A Smart Ventilation Solution for Comfortable Indoor Environment

1. AIM

- The aim of this project is to develop a smart temperature-controlled exhaust system using IoT technology and the Blynk application.
- This system will automatically activate an exhaust fan when the temperature exceeds a certain threshold, ensuring efficient ventilation and comfort in enclosed spaces.

2. ABSTRACT

The Smart Temperature Controlled Exhaust System represents an innovative solution for effectively regulating indoor environmental conditions in various settings, including residential, commercial, and industrial spaces.Traditional exhaust systems often lack the intelligence to adapt to changing environmental factors, leading to inefficient operation, energy wastage, and suboptimal indoor comfort levels.In response to this challenge, this project aims to develop an intelligent IoT-based system capable of real-time monitoring and control of temperature and humidity levels to ensure optimal indoor conditions.The project involves the integration of hardware components such as microcontrollers, temperature and humidity sensors, relay modules, and exhaust fans, along with software development for data acquisition, processing, and remote communication. The system utilizes the ESP8266 Wi-Fi module for IoT connectivity, enabling seamless integration with the Blynk platform for remote monitoring and control via a mobile application or web interface.

3. INTRODUCTION

- In indoor environments such as residential buildings, commercial spaces, and industrial facilities, maintaining optimal temperature and humidity levels is crucial for ensuring occupant comfort, health, and productivity.
- Excessive heat or humidity can lead to discomfort, decreased air quality, and potential health hazards.
- Traditional exhaust systems often lack the intelligence to adapt to changing environmental conditions efficiently.
- Therefore, there is a need for an intelligent solution that can monitor temperature and humidity levels in real-time and activate an exhaust system when necessary to regulate indoor conditions effectively.

4. PROBLEM STATEMENT

- Traditional exhaust systems often operate on fixed schedules or manual controls, leading to energy wastage and inefficient ventilation.
- Moreover, in environments where temperature fluctuations are significant, maintaining a comfortable atmosphere becomes challenging.
- Our project addresses these issues by providing an automated solution that adapts to changing temperature conditions, ensuring optimal ventilation while minimizing energy consumption

5. OBJECTIVE

The objective of this project is to develop a Smart Temperature Controlled Exhaust System using IoT technology to monitor and regulate indoor temperature and humidity levels automatically. The system should be capable of:

- To develop a system comprising DHT11 sensors to continuously monitor indoor temperature and humidity levels in real-time.
- To implement algorithms tailored to detect specific temperature and humidity thresholds indicative of uncomfortable or potentially hazardous indoor conditions.
- To create logic to automatically activate the exhaust fan or ventilation system upon surpassing predefined temperature or humidity thresholds, ensuring prompt environmental regulation.
- To integrate remote monitoring and control features via a user-friendly Blynk mobile application or web interface, enabling users to monitor temperature and humidity levels and adjust DHT11 sensor settings remotely.
- To employ energy-efficient algorithms to regulate the operation of the exhaust fan, minimizing energy consumption while maintaining optimal indoor conditions.
- To incorporate fail-safe mechanisms, error handling protocols, and emergency shutdown procedures to guarantee the reliability and safety of the exhaust fan system, mitigating the risk of malfunctions or accidents.

6. ARCHITECHTURE

The following is a architectural design that depicts a circuit that controls a fan using an NodeMCU board based on temperature readings from a sensor.

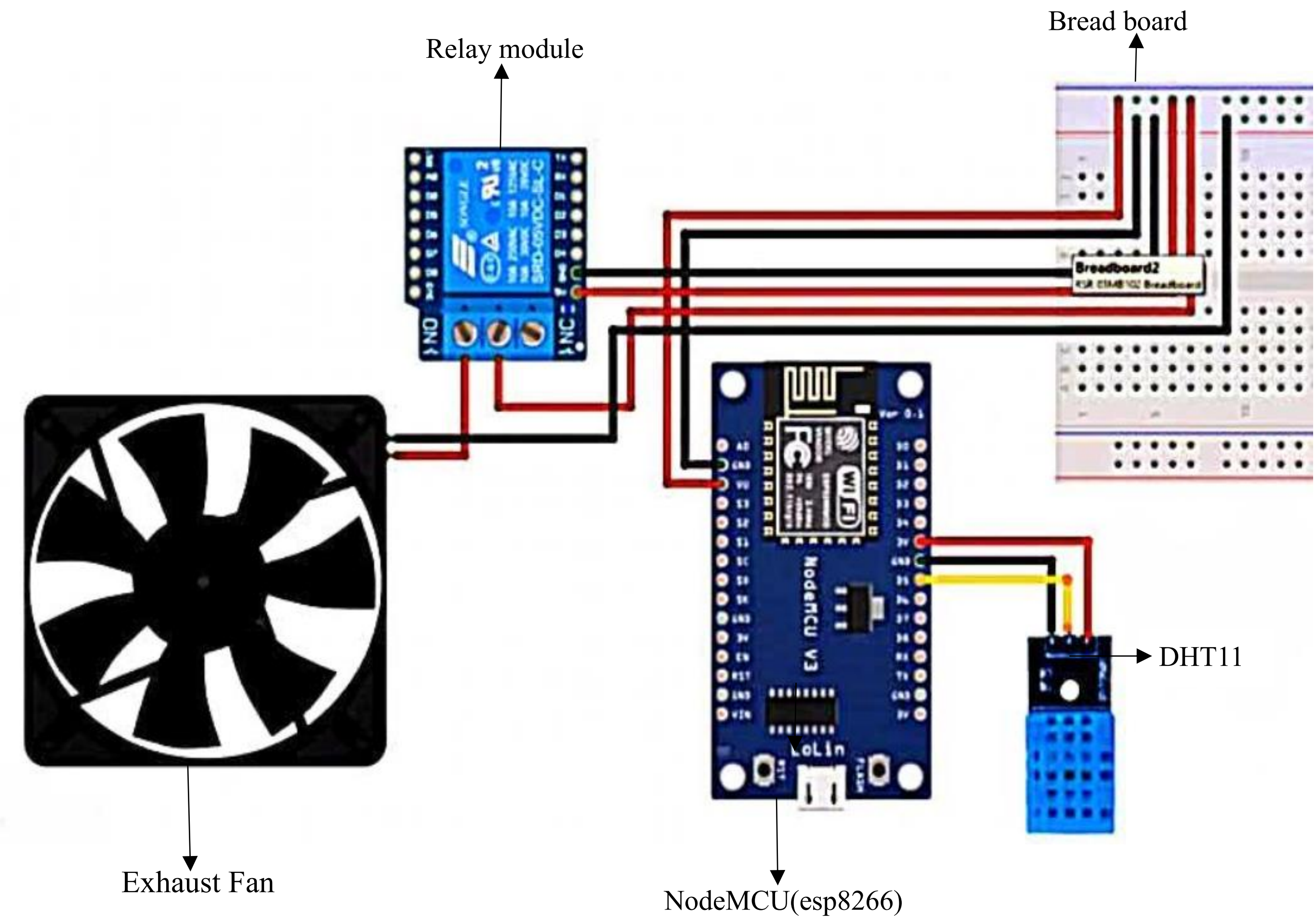

Fig:1 Smart Temperature Controlled Exhaust System

7. PROPOSED METHOD

Libraries:

• ESP8266WiFi: This library provides support for connecting the ESP8266 Wi-Fi module to a Wi-Fi network.
• BlynkSimpleEsp8266: This library enables communication between the ESP8266 module and the Blynk IoT platform, allowing for remote monitoring and control of the system.
• DHT: This library facilitates communication with the DHT temperature and humidity sensor.

Sensors:

• DHT Temperature and Humidity Sensor: Measures ambient temperature and humidity levels and provides digital readings to the NodeMCU board.
• Exhaust Fan Relay: Controls the activation of the exhaust fan based on temperature readings. When the temperature exceeds the threshold, the relay switches on the exhaust fan to regulate temperature and maintain comfort levels.

8. PROCEDURE

Step 1: Circuit Assembly

- **Connect Sensors:** Wire the temperature and humidity sensors to the NodeMCU board according to their pin configurations. Typically, they require connections for power, ground, and data.
- **Attach Relay Module:** Connect the relay module to the NodeMCU board. Ensure proper wiring to control the exhaust fan.
- **Power Supply:** Connect the power source to the NodeMCU board to provide sufficient power for the entire system.
- **Mounting:** Securely mount the sensors and the relay module, ensuring they are positioned optimally for accurate sensing and efficient operation.

Step 2: Programming the NodeMCU

- **Install Arduino IDE:** Download and install the Arduino Integrated Development Environment (IDE) on your computer.
- **Write Code:** Develop the NodeMCU sketch (code) to read data from both the temperature and humidity sensors and control the exhaust fan through the relay module.
- **Temperature Control Logic**: Implement logic to activate the exhaust fan when the temperature exceeds a predefined threshold. You may choose to vary the fan speed based on the degree of temperature rise for finer control.
- **Humidity Control Logic:** Incorporate logic to activate the exhaust fan if the humidity level surpasses a specified threshold. This prevents moisture accumulation, especially in environments prone to dampness.
- **Safety Checks:** Include error handling and safety checks in the code to ensure reliable operation and prevent system malfunctions or damage.
- **Upload Code:** Upload the compiled NodeMCU sketch to the NodeMCU board using the Arduino IDE and verify successful programming.

Step 3: Testing and Calibration

- **Power On:** Switch on the power supply to activate the system.
- Sensor Calibration: Calibrate the sensors if necessary to ensure accurate readings. This may involve adjusting calibration factors in the code or using external calibration tools.
- **Operational Testing:** Monitor the system's performance under various temperature and humidity conditions. Verify that the exhaust fan activates and deactivates as expected in response to changing environmental parameters.
- **Fine-Tuning:** Fine-tune the control thresholds and parameters as needed to optimize system efficiency and responsiveness.

Step 4: Integration and Deployment

- **Enclosure:** Enclose the circuitry in a suitable housing or enclosure to protect it from environmental factors and ensure long-term reliability.
- **Installation:** Install the Smart Temperature Controlled Exhaust System in the desired location, such as bathrooms, kitchens, or industrial facilities, where temperature and humidity regulation is critical.
- **Validation:** Validate the system's performance in real-world scenarios, ensuring it meets the specified requirements and delivers the intended functionality.
- **Maintenance:** Establish a maintenance schedule to periodically inspect and service the system, including sensor calibration, component checks, and firmware updates.

By following these steps, you can design, develop, and deploy a robust Smart Temperature Controlled Exhaust System using a NodeMCU, effectively managing environmental conditions through intelligent sensor-based control.

9. REQUIREMENTS

This project depicts a configuration involving an exhaust fan, an Node MCU, and a red temperature sensor, which collectively form a system likely designed for environmental temperature control in electronic applications. Each component plays a pivotal role in ensuring efficient thermal management and system stability.

COMPONENTS REQUIRED:

1. Node MCU (or compatible microcontroller)
2. Temperature sensor (e.g., DHT11 or DHT22)
3. Humidity sensor (e.g., DHT11 or DHT22)
4. Relay module
5. Exhaust fan
6. Power source (battery pack or wall adapter)
7. Breadboard and jumper wires

S.NO	COMPONENTS	HARDWARE SPECIFICATION
1.	Node MCU	• Model: ESP8266 development board • Processor: ESP8266 microcontroller • Operating Voltage: 3.3V • Digital I/O Pins: 11 • Analog Input Pins: 1 • Flash Memory: 4MB • Wi-Fi Connectivity: Integrated Wi-Fi module (IEEE 802.11b/g/n)
2.	Temperature Sensor	• Model: DHT11 • Measurement Range: 0°C to 50°C • Temperature Accuracy: ±2°C
3.	Humidity Sensor	• Model: DHT11 • Measurement Range: 20% to 80% relative humidity • Humidity Accuracy: ±5%
4.	Relay Module	• Type: Single-channel relay module • Voltage Compatibility: 3.3V to 5V • Maximum Switching Voltage: 250VAC / 30VDC • Maximum Switching Current: 10A
5.	Exhaust Fan	• Type: AC exhaust fan • Voltage: 180-240V AC • Current: 0.1A • Power Consumption: 10 watts

Pin Configuration:

- NodeMCU ESP8266 Ground (GND) pin is connected to Relay Ground (GND) pin, providing a common ground connection to ensure proper circuit operation.

- NodeMCU ESP8266 VIN pin is connected to Relay VCC pin, supplying power to the relay module for its operation.

- D2 pin on the NodeMCU board is connected to the control input of the relay module, allowing the NodeMCU to toggle the relay and control the exhaust fan.

- D5 pin on the NodeMCU board is connected to the data pin of the DHT11 temperature sensor, enabling the NodeMCU to read temperature data from the sensor.

- The NO (Normally Open) or COM (Common) terminal of the relay module is connected to one terminal of the exhaust fan.The other terminal of the exhaust fan is connected to an appropriate power source, such as a wall outlet or power supply.

- NodeMCU 3V pin is connected to the power supply pin of the DHT11 temperature sensor, providing the necessary power for sensor operation.NodeMCU Ground (GND) and VIN pins may be connected to an external power source.

Operational Workflow of this Integrated System:

1. **Continuous Temperature Monitoring:** The temperature sensor diligently monitors ambient temperature levels, generating corresponding electrical signals proportional to the detected values.

2. **Arduino Programming Logic:** Equipped with a meticulously crafted program, the Arduino board interprets incoming temperature data, applying predefined logic to determine appropriate actions based on preset thresholds.

3. **Threshold-Based Activation:** Upon reaching a designated temperature threshold, as dictated by the programmed logic, the Arduino triggers a relay mechanism, initiating the activation sequence for the computer fan.

4. **Fan Activation and Cooling:** In response to the relay signal, the computer fan springs into action, swiftly commencing airflow circulation to dissipate excess heat and restore equilibrium within the system.

5. **Temperature Stabilization:** As the fan operates, the temperature sensor continues to monitor ambient conditions, facilitating dynamic adjustments as needed until the desired temperature range is achieved and maintained.

In Terms of System Requirements and Considerations, Several Factors Warrant Attention:

1. **Power Supply:** A reliable power source is essential to sustain uninterrupted operation of the Arduino board and associated components. This may involve utilizing a battery pack or a mains power adapter, depending on the intended application and portability requirements.

2. **Programming Proficiency:** Proficiency in programming the Arduino board is indispensable for configuring the system's behavior, including temperature threshold settings, relay control, and feedback mechanisms. A thorough understanding of Arduino's integrated development environment (IDE) and programming syntax is recommended for optimal customization and performance.

3. **Safety Protocols:** Adherence to stringent safety protocols is paramount to mitigate risks associated with electrical connections and component interactions. Proper insulation, grounding, and component isolation measures should be implemented to prevent short circuits, overheating, or other potential hazards.

10.SOURCE CODE

```
#define BLYNK_TEMPLATE_ID "TMPL3lPdSgRaM"
#define BLYNK_TEMPLATE_NAME "exhauster"
#define BLYNK_AUTH_TOKEN "GRQWgR10hEqytf5K4HzdQEEBhTm6soFJ"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#define DHTPIN D5     // Digital pin connected to the DHT sensor
// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
#define FAN_PIN D2   // FAN RELAY
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Itel vision 2S";
char pass[] = "31012004vnr";
float humDHT = 0;
float tempDHT = 0;
// Initialize DHT sensor.
DHT dht(DHTPIN, DHTTYPE);
void setup() {
 Serial.begin(115200);
 pinMode(FAN_PIN, OUTPUT);
 digitalWrite(FAN_PIN, LOW);
 Serial.println(F("DHTxx test!"));
 dht.begin();
 Blynk.begin( auth, ssid , pass );
}
 void loop() {
 Blynk.run();
```

```
// Wait a few seconds between measurements.
delay(2000);
// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
humDHT = dht.readHumidity();
// Read temperature as Celsius (the default)
tempDHT = dht.readTemperature();
// Check if any reads failed and exit early (to try again).
if (isnan(humDHT) || isnan(tempDHT)) {
Serial.println("Failed to read from DHT sensor!");
return;
}
Serial.print(F("Temperature: "));
Serial.print(tempDHT);
Serial.print(F("°C "));
Serial.println();
Serial.print(F("Humidity: "));
Serial.print(humDHT);
Serial.println(F("%"));
// Debugging information for relay state
Serial.print("Relay State: ");
Serial.println(digitalRead(FAN_PIN));
// Compare Threshold value from Blynk and DHT Temperature value.
if (tempDHT > 31) {
 Serial.println("Fan On");
 } else {
 digitalWrite(D2, HIGH);
 Serial.println("Fan Off");
 }
 // Virtual write to Blynk
 Blynk.virtualWrite(V1, tempDHT);
```

```
  Blynk.virtualWrite(V2, humDHT);
}
```

11. OUTPUT

A Smart Temperature Controlled Exhaust System automatically adjusts fan speed based on real-time temperature readings, optimizing ventilation and energy efficiency.

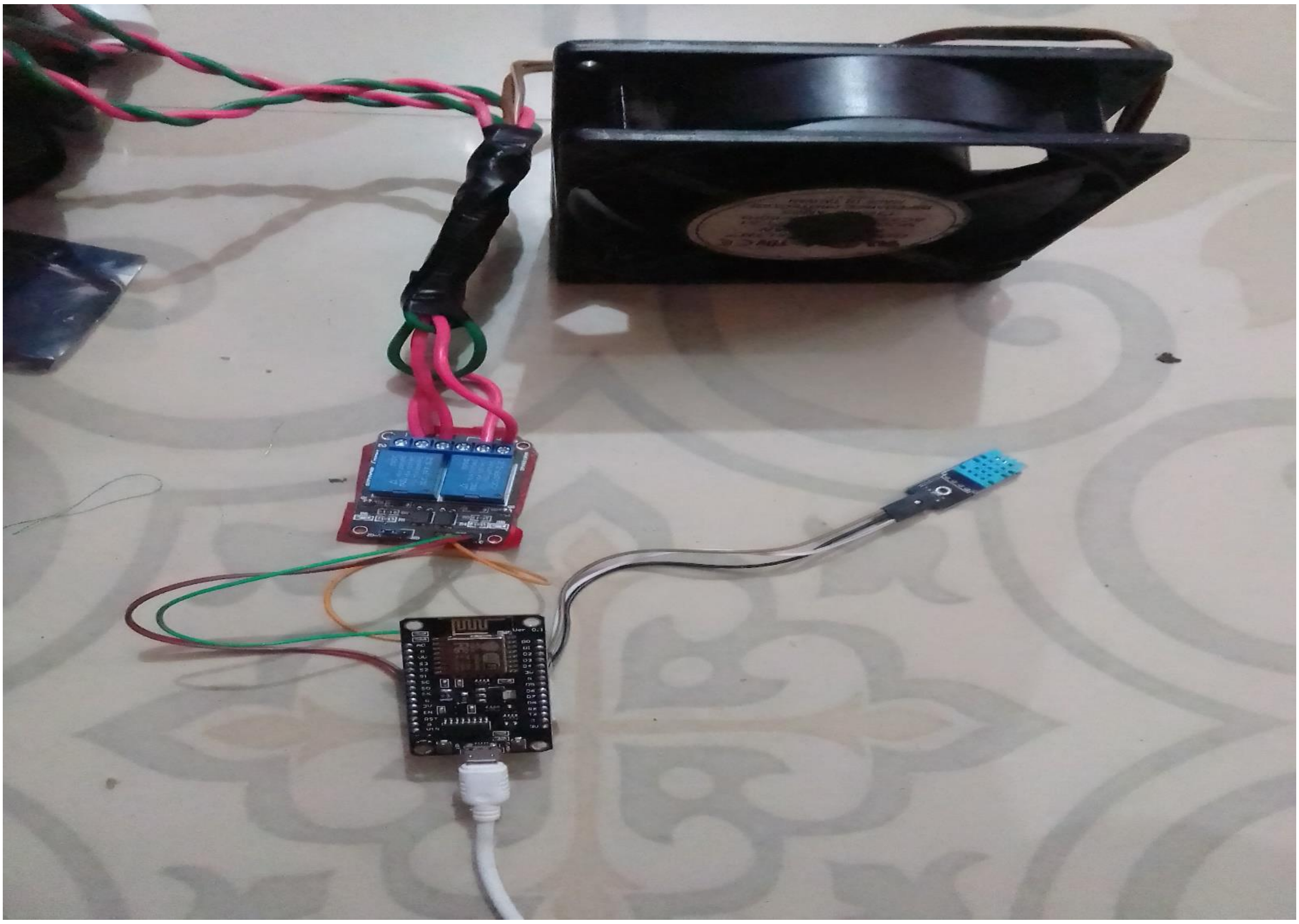

Fig 2 Automating exhaust fan output based on temperature

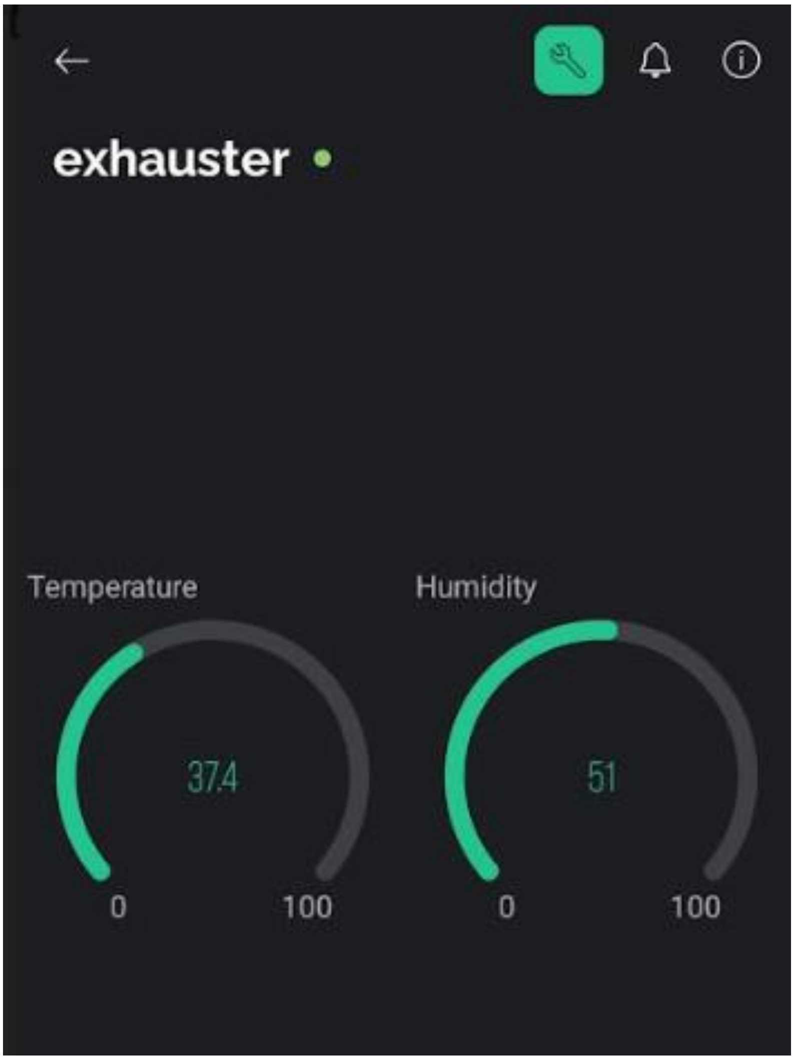

Fig 3 Blynk app for displaying of temperature and humidity

12. REAL TIME USAGE BENEFITS

The real-time usage benefits of a smart temperature-controlled exhaust system using IoT technology and the Blynk app include:

- **Energy Efficiency:** By automatically activating the exhaust fan only when necessary (i.e., when the temperature exceeds a certain threshold), the system helps to conserve energy. Traditional exhaust systems may run continuously or on fixed schedules, leading to unnecessary energy consumption.
- **Cost Savings**: Reduced energy consumption translates to cost savings on utility bills. Since the exhaust fan operates only when needed, you avoid wasting electricity on unnecessary ventilation.
- **Improved Comfort:** The system ensures a comfortable indoor environment by maintaining optimal temperature levels. When the temperature rises above the set threshold, the exhaust fan kicks in to remove excess heat, improving air circulation and comfort.
- **Remote Monitoring and Control:** With the Blynk app, users can remotely monitor the temperature status and control the exhaust fan from anywhere with an internet connection. This provides convenience and flexibility, allowing users to adjust settings according to their preferences or changing environmental conditions.

- **Enhanced Safety:** Effective ventilation is crucial for maintaining air quality and preventing the buildup of harmful gases or pollutants indoors. By automatically activating the exhaust fan when the temperature rises, the system helps to ensure proper ventilation and a healthier living or working environment.
- **Smart Automation:** The integration of IoT technology allows for smart automation of the exhaust system. Instead of manual intervention, the system autonomously responds to temperature fluctuations, providing a seamless and hassle-free experience for users.
- **Customization and Scalability:** The Blynk app offers customization options, allowing users to set their desired temperature threshold and adjust fan control settings as needed. Additionally, the system can be scaled up or expanded to integrate additional sensors or devices for comprehensive environmental monitoring and control.

Overall, a smart temperature-controlled exhaust system offers numerous benefits in terms of energy efficiency, cost savings, comfort, convenience, safety, and automation, making it a valuable addition to residential, commercial, or industrial spaces.

13.APPLICATIONS

The application of this smart temperature-controlled exhaust system with Blynk integration spans across various sectors and scenarios:

- **Residential Buildings:** Implementing the system in homes ensures comfortable living conditions by automatically regulating ventilation based on temperature fluctuations.
- **Commercial Spaces:** Offices, shops, and restaurants benefit from improved air quality and energy efficiency, enhancing the comfort and productivity of occupants.
- **Industrial Facilities:** Factories and warehouses utilize the system to maintain optimal working conditions for employees and to safeguard equipment from heat-related damage.
- **Greenhouses:** The system helps control the temperature inside greenhouses, ensuring optimal growing conditions for plants and crops.
- **Data Centers:** Critical infrastructure like data centers require precise temperature control to prevent equipment overheating. The system ensures reliable cooling while minimizing energy consumption.
- **Server Rooms:** Similar to data centers, server rooms benefit from the system's ability to maintain stable temperatures, protecting sensitive equipment and reducing cooling costs.
- **Vehicle Interiors:** Automotive applications involve integrating the system into vehicles to regulate cabin temperature automatically, enhancing passenger comfort.
- **Healthcare Facilities:** Hospitals and clinics utilize the system to maintain hygienic environments and prevent the spread of airborne contaminants.

14. RESULTS AND DISCUSSION

- Upon implementation, the smart temperature-controlled exhaust system successfully demonstrated its ability to regulate ventilation based on temperature variations.
- The system reliably detected changes in temperature and activated the exhaust fan accordingly, ensuring a comfortable indoor environment.
- The integration with the Blynk application allowed users to monitor and adjust settings remotely, enhancing convenience and control.

15. CONCLUSION

- In conclusion, the development of a smart temperature-controlled exhaust system using IoT technology and the Blynk ap plication represents a significant advancement in the field of environmental automation.
- By leveraging sensor data and cloud connectivity, we have created a solution that optimizes ventilation while minimizing energy consumption.
- Future enhancements could involve integrating additional sensors for comprehensive environmental monitoring and exploring advanced control algorithms for further efficiency improvements.

16. REFERENCES

1. S. Das, N. Debabhuti, R. Das, S. Dutta, and A. Ghosh, "Embedded system for home automation using SMS," in Automation, Control, Energy and Systems (ACES), 2014 First International Conference on, 2014, pp. 1-6: IEEE.
2. Mohite, S., Adsule, S., Patil, R., & Dhawas, N. (2020). Automatic Temperature Based Fan Speed Controller Using Arduino. SSRN Electronic Journal. doi:10.2139/ssrn.3645388
3. Conference: 3rd IEEE International Conference on Electrical, Computer and Communication Technologies (IEEE ICECCT 2019)At: SVS College of Engineering, Coimbatore, Tamil Nadu, India – 642 109Volume: 3
4. A. V. Livchak, C. Raczewski, and D. W. Shrock, "Exhaust flow control system and method," ed: Google Patents, 2016.
5. S. K. Melink, "Kitchen exhaust optimal temperature span system and method," ed: Google Patents, 2006.

TEAM MEMBERS: 1.Ashwini S (910021104005)

2.Jayanthi E (910021104016)

3.Keerthiga VNR (910021104018)

4.Kiruthiga K (910021104019)

5.Priyadharshini M (910021104026)

DEPARTMENT: B.E Computer Science and Engineering

YEAR: III rd year

STAFF SIGNATURE **HOD SIGNATURE**