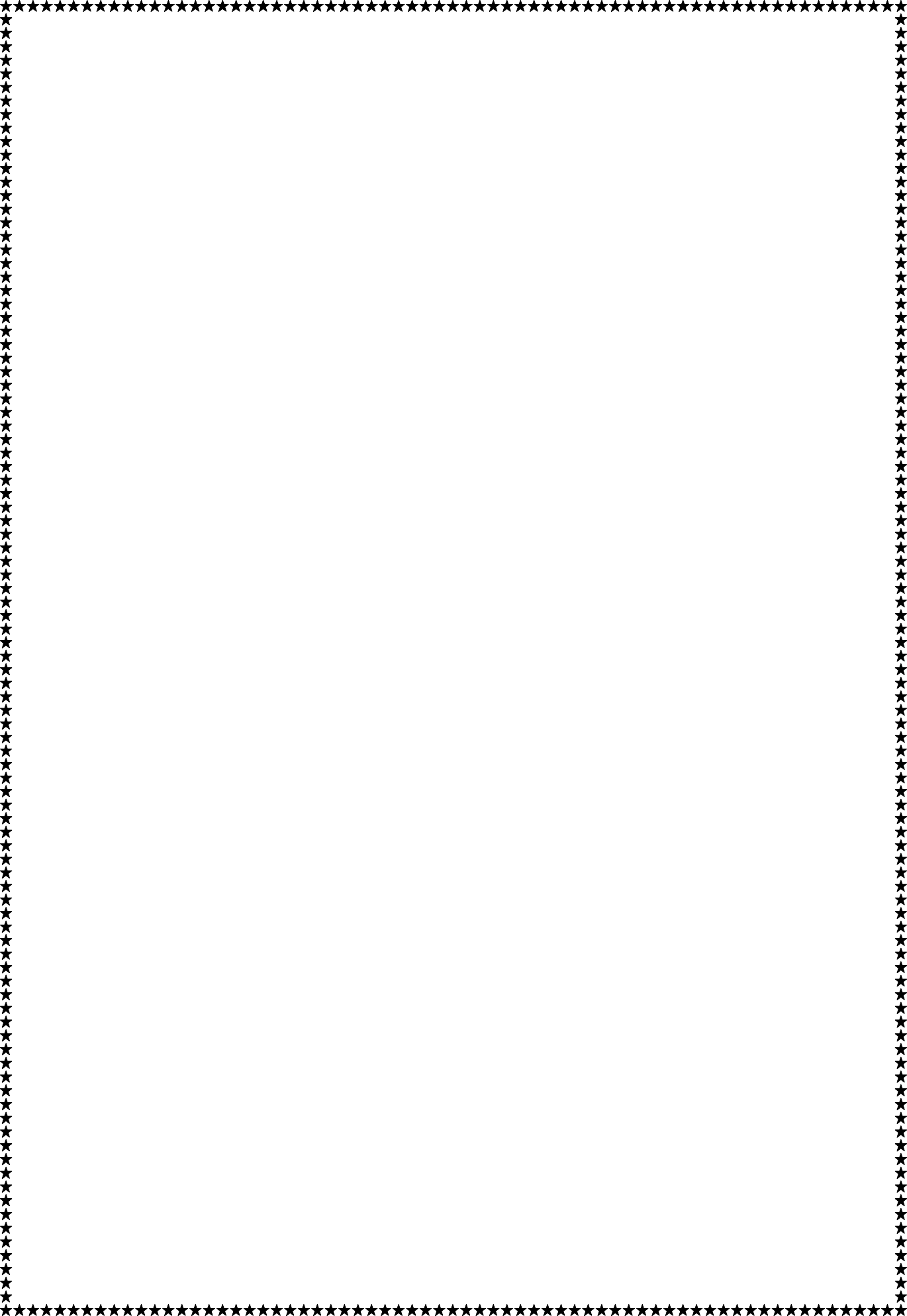
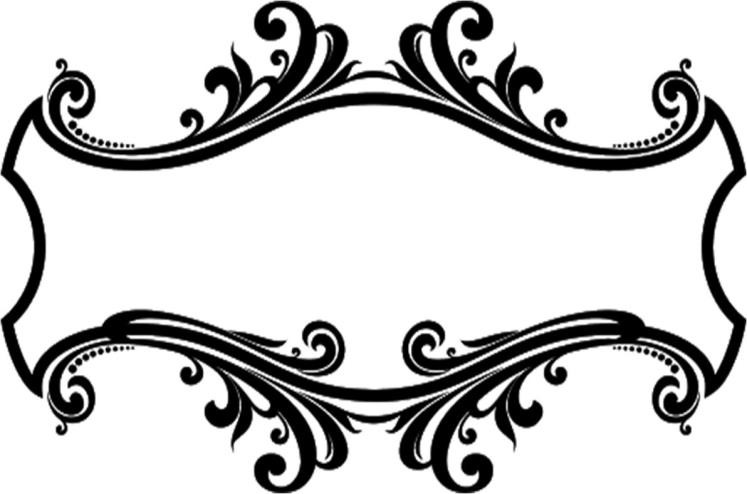
z



SESSION 2022-23

**SUBJECT:-**

# INTRODUCTION TO CLOUD TECHNOLOGIES

**PROJECT**

**COURSE:- BCA sem 4th**

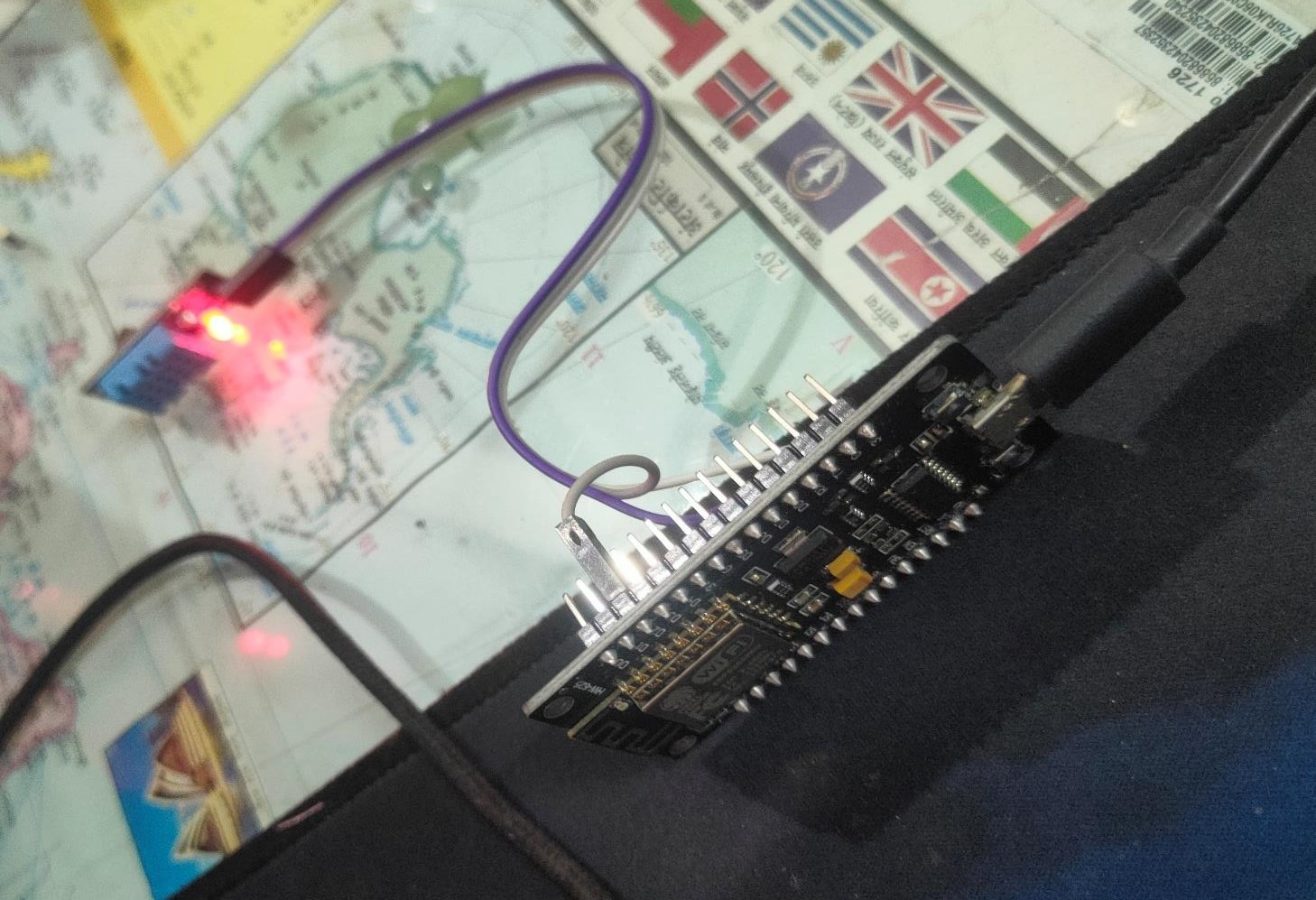
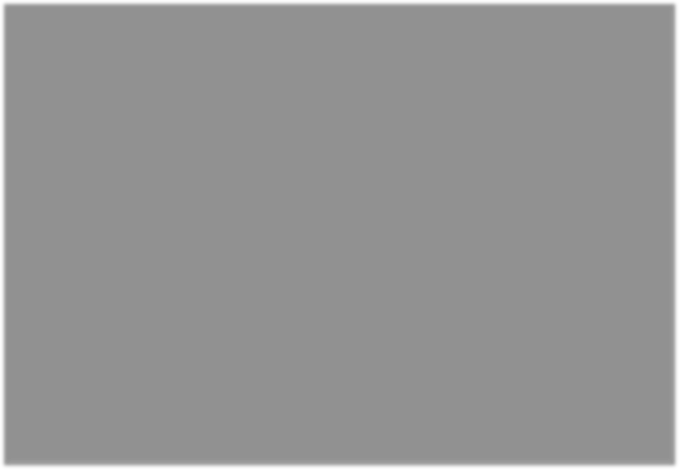
**ROLL NO :- 41221131,41221201,41221139,412211,41221138**

**SUBMITTED BY :- SUBMITTED TO:-**

**1. RITIK KUMAR , Ms.KOMAL DHINGRA**

1. **SACHIN RAJBHAR,**
2. **KULDEEP,**
3. **SACHIN KUMAR SINGH**

# DHT11 Humidity Sensor with ESP8266 and ThingSpeak **Overview**



DHT11 is a low-cost Humidity and Temperature Sensor. Since it has both the temperature and humidity sensors, the DHT11 Sensor is sufficient to implement your first IoT Weather Monitoring System.

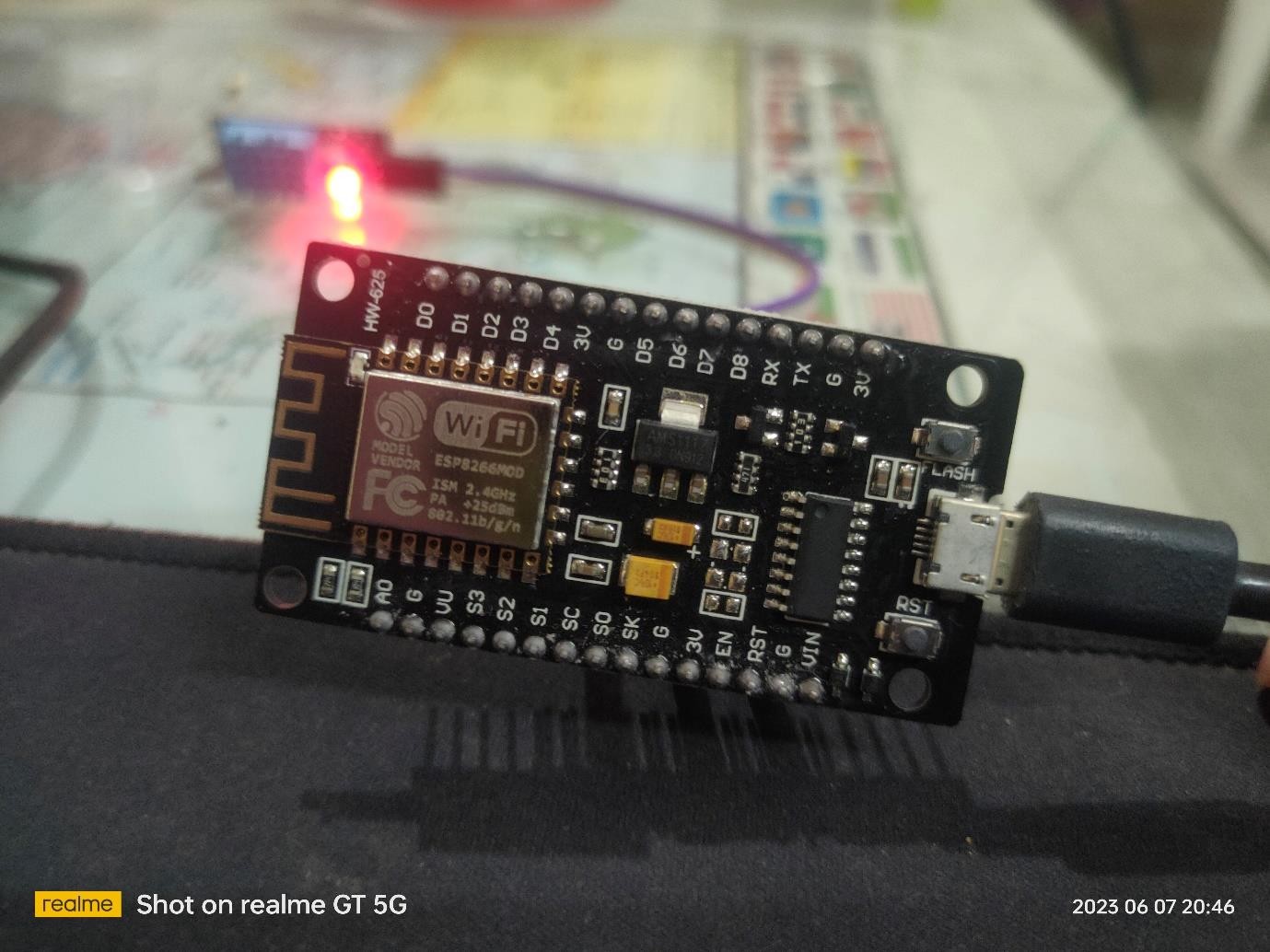
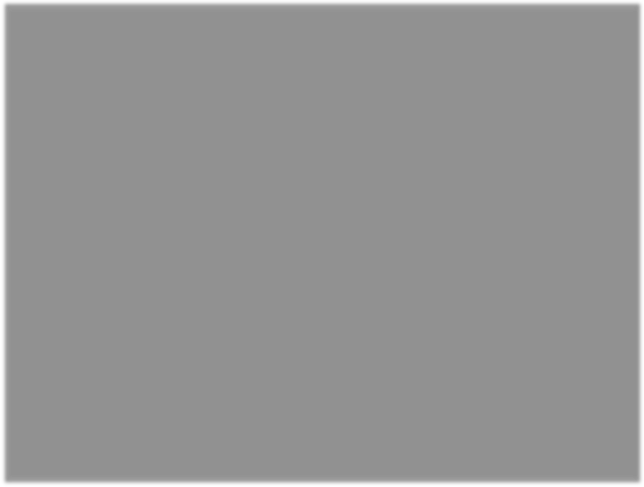
When it comes to IoT, the combination of ESP8266 and ThingSpeak is an excellent way for beginners and hobbyists to dive into your IoT related projects.

If you are planning to build your own weather station, then this project could be your first step in that path. In this project, I’ll talk about the DHT11 Sensor in brief and also explain how to interface the DHT11 Humidity and Temperature Sensor with ESP8266 and ThingSpeak.

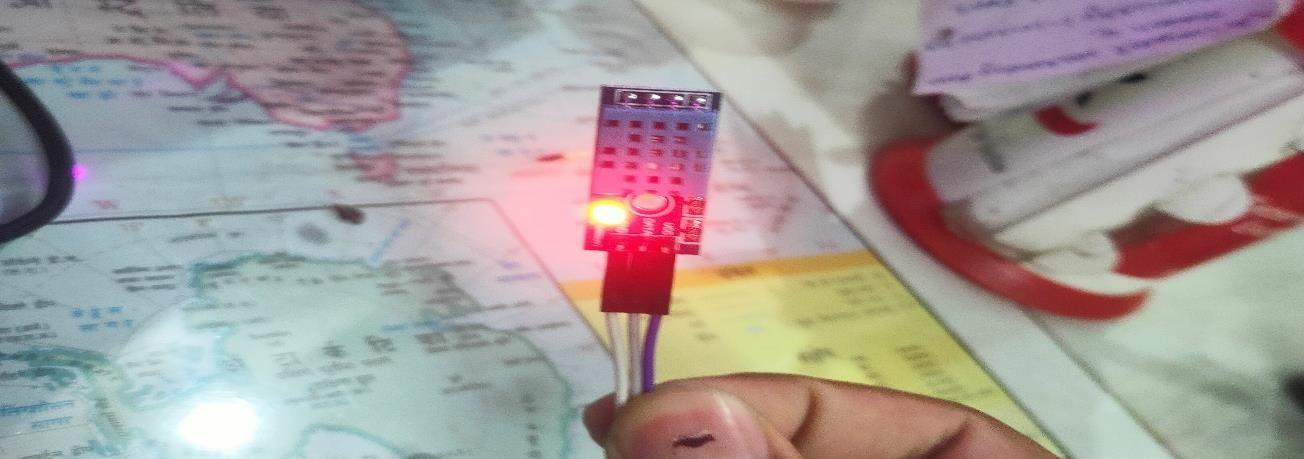
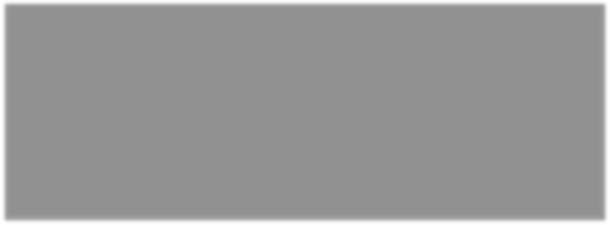
### A Brief Note on NodeMCU ESP8266

NodeMCU is an open-source development board based on the ESP8266 Wi-Fi module. It combines the functionality of a microcontroller with Wi-Fi connectivity, making it a popular choice for Internet of Things (IoT) projects. Here's a brief overview of NodeMCU:

ESP8266 Module: NodeMCU is built around the ESP8266 Wi-Fi module, which is a low-cost, low-power, and highly integrated chip capable of connecting to Wi-Fi networks. It features a microcontroller unit (MCU) and built-in Wi-Fi, making it suitable for IoT applications.



### A Brief Note on DHT11 sensor



The DHT11 sensor is a popular and affordable digital temperature and humidity sensor used in various projects and applications. Here's a brief overview of the DHT11 sensor:

* Temperature and Humidity Measurement: The DHT11 sensor is designed to accurately measure temperature and relative humidity in the surrounding environment. It utilizes a capacitive humidity sensor and a thermistor to measure these parameters.
* Low Cost and Availability: One of the key advantages of the DHT11 sensor is its low cost, making it widely accessible for hobbyists, students, and DIY enthusiasts. It is commonly available and can be easily purchased from various electronics suppliers.
* Limited Accuracy: While the DHT11 sensor provides temperature and humidity measurements, it is important to note that its accuracy is relatively lower compared to more advanced sensors. The temperature accuracy is typically within ±2°C, and the humidity accuracy is around

±5%. Therefore, it may not be suitable for applications that require high precision.

* Operating Voltage: The DHT11 sensor operates at a voltage of 3.3V to 5V, which makes it compatible with a wide range of microcontrollers and development boards. It consumes very low power, making it suitable for battery-powered applications.
* Simple Usage: The DHT11 sensor is relatively easy to use. It usually comes in a small module with three pins: VCC (power supply), GND (ground), and DATA (data line). By providing power and connecting the data pin to a microcontroller's input/output pin, you can read the temperature and humidity values using the appropriate library or code.

Cloud platform using Thingspeak

ThingSpeak is an open-source Internet of Things (IoT) platform and cloud service that allows you to collect, analyze, and visualize data from connected devices. Here's a brief overview of the ThingSpeak cloud platform:

**Data Collection:** ThingSpeak provides an easy-to-use interface and APIs that allow you to collect data from IoT devices and sensors. It supports various communication protocols such as HTTP, MQTT, and ThingSpeak's own API for sending data to the cloud.

**Channels and Fields:** In ThingSpeak, data is organized into channels. Each channel represents a collection of data from a specific device or sensor. Within a channel, you define fields to store different types of data such as temperature, humidity, pressure, etc. You can have multiple fields within a channel to accommodate different data streams.

**Data Storage:** ThingSpeak stores the collected data in its cloud database, making it readily available for analysis and visualization. The platform provides built-in data retention policies, allowing you to specify how long the data should be stored.

**Data Visualization:** ThingSpeak offers built-in tools to visualize the collected data in the form of charts, graphs, and gauges. You can create customizable visualizations to gain insights and monitor the behavior of your IoT devices over time. Real-time and historical data can be visualized to track trends and anomalies.

**Data Analysis:** ThingSpeak enables you to perform data analysis using MATLAB, a powerful programming language for mathematical computations and data analysis. You can apply algorithms, perform calculations, and generate insights based on the collected data. MATLAB's analytical capabilities make ThingSpeak suitable for more advanced data processing and decision- making.

**IoT Integrations**: ThingSpeak supports integration with various IoT platforms and services, allowing you to connect and exchange data with other cloud platforms, web services, or applications. This enables you to leverage the

capabilities of different IoT ecosystems and extend the functionality of your IoT applications.

**Open and Customizable:** ThingSpeak is an open-source platform, which means its source code is available for modification and customization. You have the flexibility to extend the platform's features or integrate it with your existing systems according to your specific requirements.

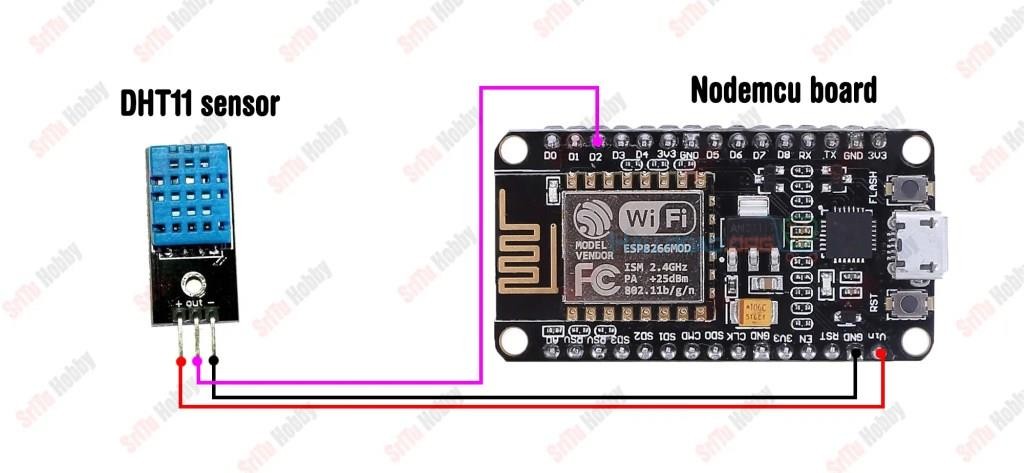
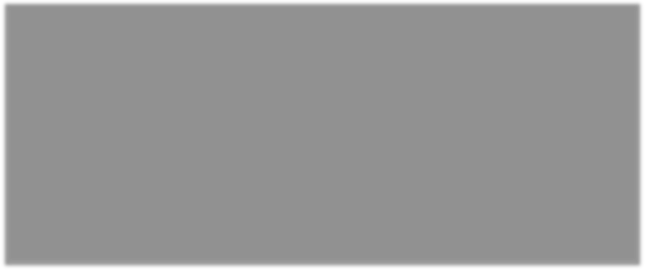
Setting ThingSpeak & Getting API Key:

* Go to <https://thingspeak.com/>and set up an account if you do not have one. Login to your account.
* Create a new channel by clicking on the button. Enter the basic details of the channel. Then Scroll down and save the channel. You can follow the video guide below.
* Then go to API keys, copy and paste this key in a separate file. You will require it again while programming.

## Circuit Diagram:

The required components are given below.

* Nodemcu (ESP8266) board x 1
* DHT11 sensor x 1
* Breadboard x 1
* Jumper wires



## Source code:

#include <ESP8266WiFi.h> #include <DHT.h> #include <ThingSpeak.h>

const char \*ssid = "realme GT 5G"; const char \*pass ="12345678";

DHT dht(D2, DHT11);

WiFiClient client;

long myChannelNumber = 2179515;

const char myWriteAPIKey[] = "L8XBOF6PHP9HULOL";

void setup() {

// put your setup code here, to run once: Serial.begin(9600);

WiFi.begin(ssid, pass); while(WiFi.status() != WL\_CONNECTED)

{

delay(200); Serial.print("..");

}

Serial.println();

Serial.println("NodeMCU is connected!"); Serial.println(WiFi.localIP());

dht.begin(); ThingSpeak.begin(client);

}

void loop() {

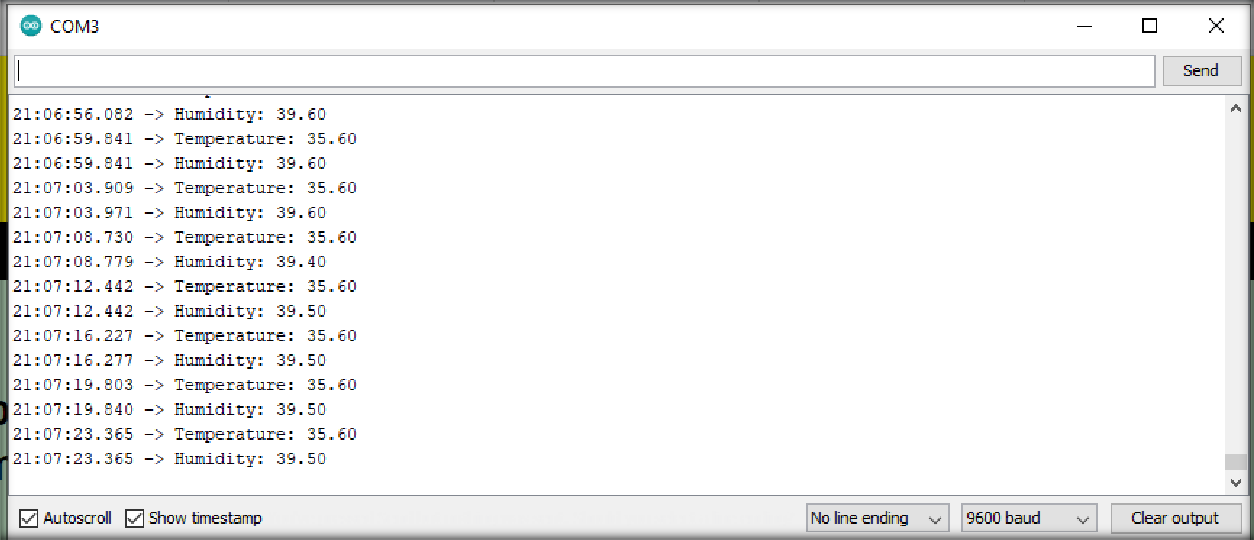
// put your main code here, to run repeatedly: float h = dht.readHumidity();

float t = dht.readTemperature(); Serial.println("Temperature: " + (String) t); Serial.println("Humidity: " + (String) h);

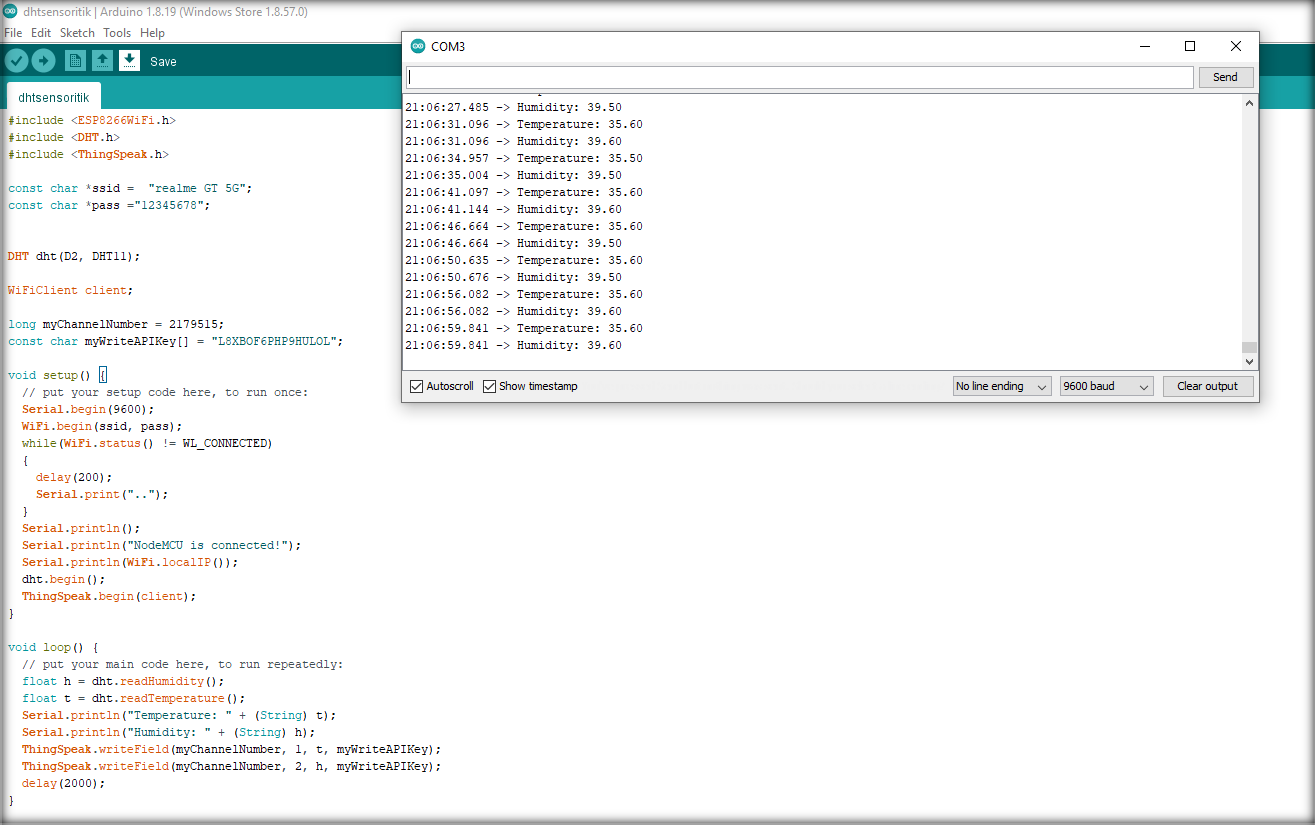
ThingSpeak.writeField(myChannelNumber, 1, t, myWriteAPIKey); ThingSpeak.writeField(myChannelNumber, 2, h, myWriteAPIKey); delay(2000);

}

## Working

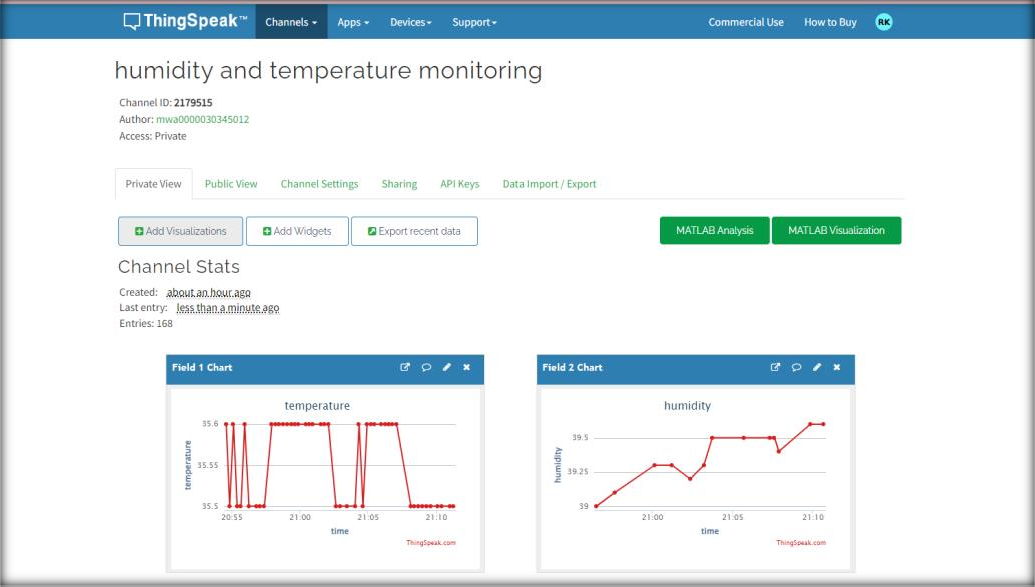


### Working on arduino ide



1. Working on thingspeak cloud

On graph



### On digital

