<u>Project title</u>: <u>IoT based cold room monitoring system</u>

<u>Team members</u>:

ISHIMWE Viviane	Reg.No:220014846
HABUFITE James	Reg.No: 220014318
NDAYISENGA Gilbert	Reg.No: 220014134
NIRERE Gaudence	Reg.No: 220013253
MURANGWA Aimable	Reg.No: 220014131

Project supervisor: Prof. KAYALVIZHI Jayavel

Abstract:

When a crop is harvested, it is almost impossible to improve its quality. A large quantity of crops get damaged after harvesting due to improper storage. Proper storage conditions in terms of *temperature and humidity* are needed to ameliorate the storage quality which leads to the good quality of the crop.

Even if crops are removed to the plant during harvesting but they remain as living tissues. They breathe, just as humans do, and their composition and physiology continue to change after harvest. They continue to ripen and, finally, they begin to die. Cellular breakdown and death are inevitable, but can be slowed with optimal storage conditions. Fresh fruits and vegetables need low temperatures (32 to 55°F) and high relative humidity (80 to 95 percent) to lower respiration and to slow metabolic and transpiration rates[1]. By slowing these processes, water loss is reduced and food value, quality and energy reserves are maintained.

1. <u>Objective:</u>

Our main objective is to monitor various temperature conditions in the cold room so as to maintain the quality of the items.

2. <u>Requirements:</u>

2.1.Hardware requirements:

To implement this system we need:

- **Three DHT sensors** for detecting the temperature and humidity in the three separate rooms since we are having :
    - A **cold storage room**: in which will store items such as (meat, dairy products) that need from 32 to 36^0 F for maintaining their quality.

    - A **cool storage room**: in which we will store items such as (fruits) that need from 40 to 55^0 F for maintaining their quality.
    - A **warmer storage room**: that will deal with items (sweet potatoes, squash, and pumpkin) that need from 55^0 F to 60 ^{0}Ffor maintaining their quality.
- **ESP8266:** for processing purpose and providing the Wi-Fi since it is incorporated with the Wi-Fi module.
- **LED indicators:** each room has two indicators:
    - **A green LED indicator:** to show that items are kept in normal conditions of storage as it is indicated in the range.
    - **A red LED indicator:** to show that the temperature in the room is out of the range required.
- **A switch:** to turn ON and OFF the cooling system accordingly.
- **A personal computer:** in which our software will be installed.
- **A USB:** for linking the PC with the circuit.
- **The bread board:** where the components are placed for being connected accordingly.
- **Jumper wires:** to interconnect the components.
- **Smart phone:** to install Blynk application and view the data through it.

2.2.Software requirements:

-Arduino Integrated Development Environment: that the software in which we need to type the source codes.

- **Blynk application:** an application availed on the smart phone play store, for viewing the state of ach room so as to take decisions accordingly.

- **Firebase:** a cloud used for storage purpose, data can even be viewed from there.

- **Fritzing:** used for drawing the circuit diagram.

3. Methodology:

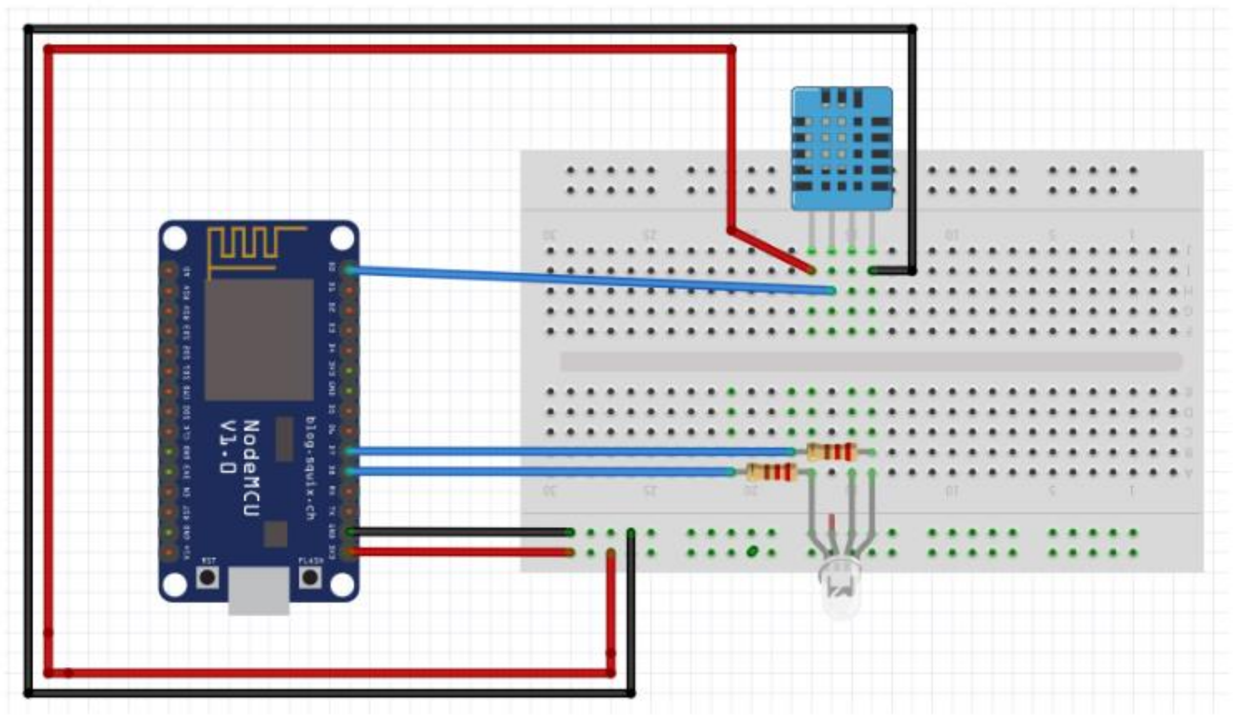
- We will start by designing our circuit using Fritzing.
- Then connect our circuit on the breadboard.
- Download and install Arduino IDE (from internet)
- Install the ESP8266 library.
- Type the source codes in the Arduino IDE.
- Download and add Firebase library in the Arduino IDE
- Download and install Blynk library in the Arduino IDE
- Open the Firebase console, create an account
- Add our project in the firebase

- Get the secret and authentication key to add in the source code
- Install Blynk application on the smart phone through the play store
- Create a Blynk account, get the SSID, add it in the source code so as to view the data
- Supply your circuit, upload the program, and view the output temperature and humidity in Firebase and Blynk.

4. How it works:

The DHT works by sensing the ambient temperature and the humidity, if the data reach the maximum limit set (i.e when the room is cold enough or when the room got hot) these data are displayed on both fire base and Blynk.

4.1.Circuit diagram:



4.2. Coding:

```
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>// blynk library
#include <ESP8266WiFi.h>    // esp8266 library
#include <FirebaseArduino.h>// firebase library
#include <DHT.h>  // dht11 temperature and humidity sensor library

#define FIREBASE_HOST "cold-room-9b2fd.firebaseio.com"// host name
#define FIREBASE_AUTH "r22bcsV8qv3VVtiI3SOMti7Mmd88pR07HJiJka3V" //
secret key of firebase
```

```
char auth[] = "6z-rFKQFJAnmJJyaxG2bD5kMEGu7JmHO"; // key of blynk
#define WIFI_SSID "Rugema"
#define WIFI_PASSWORD "ru123456" //password of wifi ssid
BlynkTimer timer;
#define DHTPIN D2          // what digital pin we're connected to
#define DHTTYPE DHT11      // select dht type as DHT 11 or DHT22
DHT dht(DHTPIN, DHTTYPE);// construct with datapin and dhtype as parameters
int greenled=D3;  // assign  red led to digital pin 3
int redled=D4; // assign green led to digital 4
float t,h=0; // declare variable to hold sensor as input
// create method to send data to blynk
void sendSensor()
{
  Blynk.virtualWrite(V1, t); // write the temperature from the DHT to virtual pin
  delay (200);  // pause program 200 ms
  Blynk.virtualWrite(V2, h);// write the humidity from the DHT sensor
  delay (200);
 }
void setup()
{
  pinMode (D3,OUTPUT); // set D3 as output pin
  pinMode (D4, OUTPUT);// set D4 as output pin
  Serial.begin(9600);  // set the baud rate of serial monitor
  delay(1000);         // pause program 1 second
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //start the wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);   // print on serial monitor ssid name
  // scann the network status
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("."); // print dot while network is not available
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());      //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);  // connect to firebase
  Blynk.begin(auth, WIFI_SSID, WIFI_PASSWORD);
  timer.setInterval(1000L, sendSensor);
  dht.begin();                          //Start reading dht sensor
}
```

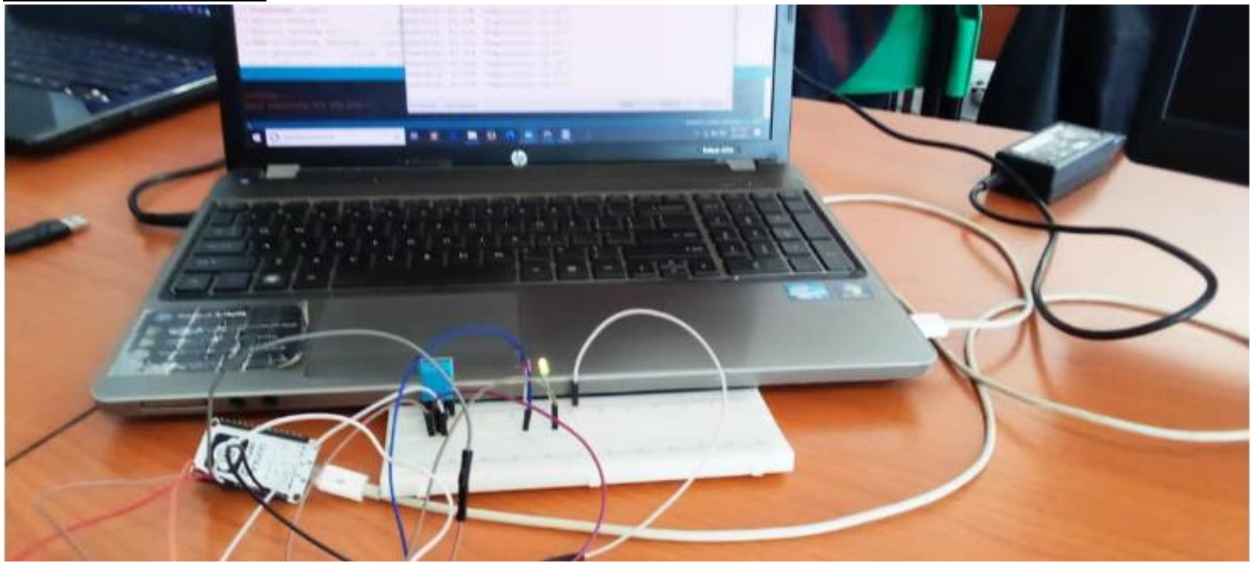
```
void loop() {
    Blynk.run();
  timer.run();
   h = dht.readHumidity();    // Reading temperature or humidity takes about 250
milliseconds!
   t = dht.readTemperature();  // Read temperature as Celsius (the default)

  if (isnan(h) || isnan(t)) {
    // Check if any reads failed and exit early (to try again).
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
    delay(2000);
}

  Serial.print("Humidity: ");  Serial.print(h);  // print on serial monitor temp, and
humidity
  String fireHumid = String(h) + String("%");     //convert integer humidity to string
humidity
  Serial.print("%  Temperature: ");  Serial.print(t);  Serial.println("°C ");
  String fireTemp = String(t) + String("°C");      //convert integer temperature to
string temperature
  delay(4000);

  Firebase.pushString("/DHT11/Humidity", fireHumid);  //setup path and send
readings
  Firebase.pushString("/DHT11/Temperature", fireTemp);//setup path and send
readings
 if(t>25)
 {
  digitalWrite(redled,HIGH);  // blink green led if temperature goes up 25
  }
  else
  {
    digitalWrite(greenled,1);  // blink red led if temperature goes bellow 25
    }
}
```

5. **Special experience faced**

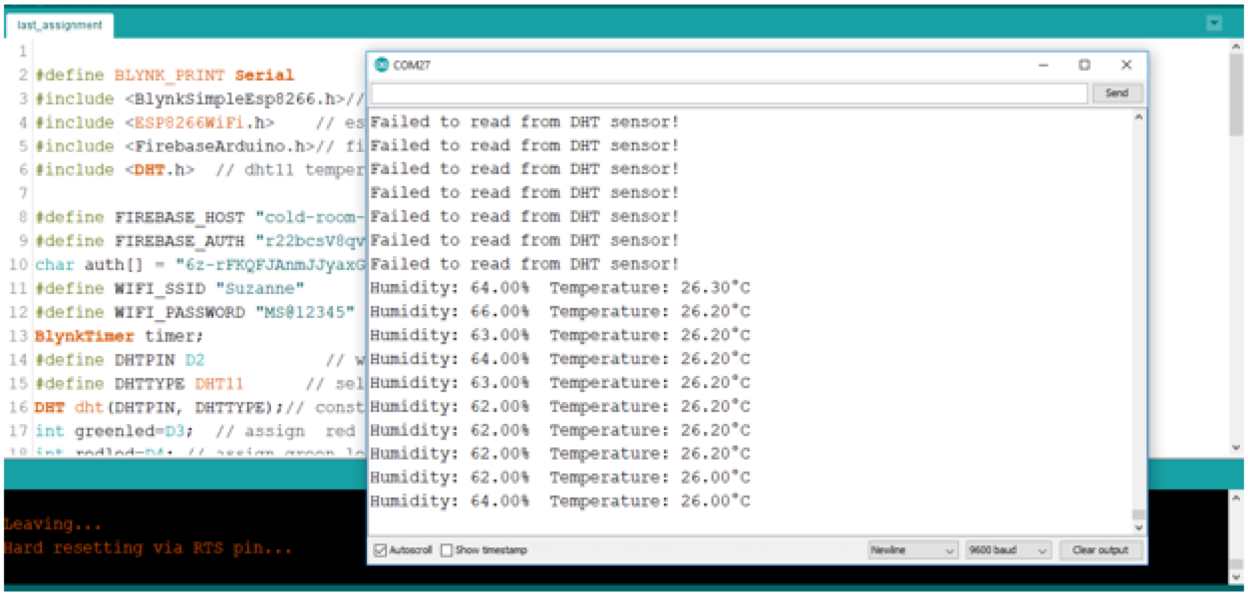
We needed to install **Arduinojson library** when we were installing the Firebase library because that error was shown while compiling.

6. Snapshots:

6.1. Connected circuits



6.2. Serial monitor photo

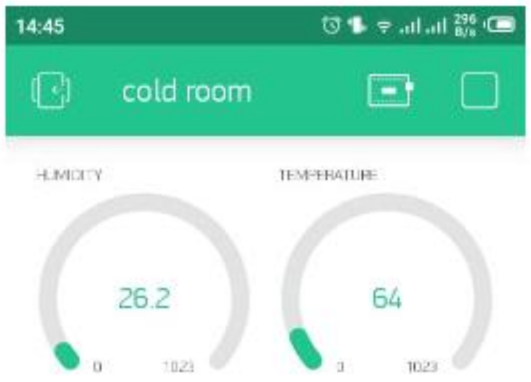


```

#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>//
#include <ESP8266WiFi.h>     // es
#include <FirebaseArduino.h>// fi
#include <DHT.h>  // dht11 temper

#define FIREBASE_HOST "cold-room-
#define FIREBASE_AUTH "r22bcsV8qv
char auth[] = "6z-rFKQFJAnmJJyaxG
#define WIFI_SSID "Suzanne"
#define WIFI_PASSWORD "MS@12345"
BlynkTimer timer;
#define DHTPIN D2              // w
#define DHTTYPE DHT11        // sel
DHT dht(DHTPIN, DHTTYPE);// const
int greenled=D3;  // assign  red
int redled=D4; // assign green le
```

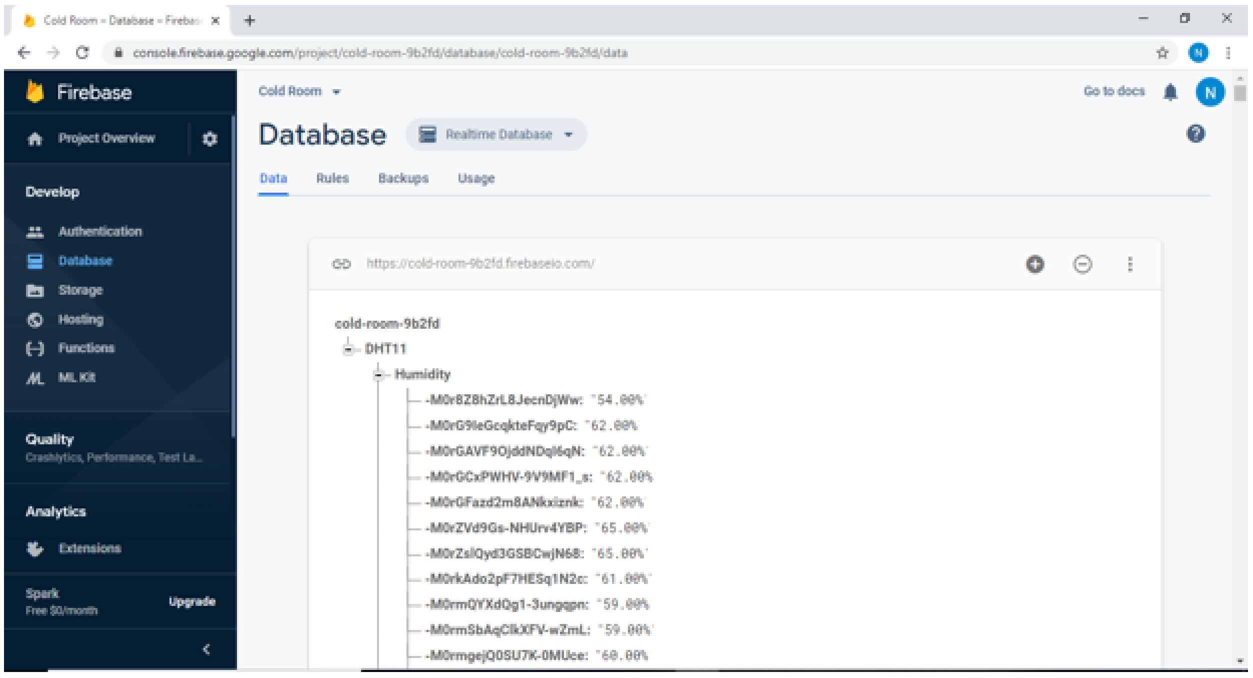
```
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Failed to read from DHT sensor!
Humidity: 64.00%  Temperature: 26.30°C
Humidity: 66.00%  Temperature: 26.20°C
Humidity: 63.00%  Temperature: 26.20°C
Humidity: 64.00%  Temperature: 26.20°C
Humidity: 63.00%  Temperature: 26.20°C
Humidity: 62.00%  Temperature: 26.20°C
Humidity: 62.00%  Temperature: 26.20°C
Humidity: 62.00%  Temperature: 26.20°C
Humidity: 62.00%  Temperature: 26.00°C
Humidity: 64.00%  Temperature: 26.00°C
```

```
Leaving...
Hard resetting via RTS pin...
```

6.3.Blynk snapshot



6.4.Firebase snapshot



```
https://cold-room-9b2fd.firebaseio.com/

cold-room-9b2fd
  DHT11
    Humidity
      -M0r8Z8hZrL8JecnDjWw: "54.00%"
      -M0rG9leGcqkteFqy9pC: "62.00%"
      -M0rGAVF9OjddNDql6qN: "62.00%"
      -M0rGCxPWHV-9V9MF1_s: "62.00%"
      -M0rGFazd2m8ANkxiznk: "62.00%"
      -M0rZVd9Gs-NHUrv4YBP: "65.00%"
      -M0rZslQyd3GSBCwjN68: "65.00%"
      -M0rkAdo2pF7HESq1N2c: "61.00%"
      -M0rmQYXdQg1-3ungqpn: "59.00%"
      -M0rmSbAqClkXFV-wZmL: "59.00%"
      -M0rmgejQ0SU7K-0MUce: "60.00%"
```

7. Final result

Due to the lack of components, we have used one DHT sensor and we are getting the notification for all temperatures which are above 25 degrees Celsius and then decisions can be taken accordingly.

This project can be used in Shopping moles, super markets agro-harvest shops where they need to conserve fresh items like vegetables, dairy products, meat. It is an efficient design in terms of power consumption since it turns the cooler ON and OFF according to the detected temperature.

8. References

[1] S. Conditions, “Bulletin # 4135 , Storage Conditions : Fruits and Vegetables.”

https://youtu.be/A6NkNL_YcDA

9. Team photos: