WEATHER MONITORING SYSTEM USING IOT IN REAL-TIME

A Report submitted

in partial fulfillment for the Degree of

Bachelor of Technology

in

Computer Engineering

by

SHEIKH MOHD SABIR (19BCS058)

MD ASFAK (19BCS051)

Department of Computer Engineering
Faculty of Engineering & Technology
Jamia Millia Islamia, New Delhi –110025
Nov, 2022

CERTIFICATE

This is to certify that the project report entitled **Weather Monitoring System Using IoT In Real-Time** submitted by **SHEIKH MOHD SABIR 19BCS058, MD ASFAK 19BCS051** to the Department of Computer Engineering, F/O Engineering & Technology, Jamia Millia Islamia New Delhi – 110025 in partial fulfillment for the award of the degree of **B. Tech in (Computer Engineering)** is a bonafide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Dr. Mohammad Amjad	**Dr. Bashir Alam**
Supervisor	Head of Department
Department of Computer Engineering	Department of Computer Engineering

New Delhi
Nov, 2022

DECLARATION

I declare that this project report titled **Weather Monitoring System Using IoT In Real-Time** submitted in partial fulfillment of the degree of B. Tech in (Computer Engineering) is a record of original work carried out by me under the supervision of **Dr. Mohammad Amjad** , and has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited

SHEIKH MOHD SABIR
19BCS058

MD ASFAK
19BCS051

New Delhi-110025
28th Nov, 2022

ACKNOWLEDGMENTS

We would like to thank our mentor **Dr Mohammad Amjad** for giving us the opportunity to take up this project. We thank him for his invaluable guidance and overwhelming moral support. We are also very grateful to our project coordinator **Dr Danish Raza Rizvi** and **HOD, Prof. Bashir Alam** for his valuable support throughout the project. We also thank our entire faculty for their teaching, guidance and support when required. We regret any inadvertent omissions.

Sheikh Mohd Sabir (19BCS058)
Md Asfak (19BCS051)

ABSTRACT

In the present time, there's a need to monitor weather conditions as rapid changes in weather conditions bring many hurdles in the development of the world. These changes then contribute to climate change which may lead to increased temperatures, sea level rise, changes in precipitation, and more frequent or intense rainfall. This Weather Monitoring System based on IoT removes this problem since it monitors the weather conditions in Real-Time. The technology behind this system i.e. IoT (Internet of Things) is an efficient solution for monitoring changes in a particular region like Okhla, Dilshad Garden, Chandni Chowk, etc.

The proposed system uses electronic gadgets and sensors to monitor temperature, humidity, pressure and quantity of rainfall of the present environment. The data captured through sensors then sent to the node MCU controller. All sensed data is uploaded using Arduino IDE. The nodeMCU calculates and displays weather parameters on the LCD display. Then this data is sent to the internet using IoT techniques. In this case, Weather Monitoring using SMS has some limitations. Since it sends SMS to few numbers and the time for sending SMS increases as the number of phone numbers increases. It is better to know the information about the weather of a particular place then they have to visit that particular site from where anyone can see it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Climate change and environmental monitoring have drawn too much attention in the present developing world. Monitoring weather's condition plays a substantial role in every person's life. To monitor and evaluate the circumstances in case of exceeding the prescribed level of parameters (such as temperature, humidity, pressure and rainfall), an effective environmental monitoring system is needed. Everyone wants to stay updated about the latest weather conditions of any place such as a parking garage, college campus, office or any other place.

As the world is changing rapidly so there should be weather stations in places of concern. In this proposed system, we present a weather station which is helpful for any place and it is based on IoT. It is equipped with sensor devices to predict the behavior of a specific area of interest. Then report the measured data in real time on serial monitor and also on cloud once data is sent to it.

For this system, we used nodeMCU and different environmental sensors like DHT11, BMP180 and rain drop sensor. The system can also be very useful in Smart Cities by giving the weather update of any particular place like a particular office or room.

1.1.1 Internet of Things (IoT)

It refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud. It is the future technology of connecting the world at one place. A typical IoT system works through the real-time collection of data. The worldwide number of IoT-connected devices is projected to increase to 43 billion by 2023.

1.1.2 Motivation of the study

In the present time, there's a need to monitor weather conditions as rapid changes in weather conditions bring many hurdles in the development of the world. These changes then contribute to climate change. Weather monitoring is important for forecasts because they are used to protect life and property. Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets.

1.1.3 Objective

It continuously measures the meteorological data such as temperature, humidity, pressure etc and stores the collected data in the cloud. Monitor the Weather in Real-Time using cloud data visualizations

CHAPTER 2
RELATED LITERATURE

In today's world, pollution monitoring is designed by different environmental parameter system models and is presented as an IOT based Weather and reporting system where you can collect, analyze, and present your measured data on w Wireless sensor network management model co device, router, gateway node and management center. End device is responsible for collecting sensor network data,and sending them to parent data is sent to the gateway node from the parent node by router. After receiving the data from the wire network, gateway node extracts data after packaging them into Ethernet formatdata,and sends the server. Less formally, any device that runs a server could be considered a server as well. Servers are used to manage network resources. The services or information provided through the Internet that are connected through LAN and made available for users via smart phones, web browser or other web browser devices to make the system more intelligent, adaptable and efficient.

The author in [1], proposed a system that monitors and predicts the weather conditions by which anyone can plan for our day-to-day life. This activity became helpful in every field either in agriculture or industry. So as to achieve monitoring and predicting weather info, the author uses 2 stages of the weather management system.In which they amalgamate the information from the sensors, bus mobility, and deep learning technology is used to allow a weather reporting system in stations and buses in real-time. Forecasting of weather is achieved through the friction model. Depending upon the sensing measurement from vehicles like buses, the work incorporates the strength of local information processing.

The author talks about in stage-I, sensing of weather's condition, multilayer perceptron model and long-term memory are trained and then it will verify using temperature data, humidity, air pressure of the test environment. In Stage-II, the training is applied to learn the time series of weather information. To get accurate data or not, to check the system performance, the author compared the predicted weather data and actually obtained data from the Environment Protection Administrator and central Baeuro of Taichung observation system that calculate the prediction of accuracy. The author finally talks about how the proposed system has reliable performance on monitoring weather. And this model also proposed a one-day weather forecast or prediction via the training model. So finally, the author demonstrates that this system presents a real-time weather monitoring and prediction system using bus information management. The author represents 4 basic components 1- Information management. 2- Interactive bus stop 3- Machine learning predictive model 4- weather information platform. In this, information shown via dynamic chart.

In the paper [2], The author proposed an "Intelligent weathering system using the internet of thing" research in which, with the help of machine learning algorithms and IoT technology, climate conditions are measured and predict the next upcoming condition. The author presents a system that is a very cost-effective and efficient method for controlling the weather condition and sends it to the cloud so that the value is fetched and available on the internet and can be seen anywhere by connecting through the internet. There is a significant role of temperature, humidity, and pressure in the system. also have a different area used in the system such as agriculture, logistics, etc. Weather monitoring, and forecasting is important for the advancement of these industries. IoT technology used in the formation and development of this research, is an advanced & efficient approach for connecting the sensor to the cloud that stores the real-time data and connects the whole world of things through a network. The thing might be anything as the sensor, electronic gadget, and automotive electronic equipment as well. The whole system deals with the monitoring and controlling of environmental conditions like temperature, smoke, gas, wind, pressure, relative humidity level, and many other gasses with sensors that transfer the data or information to the cloud platform and store the data on it. A machine learning prediction algorithm is used in this system. Machine learning is a branch of AI (Artificial Intelligent) which deals with analyzing and predicting the given dataset. The collected data is analyzed. The real-time data sent through the sensor can be reachable through the internet in the entire world. The methodology which is used by the author as node MCU microcontroller is used to gather the information from discrete sensor which is basically on the code dumped in Nodemcu, send the data to the specific cloud as Thingspeak is used for displaying both the data with web view and also application view. Each sensor is connected with nodeMCU. It transmits the sensed data from the cloud and data analysis is performed and shows the result in the resultant view on the website. Data is fetched from the cloud in the CSV file format. And after that for further processing, it includes the data from the cloud in R Studio, and then data is processed using various algorithms

CHAPTER 3
MATERIAL AND METHOD

3.1 Material and Tools

3.1.1 NodeMCU

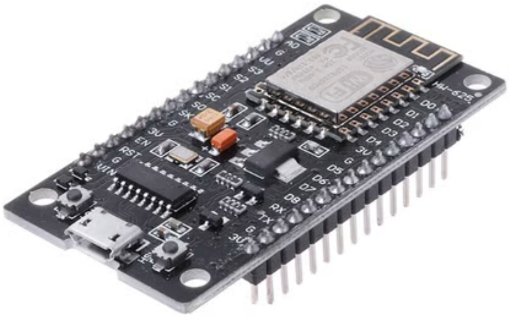

Fig3. 1: Node MCU

In this system we used NodeMCU, which is a low-cost open source IoT device and is used as a controller. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. But later it added support for the ESP32 32-bit MCU. Its memory is 128KB and storage is 4MB (i.e. flash Memory). In the proposed model, ESP8266-EX microcontroller-based WeMos D1 board is used and executed on Arduino IDE, retrieving the data from the cloud. WeMos D1 is a wifi module that is developed on ESP- 8266EX microcontroller. After the connection, it will store the data on the cloud, for storing data a thingspeak website is used to display the data regarding weather. The system displays the data on LCD and thingspeak cloud.

3.1.2 Temperature & Humidity Sensor

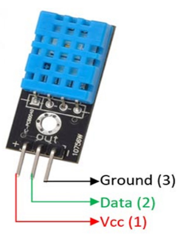

Fig 3.2: DHT11

This system utilizes temperature and moisture sensor DHT11. This DHT11 is a Temperature and Humidity sensor that features digital signal output. It provides out-standing quality, quick response, capacity to prevent interference, and cost-effectiveness. It is integrated with a high-performance 8-bit microcontroller. It has excellent quality, fast response, high performance and anti-interference ability. Each DHT11 sensor features extremely accurate calibration of the humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, enabling a variety of applications. The product is a 4-pin single row pin package. Convenient connection, special packages can be provided according to the user's needs. It uses 3 to 5V power and a max current usage of 2.5mA. It can measure 20-80% humidity levels with accuracy of 5%. It is good for 0-50° Celsius.

3.1.3 Pressure Sensor

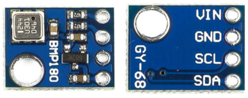

Fig 3.3: BMP180

Barometric pressure sensors measure the absolute pressure of the air around them. Depending on how you interpret the data, you can monitor changes in the weather, measure altitude, or any other tasks that require an accurate pressure reading. In this system BMP180 is used, which is an I2C standard device and a digital barometric pressure sensor. This chip uses I2C 7- bit address 0X77. It's a 4-pin tool, i.e., SDA, SCL, VIN, GND. It has ultra-low power consumption down to 3µA. It operates on 3 to 5V dc voltage. The pressure-sensing rate of BMP180 is 300-1100 hPa (9000 to -500m above the sea level). The operational range of BMP is -40° to 85° Celsius.

3.1.4 Rain-drop Sensor

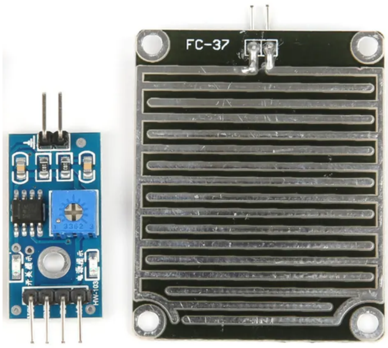

Fig 3.4: Rain-drop Sensor

It is used for rain detection and can also measure rain intensity. The raindrop sensor detects the water beyond what a humidity sensor detects. The detection of water is done on its sensor board. Connected to a 5V power supply, the LED will turn on when the induction board has no rain drop, and DO output is high. When dropping a little amount of water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and where restored to the initial state, outputs high level. A rain sensor or rain switch, is a switching device activated by rainfall.

3.2 Method

The IOT enabled weather monitoring system project, NodeMCU measures four weather parameters using 3 different sensors. These sensors are temperature and humidity sensor, pressure sensor and rain-drop sensor. These 3 sensors are directly connected to NodeMCU. NodeMCU has inbuilt Analog to digital converter. Sensors are connected with the NodeMCU ESP8266 in the architecture. It calculates and displays these weather parameters on the LCD display. Then it sends these parameters to the Internet using IoT techniques. The process of sending data to the internet using Wi-Fi is repeated after constant time intervals. Then the user can visit Thingspeak to view this weather data. Hence, the user gets Live reporting of weather conditions. Internet connectivity or Internet connection with Wi-Fi is compulsory in this IOT based weather monitoring reporting system project. The proposed system is an effective weather monitoring system with less no. of sensors. It is reliable and also free and available publicly. The proposed model measures the accurate value of the environmental condition.

Fig.1 represents the architecture diagram. With the help of this architecture, Every sensor is connected with node MCU pins. The power supply is given to the node MCU by USB which is connected with pc.

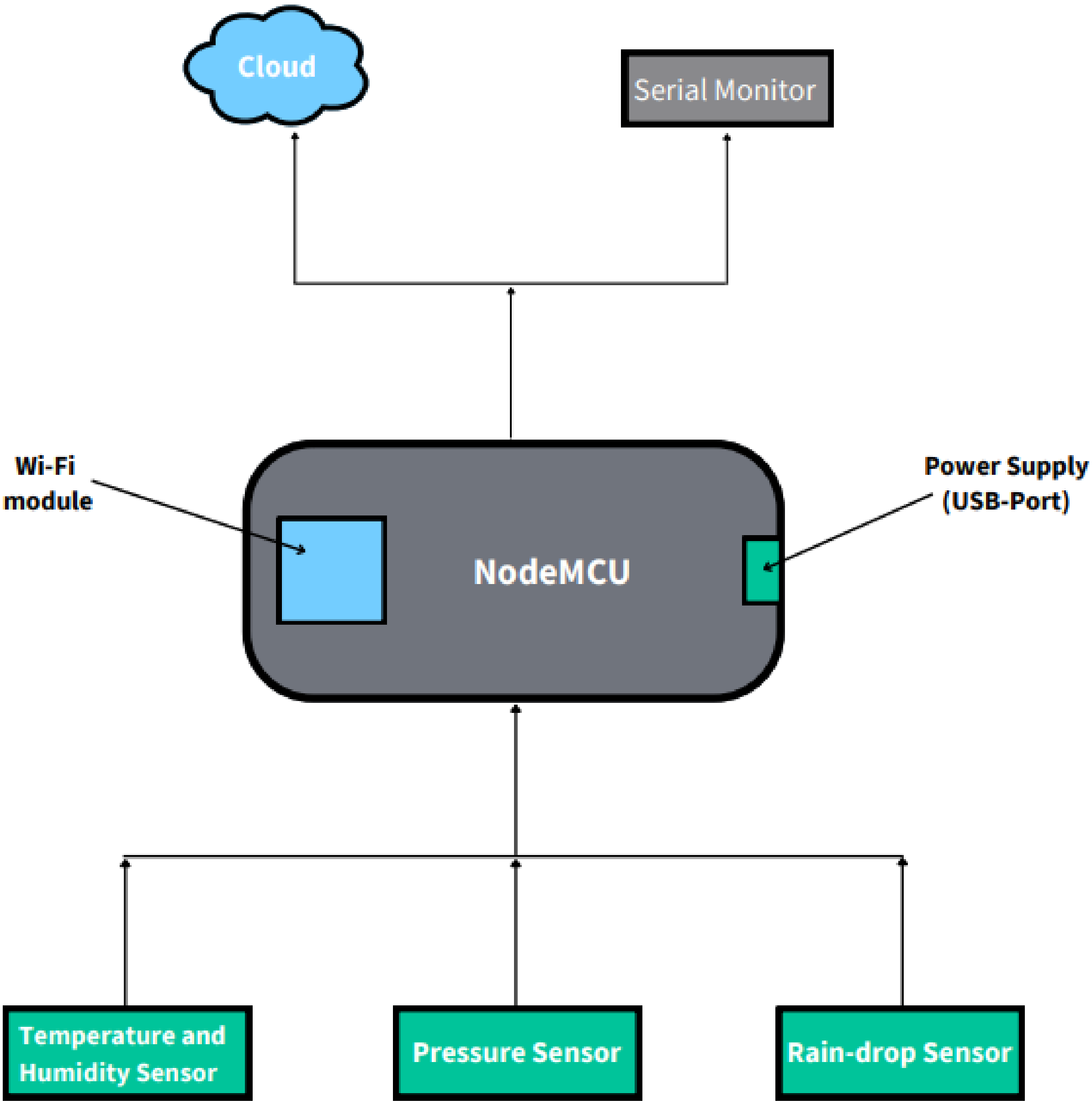

Fig 3.5: The Architectural model of Weather Monitoring System

3.2.1 Circuit Diagram

The circuit diagram of our proposed system is given below. The diagram represents the connection of the sensor and how the connection will be done.

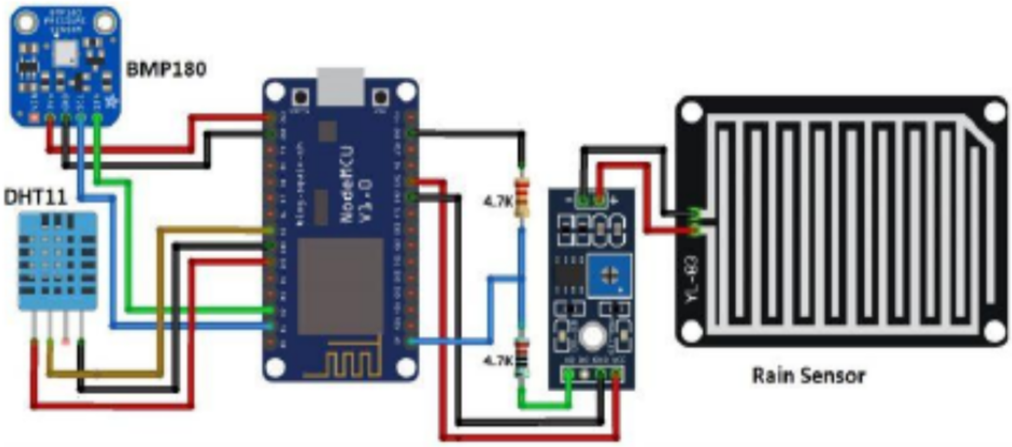

Fig.3.6. Circuit diagram of the weather monitoring system

Table3. 1. Pin configuration between nodeMCU and DHT11

VCC	3V3
DATA	D5
GND	GND

Table 3.2. Pin Configuration between nodeMCU and BMP180 sensor

SDA	D2
SCL	D1
GND	GND
VIN	3V3

Table 3.3. Pin Configuration between nodeMCU and Rain sensor:

A0	GND (including with 4.7k register)
GND	GND
VCC	3V3

CHAPTER 4

4.1 Results

Table 4.1 : Experimental results obtained at Delhi - Sadar Bazaar

Date:- 17 Nov, 2022

S.No.	Time (hh:mm)	Pressure value	Temperature value	Humidity value	Rain value	Weather
1.	17:44	988.64	25.8 °C	46%	24%	Foggy
2.	18:00	988.60	25.8 °C	45%	23%	Foggy
3.	21:41	988.60	25.8 °C	45%	23%	Foggy

Table 4.2 :Experimental results obtained at Delhi - JMI Campus

Date:- 18 Nov, 2022

S.No.	Time (hh:mm)	Pressure value	Temperature value	Humidity value	Rain value	Weather
1.	14:31	990.75	23.8 °C	44%	27%	Cold
2.	15:47	990.21	24.8 °C	45%	23%	Cold
3.	16:58	990.66	25.8 °C	45%	28%	Foggy
4.	17:13	990.64	24.8 °C	47%	26%	Cold

https://thingspeak.com/channels/1937721

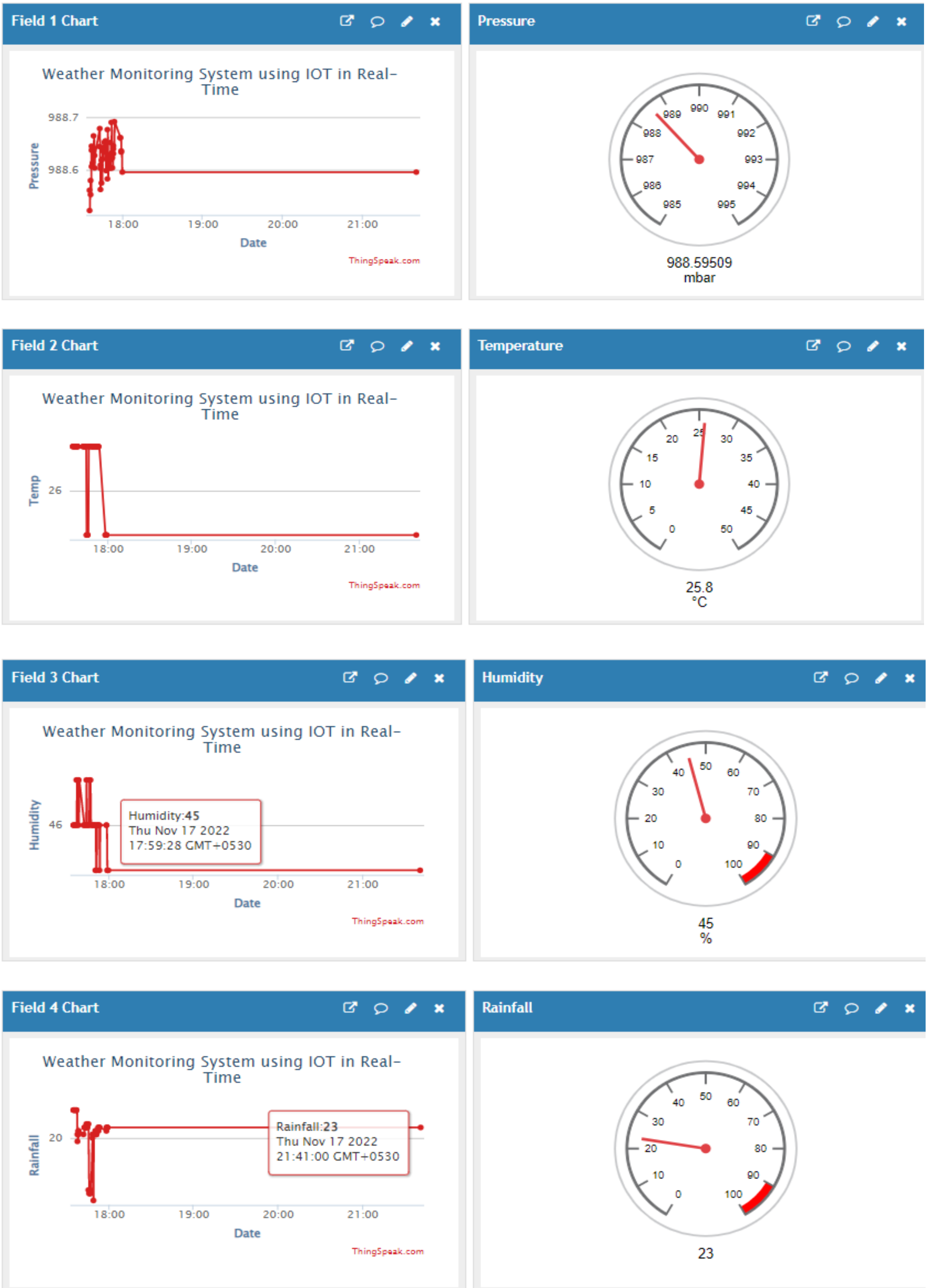

Fig 4.1: Flow Charts and Gauge meters

https://thingspeak.com/channels/1957207

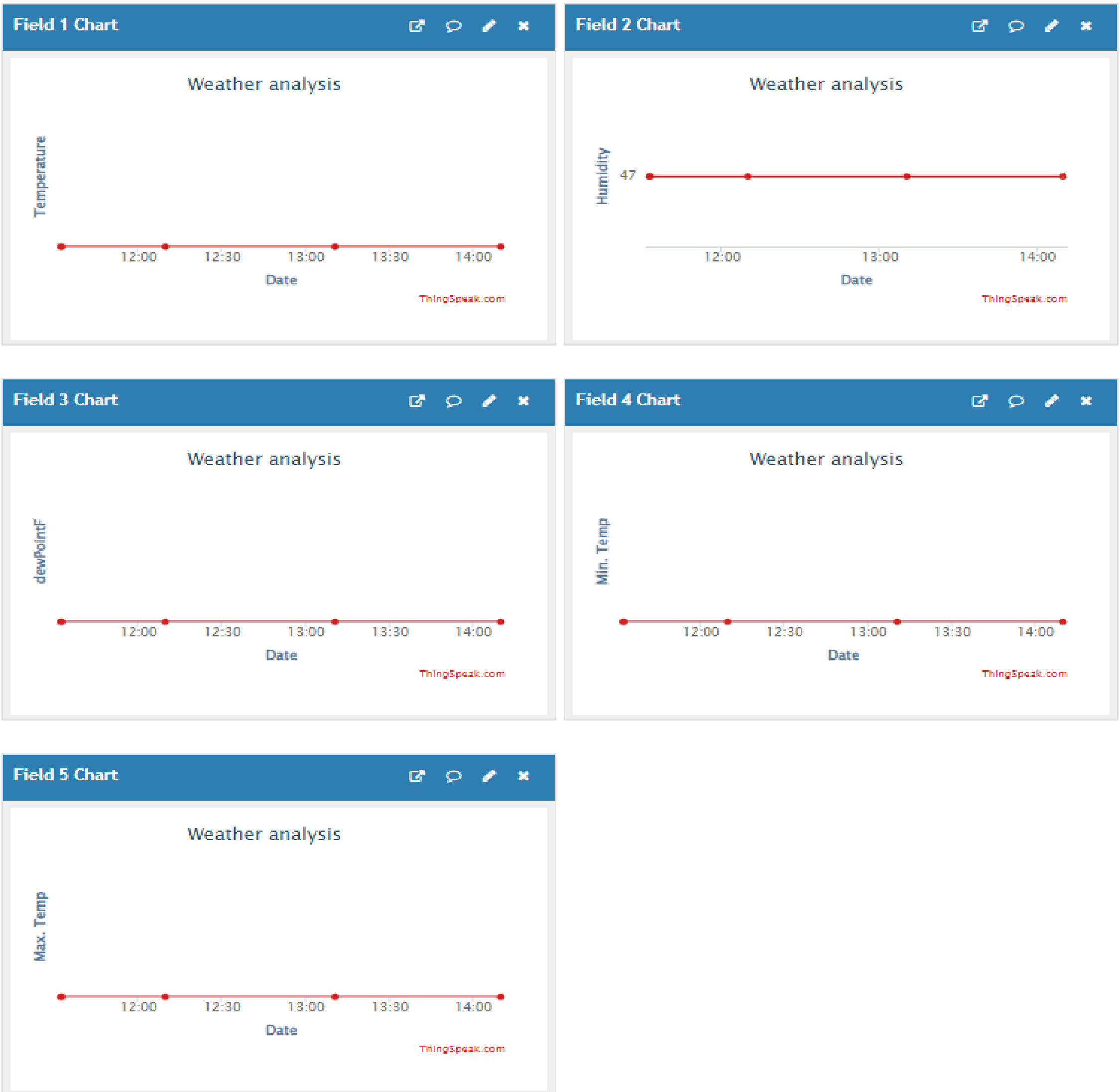

Fig 4.2: Flow Chart of analyzed data

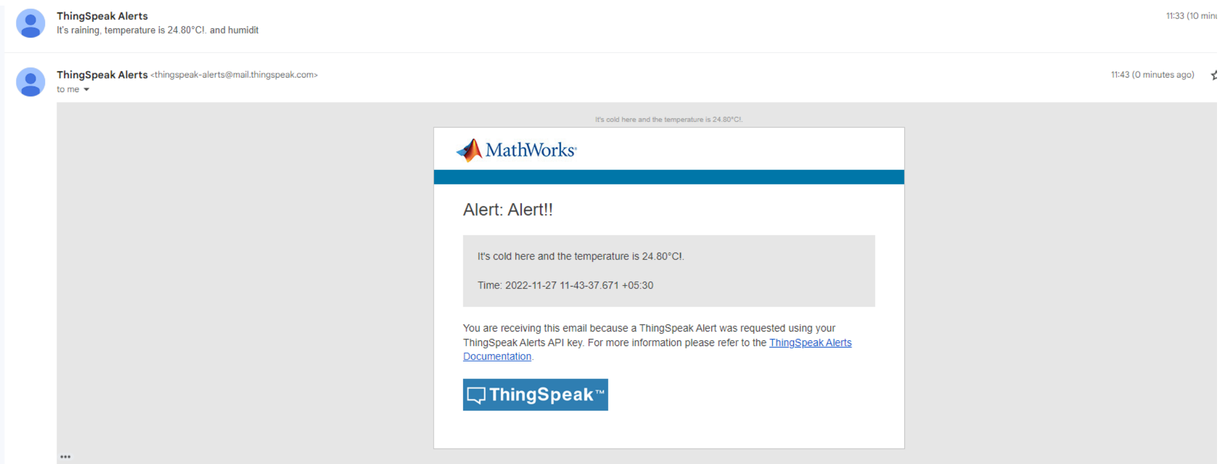

Fig 4.3: Alert Email to the client

4.2 Source Code

—> Arduino Code

```
// the program code
#include <ESP8266WiFi.h>
#include <SFE_BMP180.h>
#include <ThingSpeak.h>
#include <WiFiClient.h>
#include <Wire.h>
#include "DHT.h"

#define DHTPIN D5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
SFE_BMP180 bmp;
double T, P;
char status;
const char* ssid = "wifipcg";  // replace with your wifi ssid and wpa2 key
const char* password = "hello123";
const char* server = "api.thingspeak.com";
unsigned long myChannelNumber = 1937721;
const char* myWriteAPIKey = "ZG62CHYWJONCUPK8";
WiFiClient client;

void setup() {
    Serial.begin(115200);
    delay(10);
    dht.begin();
    bmp.begin();
    Wire.begin();

    // Connect to WiFi network
    WiFi.disconnect();
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    ThingSpeak.begin(client);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
```

```
    Serial.println("WiFi connected");
    pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

void loop() {
    // Blink nodemcu LED
    digitalWrite(LED_BUILTIN, LOW);   // Turn the LED on (Note that LOW is the
voltage level but actually the LED is on; this is because it is active low on the
ESP-01)
    delay(1000);                      // Wait for 1 sec
    digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage
HIGH

    // BMP180 sensor
    status = bmp.startTemperature();
    if (status != 0) {
        delay(status);  // Wait for the measurement to complete:
        status = bmp.getTemperature(T);
        status = bmp.startPressure(3);  // Calculates pressure values from 0 to 3
        if (status != 0) {
            delay(status);
            status = bmp.getPressure(P, T);
            if (status != 0) {
            }
        }
    }
    delay(1000);  // wait for 1sec

    // DHT11 sensor
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("failed to read DHT");
    }
    delay(1000);  // wait for 1sec

    // Rainfall Sensor
    int r = analogRead(A0);       // Gets the rainfall values through the sensor
    r = map(r, 0, 1024, 0, 100);  // Converts these values from 0 to 100
    int R = 100-r;

    // Display data to Serial Monitor
```

```
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println("'C");
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.println("%");
    Serial.print("Absolute Pressure: ");
    Serial.print(P, 2);
    Serial.println("mbar");
    Serial.print("Rainfall: ");
    Serial.println(R);
    Serial.println("");

    // Sent data to ThingSpeak.com cloud
    ThingSpeak.setField(1, (float)P);
    ThingSpeak.setField(2, (float)t);
    ThingSpeak.setField(3, (int)h);
    ThingSpeak.setField(4, (int)R);
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);  // Update in
ThingSpeak

    Serial.println("Waiting.........");
    delay(15000);  // thingspeak needs minimum 15 sec delay between updates
}
```

—> **MATLAB Analysis Code**

```
% matlab analysis for, curr.Temp, curr.Humidity, dewpointC, minTemperatureC,
 maxTemperatureC
readChannelID = 1937721; % Channel ID from which you are reading the data
readAPIKey = 'UIMD71RM0ERIBGL8'; % Channel read API key from which you are reading
 the data
TemperatureFieldID = 2; % Temperature Field ID
HumidityFieldID = 3; % Humidity Field ID
[tempC,time] = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID, ...

 'NumPoints',1,'ReadKey',readAPIKey); % Read the temperature data
humidity = thingSpeakRead(readChannelID,'Fields',HumidityFieldID, ...

 'NumPoints',1,'ReadKey',readAPIKey); % Read the humidity data
```

```
[maxTempC,maxTempIndex] = max(tempC);
[minTempC,minTempIndex] = min(tempC);

b = 17.62; % constant for water vapor (b)
c = 243.5; % constant for barometric pressure (c).
gamma = log(humidity/100)+b*tempC./(c+tempC);
dewPointC = c*gamma./(b-gamma)
display(dewPointC,'Dew Point is');
writeChannelID = 1957207; % Write channel ID
writeAPIKey = 'AGTRZJS3ERAPDCBB'; % Write channel API Key
thingSpeakWrite(writeChannelID,[tempC,humidity,dewPointC,minTempC,maxTempC],'Field
 s',[1,2,3,4,5],'WriteKey',writeAPIKey);
```

—> **MATLAB Alert Code**

```
% matlab alert code, for email alerts of weather conditions.
readChannelID = 1937721; % Channel ID from which you are reading the data
readAPIKey = 'UIMD71RM0ERIBGL8'; % Channel read API key from which you are reading
  the data
TemperatureFieldID = 2; % Temperature Field ID
HumidityFieldID = 3; % Humidity Field ID
RainFieldID = 4; % rainlevel Field ID
[tempC,time] = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID, ...

  'NumPoints',1,'ReadKey',readAPIKey); % Read the temperature data
humidity = thingSpeakRead(readChannelID,'Fields',HumidityFieldID, ...

  'NumPoints',1,'ReadKey',readAPIKey); % Read the humidity data
rain = thingSpeakRead(readChannelID,'Fields',RainFieldID, ...

  'NumPoints',1,'ReadKey',readAPIKey); % Read the rainlevel data
[maxTempC,maxTempIndex] = max(tempC);
[minTempC,minTempIndex] = min(tempC);

alertApiKey='TAKWgsT7Ri1a26AXlBN';
alertURL = "https://api.thingspeak.com/alerts/send";
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey ]);
alertSubject = sprintf("Alert!!");
```

```
if humidity >= 90
   alertBody = sprintf("It's highly humid, raining and cold here, temperature is
  %0.2f°C!. and humidity is %0.2f %", minTempC, humidity);
elseif rain < 24
   alertBody = sprintf("It's raining, temperature is %0.2f°C!. and humidity is
  %0.2f %", minTempC, rain);
else
   if minTempC <= 25
       alertBody = sprintf("It's cold here and the temperature is %0.2f°C!.",
  minTempC);
   elseif minTempC <= 28
       alertBody = sprintf("It's Foggy here and the temperature is %0.2f°C!.",
  minTempC);
   else
       alertBody = sprintf("It's hot here and the temperature is %0.2f°C!.",
  maxTempC);
   end
end

try
webwrite(alertURL, "body", alertBody, "subject", alertSubject, options);
catch
   % Code execution will end up here when an error 429 (error due to frequent
  request) is caught
end
```

4.3 Conclusions

Devices that monitor weather's parameters with minimum cost in our proposed system. The proposed system works on the client-side architecture model. The proposed approach observed various environmental information using multiple sensors. The system which is designed has used less sensors than the existing model. The main aim of our proposed model is to make the system cost-effective, affordable. So that everyone can use it freely.

This System monitors the changes happening over the environment and provides enough ways for the users to access the information from anywhere through the cloud. The temperature and humidity sensor, pressure and rain-drop sensor will monitor and give the details about the changes happening over the climate.

The proposed model is not only collecting data but also making decisions on the basis of observed data. It is very much helpful for farmers because it collects environmental data making an interpretation for the former. But it is also very important for making other decisions such as our industry work purpose, transportation. The accuracy of the proposed model is nearly accurate with real data.

4.4 Future Work

- Adding more sensor to monitor other environmental parameters such as CO2 and Oxygen Sensors etc.
- One can implement a few more sensors and connect it to the satellite as a global feature of this System.
- It can also be implemented in hospitals or medical institutes for the research & study in "Effect of Weather on Health and Diseases, hence to provide better precaution alert.

REFERENCES

1. Zi-Qi Huang, Ying-Chih Chen and Chih-Yu Wen, "Real-Time Weather Monitoring and Prediction Using, City Buses and Machine Learning", Vols. 3 to 21, Published 10 September (2020)
2. M. Prasanna, M. Iyapparaja, M. Vinothkumar, B Ramamurthy, S.S. Manivannan, "An Intelligent Weather Monitoring System using Internet of Things", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November (2019)
3. Ferdin Joe John Joseph "IoT Based Weather Monitoring System for Effective Analytics" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April (2019)
4. Md. Majharul Islam "Weather Monitoring System Using Internet of Things", Juniper Publishers Volume 3 Issue 3 - October (2019)
5. Anita M. Bhagat1, Ashwini G. Thakare1, Kajal A. Molke1, Neha S. Muneshwar1, Prof. V. Choudhary "IOT Based Weather Monitoring and Reporting System Project", IJTSRD Volume-3, Issue- 3, Mar-Apr (2019)
6. Puja Sharma1 and Shiva Prakash2 "Real Time Weather Monitoring System Using Iot", ITM Web of Conferences 40, 01006 (2021)
7. https://create.arduino.cc/projecthub/tarantula3/weather-station-using-arduino-and-nodemc u-d2b9d3
8. https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/ growing-opportunities-in-the-internet-of-things
9. https://www.hackster.io/techmirtz/using-16x2-lcd-with-arduino-d89028
10. https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-/all
11. https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics