VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

IOT FUNDAMENTALS

ECE 3501

J COMPONENT REPORT

TITLE: WEATHER MONITORING SYSTEM

TEAM MEMBERS

KARTHIKEYAN BM (20BEC0503)

YOGESHWAR P (20BEC0606)

SLOT: L3 + L4

FACULTY: Dr. SANGEETHA A

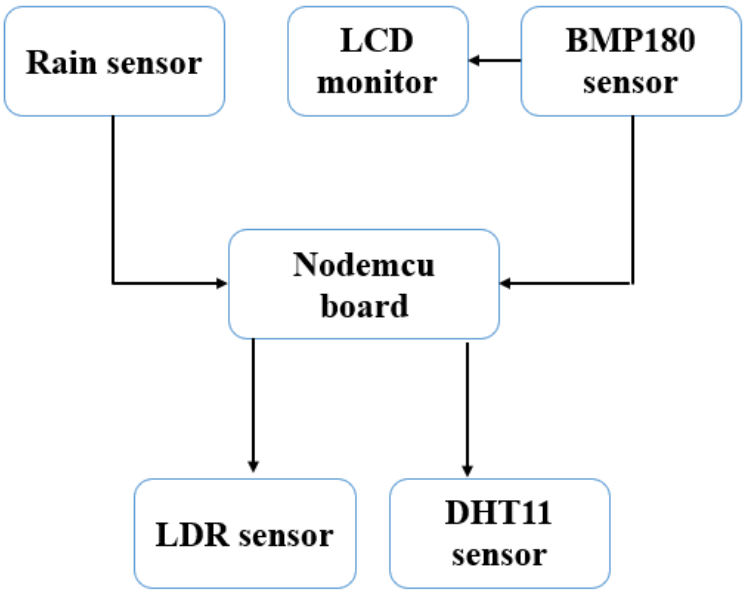
SYNOPSIS:

- Introduction
- Block schematic
- Specification of components
- Circuit diagram
- Code
- Results
- Reference

INTRODUCTION:

- The main aim of the project is to design and build a Weather Monitoring System using Nodemcu, Temperature sensor, rain sensor and BMP180 sensor, LDR sensor etc.
- The system functions with tracking and monitoring environmental circumstances such as temperature, relative humidity, light intensity, pressure and quantity of rainfall with sensors and it will accessed through cloud with Wi-Fi and also we can see the conditions through our mobile.
- The technological advances behind this is due to Internet of Things (IoT), which is an efficient and effective solution for linking the things to the web and to connect the entire world of things in a network.

BLOCK SCHEMATIC:



SPECICFICATIONS OF COMPONENTS:

Nodemcu:

- Nodemcu has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

DHT11 Sensor:

- The DHT11 is a commonly used Temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data

BMP180 Sensor:

- The BMP180 sensor is mainly used to measure atmospheric pressure or biometric pressure. The working principle of the air pressure sensor is very simple, it works based on the weight of air. Because the air around us has a certain weight, and this weight has a specific pressure

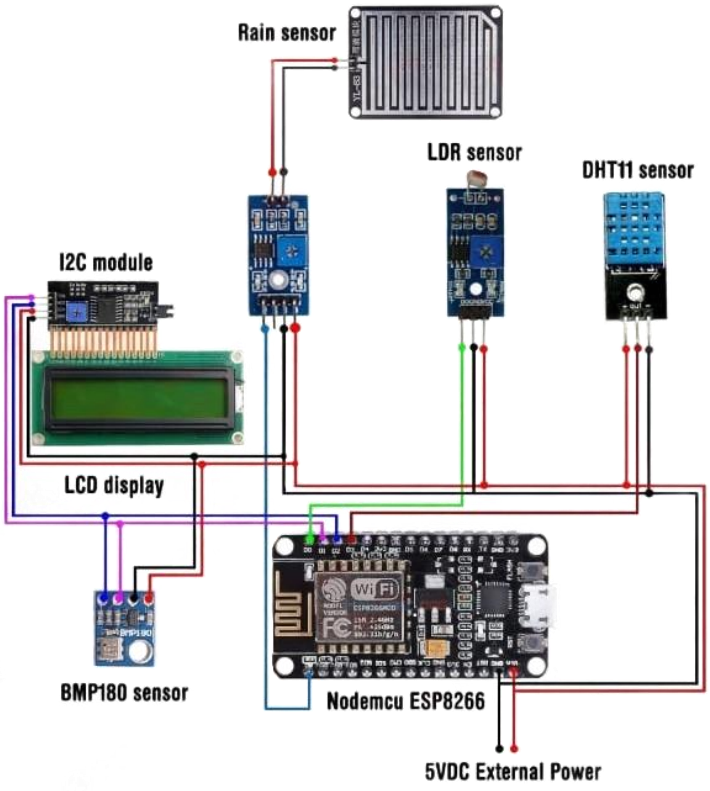
Rain Sensor:

- A sensor that is used to notice the water drops or rainfall is known as a rain sensor. This kind of sensor works like a switch. This sensor includes two parts like sensing pad and a sensor module.

LDR Sensor:

- LDR sensor module is a low-cost digital sensor as well as analog sensor module, which is capable to measure and detect light intensity. This sensor also is known as the Photoresistor sensor. This sensor has an onboard LDR (Light Dependent Resistor), that helps it to detect light.

CIRCUIT DIAGRAM:



CODE:

```
//Include the library files
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <SFE_BMP180.h>



//Initialize the LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Create an object for the BMP180 sensor
SFE_BMP180 bmp;

char auth[] = "PSDOIPBuB0VkgH1t95g-R2tJ4SIuIWcG";//Enter your Auth token
char ssid[] = "realme 7 pro";//Enter your WIFI name
char pass[] = "Karthi123@";//Enter your WIFI password

DHT dht(D3, DHT11);//(sensor pin,sensor type)
BlynkTimer timer;

//Define component pins
#define rain A0
#define light D0
```

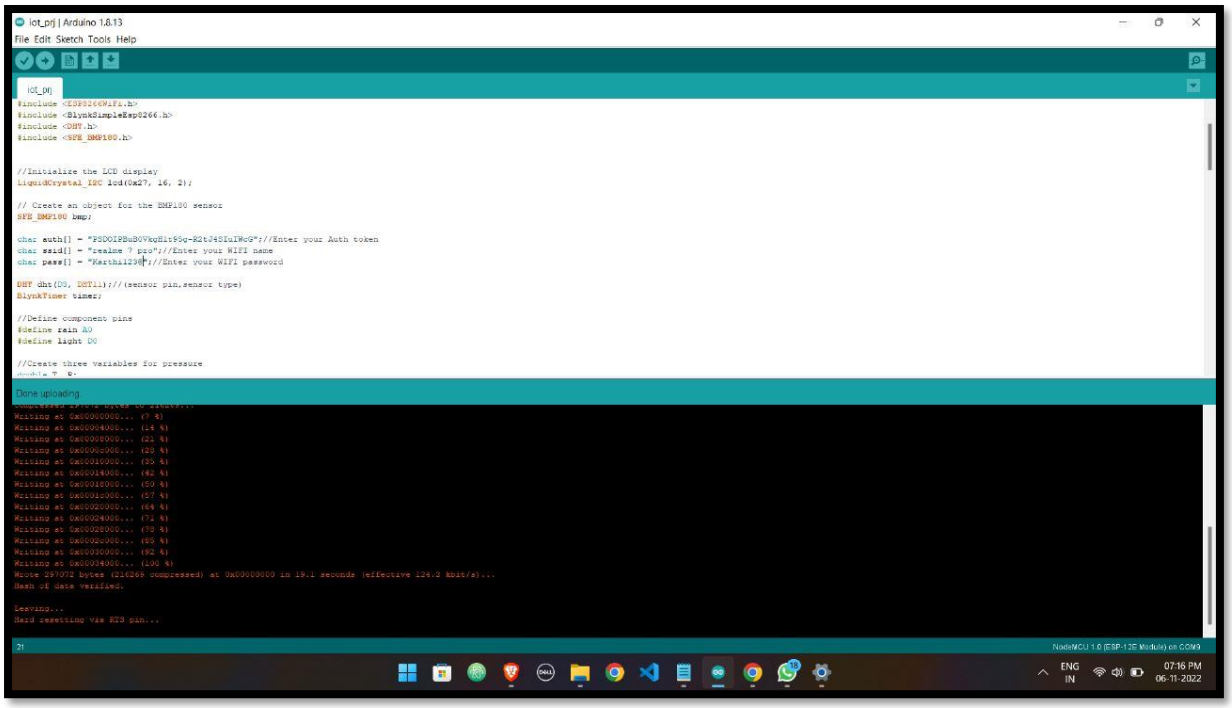
```
//Create three variables for pressure
double T, P;
char status;

void setup() {
  Serial.begin(9600);
  bmp.begin();
  lcd.init();
  lcd.backlight();
  pinMode(light, INPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  dht.begin();

  lcd.setCursor(0, 0);
  lcd.print("Weather Monitor");
  lcd.setCursor(4, 1);
  lcd.print("System");
  delay(4000);
  lcd.clear();

  //Call the functions
  timer.setInterval(100L, DHT11sensor);
  timer.setInterval(100L, rainSensor);
  timer.setInterval(100L, pressure);
  timer.setInterval(100L, LDRsensor);
}

//Get the DHT11 sensor values
void DHT11sensor() {
```

```
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);

  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(t);

  lcd.setCursor(8, 0);
  lcd.print("H:");
  lcd.print(h);

}

//Get the rain sensor values
void rainSensor() {
  int value = analogRead(rain);
  value = map(value, 0, 1024, 0, 100);
  Blynk.virtualWrite(V2, value);

  lcd.setCursor(0, 1);
  lcd.print("R:");
  lcd.print(value);
  lcd.print(" ");
```

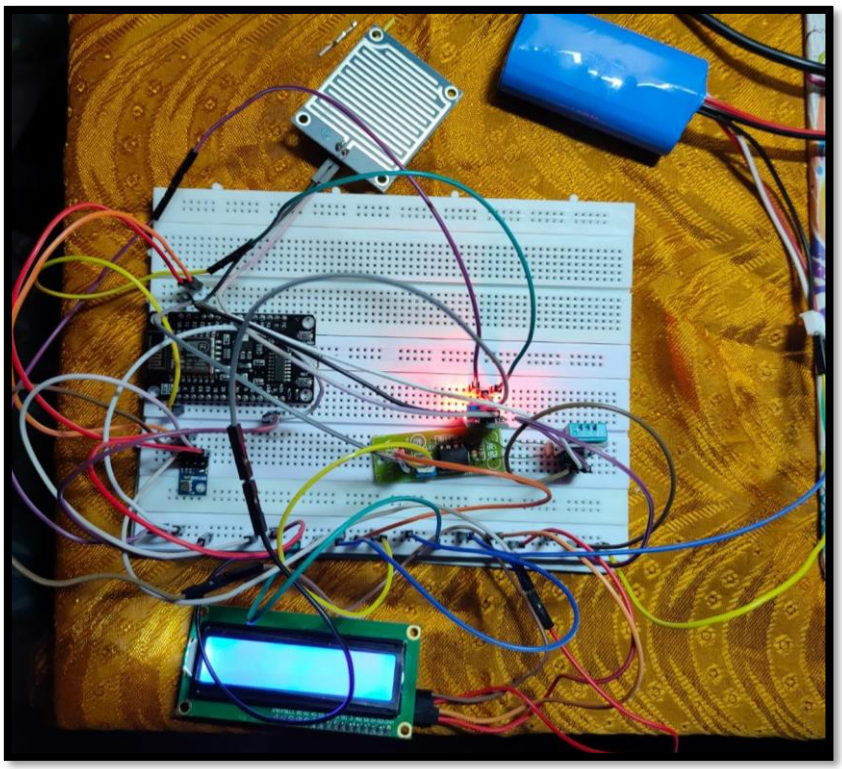
```
}

//Get the pressure values
void pressure() {
  status =  bmp.startTemperature();
  if (status != 0) {
    delay(status);
    status = bmp.getTemperature(T);

    status = bmp.startPressure(3);// 0 to 3
    if (status != 0) {
      delay(status);
      status = bmp.getPressure(P, T);
      if (status != 0) {

      }
    }
  }

  Blynk.virtualWrite(V3, P);
  lcd.setCursor(8, 1);
  lcd.print("P:");
  lcd.print(P);
}

//Get the LDR sensor values
void LDRsensor() {
  bool value = digitalRead(light);
  if (value == 0) {
    WidgetLED LED(V4);
```

```
    LED.on();
  } else {
    WidgetLED LED(V4);
    LED.off();
  }

}

void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```

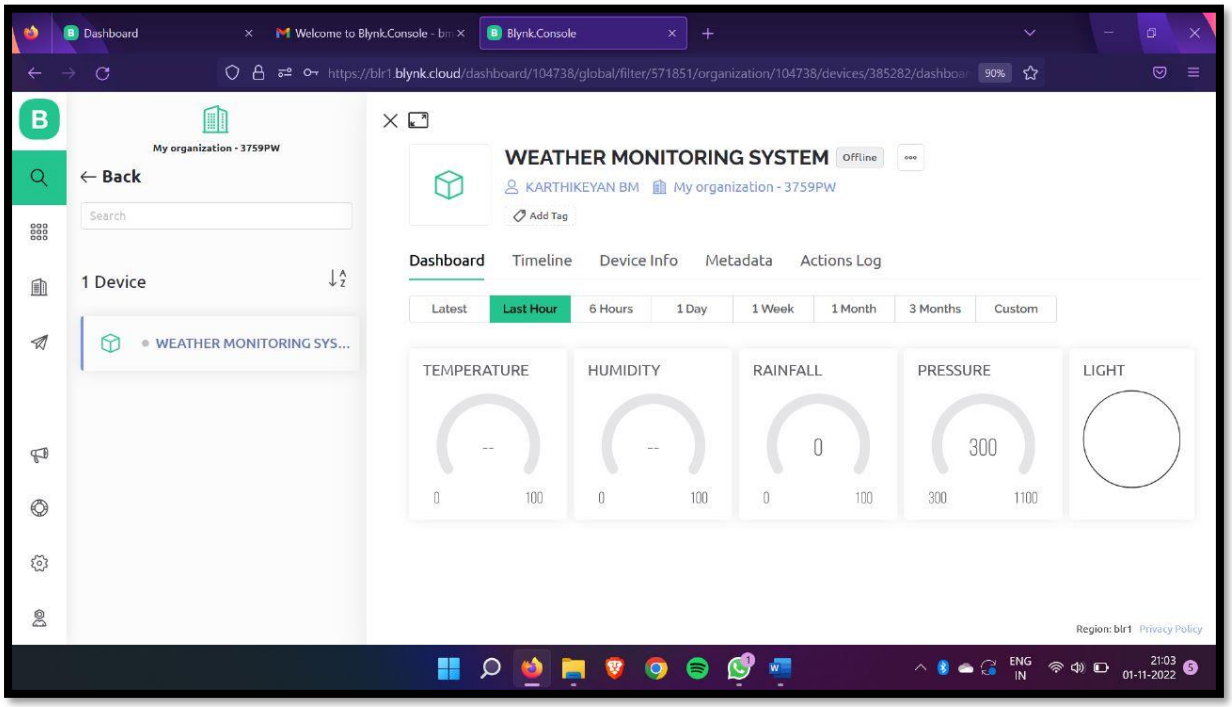
ARDUINO IDE:



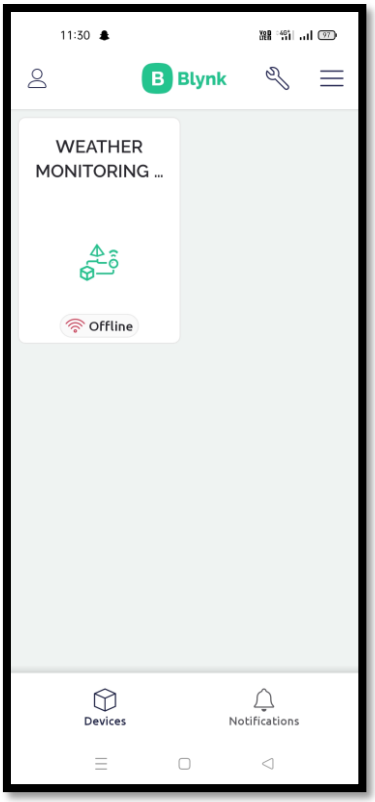
HARDWARE:



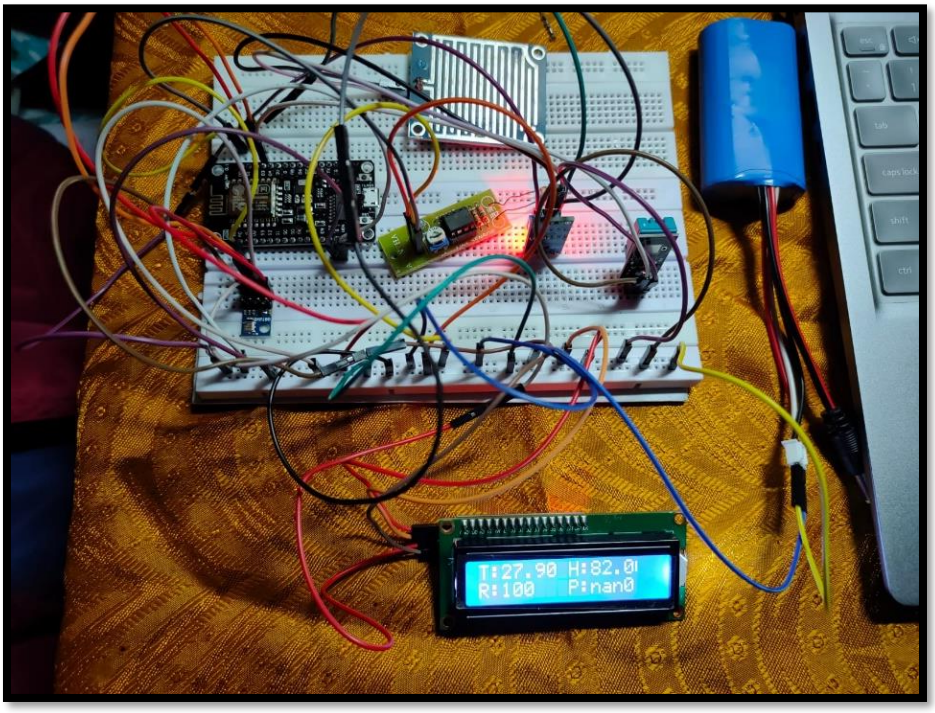
BLYNK CLOUD CONSOLE:



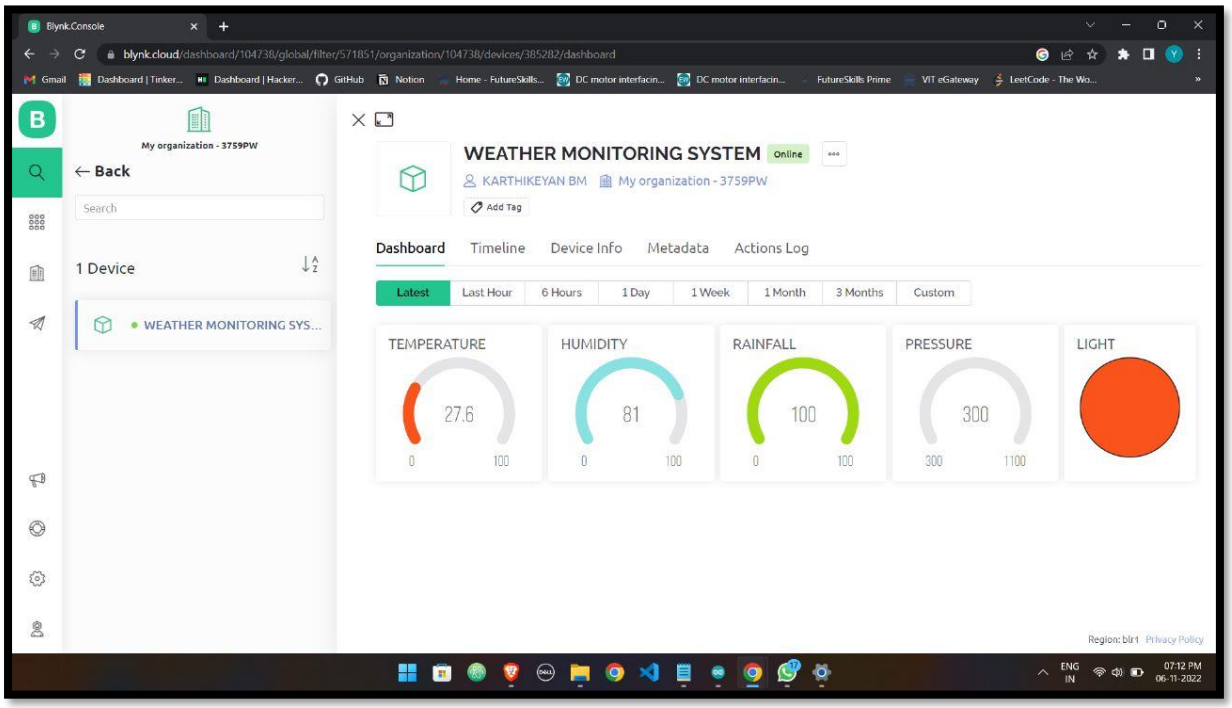
BLYNK MOBILE CONSOLE:

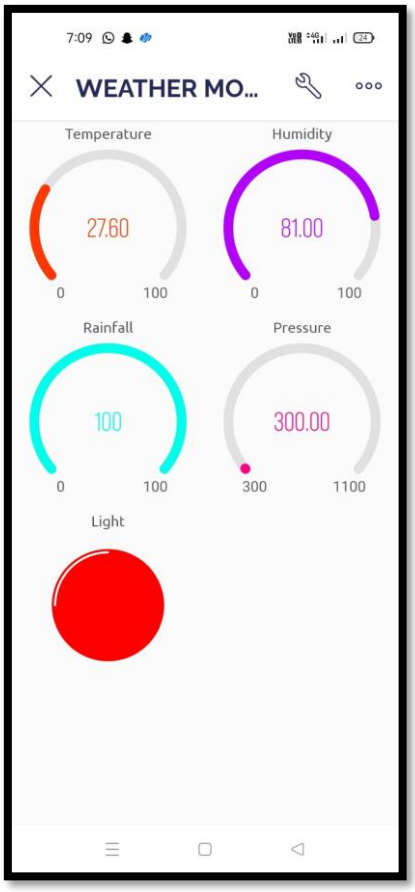


RESULTS OF THE PROJECT:

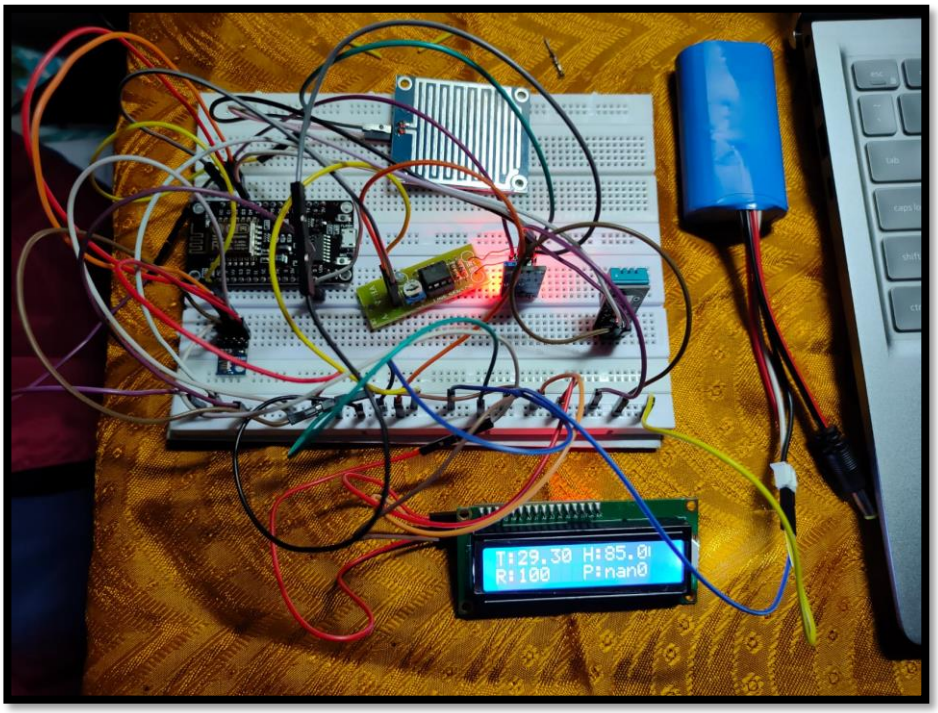


MONITORED IN CLOUD USING BLYNK:

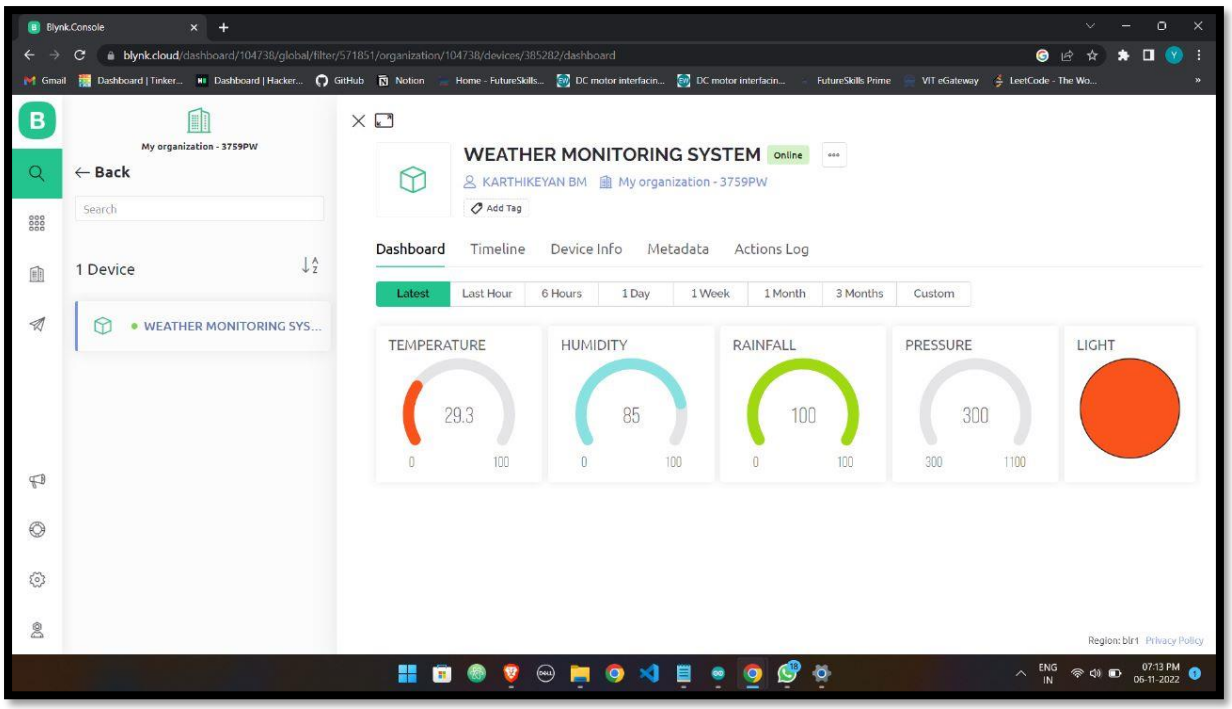




RESULTS FOR DIFFERENT CONDITIONS:



MONITORED IN CLOUD USING BLYNK:



REFERENCES:

- R. K. Kodali and S. Mandal, "IoT based weather station," 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2016, pp. 680-683, doi: 10.1109/ICCICCT.2016.7988038.
- N. Kumari, Sakshi, S. Gosavi and S. S. Nagre, "Real-Time Cloud based Weather Monitoring System," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 25-29, doi: 10.1109/ICIMIA48430.2020.9074848.