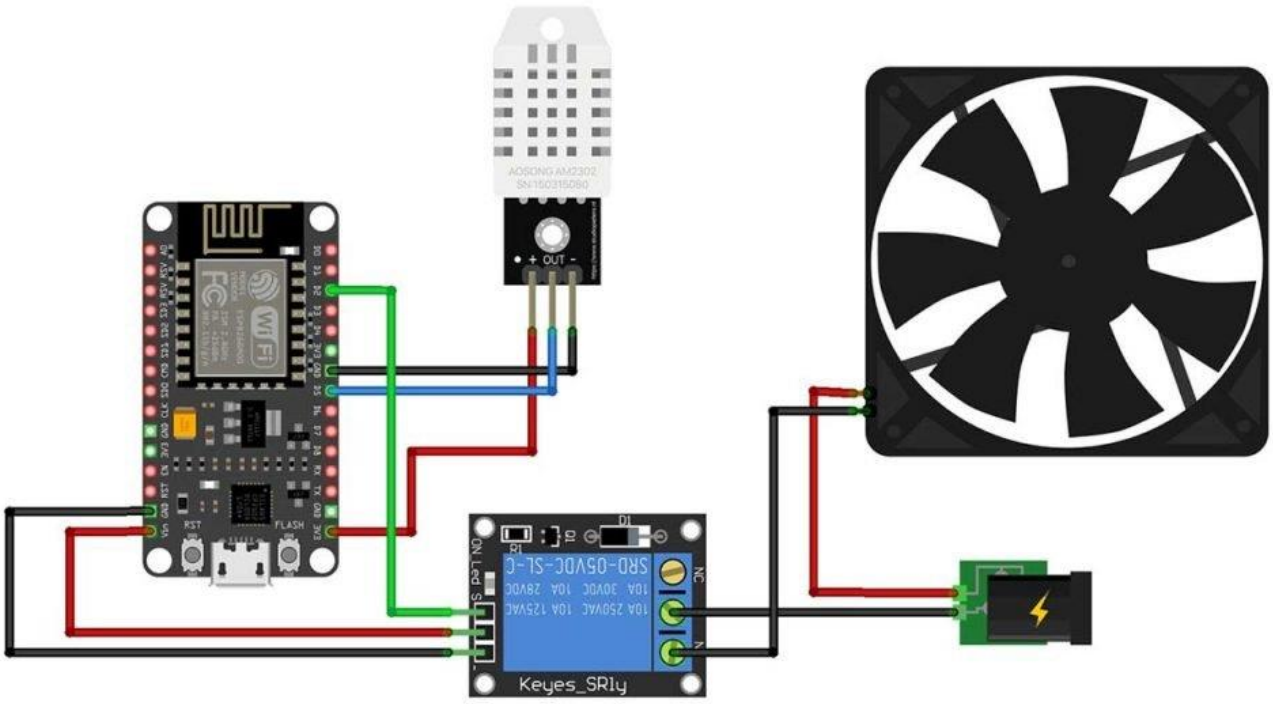
Arduino IDE

Arduino IDE is the primary development environment used in this project for programming and uploading code to the ESP32 microcontroller. It provides an easy-to-use interface for writing, compiling, and debugging the code that controls the four-way traffic light system. With its extensive library support and compatibility with ESP32, the Arduino IDE streamlines the development process, allowing for efficient code iteration and testing.

Thingzmate Cloud

ThingzMate enables real-time simulation of a 4-way traffic light control system, allowing you to monitor and manage traffic lights via cloud connectivity.It provides tools for configuring traffic light sequences, ensuring accurate simulation of traffic flow and light transitions.With ThingzMate, you can easily visualize and control traffic light statuses, optimizing traffic management and ensuring efficient simulation scenarios.

Block Diagram



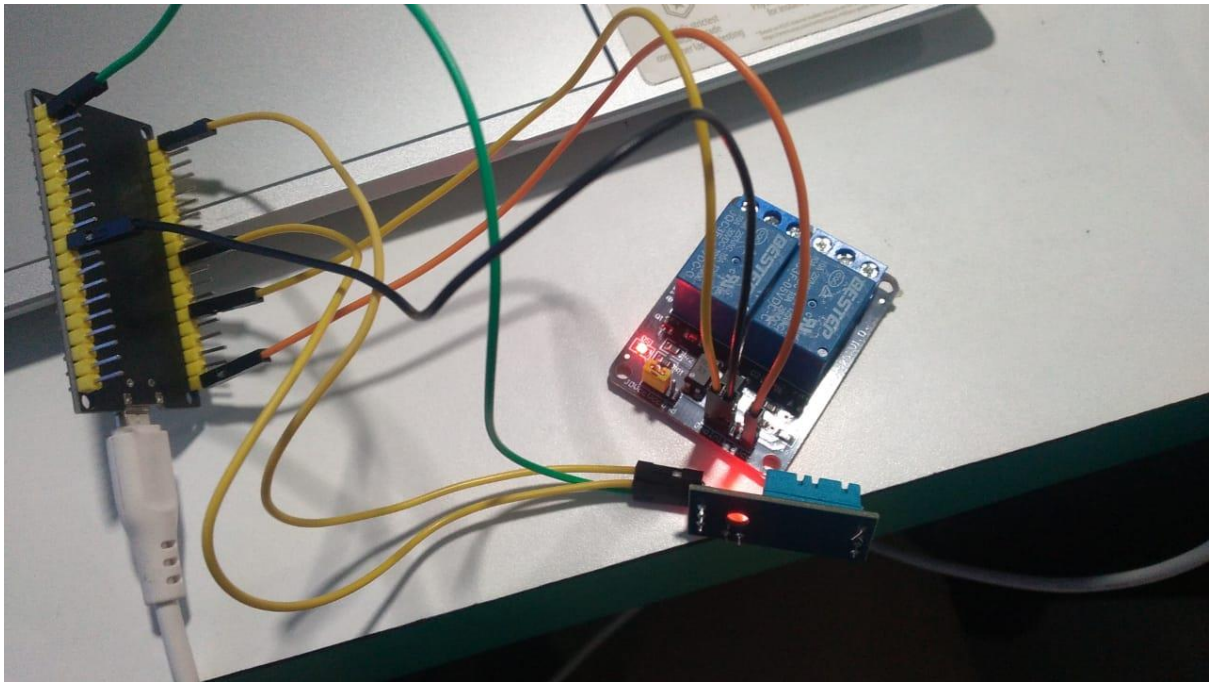
Code

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <DFRobot_DHT11.h>

// WiFi credentials
#define WIFI_SSID "YOUR DEVICE NAME"
#define WIFI_PASSWORD "YOUR PASSWORD"

// Define the DHT11 sensor and pin
DFRobot_DHT11 DHT;
#define DHT11_PIN 26

// Define the relay pin
#define RELAY_PIN 5  // You can change this to any other GPIO pin

// Server endpoint and Authorization token
const char *serverUrl = "https://console.thingzmate.com/api/v1/device-
types/dht1112/devices/dht1112/uplink";
String AuthorizationToken = "Bearer dabe9aa0d02bfc3eb230201ebf623dda";

// Define the temperature threshold to control the fan
#define TEMPERATURE_THRESHOLD 32.0

void setup() {
  Serial.begin(115200);

  // Initialize the relay pin
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);  // Ensure the relay is initially off

  // Connect to WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  // Read the temperature and humidity
  DHT.read(DHT11_PIN);
  Serial.print("temp: ");
  Serial.print(DHT.temperature);
```

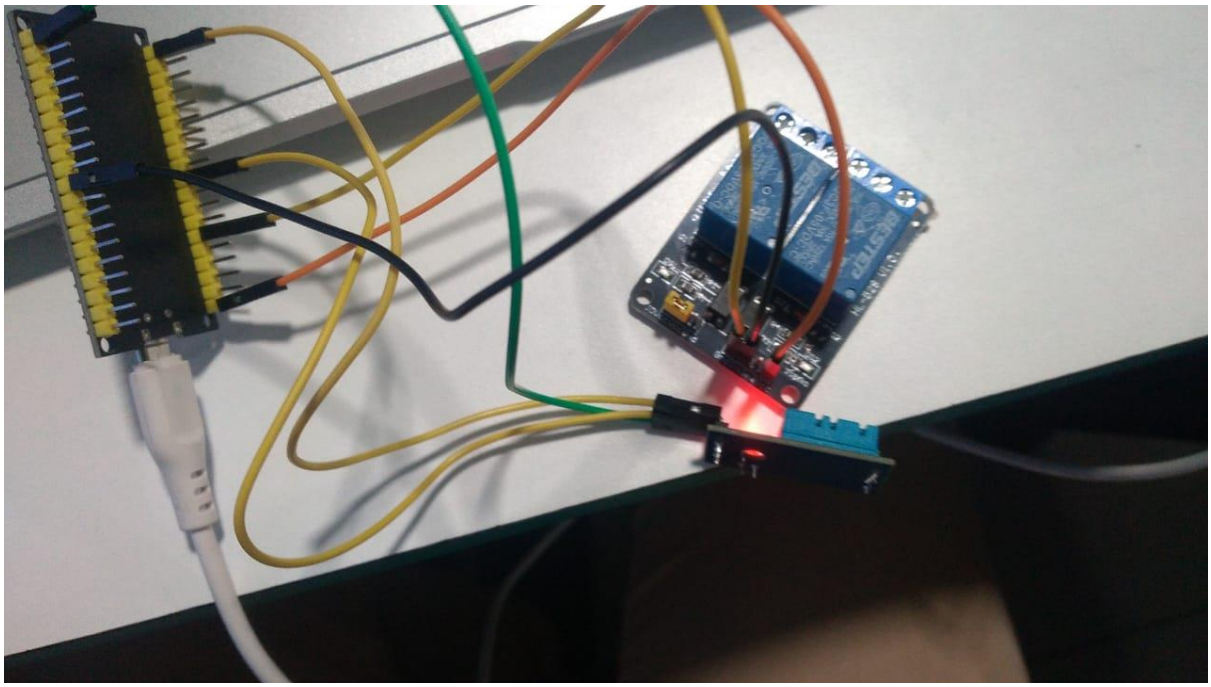
```
  Serial.print("  humi: ");
  Serial.println(DHT.humidity);

  // Control the relay based on temperature
  if (DHT.temperature < TEMPERATURE_THRESHOLD) {
    digitalWrite(RELAY_PIN, LOW); // Turn the fan OFF
    Serial.println("Fan OFF");
    uploadStatus("Fan OFF");
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Turn the fan ON
    Serial.println("Fan ON");
    uploadStatus("Fan ON");
  }

  delay(1000);  // 1 second delay between readings
}

void uploadStatus(String status) {
  HTTPClient http;
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");
  http.addHeader("Authorization", AuthorizationToken);

  // Create JSON payload
  String payload = "{\"status\":\"" + status + "\"}";

  // Send POST request
  int httpResponseCode = http.POST(payload);

  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("HTTP Response code: " + String(httpResponseCode));
    Serial.println(response);
  } else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }

  http.end();  // Free resources
}
```

Output Results

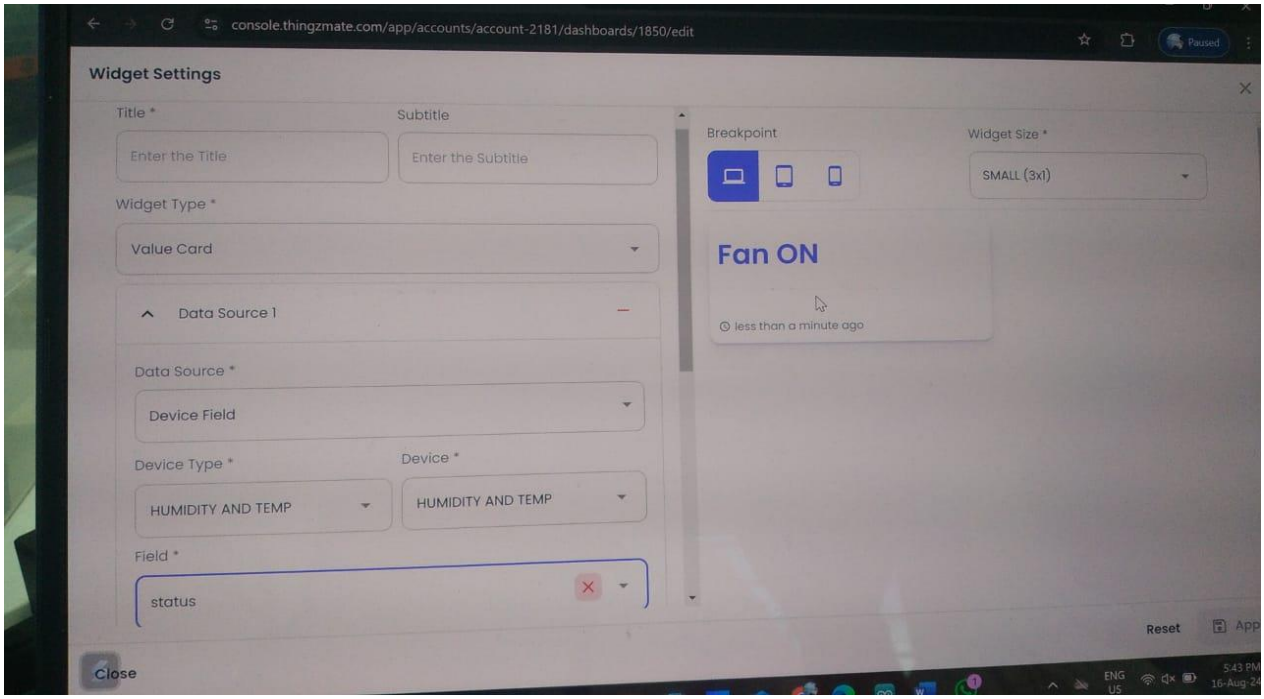
HOT ROOM TEMPERATURE – TURN ON THE FAN / TURN ON THE RELAY

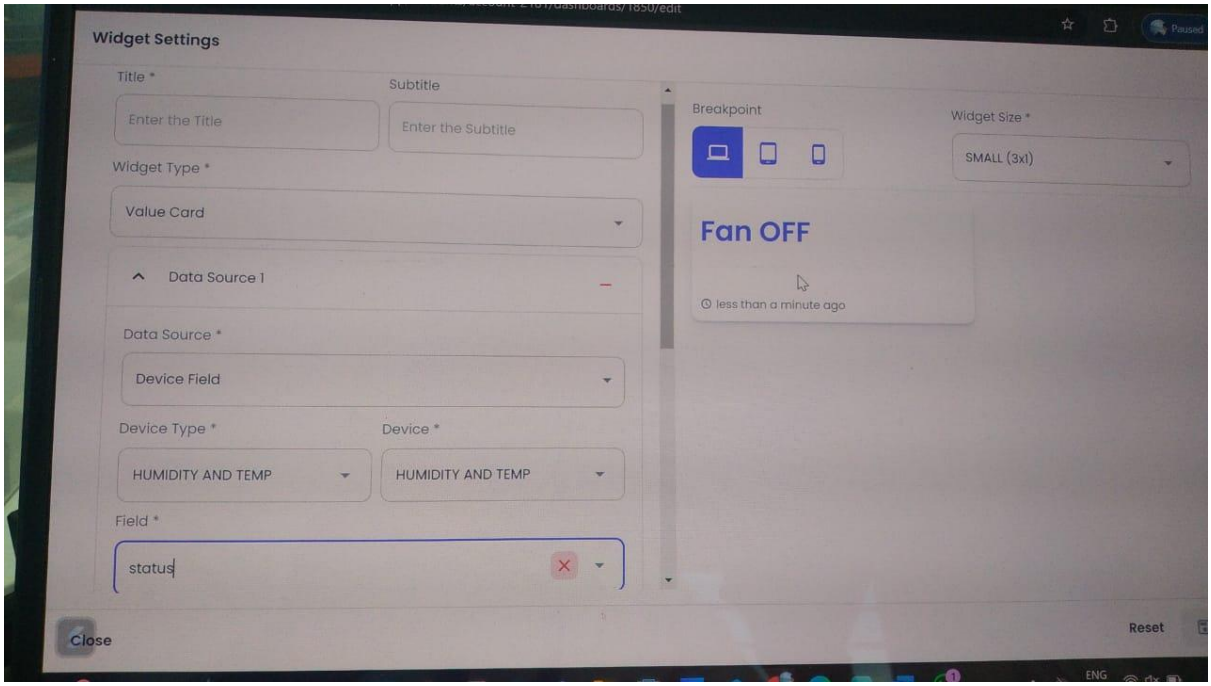


COLD ROOM TEMPERATURE – TURN OFF THE FAN / TURN OF THE RELAY



Cloud Output





Conclusion

The implementation of the smart temperature-based fan control system met the project objectives effectively. The integration of the ESP32 with the DHT11 sensor and relay module provided a reliable and automated solution for controlling a fan based on ambient temperature. The successful integration with the ThingzMate cloud platform enabled remote monitoring and control, enhancing the system's functionality and user convenience.
Future improvements could include expanding the system to control additional devices, integrating more sensors for comprehensive environmental monitoring, or optimizing the system for energy efficiency and reliability.