IOT based manual and automated monitoring and controlling for cold storage system

This project report is submitted to

Yeshwantrao Chavan College of Engineering
(An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

In partial fulfillment of the requirement
For the award of the degree

Of

Bachelor of Technology
in
Computer Science and Engineering

By

Harshal Borkar
Sankit Binkar
Aditya Pethe
Soniya Rangari

Under the guidance of

Prof. Jiwan N. Dehankar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Nagar Yuwak Shikshan Santa's

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING,
(An autonomous institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)
NAGPUR – 441 110
2023-24

CERTIFICATE OF APPROVAL

Certified that the project report entitled "**IOT based manual and automated monitoring and controlling for cold storage system**" has been successfully completed by **Harshal Borkar, Sankit Binkar, Aditya Pethe, Soniya Rangari** under the guidance of Prof. Jiwan N. Dehankar in recognition to the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, **Yeshwantrao Chavan College of Engineering(*An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University*)**

Signature	Signature	Signature	Signature
Prof. Jiwan Dehankar	Prof. Priyanka Pitale	Prof. Amruta Paul	Dr. Lalit Damahe
(Guide)	(Project Coordinator)	(Project Coordinator)	(Head of Department)

Signature of External

Examiner Name :

Date of Examination:

DECLARATION

We certify that

a. The work contained in this project has been done by me under the guidance of my supervisor(s).
b. The work has not been submitted to any other Institute for any degree or diploma.
c. We have followed the guidelines provided by the Institute in preparing the projectreport.
d. We have conformed to the norms and guidelines given in the Ethical Code of Conductof the Institute.
e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the reportand giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Name of the Student   Signature of Student

1.Harshal Borkar

2.Sankit Binkar

3.Aditya Pethe

4.Soniya Rangari

ACKNOWLEDGEMENT

We would like to thank our guide **Prof. Jiwan N. Dehankar** and industry mentor **Mr**. **Sandeep Sonaskar** and **Mr**. **Kunal Khawashi** for thorough guidance in the project. We are extremely gratefuland indebted to them for their expert, sincere, valuable guidance and encouragement which was of immense help to us.

We would like to express our sincere gratitude to **Dr. L B Damahe**, Head, Department of Computer Science & Engineering for his constant encouragement towards the successful completion of our work.

We wish to express our sincere thanks to **Dr U.P. Waghe**, the Principal of our college for providing us with all the necessary facilities and the infrastructure without which we would not have been able to complete our project successfully.

We would also like to thank our Project Coordinators **Prof. Priyanka Pitale** and **Prof. AmrutaT Paul** for their continuous guidance owing to which the project could take shape.

We would like to thank technical assistant for providing necessary technological support. Last, but not the least, we would like to thank all the faculty members and non-teaching staff members who helped us despite of their busy schedule.

Certificate of collaboration (industry/research organization)

Certificate of Completion

This is to certify that following students of final year Computer Science & Engineering Department, Yeshwantrao Chavan College of Engineering, Nagpur, have successfully completed Live/Industry/Joint research major project titled "**IoT based manual and automated monitoring and controlling for cold storage system**" under the guidance of **Prof Jiwan N. Dehankar** and Co-guide **Mr**. **Sandeep Sonaskar, Mr**. **Kunal Khawashi** with **V S Informatics Private Limited** for the session 2023-24.

Name of student 1: Harshal Borkar Enrolment No: 20010850
Name of student 2: Sankit Binkar Enrolment No: 20010825
Name of student 3: Aditya Pethe Enrolment No: 20010659
Name of student 4: Soniya Rangari Enrolment No: 20011028

Name and Signature of Industry Guide with Seal

Mr. Sandeep Sonaskar Mr. Kunal Khawashi

Table of Contents

List of Figures

Yeshwantrao Chavan College of Engineering
Department of Computer Science and Engineering
Project Cos

Upon completion of this course, the students will be able to:

CO-1: Investigate the Problem domain knowledge in their area of study and analyse the problem domain for executing the projects [PO-1, PO-2, PO-4] [PSO-1, PSO-2]

CO-2: Design and develop the solution using appropriate tools and techniques and communicate effectively with society at large for betterment. [PO-3, PO-5, PO-6, PO-7, PO-10] [PSO-1, PSO-2]

CO-3: Analyze and interpret the obtained results using acquired research-based knowledge. [PO-4, PO12] [PSO-1, PSO-2]

CO-4: Function effectively as an individual, and as a member or leader in a team under multidisciplinary settings following ethical practices. [PO-8, PO-9, PO-11, PO-12] [PSO-1, PSO-2]

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO-1: Investigate the Problem domain knowledge in their area of study and analyse the problem domain for executing the projects	3	3		3									3	3
CO-2: Design and develop the solution using appropriate tools and techniques and communicate effectively with society at large for betterment.			3		3	3	3			3			3	3

CO-3: Analyze and Interpret the obtained results using acquired research-based knowledge.				**3**								**3**	**3**	**3**
CO-4: Function effectively as an individual, and as a member or leader in a team under multidisciplinary settings following ethical practices								**3**	**3**		**3**	**3**	**3**	**3**

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Overview

These banana ripening chambers are safer than the use of calcium carbide to expedite the ripening process. Calcium carbide, which is generally used in welding applications, has been proven to be toxic and carcinogenic. It has been, therefore, banned by policymakers and its use is prohibited under section 44 A of the Prevention of Food Adulteration Act.

Ethylene gas on the other hand is a scientifically proven and globally accepted method of ripening. Usually, a fruit is exposed to a very small amount of ethylene to stimulate its ripening process until the fruit starts producing ethylene naturally to ripen in a controlled environment, producing the best result. Moreover, since ethylene is a natural hormone in plants, it does not pose any health hazard and is also a detraining agent which turns the peel from green to perfect yellow, maintaining the sweetness and aroma of the fruit.

❖ What Are the Components of a Banana Ripening Chamber?

A standard banana ripening chamber consists of the following components.

- Airtight insulated ripening chamber
- Refrigeration rack system with optional standby compressor
- Air-cooled or water-cooled condensers
- Powder-coated evaporator casing
- Electrical control panel
- Safety system for running equipment such as high or low-pressure switches, differential pressure switches, overload protection, and more
- Ethylene generators or gas-discharge systems suitable for ripening processes
- CO2 and ethylene analyzers for monitoring gas levels

❖ Banana Ripening Chamber Working Cycle

For bananas, a temperature range of 13 to 14°C and 90 to 95% RH (relative humidity) is considered ideal for storage and transport and a temperature of 15 to 20°C with 90 to 95% RH is ideal for ripening.

Usually, the banana ripening chamber is operated between four and six days for one cycle. The cycle may consist of the following steps.

1. For operations, ideal conditions are—18 to 21°C and >90% RH and pull down time of 16 to 20 hrs to 20°C.
2. Ethylene injection is given after 24 hours at a constant room temperature of 18 to 19°C and >90% RH. The amount of ethylene gas required for a ripening room is normally calculated based on the free air space after the bananas have been loaded. For instance, if bananas take up 35% of the room space, the amount of ethylene required is calculated based on the remaining 65% free air space.
3. Next is the holding period of 12 to 16 hours (a total of 24 hours from the start of ethylene injection) with temperature maintained at 18 to 19°C and >90% RH.
4. Ventilation is initiated after 24 hrs post ethylene injection. Ethylene and CO_2 are expelled out and fresh air is injected into the cold room. CO_2 levels should not exceed 5000 ppm during the ripening process.
5. This is followed by a holding period of three to four days until the required yellow colouration is achieved.
6. Temperature is then gradually reduced to 14 to 16°C for increasing the shelf-life of the fruit.
7. An air distribution system for uniform ripening of bananas is installed in palletized crates with a suitable air bypass sealing system.

To meet the varying needs of traders and farmers, these ripening chambers are available in different sizes and refrigeration capacities.

Fig 1.1: Banana Ripening Chamber

Hence, the implementation of an automatic system for measuring freezer temperature is significant in alleviating the workload of the crew when it comes to monitoring freezer temperature. The boosting demand for viable and efficient cold storage has led to the advancement of new techniques and technologies. Vintage refrigeration systems are known for their high energy consumption and substantial aid to the release of greenhouse gases. This work introduced a novel optimization tactic for cold storage, incorporating Internet of Things (IoT) technology and solar panels along with Image Processing machine learning based approach to minimize both energy expenses and the environmental influence of greenhouse gas emissions and classify ripe and unripe fruits/ vegetables. The proposed work comprises essential components, including a DHT11 temperature and humidity sensor, Relays, a gas sensor, and a microcontroller, responsible for data collection and refrigeration system.

1.2 Literature Survey

1. Maged Mohammed, Khaled Riad, Nashi Alqahtani

Title: "**Design of a Smart IoT-Based Control System for Remotely Managing Cold Storage Facilities**"

Source: Sensors, 2022, 22(13), 4680

Abstract: This paper proposes an IoT-based control system for managing cold storage facilities remotely. The system integrates sensors, actuators, and communication modules to monitor and regulate temperature, humidity, and other parameters. The authors discuss the architecture, implementation, and benefits of their solution.

2. Yanlin T, Zuoxin Hu, Tianyu T, Xin Gao

Title: "**Effect of Goods Stacking Mode on Temperature Field of Cold Storage**"

Source: Proceedings of EEEP 2020 IOP Conference Series: Earth and Environmental Science, Volume 675(1), 5th ICEEEP, IOP Publishing, Orlando, FL (2021)

Abstract: Investigating the impact of goods stacking patterns on temperature distribution within cold storage facilities, this study explores how different stacking modes affect energy efficiency and temperature uniformity. The findings contribute to optimizing storage layouts.

3. Hina Afreen, Imran Sarwar Bajwa

Title: "**An IoT-Based Real-Time Intelligent Monitoring and Notification System of Cold Storage**"

Source: IEEE Access, Volume 9, ISSN: 2169-3536 (2021)

Abstract: Focusing on real-time monitoring and alerts, this research presents an IoT-based system for cold storage facilities. The system continuously collects data from sensors, analyzesit, and notifies stakeholders of critical events. The authors emphasize its practical applicability.

4. Santoso Budijono, Felita

Title: "**Smart Temperature Monitoring System Using ESP32 and DS18B20**"

Source: 4th International Conference on Eco Engineering Development 2020, Banten, Indonesia, IOP Conf. Series: Earth and Environmental Science, Volume 794, 012125 (2021)

Abstract: This study introduces a low-cost temperature monitoring system based on ESP32 and DS18B20 sensors. The system provides real-time temperature data, enabling efficient cold storage management.The authors discuss the hardware setup, data transmission, and implications.

1.3 Problem Statement

1. The project is based upon the industry that includes the process of banana ripening which needs to be kept at frozen temperature.
2. There are 2 systems one is Ethylene gas valve and Air condition valve.
3. Both these valves are set at a particular time mode, currently it is done manually to set the timer so the gas valve and AC valve turn ON off at the set time.
4. In order to fix this problem, monitoring of temperature of cold storage unit and setting the timer online for both the valve can be done by IoT.
5. The aim is to design the secure website and control panel to access the data as well as to control the valve online and get alert on mobile app.

1.4 Thesis Objective

To create a sophisticated monitoring and control solution tailored for cold storage chambers, we aim to merge cutting-edge technologies. This fusion involves integrating a DHT11 sensor, renowned for its accurate temperature and humidity measurement capabilities, alongside an MQ-3 gas sensor for Ethylene gas detection. To deliver 12 instantaneous insights, an LCD is incorporated. The system's core function involves capturing sensor data, which is subsequently showcased on an LCD screen. Concurrently, this recorded dataset is transmitted via IoT technology to a dedicated website, enabling remote surveillance. Additionally, a potent IoT-linked notification system ensures real-time alerts. Further elevated this innovation, by introducing an advanced automated machine learning module which is meticulously crafted to impeccably differentiate between mature and immature fruits and vegetables, achieving exceptional classification precision. The ultimate goal is to streamline quality control and sorting procedures through heightened accuracy.

1.5 Thesis Contribution

Name of Student	**Contribution**
Harshal Borkar	Integration of hardware and software, NodeMcu programming, Backend and Alert system unit
Sankit Binkar	Integration of hardware components, Frontend, Poster design and documentation
Aditya Pethe	Integration coding of IoT Components, PPT and Report writing
Soniya Rangari	Frontend

CHAPTER 2
REVIEW OF LITERATURE

CHAPTER 2
REVIEW OF LITERATURE

2.1 Literature Review

In 2022, authors of Development of Fruit Cold Storage Monitoring Controller using IoT research suggestan IoT-based monitoring system that displays the maintenance status in cold chain plants. ESP microcontroller is used to collect environmental data from DHT 11 sensors, such as temperature and humidity, and cloud server module is used to deliver communication data [1].

In crucial situations, early warning alerts and notifications are the methodology suggested by the paper's authors. It allows for end-to-end responsibility and visibility along the full product value chain. To get to their destination, perishable commodities frequently need to travel thousands of kilometers via land, air, and marine transportation methods [2].

This study introduces a low-cost, autonomous cold storage monitoring and regulating system based on theInternet of Things. The suggested system consists of a sensor that can measure both temperature and humidity, a microcontroller, a power supply module based on a DC-DC step down converter, a cooling fan to lower the temperature, and an app to monitor and regulate the cold storage system's temperature [3].

The authors of Solar-Powered Cold Storage System want to reduce energy usage in Gaza, which experiences power shortages, by designing and developing a solar powered cold storage system. With three evaporators operating at various temperatures and a single compressor, the system operates in a vaporcompression refrigeration cycle (VCR) [4].

The article Technologies Applied in the Monitoring and Control of The Temperature in the Cold Chain proposes is an analysis of the technologies used to track and regulate the temperature of perishable goodsthroughout the cold chain. Additionally, a basic description of the stages of the process is made, starting with production, and ending with final consumer, transit, storage, and distribution. The goal of this work is to discover the current technological trends that can be fully applied in a cold chain to save money, a lotof time, and lessen food waste in businesses [5].

In order to prevent food goods from rotting due to an increase in temperature and humidity, the author ofthis research suggests adopting Wireless Sensor Networks (WSN) based continuous monitoring of temperature and humidity of the cold storage warehouses [6].

Chapter No.2: Literature Review 6 A design based on ZigBee wireless sensor network (WSN) features

and the actual scenario of cold storage, which includes ZigBee network hardware and software architecture. The ZigBee WSN routing protocol Cluster Tree is upgraded; neighbor table information is integrated into the Cluster Tree algorithm, and the selection of nodes in the neighboring table is explored;this is advantageous in finding an ideal path and reducing data transmission delay [7].

India is the world's largest grower of fruit and the world's second greatest producer of veggies. Production is gradually increasing as a result of the different agro-climatic conditions and increased availability of package methods. Despite the fact that there is a large potential for growing production, the absence of suitable cold storage and cold chain facilities is becoming a key impediment in realizing the promise. Thiscan be avoided by using a wireless sensor network-based cold storage system [8].

The cold chain temperature monitoring system architecture, software design, system flow method, and other technical benefits are all fully utilized by this application, together with RFID tags, temperature sensors, GPS systems, and other advantages. To ensure the quality of cold chain products across the supplychain, the system can track the location and temperature of those products in real-time [9].

The main causes of this are difficulties with post-production and the ineffectiveness of cold storage facilities. Potatoes are the fourth-largest crop produced in India [10].

2.2 Preliminary Investigation Report

Title of the Project:

IOT based manual and automated monitoring and controlling for cold storage system

Area of Project Work:

IOT, Cloud and Web Development

Problem Statement:

To monitor and control the environment of the cold storage system for maintaining quality of item and increase the life span.

Prior Art (Patent Search):

Patent Application No.	Title of Patent	Existing Solutions (Abstract of Patent)
US1515546A	COLD-STORAGE SYSTEM	Refrigeration or cold storage of perishable food products of various kinds which require the presence of the proper amount of air, the proper relative humidity, and correct uniform temperature, for their preservation
US4989417A	Cold-storage warehouse	The cold storage warehouse design features an inner grouping of cells for below-zero temperature storage, surrounded by an outer ring of cells for above-freezing temperature storage. Temperature control is achieved through a system of finned ducts suspended beneath the ceiling, with reversible ceiling fans aiding in air circulation

Literature Review:

Title of Paper	Details of Publication with Date and Year	Literature Identified for Project
Maged Mohammed, Khaled Riad, Nashi Alqahtani, “Design of a Smart IoT-Based Control System for Remotely Managing Cold Storage Facilities”. Sensors 2022, 22(13), 4680 (2022).	2022	• Displays the maintenance status in cold chain plants. • Collect environmental data such as temperature and humidity. • Cloud server module is used to deliver communication data
Yanlin T, Zuoxin Hu, Tianyu T, Xin Gao, “Effect of goods stacking mode on temperature field of cold storage”. Proceedings of EEEP 2020 IOP Conference Series: Earth and Environmental Science. Volume 675(1): 012052, 5th ICEEEP, IOP Publishing. Orlando, FL (2021).	2021	• Adopt WSN based continuous monitoring of temperature and humidity of the cold storage warehouses

Hina Afreen, Imran Sarwar Bajwa, “An IoT-Based Real-Time Intelligent Monitoring and Notification System of Cold Storage”. IEEE Access PP(99): 1-1. (Volume 9), ISSN: 2169-3536, (2021).	2021	• Use of API for alert system

Current Limitation:

There is no cold storage which can be controlled remotely through web application , also all systems are to be monitored and controlled manually by the user.

Proposed Solution:

We have developed IoT based manual and automated monitoring and controlling for cold storage system. The integration of Master NodeMCU, Slave controller, sensors, and relays ensures accurate monitoring and control of critical environmental parameters such as temperature, humidity, and gas levels. The successful operation of the Air Conditioner and Ethylene gas cylinder demonstrates the system's ability to maintain optimal storage conditions and prevent premature ripening of bananas. The user-friendly web-based Control Panel and LCD display enhance accessibility and provide a convenient interface for users to monitor and manage the storage environment. If the user is not available and there is any change in the system then the software is capable of handling it and make appropriate changes.

Objectives and Scope of Work:

- To monitor and control the cold storage environment remotely through web application.
- To maintain quality and increase life span of food by controlling the growth of bacteria & micro-organisms.
- To develop alert system and automated control for client.
- To maximize profit

Feasibility Assessment:

I. Expected Outcomes of the Project

- Monitor and control the cold storage environment remotely through web application.
- Maintain quality and increase life span of food by controlling the growth of bacteria & micro-organisms.
- Send alert system and automatically controls system for client.
- Maximized profit

II. Innovation Potential

- Reduce delay between hardware and software.
- Reduce latency between IOT components.
- Increase efficiency of software.

III. Task Involved

- Working on software efficiency.
- Switching on private database and premium software.

IV. Expertise Required

- Full Stack Web Developer.
- IOT expert

V. Facilities Required

- IOT components if required.
- Premium softwares for hosting and IOT.

CHAPTER 3
WORK DONE

CHAPTER 3
WORK DONE

Proposed Methodology
3.1 Overview

The project's operation hinges on IoT capabilities. To materialize this idea, two vital components are required: a sensor in the environment for data collection and a cloud service for data storage. Our cold storage monitoring system resorts to a network of interconnected sensors that gather and upload data to the cloud, ensuring remote accessibility. The monitoring system comprises two subsystems: one for overseeing the cold storage conditions within facilities and another for monitoring logistics processes. These subsystems are based on the Node-MCU prototyping platform, each serving dual purposes:measuring environmental temperature and humidity, and trailing item movements. In the cloud, facility data abides by specific routes to reach three monitoring personnel. Temperature data is admittable via a web portal, while inventory details are retained. The web server, equipped with a user-friendly interface, permits real-time appliance monitoring from anywhere. To achieve this, the system relies on the Node- MCU ESP controller, an open-source platform with user-friendly hardware and software, and an integratedWi-Fi module, ESP8266. Programming is done using the Arduino language within the Arduino Integrated Development Environment. It utilizes the DHT11 digital temperature sensor, programmable to interact with the controller's digital pins.

3.2 Method

The system uses Master NodeMcu to send data to cloud server. On the web-based Control Panel fields are there which display recorded values of Temperature, Humidity and Gas Sensor. DHT11 Temperature and MQ-3 gas sensor are connected to the Slave controller. These sensors are used to measure and monitor temperature, humidity and gas values present in the cold storage chamber, respectively. Relay 1 is connected to slave ESP controller to turn Air Conditioner ON and OFF. LCD display of 16*2 is used to display temperature and gas values sensed by sensors. Relay 2 is connected to master ESP controller for operation of Ethylene gas cylinder.

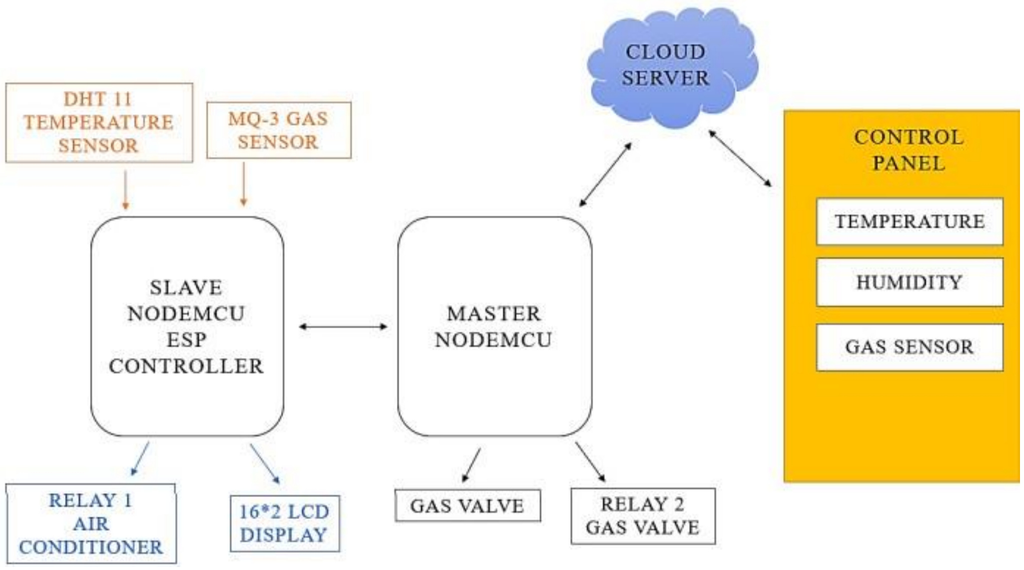


Fig 3.1: Block diagram of proposed system

3.3 Tools

- **ESP8266 Microcontroller:** The ESP8266 module enables microcontrollers to connect to 2.4 GHz Wi-Fi, using IEEE 802.11 bgn. It can be used with ESP-AT firmware to provide Wi-Fi connectivity to external host MCUs, or it can be used as a self-sufficient MCU by running an RTOS-based SDK.

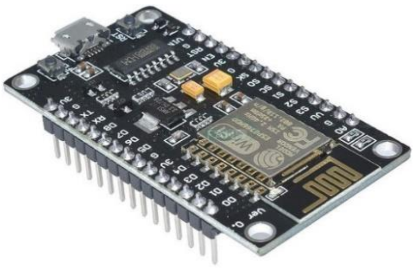


Fig 3.2: ESP8266 Microcontroller

- **DHT11:** The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed.

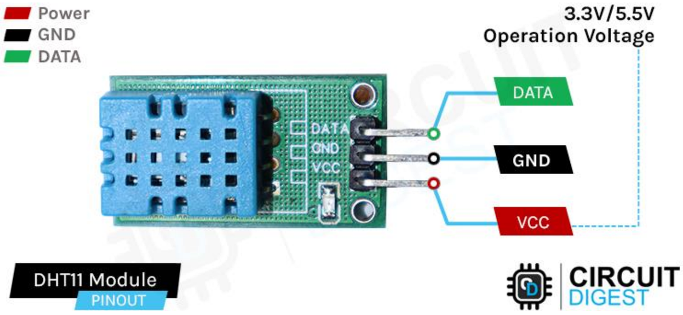


Fig 3.3: DHT11

- **MQ-3:** The Grove - Gas Sensor (MQ3) module is useful for gas leakage detection (in home and industry). It is suitable for detecting Alcohol, Benzine, CH4, Hexane, LPG, CO. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible.

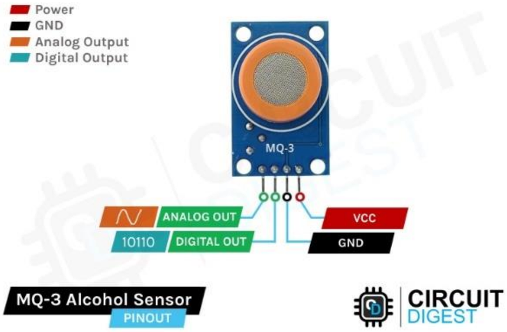


Fig 3.4: MQ-3

- **Relay Switch:** Relays are electrically operated switches that open and close the circuits by receiving electrical signals from outside sources. They receive an electrical signal and send the signal to other equipment by turning the switch on and off.

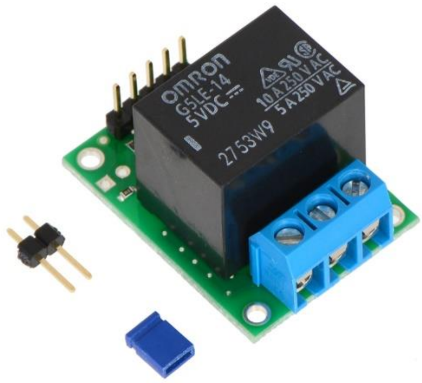


Fig 3.5: Relay Switch

➢ **LCD Display:** LCD screens can be used to display real-time data from IoT sensors or to provide a user interface for controlling IoT devices.

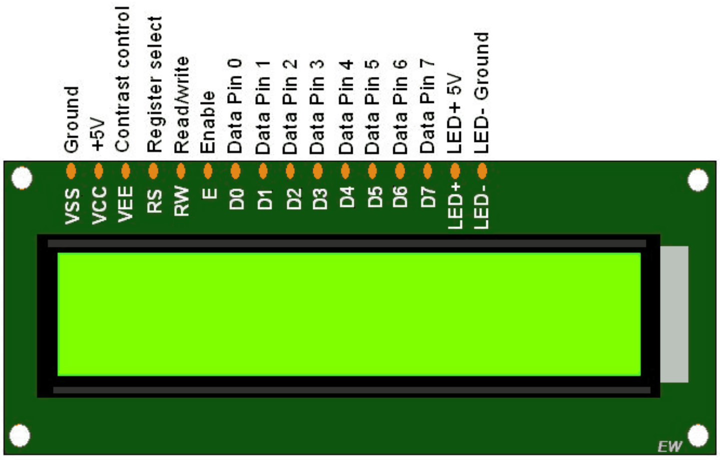


Fig 3.6: LCD Display

➢ **Gas Emitter:** This is a solenoid valve gas emitter which is used to emit gas in the cold storage chamber. It can contain various types of gases which can emitted manually/automatically from the web application by the user.

Fig 3.7: Gas Emitter

➢ **Exhaust Fan:** This is a 12V fan. It can be used to maintain gas and humidity in the chamber.

Fig 3.8: Exhaust Fan

- **Air Conditioner:** It is used to maintain the temperature of Cold Storage system.
- **SMPS:** SMPS is **Switched Mode Power Supply** also known as **Switching Mode Power Supply**. SMPS is an electronic power supply system that makes use of a switching regulator to transfer electrical power effectively.

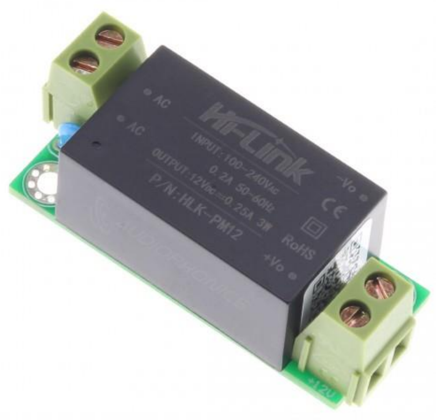


Fig 3.9: SMPS

- **Battery:** It is used to power the sensors, IOT components and other electrical components. We have used two 12V batteries.
- **MySQL Data Base:** MySQL is a valid choice in database platforms when you are designing a system that interacts with the IoT. The point of the Internet of Things is to gather or exchange data to facilitate physical processes or gain insight into complex abstract or tangible systems.
- **Arduino IDE Software:** The Arduino IDE (Integrated Development Environment) is used to write the computer code and upload this code to the physical board.

3.4 PLAN OF ACTION

3.4.1 Testing of Gas Sensor MQ3

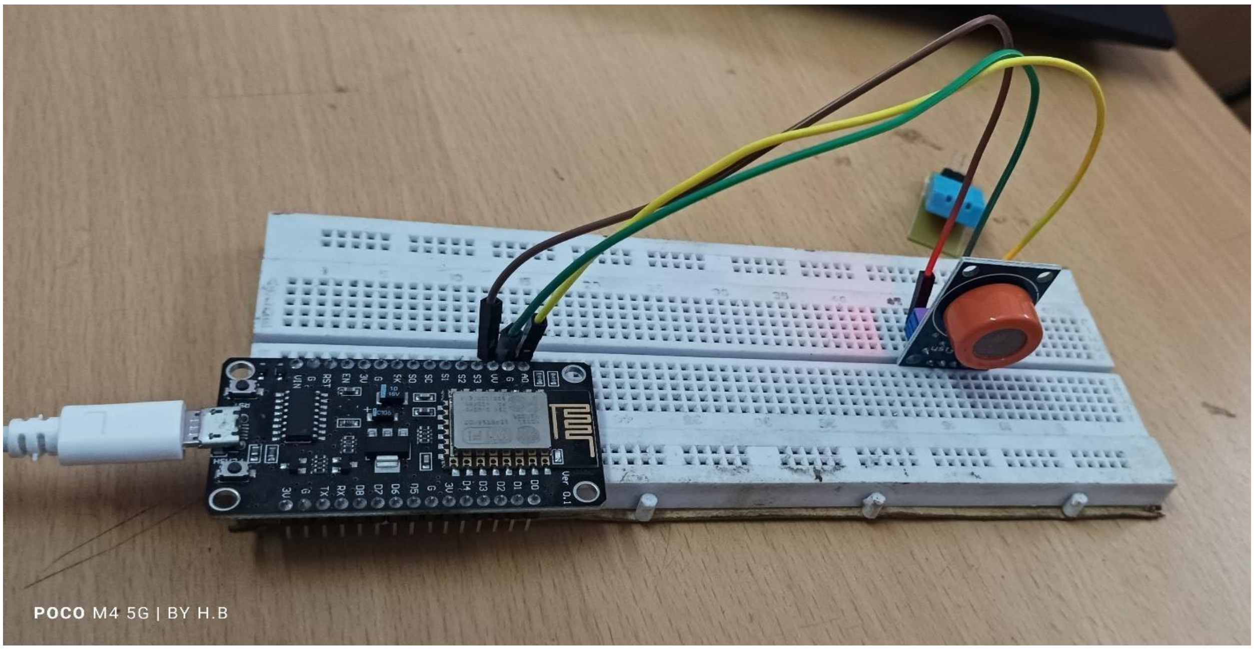


Fig.3.10 Testing of Gas Sensor MQ3

Testing the MQ-3 gas sensor involves connecting it to a microcontroller, calibrating it with clean air to establish a baseline reading, exposing it to alcohol vapor at varying concentrations, recording its responses for analysis, and validating its accuracy against known standards. Documentation of the testing procedures, results, and any observations made during testing is essential for future reference and troubleshooting. Additionally, iterative testing may be necessary to fine-tune the sensor's performance or ensure long-term stability in its intended application.

3.4.2 Testing of Display Unit – LCD

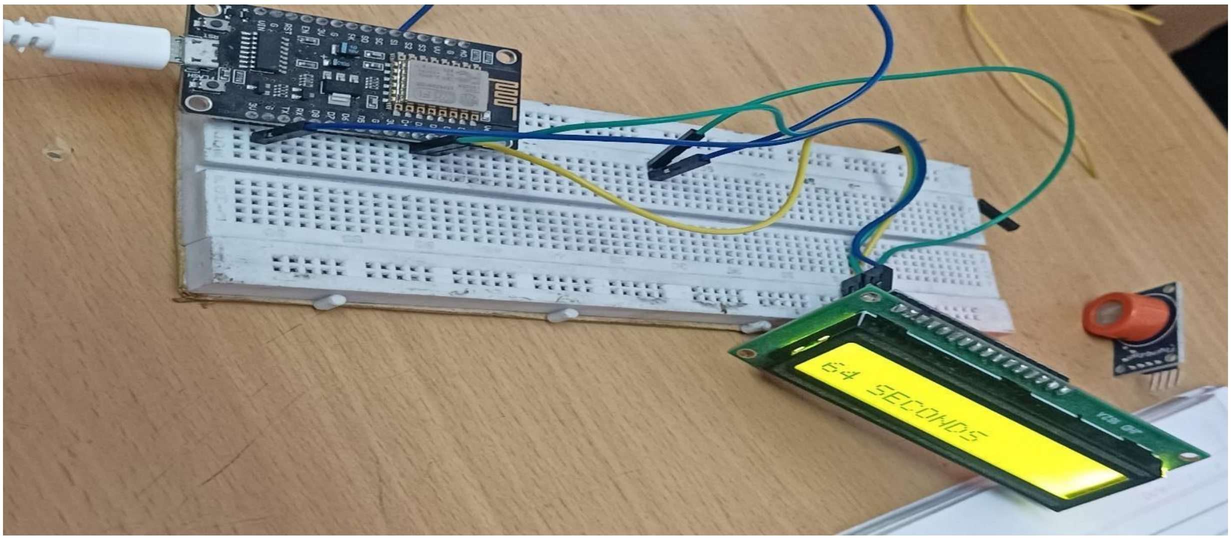


Fig.3.11 Testing of Display Unit – LCD

Testing the LCD display unit involves several steps to ensure its functionality and reliability. Begin by connecting the LCD to a compatible microcontroller and verifying proper electrical connections. Then, execute a test script to display predefined patterns or messages on the LCD screen. Validate the readability, contrast, and alignment of characters or graphics. Next, assess the display's response to various environmental conditions, such as temperature and humidity, to ensure performance under different circumstances. Finally, conduct a longevity test to evaluate the display's durability over time, simulating continuous usage. Document all testing procedures and results for future reference and quality assurance purposes.

3.4.3 Testing of Humidity and Temperature sensor- DHT11

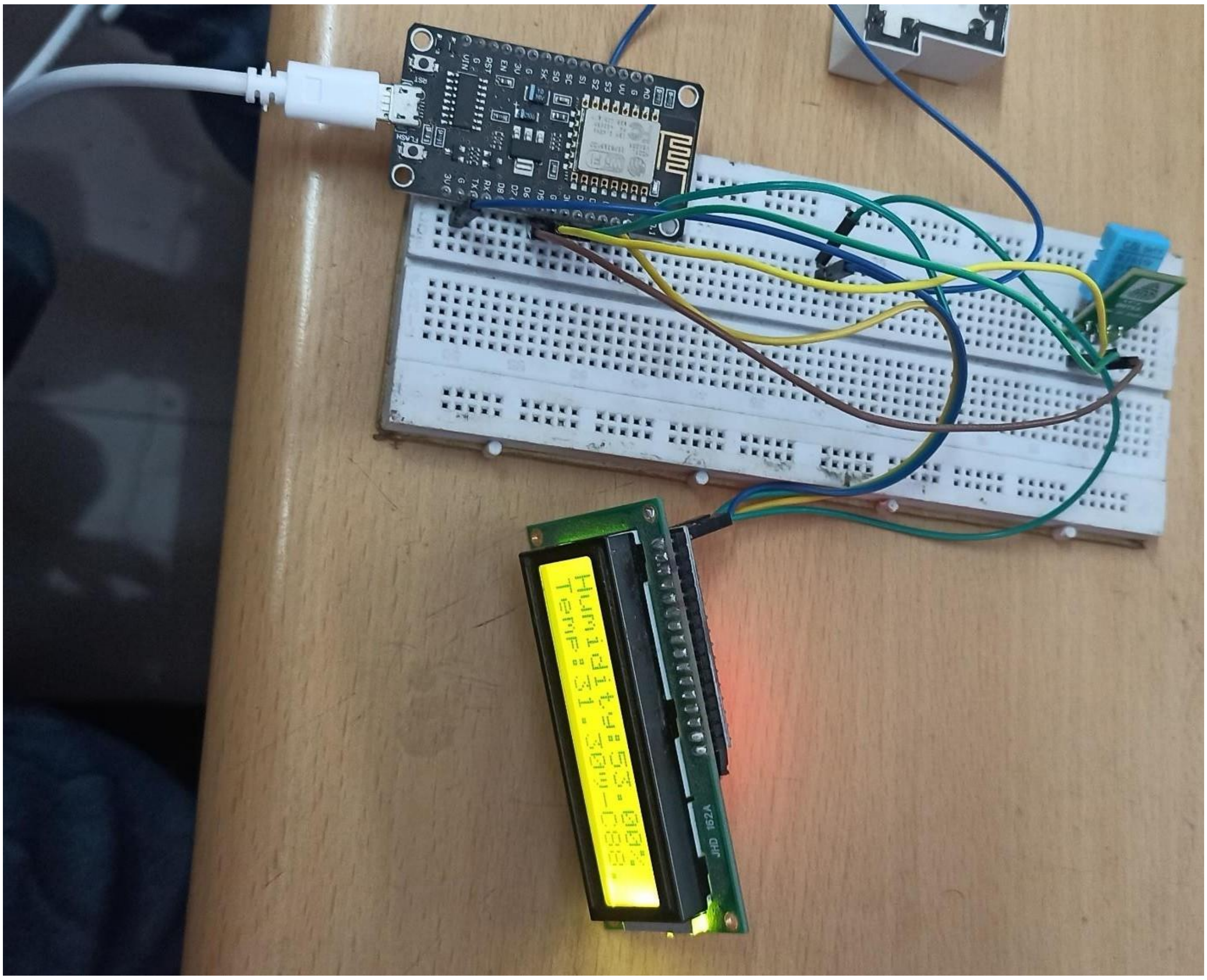


Fig.3.12 Testing of Humidity and Temperature sensor-
DHT11

Testing the DHT11 humidity and temperature sensor encompasses several key steps to ensure accurate and reliable performance. Begin by connecting the sensor to a microcontroller and verifying the wiring according to the sensor's datasheet. Execute a test script to read data from the sensor and validate the humidity and temperature readings. Compare the sensor's measurements with a known reference, such as a calibrated instrument, to assess accuracy. Additionally, subject the sensor to different environmental conditions to evaluate its response and reliability under various scenarios. Document the testing procedures and results for future reference and quality control purposes.

3.4.4 Integration of Control Unit

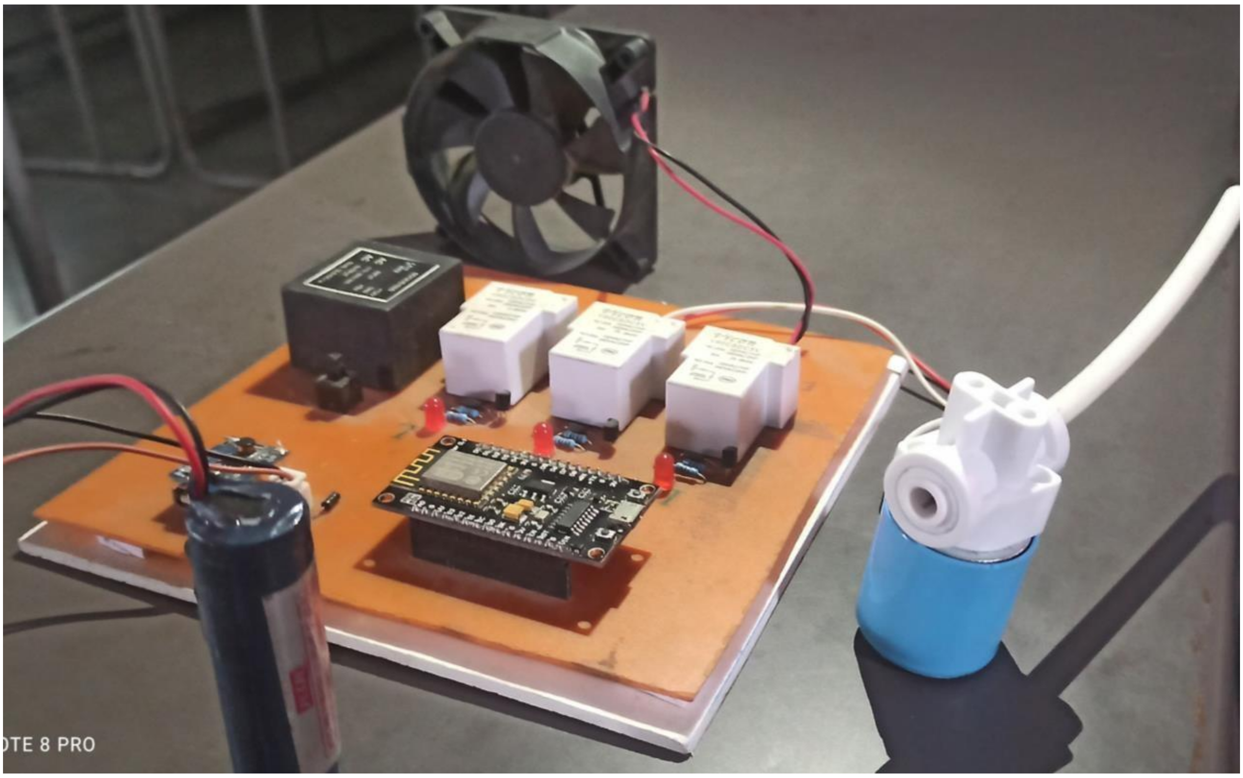


Fig.3.13 Integration of Control Unit

In this Fig 3.5, there is a NodeMcu which is connected to Wi-Fi also there are three relays to control air conditioner, fan and gas emitter and a SMPS (Switched Mode Power Supply) to convert AC to DC. The microcontroller gets data from Thinkspeak (which is taken from the database) and accordingly switch on/off the three respective components. This can be regulated manually by the user from the web application or automatically by the software if it remains idle for a specific amount of time.

Circuit Diagram of Control unit:

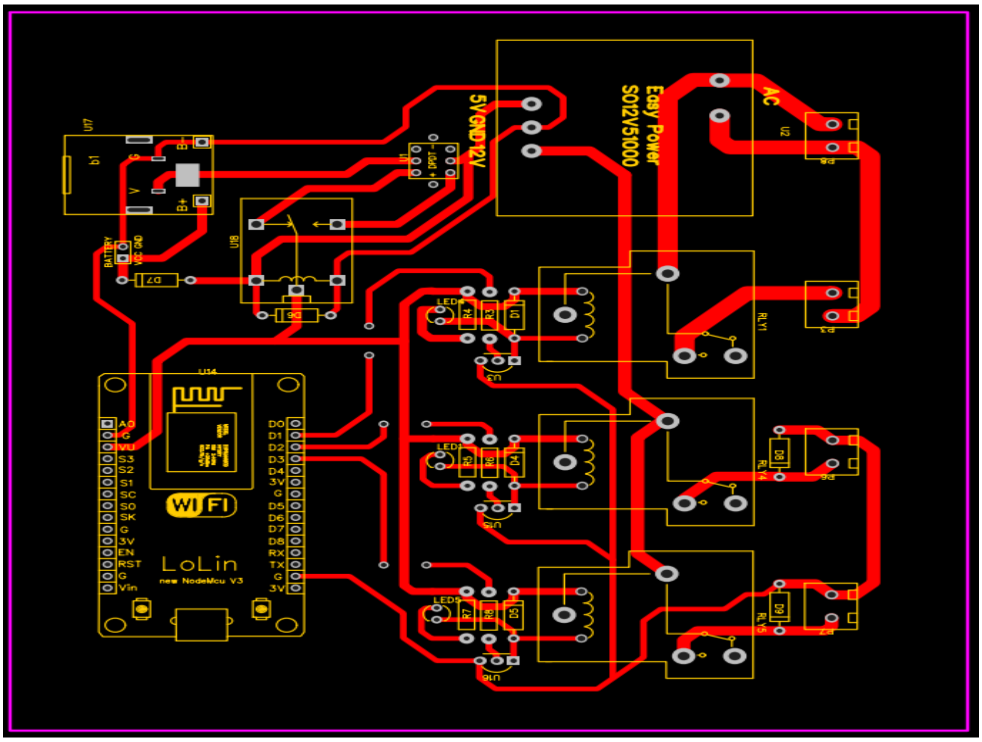


Fig. 3.14 Connection Diagram of Monitoring Unit

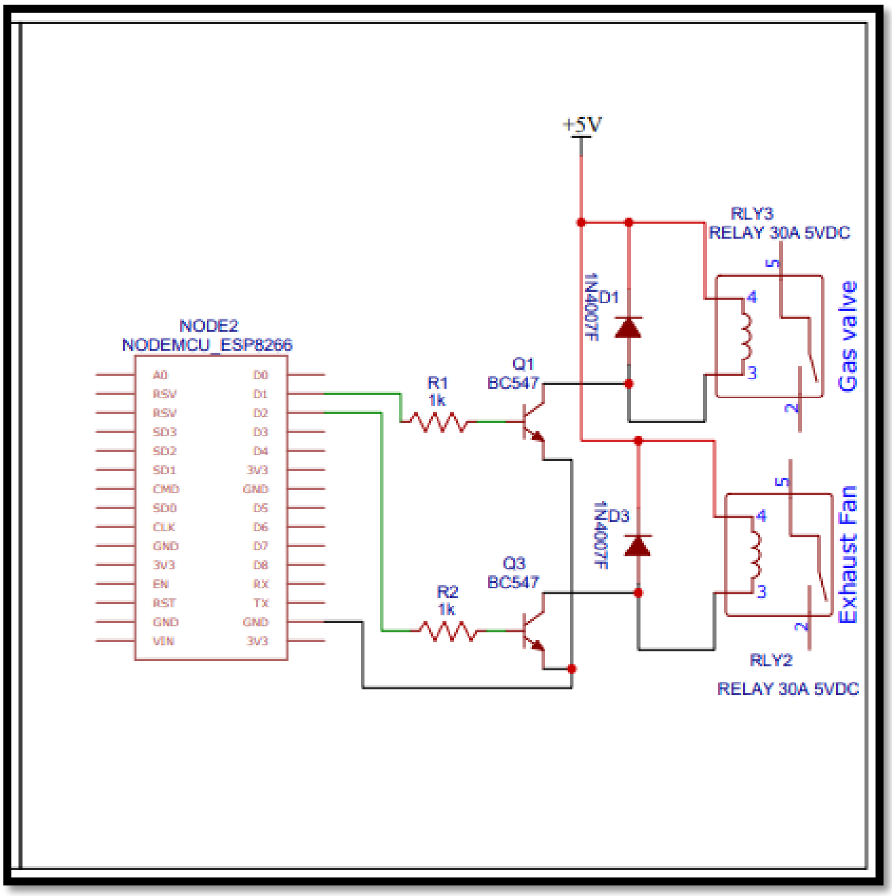


Fig. 3.15 Circuit Diagram of Control Unit

- **Control Unit Coding:**

```
// Adding libraries
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

// Setting wifi password and name
const char* ssid = "POCO M4 5G";
const char* password = "@np@s.....@";

// Creating wifi and http client object
WiFiClient  client;
HTTPClient http;

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

// Declaring variables
int statusCode = 0,statusCode1 = 0,statusCode2 = 0;
```

```
int device_status1,device_status2,device_status3;
int httpCode1,httpCode2,httpCode3;
int device1_pin=D1,device2_pin=D2,device3_pin=D5;

String payload,payload2,payload3;

long
currentMillis=0,currTime=0,onTime1=0,onTime2=0,onTime3=0,offTime1=0,offTime2=0,offTime
3=0;

int r1=0,r1on=0,r1off=0,dev1on,dev1off,offTimeStart1=0;
int r2=0,r2on=0,r2off=0,dev2on,dev2off,offTimeStart2=0;
int r3=0,r3on=0,r3off=0,dev3on,dev3off,offTimeStart3=0;


// Setting Thingspeak details
unsigned long counterChannelNumber = 2473069;          // Channel ID
const char * myCounterReadAPIKey = "FW6PX9J38PRLL7JP"; // Read API Key

void getTime(){
  timeClient.update();
  String formattedTime = timeClient.getFormattedTime();
  Serial.print(formattedTime);
}

void setup(){
  Serial.begin(9600);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  pinMode(device1_pin,OUTPUT);
  digitalWrite(device1_pin,LOW);
  pinMode(device2_pin,OUTPUT);
  digitalWrite(device2_pin,LOW);
  pinMode(device3_pin,OUTPUT);
  digitalWrite(device3_pin,LOW);
  if (WiFi.status() != WL_CONNECTED){
    Serial.println("Connecting to " + String(ssid));
    WiFi.begin(ssid, password);
    while(WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println();
    Serial.println("Connected to Wi-Fi Succesfully.");
  }
  // Initialize a NTPClient to get time
  timeClient.begin();
  timeClient.setTimeOffset(19800);
}

void loop(){
```

```
  Serial.println();
  // currTime=millis();
  Serial.println(millis());

//DEVICE 1
  httpCode1=0;
  http.begin(client,"http://coldstoragemonitor.000webhostapp.com/devicestatus.php?id=1");
  httpCode1 = http.GET();
  if(httpCode1 > 0) {
    payload = http.getString();
    device_status1 = payload.toInt();
  }

  if(device_status1 == 0){
    Serial.println("Device 1 AC Timer ON");

      //---------------- Device 1 ON TIME ----------------//
    statusCode=0;

    do{
      dev1on = ThingSpeak.readLongField(counterChannelNumber, 1, myCounterReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);
    dev1on=dev1on*1000;

    Serial.println("Dev1 on time: "+ String(dev1on));

    //---------------- Device 1 OFF TIME ----------------//
    statusCode =0;
    do{
      dev1off = ThingSpeak.readLongField(counterChannelNumber, 2, myCounterReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);
    dev1off=dev1off*1000;
    Serial.println("Dev1 off time: "+ String(dev1off));

    if(dev1on>0 && r1==0 && offTimeStart1==0){
      onTime1 = millis();
      Serial.println("DEV 1 ONTIME "+ String(onTime1));
      r1=1;
      offTime1=0;
    }

    if(r1==1){
      digitalWrite(device1_pin,HIGH);
      Serial.println("ON");
    }

    if(currTime-onTime1>=dev1on && r1==1 && onTime1>0){
```

```
      offTime1 = millis();
      Serial.println("DEV 1 OFFTIME "+ String(offTime1));
      r1=0;
      offTimeStart1=1;
    }

    if(r1==0){
      digitalWrite(device1_pin,LOW);
      Serial.println("OFF");
    }

    if(currTime-offTime1>=dev1off && offTime1>0){
      r1=0;
      offTimeStart1=0;
      Serial.println("CYCLE FOR DEV 1 OVER");
    }

  }
  else{
    digitalWrite(device1_pin,HIGH);
    r1=0;
    onTime1=0;
    offTime1=0;
    offTimeStart1=0;
    Serial.println("Device 1 AC Timer OFF");
  }



//DEVICE 2
  httpCode2=0;
  http.begin(client,"http://coldstoragemonitor.000webhostapp.com/devicestatus.php?id=2");
  httpCode2 = http.GET();
  if (httpCode2 > 0) {
    payload = http.getString();
    device_status2 = payload.toInt();
  }

  if(device_status2 == 0){
    Serial.println("Device 2 GAS TIMER ON");

      //---------------- Device 2 ON TIME ----------------//
    statusCode1=0;
    do{
      dev2on = ThingSpeak.readLongField(counterChannelNumber, 3, myCounterReadAPIKey);
      statusCode1 = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode1!=200);
    dev2on=dev2on*1000;
    Serial.println("dev2 on time: "+String(dev2on));
```

```
  //---------------- Device 2 OFF TIME ----------------//
  statusCode1=0;
  do{
    dev2off = ThingSpeak.readLongField(counterChannelNumber, 4, myCounterReadAPIKey);
    statusCode1 = ThingSpeak.getLastReadStatus();
    delay(200);
  }while(statusCode1!=200);
  dev2off=dev2off*1000;
  Serial.println("dev2 off time: "+String(dev2off));

  if(dev2on>0 && r2==0 && offTimeStart2==0){
    onTime2 = millis();
    Serial.println("DEV 2 ONTIME "+ String(onTime2));
    r2=1;
    offTime2=0;
  }

  if(r2==1){
    digitalWrite(device2_pin,HIGH);
    Serial.println("ON");
  }

  if(currTime-onTime2>=dev2on && r2==1 && onTime2>0){
    offTime2 = millis();
    Serial.println("DEV 2 OFFTIME "+ String(offTime2));
    r2=0;
    offTimeStart2=1;
  }
  if(r2==0){
    digitalWrite(device2_pin,LOW);
    Serial.println("OFF");
  }

  if(currTime-offTime2>=dev2off && offTime2>0){
    r2=0;
    offTimeStart2=0;
    Serial.println("CYCLE FOR DEV 2 OVER");
  }
 }
 else{
  digitalWrite(device2_pin,HIGH);
  r2=0;
  onTime2=0;
  offTime2=0;
  offTimeStart2=0;
  Serial.println("Device 2 GAS TIMER OFF");
 }

//DEVICE 3
 httpCode3=0;
 http.begin(client,"http://coldstoragemonitor.000webhostapp.com/devicestatus.php?id=3");
```

```
httpCode3 = http.GET();
if (httpCode3 > 0) {
  payload = http.getString();
  device_status3 = payload.toInt();
}

if(device_status3 == 0){
  Serial.println("Device 3 FAN TIMER ON");

  //---------------- Device 3 ON TIME ----------------//
  statusCode2=0;
  do{
    dev3on = ThingSpeak.readLongField(counterChannelNumber, 5, myCounterReadAPIKey);
    statusCode2 = ThingSpeak.getLastReadStatus();
    delay(200);
  }while(statusCode2!=200);
  dev3on=dev3on*1000;
  Serial.println("dev3 on: "+String(dev3on));

  //---------------- Device 3 OFF TIME ----------------//
  statusCode2=0;
  do{
    dev3off = ThingSpeak.readLongField(counterChannelNumber, 6, myCounterReadAPIKey);
    statusCode2 = ThingSpeak.getLastReadStatus();
    delay(200);
  } while(statusCode2!=200);
  dev3off=dev3off*1000;
  Serial.println("dev3 off: "+String(dev3off));

  if(dev3on>0 && r3==0 && offTimeStart3==0){
    onTime3 = millis();
    Serial.println("DEV 3 ONTIME "+ String(onTime3));
    r3=1;
    offTime3=0;
  }

  if(r3==1){
    digitalWrite(device3_pin,HIGH);
    Serial.println("ON");
  }

  if(currTime-onTime3>=dev3on && r3==1 && onTime3>0){
    offTime3 = millis();
    Serial.println("DEV 3 OFFTIME "+ String(offTime3));
    r3=0;
    offTimeStart3=1;
  }
  if(r3==0){
    digitalWrite(device3_pin,LOW);
    Serial.println("OFF");
  }
```

```
    if(currTime-offTime3>=dev3off && offTime3>0){
      r3=0;
      offTimeStart3=0;
      Serial.println("CYCLE FOR DEV 3 OVER");
    }
  }
  else{
    digitalWrite(device3_pin,HIGH);
    r3=0;
    onTime3=0;
    offTime3=0;
    offTimeStart3=0;
    Serial.println("Device 3 FAN ON");
  }
  http.end();
  currTime=millis();
  Serial.println();
  Serial.println();

}
```

3.4.5 Integration of Monitoring Unit

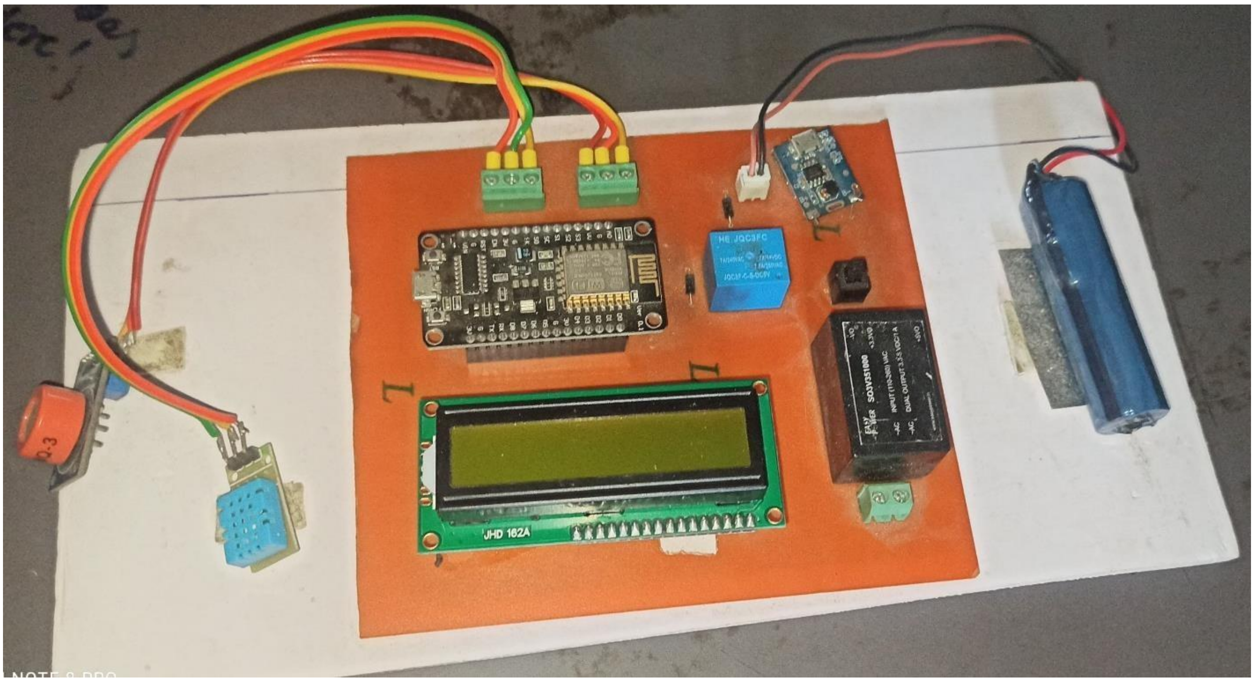


Fig. 3.16 Integration of Monitoring Unit

There are two sensors in Fig 3.8, DHT11 for measuring temperature and MQ3 for gas and humidity. There is one display which shows all the real time values collected from the sensors. These sensors are also connected to the NodeMcu which is connected to the Wi-Fi and sends all the values to the database from where it is also displayed on the website. There is a battery to power the sensors and display and a unit for it’s charging.

Circuit Diagram of Monitoring Unit:

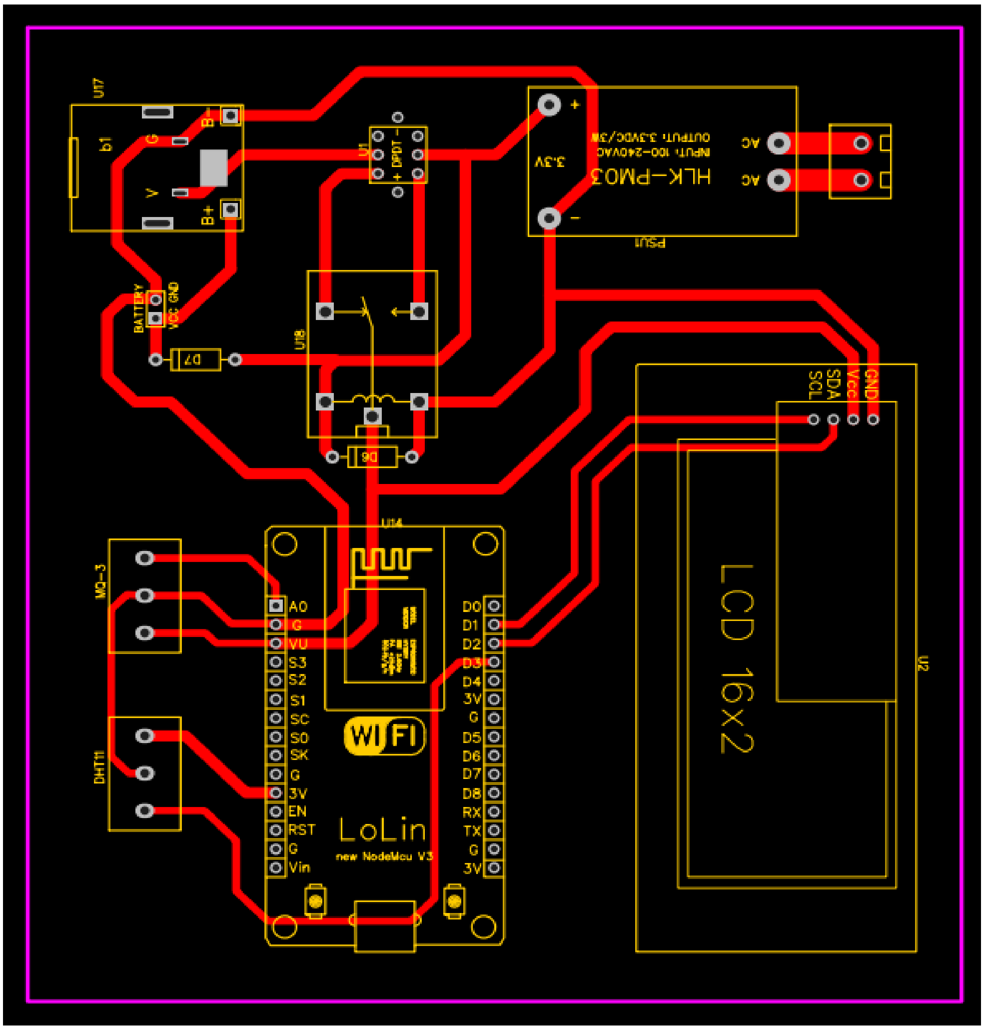


Fig. 3.17 Circuit Diagram of Monitoring Unit

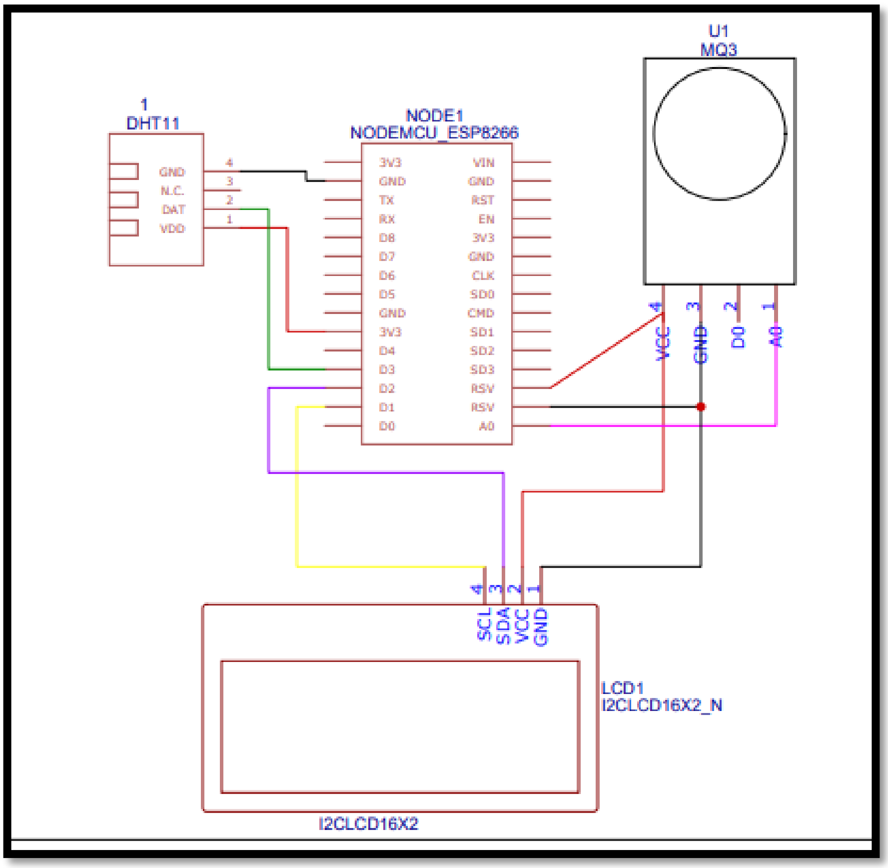


Fig. 3.18 Circuit Diagram of Monitoring Unit

❖ Monitoring Unit Coding:

```
// Including Libraries
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <WiFiClientSecure.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <UrlEncode.h>
#include "ThingSpeak.h"
#include <NTPClient.h>
#include <WiFiUdp.h>

// Define wifi & http client objects
WiFiClient client;
HTTPClient http;

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
```

```
// Setting wifi name and password
const char* ssid = "POCO M4 5G";
const char* password = "@np@s.....@";

// Defining pins for sensors
#define DHTPIN D3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Setting api keys for thingspeak
unsigned long myChannelNumber = 2473069;
const char * myWriteAPIKey = "16BD2RV42BA6HC3R";
const char * myReadAPIKey = "FW6PX9J38PRLL7JP";
String myStatus = "";

// Defining variables
float MQ3pin = A0,t,h,g,z;
const int gas = 0;
int cloud=0,statusCode, httpCode, httpResponseCode,send=0;
int bf1,bf2,bf3,bf4,bf5,bf6,af1,af2,af3,af4,af5,af6,x,thingspeak=0;

unsigned long currentMillis,getcurrtime,getSendTime,previousMillis = 0;

void getTime(){
  timeClient.update();
  String formattedTime = timeClient.getFormattedTime();
  Serial.print(formattedTime);
}

void setup(){

  // Initializing LCD
  lcd.init();
  lcd.backlight();

  // Setting serial monitor
  Serial.begin(115200);

  // Connecting to wifi
  WiFi.begin(ssid, password);
  lcd.setCursor(0,0);
  lcd.print("Connecting..");
  Serial.println("Connecting to " + String(ssid));
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Initialize a NTPClient to get time
  timeClient.begin();
```

```
  timeClient.setTimeOffset(19800);
  Serial.println();
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
  lcd.setCursor(0,0);
  lcd.print("Connected..");
  ThingSpeak.begin(client);
  Wire.begin();
  delay(2000);
  lcd.clear();
  cloud=0;
}

void loop(){

  Serial.println();
  currentMillis = millis();
  getcurrtime=millis();
  Serial.println("Current millisec: "+ String(currentMillis));

  do{
    t = dht.readTemperature();
    delay(200);
  }while(isnan(t));

  do{
    h = dht.readHumidity();
    delay(200);
  }while(isnan(h));

  do{
    g = analogRead(MQ3pin);
    delay(200);
  }while(isnan(x));

  z = (g/1024)*100;

  // Displaying Sensor data
  String curr_condition="Temp: " + String(t)+" deg Cel, Hum: "+String(h)+"%, Gas: "+String(z) +
"%";
  Serial.println(curr_condition);

  lcd.setCursor(0,0);
  lcd.print("T:");
  lcd.setCursor(2,0);
  lcd.print(t);

  lcd.setCursor(9,0);
  lcd.print("H:");
  lcd.setCursor(12,0);
  lcd.print(h);
```

```
  lcd.print(" %");

  lcd.setCursor(0,1);
  lcd.print("Gas:");
  lcd.setCursor(5,1);
  lcd.print(z);
  lcd.setCursor(10,1);
  lcd.print("%");


     //Updating database after every 30 seconds
  if(cloud==0 || currentMillis - previousMillis >= 20000 ){
    Serial.println("Database updation...");

    ThingSpeak.setField(1, t);
    ThingSpeak.setField(2, h);
    ThingSpeak.setField(3, z);
    ThingSpeak.setStatus(myStatus);

    // write to the ThingSpeak channel
    x=0;
    do{
      x = ThingSpeak.writeFields(2492979,"BV63T1LIWZUMB5FE");
      delay(200);
    }while(x!=200);


    do{
      http.begin(client,"http://coldstoragemonitor.000webhostapp.com/add.php?temp="+ String(t)+
"&hum="+String(h)+"&gas="+String(z));
      httpCode = http.GET();
      Serial.println(httpCode);
      if(httpCode!=200){
        delay(5000);
      }
    }while(httpCode!=200);

    http.end();
    cloud=1;
    previousMillis = currentMillis;
  }


  // Sending alert to user when temperature goes beyond Ideal values
  if(t<=13 || t>=20){

    if(send==0 || getcurrtime-getSendTime>=900000){ //FIRST ALERT MSG or SEND MSG
AFTER EVERY 15 Min
      Serial.println("ALERT MSG SEND");
      String message="Alert...Take Action. "+ curr_condition;
      String url = "http://api.callmebot.com/text.php?user=borkarh&text="+urlEncode(message);
```

```
httpResponseCode = 0;

do{
  http.begin(client, url);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  httpResponseCode = http.POST(url);
  delay(200);
}while(httpResponseCode!=200);

Serial.println("Message sent successfully");
http.end();


send=1;
getSendTime = getcurrtime;
thingspeak=1;

// Reading thingspeak data
statusCode=0;
do{
  bf1 = ThingSpeak.readLongField(myChannelNumber, 1, myReadAPIKey);
  statusCode = ThingSpeak.getLastReadStatus();
  delay(200);
}while(statusCode!=200);

statusCode=0;
do{
  bf2 = ThingSpeak.readLongField(myChannelNumber, 2, myReadAPIKey);
  statusCode = ThingSpeak.getLastReadStatus();
  delay(200);
}while(statusCode!=200);

statusCode=0;
do{
  bf3 = ThingSpeak.readLongField(myChannelNumber, 3, myReadAPIKey);
  statusCode = ThingSpeak.getLastReadStatus();
  delay(200);
}while(statusCode!=200);

statusCode=0;
do{
  bf4 = ThingSpeak.readLongField(myChannelNumber, 4, myReadAPIKey);
  statusCode = ThingSpeak.getLastReadStatus();
  delay(200);
}while(statusCode!=200);

statusCode=0;
do{
  bf5 = ThingSpeak.readLongField(myChannelNumber, 5, myReadAPIKey);
  statusCode = ThingSpeak.getLastReadStatus();
```

```
      delay(200);
    }while(statusCode!=200);

    statusCode=0;
    do{
      bf6 = ThingSpeak.readLongField(myChannelNumber, 6, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    Serial.println("Thingspeak before code " + String(statusCode));
    Serial.println("Before value: bf1: " + String(bf1) + " bf2: " + String(bf2) + " bf3: " +
String(bf3)+ " bf4: " + String(bf4)+ " bf5: " +String(bf5)+ " bf6: " +String(bf6) );

  }

  if(getcurrtime-getSendTime>=60000 && thingspeak==1){ //Waiting for 1 Min
    statusCode=0;
    do{
      af1 = ThingSpeak.readLongField(myChannelNumber, 1, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    statusCode=0;
    do{
      af2 = ThingSpeak.readLongField(myChannelNumber, 2, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    statusCode=0;
    do{
      af3 = ThingSpeak.readLongField(myChannelNumber, 3, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    statusCode=0;
    do{
      af4 = ThingSpeak.readLongField(myChannelNumber, 4, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    statusCode=0;
    do{
      af5 = ThingSpeak.readLongField(myChannelNumber, 5, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);
```

```
    statusCode=0;
    do{
      af6 = ThingSpeak.readLongField(myChannelNumber, 6, myReadAPIKey);
      statusCode = ThingSpeak.getLastReadStatus();
      delay(200);
    }while(statusCode!=200);

    Serial.println("Thingspeak After code " + String(statusCode));
    Serial.println("After value: af1: " + String(af1) + " af2: " + String(af2) + " af3: " + String(af3)+
" af4: " + String(af4)+ " af5: " +String(af5)+ " af6: " +String(af6) );

    thingspeak=0;

    if(af1==bf1 && af2==bf2 && af3==bf3 && af4==bf4 && af5==bf5 && af6==bf6) {

      // Writing Thingspeak data
      ThingSpeak.setField(1, 10);
      ThingSpeak.setField(2, 10);
      ThingSpeak.setField(3, 10);
      ThingSpeak.setField(4, 10);
      ThingSpeak.setField(5, 10);
      ThingSpeak.setField(6, 10);
      ThingSpeak.setStatus(myStatus);

      // write to the ThingSpeak channel
      x=0;
      do{
        x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
        delay(1000);
      }while(x!=200);

      if(x == 200){
        Serial.println("Channel update successful.");
        String msg="Automatic time updated";
        String url = "http://api.callmebot.com/text.php?user=borkarh&text="+urlEncode(msg);

        httpResponseCode = 0;

        do{
          http.begin(client, url);
          http.addHeader("Content-Type", "application/x-www-form-urlencoded");
          httpResponseCode = http.POST(url);
          delay(200);
        }while(httpResponseCode!=200);
        Serial.println("Message sent successfully");
        http.end();

      }
      else{
        Serial.println("Problem updating channel. HTTP error code " + String(x));
      }
```

```
      }
      else{
        Serial.println("User has taken action");
      }
    }
    else{
      Serial.println("Waiting for response");
    }
  }

  Serial.println();
  Serial.println();
  delay(200);
}
```

3.4.6 Web Based UI for client:

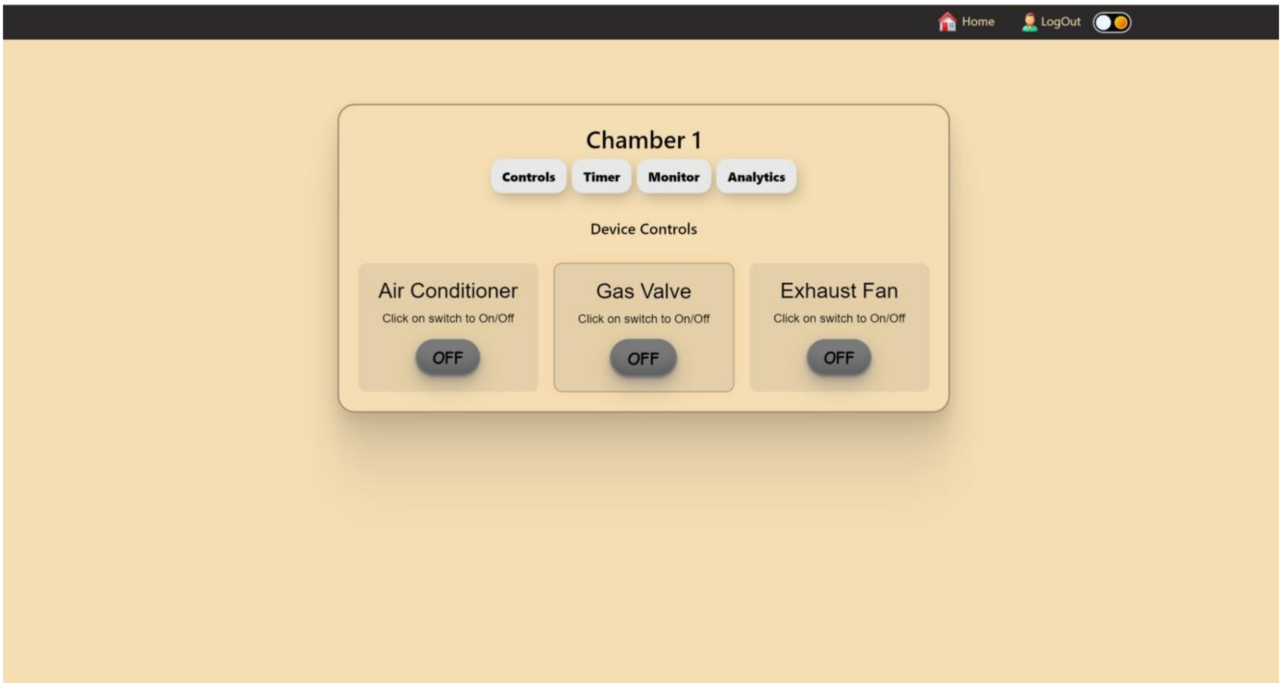


Fig. 3.19 Device Controls Panel

There are four tabs under each chamber – Controls, where the components can be manually turned ON/OFF, Timer – where the components can be set for automatic working for the set period of time, Monitor – where the environment of Cold Storage can be monitored and tracked and Analytics – where all the changes and variations in the Cold Storage system can be monitored

There are three controls under "Controls" tab - Air Conditioner, Gas Valve and Exhaust Fan. This controls can be used to turn Air Conditioner, Gas Valve and Exhaust Fan ON/OFF. These are the manual controls to turn the respective components turn ON/OFF by the user.

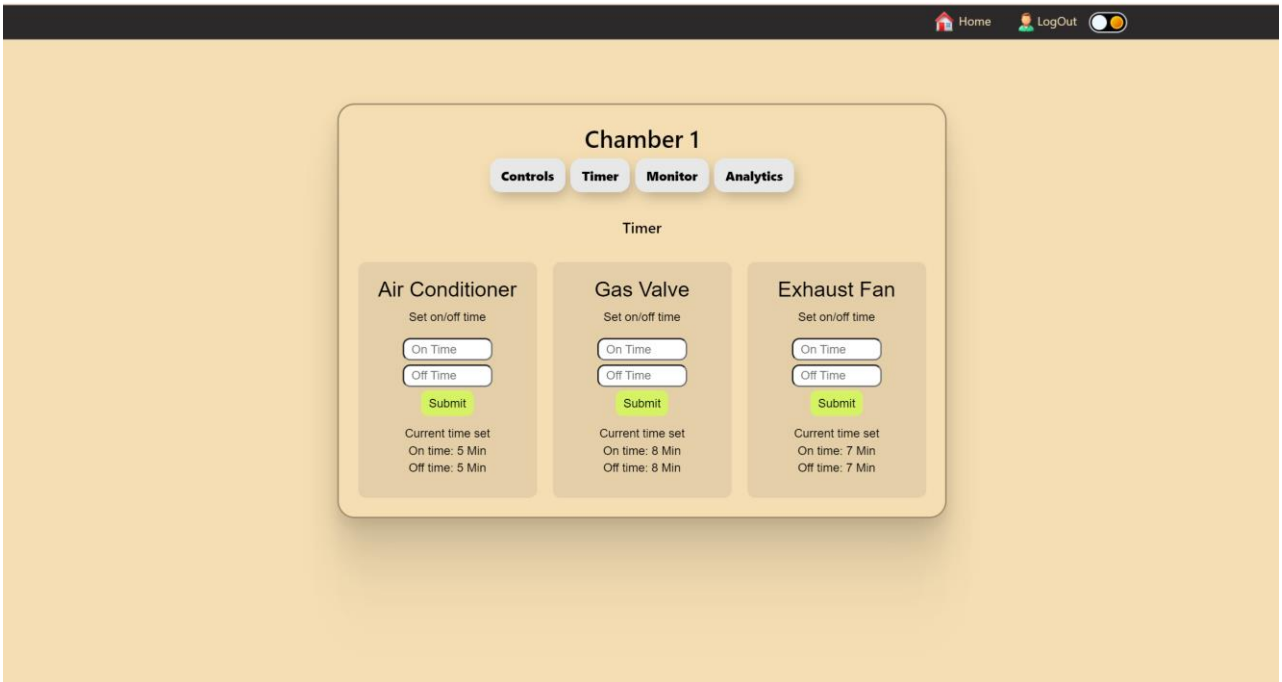


Fig. 3.20 Timer Panel

This is the “Timer” tab, all the three components can be set to work automatically for a specified time by the user. The on-time indicates the time for which the component will be ON and off-time indicates the time for which it will be OFF. It runs in a loop for specified time by the user as needed and according to the food stored in the Cold Storage system.

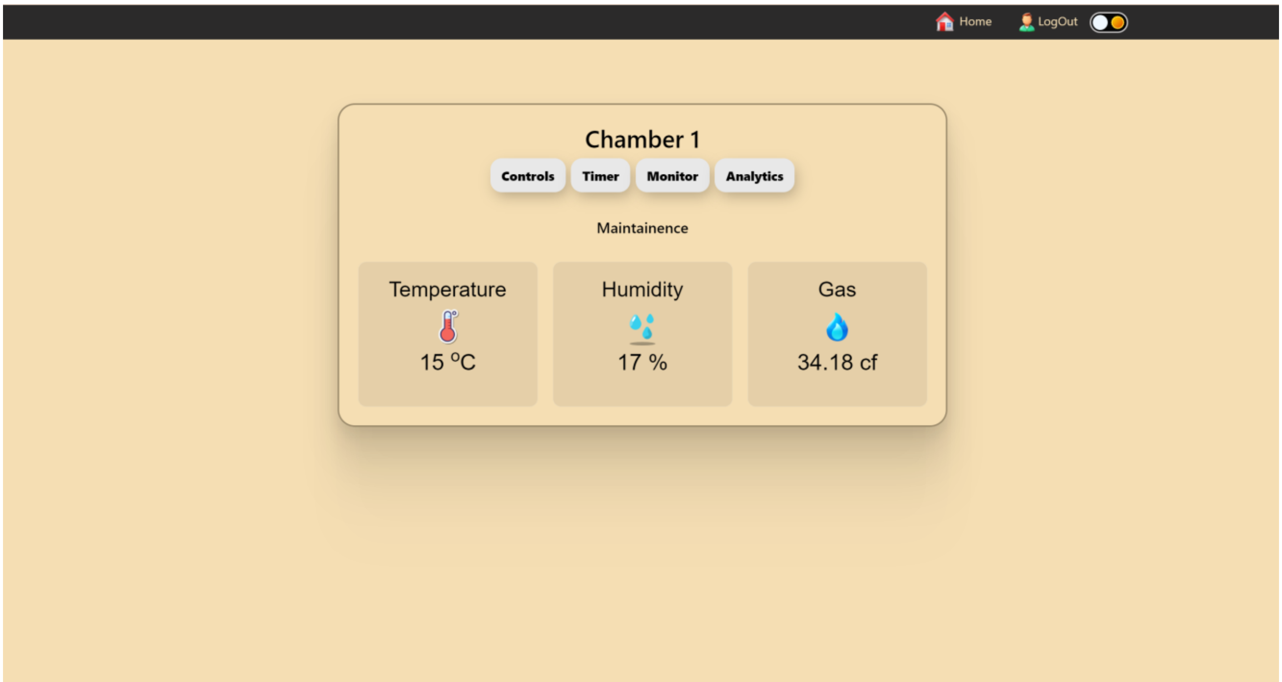


Fig. 3.21 Monitoring Panel

This the monitor tab where the user can monitor Temperature, Humidity and Gas in the respective chamber. The temperature, humidity and Gas has to be maintained at a specific value according to the food that is stored in the Cold Storage chamber. These values are also tracked for the Alert system which sends alerts to the user if there it changes over the specified for the chamber.

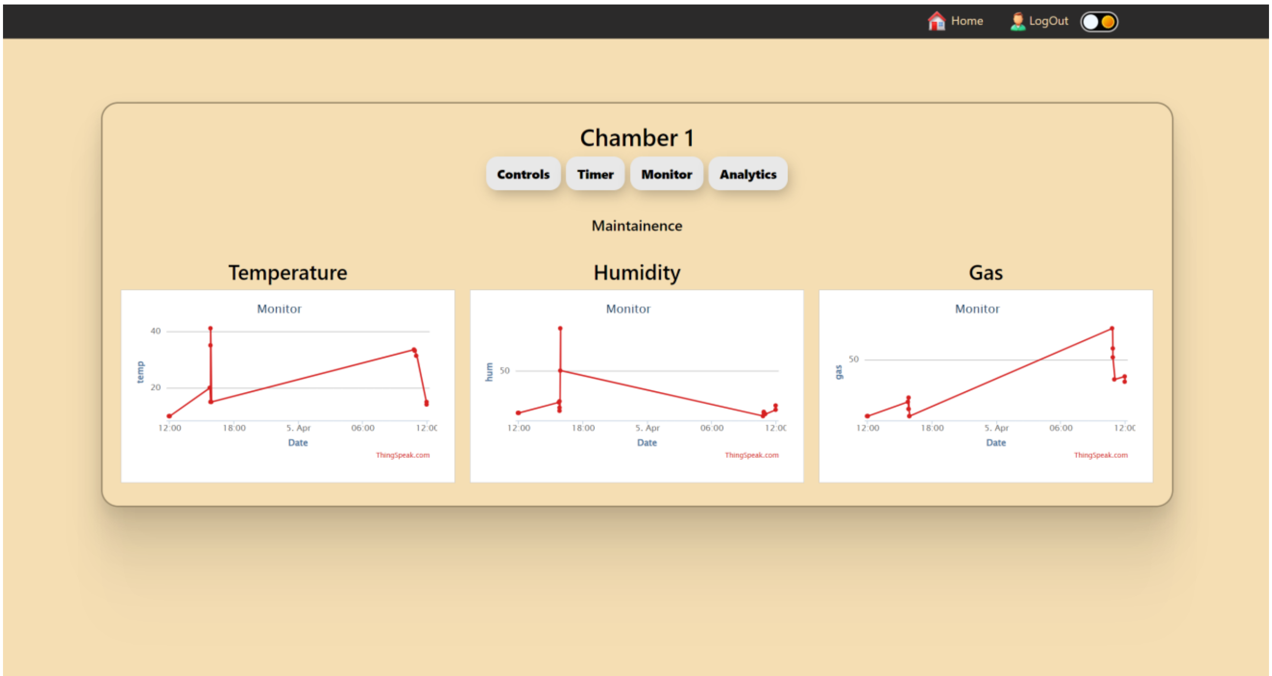


Fig. 3.22 Analytical Panel

This is the analytics tab where all the major parameters like temperature, humidity and gas are deeply monitored and tracked for analysis purpose by the system. If there is change over the specified value for the chamber the system sends an alert to the user and if the user does not take manual control of the situation then the system automatically takes control.

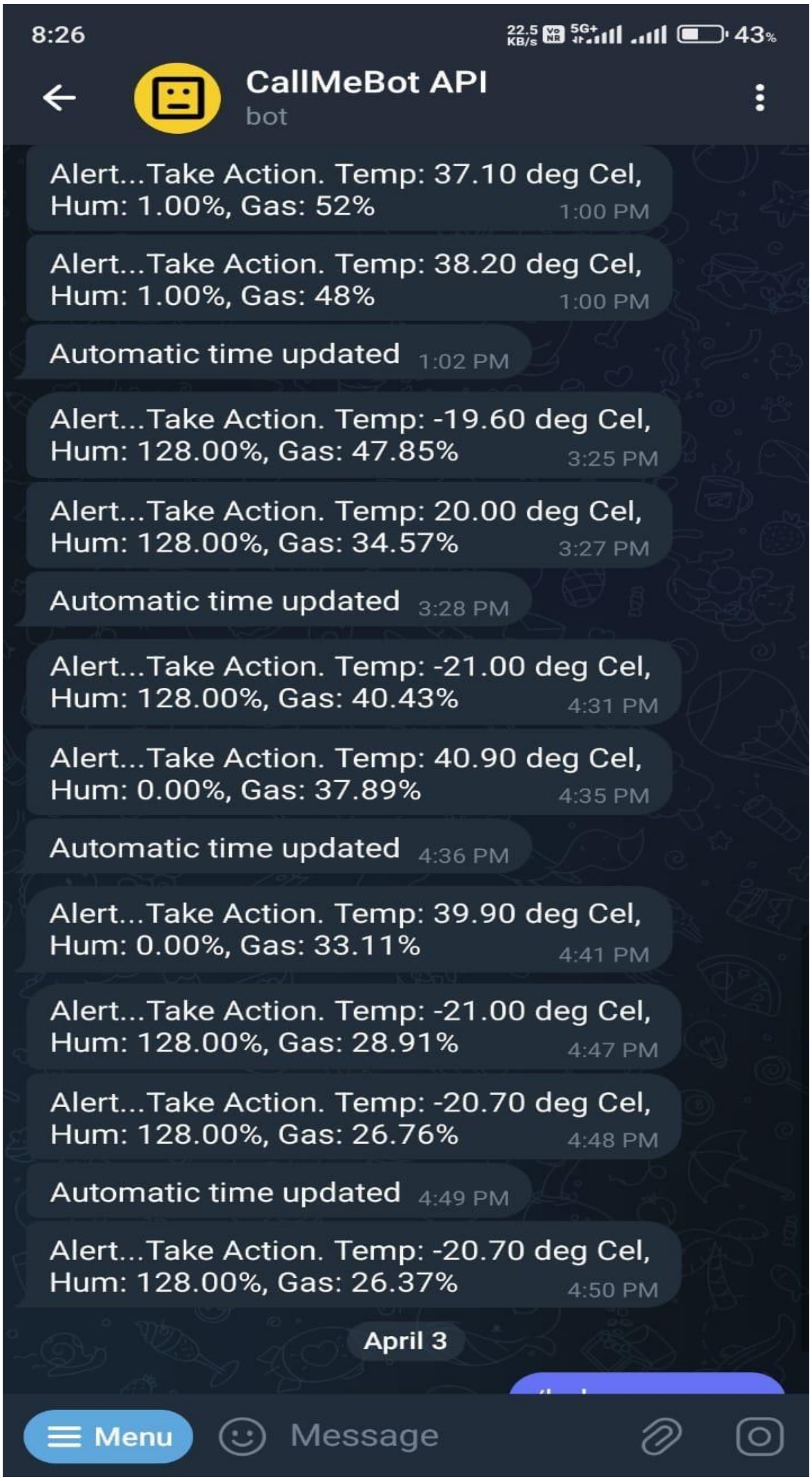


Fig. 3.23 Alert messages

CHAPTER 4
RESULT AND DISCUSSION

CHAPTER 4
RESULT AND DISCUSSION

The results of the **IoT based manual and automated monitoring and controlling for cold storage system** indicate successful integration of sensors and controllers for effective monitoring and control. The Master NodeMcu efficiently transmits data to the cloud server, enabling real-time access through the web-based Control Panel. The DHT11 Temperature andMQ-3 gas sensors on the Slave controller accurately measure and monitor temperature, humidity, and gas levels within the cold storage chamber. Relay 1 demonstrates reliable operation of the Air Conditioner, maintaining the desired storage conditions.

The 16*2 LCD display provides a user-friendly interface for observing temperature and gas values. Notably, Relay 2 on the master ESP controller effectively manages the Ethylene gascylinder, a critical element in preventing banana ripening during storage. This robust systemensures precise environmental control, offering a promising solution for banana storage andripening prevention.

In discussion, the implemented IoT system proves its efficacy in addressing the specific requirements of banana storage. The seamless communication between devices, cloud integration, and sensor accuracy contribute to a reliable and user-friendly solution. The use of multiple sensors enhances the system's capability to maintain optimal storage conditions. The successful prevention of ripening through Ethylene gas control highlights the practicality and versatility of the proposed cold storage system, showcasing its potential for broader applicationsin the food industry. Future enhancements could explore additional features and scalability for accommodating varying storage needs.

CHAPTER 5
CONCLUSION AND FUTUTRE SCOPE

CHAPTER 5
CONCLUSION AND FUTUTRE SCOPE

5.1 Conclusion

In conclusion, the developed IoT based manual and automated monitoring and controlling for cold storage system has proven to be a robust and effective solution for banana storage, specifically designed to prevent ripening. The integration of Master NodeMcu, Slave controller, sensors, and relays ensures accurate monitoring and control of critical environmental parameters such as temperature, humidity, and gas levels. The successful operation of the Air Conditioner and Ethylene gas cylinder demonstrates the system's ability to maintain optimal storage conditions and prevent premature ripening of bananas.

The user-friendly web-based Control Panel and LCD display enhance accessibility and alert system provide a convenient interface for users to monitor and manage the storage environment. Overall, the system fulfills its intended purpose of extending the shelf life of bananas by creating an ideal storage atmosphere.

5.2 Future Scope:

The presented cold storage system opens avenues for further improvements and expansions:

1. **Energy Efficiency:** Explore energy-efficient mechanisms for the Air Conditioner to enhance sustainability and reduce operational costs.
2. **Machine Learning Integration:** Incorporate machine learning algorithms to analyze historical data and optimize the system's performance over time.
3. **Mobile Application:** Develop a mobile application for enhanced accessibility, allowing users to monitor and control the cold storage system using smartphones.
4. **IoT Security:** Strengthen security protocols to safeguard the system against potential cyber threats, ensuring the integrity of stored data.

By addressing these aspects, the cold storage system can evolve into a more advanced, adaptive, and versatile solution with broader applications in the food industry

REFERENCES

[1]. Smart IoT-Based Control System for Remotely Managing Cold Storage Facilities by Maged Mohammed 1,2, *ORCID, Khaled Riad 3,4ORCID and Nashi Alqahtani 1,5

[2]. Maged Mohammed, Khaled Riad, Nashi Alqahtani, “Design of a Smart IoT-Based Control System for Remotely Managing Cold Storage Facilities”. Sensors 2022, 22(13), 4680 (2022).

[3]. Hina Afreen, Imran Sarwar Bajwa, “An IoT-Based Real-Time Intelligent Monitoring and Notification System of Cold Storage”. IEEE Access PP(99): 1-1. (Volume 9), ISSN: 2169-3536, (2021).

[4]. Narender Chinthamu, R. Reka, B. Sundaramurthy, Rajendra Singh Bisht, Pilli Lalitha Kumari, Amit Singh Rajput, “IoT-Driven Cold Chain Management Ensuring Food Safety and Quality in Processing and Distribution”. Published in European Chemical Bulletin Open Access EJournal. ISSN: 2063-5346 (2023).

[5]. Shagun Gupta, Shubham Mahajan, Amit Pandit, “A Review on Image Processing Techniques”. 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020 (2020).

[6]. Yanlin T, Zuoxin Hu, Tianyu T, Xin Gao, “Effect of goods stacking mode on temperature field of cold storage”. Proceedings of EEEP 2020 IOP Conference Series: Earth and Environmental Science. Volume 675(1): 012052, 5th ICEEEP, IOP Publishing. Orlando,FL (2021).

[7]. Mercedes Sáenz-Baños, Juan Ignacio Latorre-Biel, Eduardo Martínez-Cámara, Emilio Jiménez-Macías, Francesco L, Julio BlancoFernández, “Methodology for energy demand reduction of potato cold storage process”. Published in Journal of Food Process Engineering, Published by Wiley. Volume: 45, Issue:10/ e14127 (2022).

[8]. Gunawati, A.H.Dongoran,A.Setiawan, “Evaluation of the performance of cold storage box enveloped with phase change materials”. Published in Journal of Physics: ConferenceSeries. Conf. Ser. 1242(1): 012023. International Seminar on Science and Technology, IOP Publishing (2019).

[9]. Y.F.Kurnia1, E.L.S.Suharto, E.Purwati, “Quality of fermented goat milk with carrot juice during cold storage”. Published unde r IOP Conference Series: Earth and 87 Environmental Science, Volume 694 012 076. Published under IOP Publishing Ltd, International e-Conference on Sustainable Agriculture and Farming System, Bogor,Indonesia (2021).

APPENDIX

APPENDIX A

G-17 Project Group Members

Group Members from left to right:

1. Soniya Rangari

2. Sankit Binkar

3. Jiwan Dehankar (Guide)

4. Aditya Pethe

5. Harshal Borkar

Harshal Borkar

Mobile no.: 8668371109 | Email-id: harshalborkar10105@gmail.com

Date of Birth: 28/12/2002 | Address: 263, babul ban, garoba maidan, Nagpur, Maharashtra

Objective: Pasionate 4th-year software engineering student with experience in programming languages such as Python, Java, and C++, and a keen interest in software development and web development. Aiming to secure an internship to enhance my coding skills and contribute to innovative software solutions.

Academic Qualification:

Enrollment Number: 20010850

Branch of Study: Computer Science and Engineering

Qualification	Institute	Year	Score/SGPA
B.Tech 7^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	8.39
B.Tech 6^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	7.65
B.Tech 5^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	8.12
B.Tech 4^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	7.74
B.Tech 3^{rd} Semester	Yeshwantrao Chavan College of Engineering	2021	7.35
B.Tech 2^{nd} Semester	Yeshwantrao Chavan College of Engineering	2021	7.73
B.Tech 1^{st} Semester	Yeshwantrao Chavan College of Engineering	2020	8.14
Class XII (HSC)	Shivaji Science College	2020	80.62
Class X (SSC)	Shri Umaya Shankar Narayanji High School	2018	91

Projects:

- **Growmate:** Growmate is a crop recommendation and plant disease identification web-based application developed using machine learning algorithm and web technologies such as Python- Flask, HTML, CSS, JS, APIs.
- **IoT based Cold Storage Monitoring and Manual and Automated Controlling System**: It is an IoT based web application designed and developed using various web technologies such as HTML, CSS, JS and PHP.
- **DailyDose**: DailyDose is a News app developed using ReactJS and free Api to fetch real-time news.

Programming Language Proficiency: C, C++, Python, SQL, HTML, CSS, JavaScript, PHP.

Student Signature

Sankit Binkar

Mobile no.: 7387424290 | Email-id: sankitbinkar0904@gmail.com

Date of Birth: 09/04/2002 | Address: 96, New Balaji nagar, Nagpur, Maharashtra

Objective: Enthusiastic 4th-year software engineering student with experience in programming language Python but I keen interest in management and finance. Aiming to secure post-graduation in management to enhance my management and finance skills and contribute to growing Indian economy.

Academic Qualification:

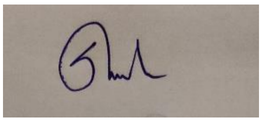
Enrollment Number: 20010825

Branch of Study: Computer Science and Engineering

Qualification	Institute	Year	Score/SGPA
B.Tech 7th Semester	Yeshwantrao Chavan College of Engineering	2023	7.63
B.Tech 6th Semester	Yeshwantrao Chavan College of Engineering	2023	6.81
B.Tech 5th Semester	Yeshwantrao Chavan College of Engineering	2022	7.3
B.Tech 4th Semester	Yeshwantrao Chavan College of Engineering	2022	6.87
B.Tech 3rd Semester	Yeshwantrao Chavan College of Engineering	2021	7.65
B.Tech 2nd Semester	Yeshwantrao Chavan College of Engineering	2021	7.66
B.Tech 1st Semester	Yeshwantrao Chavan College of Engineering	2020	7.25
Class XII (HSC)	Shivaji Science College	2020	74.56
Class X (SSC)	South Point School	2018	91

Projects:

- **Growmate:** Growmate is a crop recommendation and plant disease identification web-based application developed using machine learning algorithm and web technologies such as Python- Flask, HTML, CSS, JS, APIs.
- **IoT based Cold Storage Monitoring and Manual and Automated Controlling System**: It is an IoT based web application designed and developed using various web technologies such as HTML, CSS, JS and PHP.
- **Jarvis:** It,s my voice assistant that uses speech recognitionV pyttsx3V webbrowser and subprocess modules which listen to voice and performs the function accordingly.



Student Signature

ADITYA PETHE

Mobile no.: 9860571108 | Email-id: adityapethe07@gmail.com

Date of Birth: 21/11/2001 | Address: 225, Kelod, Nagpur, Maharashtra
Pin code: 441 112

Objective: Curious, innovative thinker, initiative taker and multi dimensional professional, logical and analytical skills. Seeking a challenging role that allows me to expand my knowledge, work on complex projects, and collaborate with a talented team to drive innovation and achieve business goals. Ability to handle multiple projects simultaneously with a high degree of accuracy.

Academic Qualification:

Enrollment Number: 20010659

Branch of Study: Computer Science and Engineering

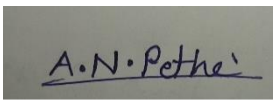
Qualification	Institute	Year	Score/SGPA
B.Tech 7^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	9.39
B.Tech 6^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	8.88
B.Tech 5^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	9.45
B.Tech 4^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	9.68
B.Tech 3^{rd} Semester	Yeshwantrao Chavan College of Engineering	2021	9.52
B.Tech 2^{nd} Semester	Yeshwantrao Chavan College of Engineering	2021	9.16
B.Tech 1^{st} Semester	Yeshwantrao Chavan College of Engineering	2020	9.23
Class XII (HSC)	Bhalerao Junior Science College Saoner	2020	90.62 %
Class X (SSC)	Bhikulal Chandak Highschool Kelod	2018	95.60 %

Projects:

1. **Social Distancing Project:** This project is made using Computer Vision and Machine Learning used in Covid Situation under Intel Unnati Industrial Training.

2. **Smart Water Level Monitoring System:** When the water tank gets full and it is successfully detected by sensor. After detecting, the message is successfully sent by the controller on users mobile, LED light is blinking and Buzzer is beeping. The water pump is switched off automatically by system. (IOT based, Mini Project)

Programming Language Proficiency: C, Python, .Net FullStack, Java OOPS, DBMS using SQL, Web Development, IOT, R language



Student Signature

Soniya Rangari

Mobile no.: 9545653179 | Email-id: rangarisoniya661@gmail.com

Date of Birth: 29/01/2003 | Address: Dabha Chowk,Nagpur, Maharashtra

Objective: Good communicator with efficient self-starter eager by actively engaging in group project. Passionate to learn about web development and data science. Offering keen attention to detail, with exceptional analytical, problem solving and interpersonal skills. Desire to enhance social events through technology.

Academic Qualification:

Enrollment Number: 20011028

Branch of Study: Computer Science and Engineering

Qualification	Institute	Year	Score/SGPA
B.Tech 7^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	5.00
B.Tech 6^{th} Semester	Yeshwantrao Chavan College of Engineering	2023	6.36
B.Tech 5^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	6.65
B.Tech 4^{th} Semester	Yeshwantrao Chavan College of Engineering	2022	5.67
B.Tech 3^{rd} Semester	Yeshwantrao Chavan College of Engineering	2021	7.80
B.Tech 2^{nd} Semester	Yeshwantrao Chavan College of Engineering	2021	8.03
B.Tech 1^{st} Semester	Yeshwantrao Chavan College of Engineering	2020	8.68
Class XII (HSC)	N.P.K.Junior College	2020	65%
Class X (SSC)	Katakwar Convent	2018	82%

Projects:

1. **Fire Detection and alarm system:** This project is based upon image processing system by the use of camera as well as image by applying machine learning algorithm which also alert using alarm system to the client.
2. **News app using python** : Here the api gives the short news regarding sports, entertainment and general . We made this app using tkinter and various data science libraries.

Programming Language Proficiency: Python, SQL, HTML, CSS, Javascript.

Student Signature

SOCIETAL IMPACT

This project has major impact on two industries, first one is the food industry for storing fruits and vegetables and second one is the medical industry for storing medicines.

We have specifically focused on food industry for storing bananas. It has many sensors which continuously monitor the environment of cold storage. It prevents growth of micro-organisms and can also release ethylene for ripening process. This system can be monitored and controlled remotely from the web application. This also has a automated feature so if the user is not available to control the system manually or system is idle for 15 minutes then the software will automatically take control of the system and changes will be made by itself if required.

This same system can be installed and used for storing medicines and other medical equipments which is required to be stored in a specific temperature or wants continuous monitoring and controlling .

It’s advance software allows the user to control the system remotely which decrease the cost of storing and maintenance and works more efficiently than manual control by humans which increases quality of products, efficient and increased time storage and profit.