CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

Project Report

Course No: ETE-416

Course Title: IoT and Industrial Automation Sessional

Name of the Project: Weather Monitoring and Remote Plant Watering system using ESP32 and Blynk Live Server and App

Date of Submission: 17.01.2023

Submitted By: Group-03

Maisha Mostaque	**ID:** 1708011
Sakawat Hossain Sakib	**ID:** 1708012
Pramit Dutta	**ID:** 1708013
Jahid Hasan Angkur	**ID:** 1708014
Hasanur Rohman	**ID:** 1708015

Submitted To: Khaleda Akhter Sathi

Lecturer,

Dept. of ETE, CUET

Remarks

Signature

TITLE: Weather Monitoring and Remote Plant Watering system using ESP32 and Blynk Live Server and App.

ABSTRACT: In Today's World of agriculture, knowing live environmental condition and take recountable actions or precautions is one of the biggest issues for farmers and agriculturist. Hence to help farmers and agriculturist,we came up with a iot based system.The proposed system in this project is an advanced solution for monitoring the weather conditions from any place by sensing temperature and humidity. The data acheived from the implemented system can be accessible and controllable in the internet from anywhere in the world using an app. The communication part (client-server) can also be done by sending and uploading the data to the server so that it can be useful for analysis. Based on the data this system allows the user to water the plants remotely until the soil reaches to its required moisture level. This project interfaced with DHT11 sensor, Nodemcu wifi module (ESP 32), relay module etc which intregated with the IoT platform by Blynk Live Server & App. The device shows an average delay of 11.67 and 7.33 seconds for temperature and humidity updatation respectively. The temperature and humidity is measured with an accuracy of 97.44% and 96.64% repectively. Each component of the device and system has been tested and also functioning according to the purpose, so the system has the potential to be used in the process of weather monitoring and watering the plants remotely incorporate with IoT platform.

INTRODUCTION:

Background study: IoT is an area of communication technology that is still developing. Since agriculture is the main industry in our nation, we strive to exploit the benefits of IoT technology in agriculture in this project. There are various factors, like temperature and humidity, that contribute to successful cultivation. But these dependencies are not well known to our farmers. Our objective is to create an IoT project that measures temperature and humidity and takes appropriate action based on the scenario in order to assist them in their work. All components of an IoT system, including sensors and devices, communication, data processing, and a user interface, are present in this end-to-end project.

Statement of the problem: The key issue that we hope to address with this project is the ability to remotely regulate a motor based on the responses of temperature and humidity sensors. This will make it easier to cultivate a field from a distance without a farmer's present. This will address the issue of the farmer being continuously observed while remaining in the field. That's why in this IoT project we build a system combining sensor (DHT11), actuator (Motor) , server (BLYNK) and a controlling UI provided by BLYNK to measure the humidity and temperature, maintaining them in a suitable range by supplying water using motor.

Project objective : The primary goal of this project is to use a DHT11 sensor to measure temperature and humidity in real-time, upload that data to a cloud server (BLYNK) that can be accessed from anywhere in the world, and create a user interface (UI) where users can decide whether to start a motor or not using a mobile application. User not required to present in the field by his own but can control from any where in the world just need an internet connection. That's why our main objective to build a semi automated IoT system that monitor the humidity and temperature and a user can maintain them in a suitable range by controlling the motor from any where.

Project hypothesis: We anticipate that keeping plants in an appropriate environment all the time by remotely adjusting the temperature and humidity will aid in their optimal growth. Staying in the field will save the farmer time compared to continuous monitoring. Production will eventually rise, and human interference will decline.

METHODOLOGY: In this project, a remote weather monitoring system with automatic plant watering system is implemented. Digital temperature and humidity sensor (DHT11) is used to measure the temperature and humidity. The sensor data is then processed using ESP32 module and later the data is send to Blynk live server via WiFi. The blynk server also have a mobile app by which the uploaded data can be monitored using a smartphone.

However, for automatic plant watering system, a relay is connected to the ESP32 module. The module receives instruction from the blynk server and operate the motor by providing a signal to the relay module.

In this way, an automated IoT system that can monitor the weather and remote controlling of a plant irrigation system is implemented.

Required equipment and software:

Hardware:

1. ESP32 Module;
2. Breadboard;
3. Jumper Wire (Male-to-Male, Male-to-Female, Female-to-Female and copper wire);
4. Ac to Dc adopter;
5. Power supply Module;
6. Digital Temperature and Humidity Sensor (DHT11);
7. Relay Module;
8. DC Motor;
9. 9V Battery;
10. Battery Cap;
11. Water Pipe;
12. Water Tank;

13. Cork Sheet and
14. Glue Stick

Software: Arduino IDE 2.0.3

Live Server: Blynk Live Server

Prototype Design: To evaluated the project's performance, a prototype is designed in which we try to imitate the natural working environment of the project. A labled protoype design of the model is given below:

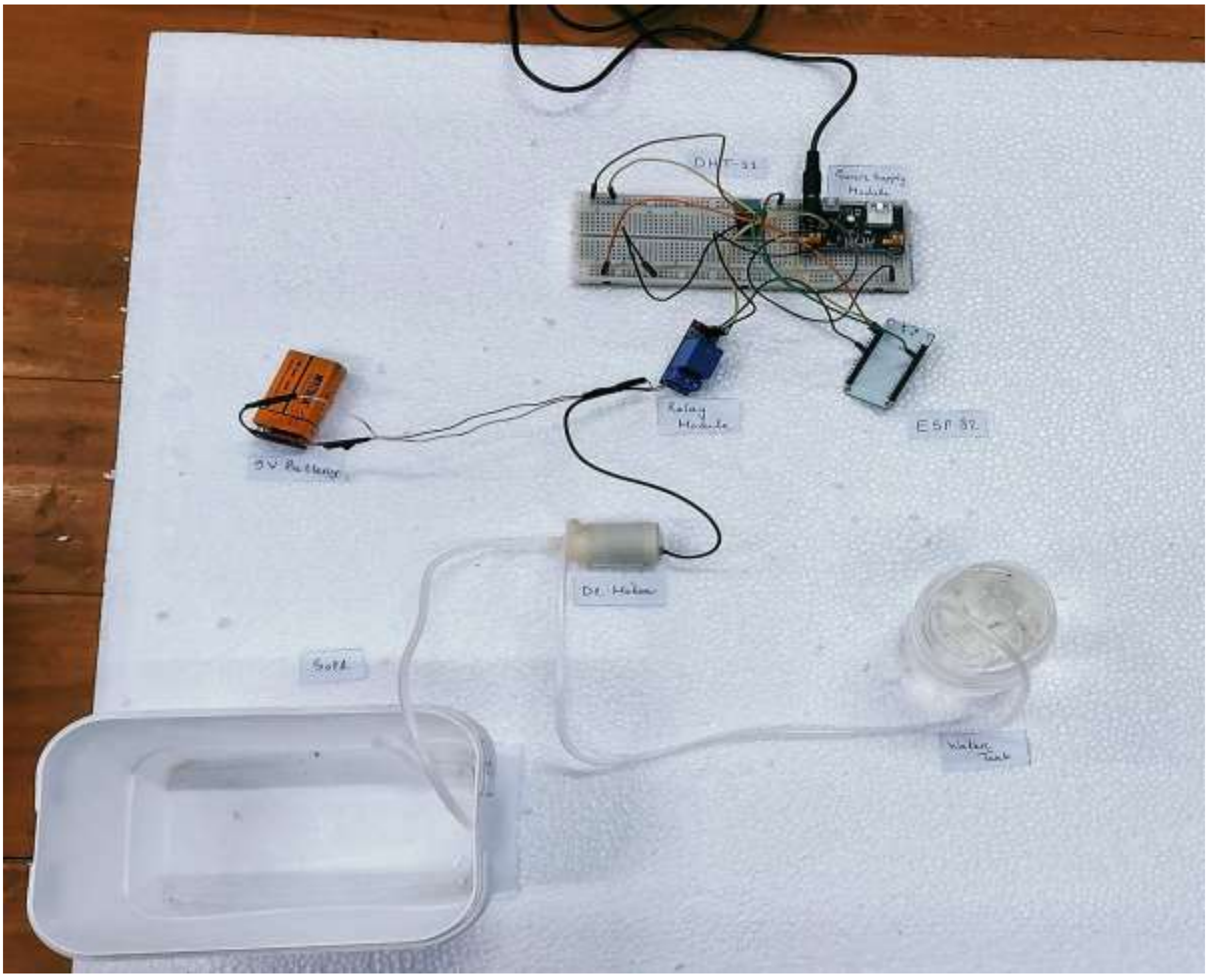


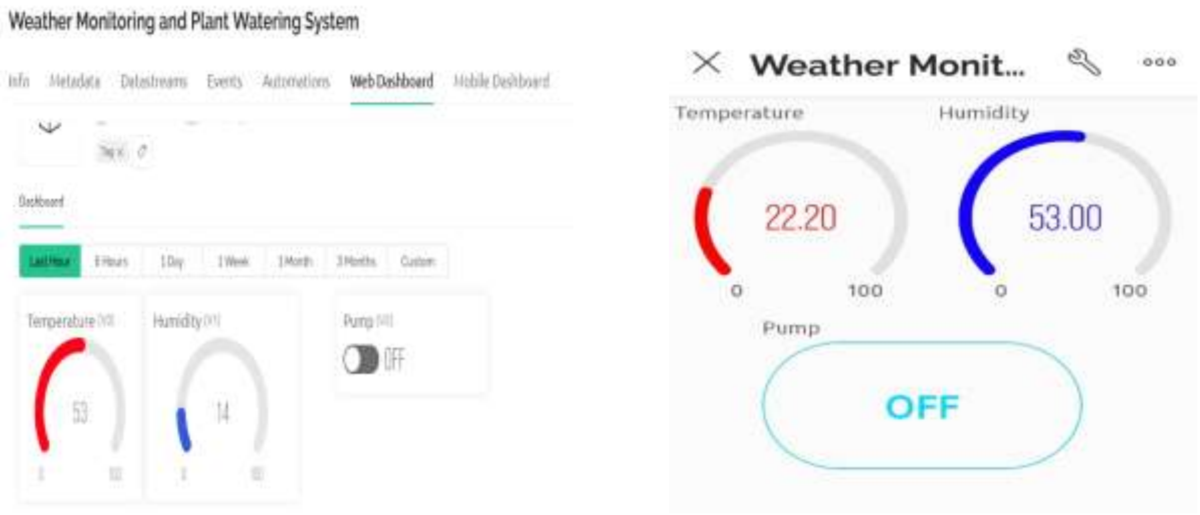
Figure-01: Labeled Prototype Design of the Proposed Model

IoT Stack: The core IoT fundamental stack has three layer- Application, Communications Network andThings: sensor and actuator. The communication in this project is performed using the ESP32 microcontroller. The application layer is implemented using Blynk Live server. Finally as sensor, DHT11 is used and on the other hand as actuator a dc motor is used.

Application Layer

The application layer in IoT refers to the services that enable the functionality of IoT service. Our proposed system perform mainly two function- monitoring and controlling. These functionality is enabled using the Blynk live server. The functionality and configuration of the Blynk live server is discussed below:

Blynk Live Server: The DHT11 sensor measure the temperature and humidity while ESP32 process and transmit it. The transmitted data needs to be stored in a server. Moreover, there should be an interface to control the motor remotely. For this reason, in this project, we make a interface in blynk to record the data and a virtual switch to turn on the motor. The web dashboard and mobile dashboard are set up as follow:



(a) Web Dashboard (b) Mobile Dashboard

Figure-02: Blynk Live Server Setup

Communications Network Layer

The communication network layer consist of protocol and technology that perform the transmission of data. The above mentioned project uses ESP32 microcontroller to perform these protocol and communication. The ESP32 microcontroller's description and the implementation of communications network layer using this microcontroller is discussed below:

ESP32 Microcontroller: Espressif Systems' ESP32 is a low-cost, low-power microcontroller with integrated Wi-Fi and Bluetooth capabilities. It is built around the ESP32 microcontroller and is programmed with the Arduino IDE. It features a dual-core 32-bit processor, 520 KB of SRAM, and a variety of communication protocol including I2C, SPI, UART, and PWM is used.

In this project, ESP32 is used as the processor. It process the DHT11 sensor value and send the measured value to blynk live server via wifi network. Moreover, this microcontroller is capable to receive data from the blynk server and use it to control the plant watering system. A table (Table-01) containing the parameters of ESP32 is given below:

Table-01: Parameters of ESP32

Parameter	Value
Data Rate	120 Mbps
Antenna Power	20.5 dBm
Operating Voltage	3.3V
Operating Current	600mA
General Purpose input output pin	36

Things: Sensors and Actuators Layer

The sensor and actuator acts like the bridge between our system and environment. In this proposed system the measured data is temperature and humidity which is measures using the DHT11 sensor. The system also has a feature to control a dc motor where the dc motor acts as an actuator for this proposed system.

Digital Temperature and Humidity Sensor (DHT11): DHT11 sensor is used to measure the temperature and humidity of an environment. This sensor measure the data and produces a digital output which can be read by a microcontroller. In this project the ESP32 reads this value and transmit the value to Blynk live server. The Table-02 contains some significant parameter of DHT11 sensor.

Table-02: Parameters of DHT11

Parameter	Value
Operating Voltage	3.3~5 V
Opearting Current	2.5~8 mA
Tempearture Range and Error	(0-50)°C with ±2°C error (-40-80)°C with ±0.5°C error
Humidity Range and Error	(20-90)% with ±5% error (0-100)% with ±2% error
Response Time	< 2s

DC Motor and 9V Battery: In this project, as the actuator a dc motor is used where the operating voltage is 9V. A PP3 9V alkaline battery produced by MOTOMA is used as the power source for this motor.

To implement the proposed system, we have to use some other component that are not included in the IoT fundamental stack. The details, working, configuration and integration of these component in our project is discussed below:

Ac to Dc adopter: This ac to dc adopter is used to provide power to the power supply module. The adopter converts 120~240V, 50~60Hz ac current to 9V, 2A dc current.

Power Supply Module: The power supply module takes 9V dc current as input and produces two output voltage- 3.3V and 5V. The power supply module has a push button which can be used to control the power supply of the whole system. In this project, the module provide 3.3V supply to the relay module and 5V supply to the sensor and ESP32 microcontroller where the module works as a buck converter which convert higher dc voltage to lower dc voltage.

Relay Module: In this project, the relay module is used to execute the instruction send by blynk live server. The relay module has a normally open (NO) channel where the we connect the dc motor and the 9V supply is connected to the common port. When the blynk live server send a signal the ESP32 receives it and trigger the relay module. After that the relay module connect power supply to the motor and thus the motor operates.

When the blynk server sends the off instruction, the ESP32 receives it and trigger the channel again and power supply gets disconnected.

Simulation: Before implementing these project, an automated watering system was designed using proteus. In this system, a DHT11 sensor was used to measure temperature and humidity. Arduino was used to process these value and based on that a servo motor was controlled. The servo motor turns on when the humidity is less that 50%. A lcd display was used to display the measured value and the status of the motor. The proteus schematic design is given below:

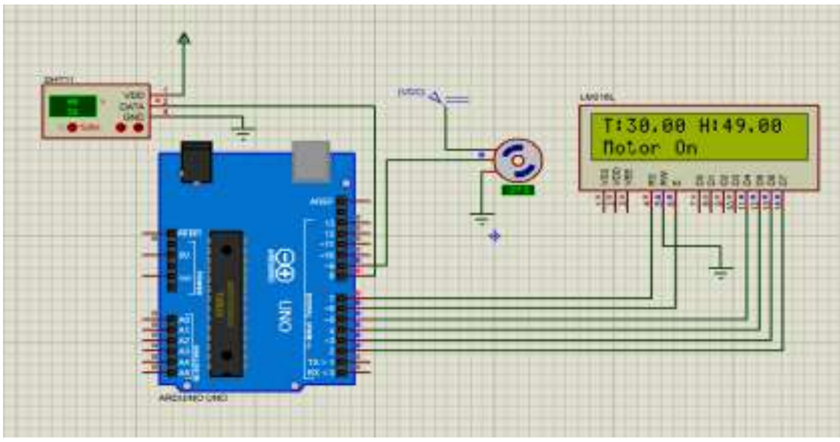


Figure-03: Proteus Schematic Design

In this automated water pump design, only one layer of the IoT fundamental stack was fully implemented. Things: Sensor and Actuator layer was implemented. In this system, there was an interaction between sensor and actuator which was basically the function of an application layer was executed. However, there was no user interface or a platform that integrated all the layers. In view of that, it can be concluded that here the Things: Sensor and Actuator layer was implemented fully and the application layer was implemented partially.

Arduino IDE 2.0.3: The ESP32 is used as the processor in this project. The Arduino IDE 2.0.3 which is the latest version is used as the coding platform and later is used to upload the code and perform debugging and troubleshooting via serial monitor.

Arduino IDE 2.0.3 is generally used to program Arduino and to make it compatible with ESP32 the IDE needs some additional configuration. Moreover, some library such as wifi library, blynk library, adafruit unified sensor library and dht sensor library is used in this project. The code for this project is given below:

```
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPLQNF8LYfz"
#define BLYNK_DEVICE_NAME "Weather Monitoring and Plant Watering System"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "DHT.h"
char auth[] = "******";     // auth token
char ssid[] = "******";    //wifi name
char pass[] = "******";  // wifi password
 #define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
bool Relay = 0;
#define waterPump 26
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V1, h);
  Blynk.virtualWrite(V0, t);
}
void setup()
{
  Serial.begin(115200);
  delay(1000);
  pinMode(waterPump, OUTPUT);
  digitalWrite(waterPump, HIGH);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  dht.begin();
  timer.setInterval(1000L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
}
BLYNK_WRITE(V2) {
  Relay = param.asInt();
  if (Relay == 1) {
    digitalWrite(waterPump, LOW);
    Serial.println("Motor On");
  } else {
    digitalWrite(waterPump, HIGH);
    Serial.println("Motor Off");
  }
  delay(5000);
}
```

In the above mentioned code there are mainly seven section. In the first part, the necessary library file is included these include library like WiFi library, blynk library and DHT library.

In the next part, the WiFi connection is configured where the wifi name and password is provided via code. Besides WiFi, in these part, the authorization token for blynk is also provided. After that, the pin for DHT11 and relay module is configured and the function for blynk and DHT is also declared. In the 4th part, the sensor value is measured and uploaded to blynk server which is performed under the function sendSensor(). Then, the whole process upto this is configured. This part is written inside the void setup() function which initialize the whole setup. After that, to run this whole procedure the loop function is used which is void loop() where two blynk function is called. Finally, in the last part, the code is written to receive data from the blynk server and act accordingly.

RESULT:

Table-03: Temperature Sensor Response Time Observation

Observation No.	Heat Source	Reading (°C)		Delay (s)
		Before Giving Heat	After Giving Heat	
01.	Candle	22.08	22.15	7s
02.	Match	23.60	23.62	16s
03.	Sunlight	23.40	23.45	12s

From Table-03, we notice that the temperature sensor works suitably according to the respective temperature changes and heat. From obs. No.1 deviation of temperature is the most (0.07°C), because the heating source candle is the highest heat glowing from those two. In such case, delay is also comparatively fast. Later from the next two observation delays become long & temperature shifting is also reduced according to the heating source. We can calculate the average delay as,

$$\text{Average Delay} = \frac{(7+16+12)s}{3} = 11.67 \text{ sec}$$

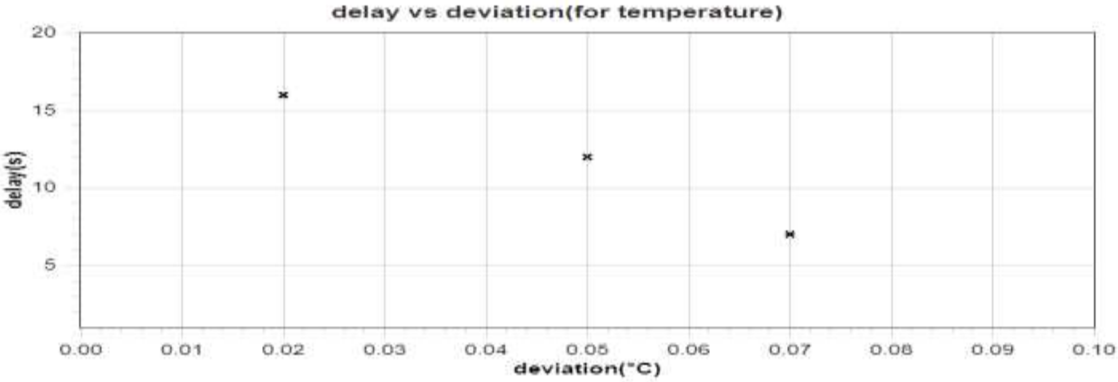
Table-04: Humidity sensor response observation

Observation No.	Reading (%)		Delay(s)
	Before GivingWater	After GivingWater	
01.	64.00	65.20	11s
02.	70.20	73.65	7s
03.	65.60	69.45	4s

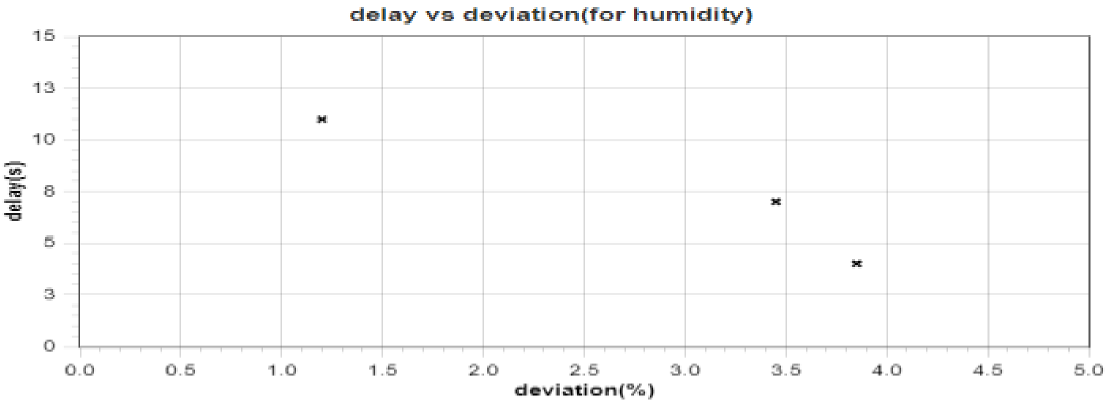
From Table-04, humidity sensor response is similar to that of temperature. It's sensing response depends on the deviation of humidity. When the deviation is large, the response is comparatively faster and vice versa. We can calculate average delay as,

Average Delay=$\frac{(11+7+4)s}{3}$ =7.33 sec.

The below curve shows the relation between delay and deviation in measurement. The delay increase for both temperature and accuracy when the deviation is large and decreased when the deviation is small.



(a) For Temperature



(b) For Humidity

Figure-04: Relation between Response and Deviation in Measurement

Table-05: Temperature sensor Accuracy Measurement

Observation No.	Measured by Mobile App(°C)	Measured by DHT11(°C)	Accuracy (%)
01.	23.59	24.20	97.41%
02.	21.65	22.50	96.07%
03.	22.59	22.85	98.84%

From Table-05, accuracy is taken from a mobile app & DHT11 sensor. DHT-11 is implemented as hardware whose task is to sense both humidity & temperature. The mobile app help us to verify the result of this sensor. In this case, the average accuracy is:

Average Temperature Accuracy = $\frac{(97.41+96.07+98.84)\%}{3}$ = 97.44%

Table-06: Humidity sensor Accuracy Measurement

Observation No.	Measured by Mobile App (%)	Measured by DHT11 (%)	Accuracy (%)
01.	53.62%	51.43%	95.92%
02.	57.37	59.97	95.47%
03.	64.60	65.54	98.54%

However, the similar mobile app is used to verify the result of DHT11 sensor. The result is recorded in Table-06. The average accuracy is calculated as:

Average Humidity Accuracy = $\frac{(95.92+95.47+98.54)\%}{3}$ = 96.64%

From the Table-02, the average error of DHT11 in terms of temperature and humidity can be observed. The result in Table-05 and Table-06 shows that the sensor is working properly

CONCLUSION: This project achieved the objectives to build a weather monitoring system that can check the weather conditions using the application, Blynk. However, Blynk isn't limited to some specific board or platform it can be used with any microcontroller. Next, the project is also able to display the current weather conditions on weather monitoring system. The implementation of a system to monitor the weather using Internet of Things (IoT) is accomplished. The system provides a low-power solution for monitoring weather and environment. The monitoring system has been tested in outdoor environment and successfully updated data from sensor. The data will be used for various types of analysis and it can be shared with other people or users. The project has the potential to be implemented for monitoring the developing cities and industrial zones, especially for pollution monitoring. In order to protect public health from pollution, the system is also able to provide an efficient and low-cost solution for the authority. It is also suitable for continuous monitoring of environment in the future. Moreover, the system can be equipped with more sensor by which they will provide important insight about weather which can be helpful for the researcher to invent new variant of crops or to optimize the cultivating technique.