My project, "Enhancing IoT-Powered Environmental Monitoring for Sustainable Healthcare," was influenced by extensive background research focused on the impact of artificial intelligence (AI) in healthcare, the role that the Internet of Things (IoT) plays in real-time data collection, the importance of blockchain data security, and the necessity of sustainability in smart healthcare. The research emphasises the significance of standards and cloud computing in IoT applications and sets the project's objectives while highlighting the potential of AI, IoT, cloud computing, and sensing technologies in enhancing healthcare.

1. Methodology & Tools

The approach, "Enhancing IoT-Powered Environmental Monitoring for Sustainable Healthcare," aims to improve environmental monitoring and maintenance in the healthcare industry while promoting sustainability. This was achieved by utilising various platforms, software tools, and hardware components, which are listed below:

- 1 x Arduino Uno
- 1 x DHT11 Temperature and Humidity sensor
- 1 x Breadboard
- 1 x 16x2 LCD display
- 13 x Breadboard jumper wires
- 6 x F-M wires (Female to Male DuPont wires)
- Potentiometer and
- Resistor
- Tinkercad
- ThingSpeak

These components have been chosen based on their unique characteristics. The central processing unit, the Arduino Uno, makes data collecting and transfer easier. Precise temperature and humidity readings are provided by the DHT11 sensor module which is essential for applications in healthcare. The potentiometer is used to adjust the LCD screen contrast, while the resistor assists in controlling the flow of current to the LCD, providing an easy-to-understand real-time data display. The parts are connected using jumper wires, ensuring smooth seamless system integration.

Furthermore, the Arduino IDE is utilised for programming the Arduino Uno. This allows for the creation of personalised codes for data collection, processing, and transmission. This software option provides the flexibility and customization needed to adjust to the unique requirements of healthcare environments. The Tinkercad platform is utilised for virtual prototyping and simulation to further increase the project's adaptability and accessibility. Tinkercad ensures a more productive and economic development process by facilitating the system's design, testing, and debugging in a virtual environment.

For storing and visualising data, the project has chosen ThingSpeak. A powerful IoT platform that enables real-time monitoring, data logging, and analysis. Thingspeak is an excellent fit for this healthcare industry environmental monitoring project because of its user-friendly interface and integration capabilities.

Considering all these, The combination of Arduino Uno, 16x2 LCD, potentiometer, resistor, jumper wires, Tinkercad, Arduino IDE, and Thingspeak with the DHT11 temperature and humidity sensor provides a strong foundation for enhancing IoT-Powered Environmental Monitoring for Sustainable Healthcare. These hardware and software platform combinations have been carefully chosen for their interoperability, adaptability, and capacity to deliver real-time data insights, thereby promoting a more sustainable and effective healthcare environment.

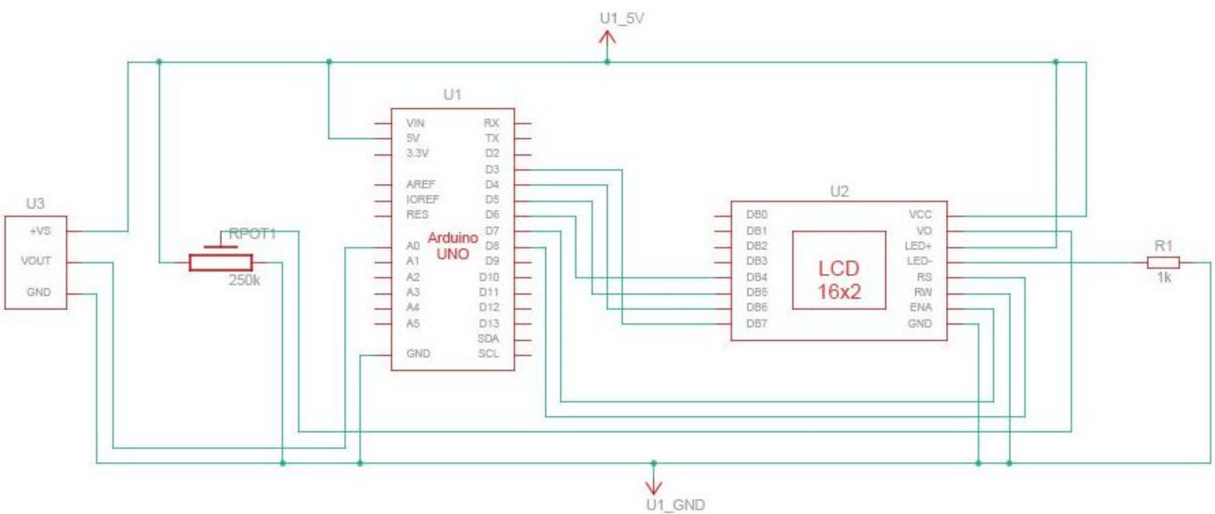
2. Design & Functionality

The integration of Internet of Things (IoT) technology with environmental monitoring is essential in the pursuit of sustainable healthcare solutions, as it ensures the safety and well-being of both patients and healthcare facilities. With the help of this technology, healthcare workers will be able to monitor temperature and humidity as well as other environmental factors, promoting a safer and healthier atmosphere for both employees and patients. To build an IoT system for this purpose, an Arduino Uno board, a DHT11 temperature and humidity sensor, a 16x2 LCD, and the Arduino IDE are used. The platform ThingSpeak is utilised to send and display data. With the system's user interface, Healthcare practitioners can monitor and evaluate environmental conditions in real time. The hardware and software design of the system ensures the goal of enhancing healthcare sustainability through the provision of an effective and user-friendly environmental monitoring solution.

Functionality of the IoT System

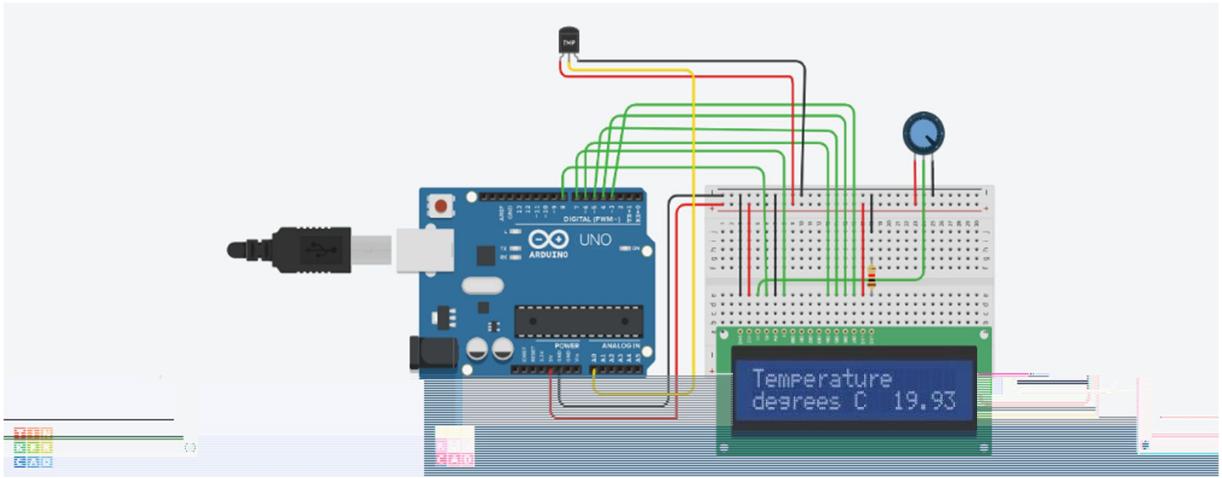
The IoT system is intended to continuously monitor the temperature and humidity in healthcare facilities and give medical personnel real-time data to aid in their decision-making. To maintain the best possible conditions for patient comfort and health, as well as the preservation of medical supplies and equipment. It plays an important role in quickly identifying possible problems, such as high humidity that might encourage the formation of mould or temperature swings that compromise patient comfort.

In healthcare facilities, temperature and humidity are crucial environmental parameters that need to be monitored. The DHT11 sensor is used to measure these parameters, and the gathered data is displayed on a 16x2 LCD that provides a clear and easily readable interface for users inside the facility. Additionally, the Arduino Uno is utilised to send the data to the ThingSpeak platform. With Thingspeak's data visualisation, remote access, and history tracking features, healthcare workers can monitor the conditions in real-time and detect any anomalies or trends with ease. To comprehensively document and showcase the development of the IoT prototype, I have included a detailed systematic diagram in Figure 1, a circuit diagram and code in Figure 2, a physical prototype and the relevant code used in testing the device in Figure 3, and a graphical user interface on thingspeak for remote monitoring and data visualisation in Figure 4.



Name	Quantity	Component
U1	1	Arduino Uno R3
U2	1	LCD 16 x 2
Rpot1	1	250 kΩ Potentiometer
R1	1	1 kΩ Resistor
U3	1	Temperature Sensor [TMP36]

Fig. 1: A systematic diagram of the Virtual Design.




```
#include "LiquidCrystal.h"
LiquidCrystal lcd(8,7,6,5,4,3);
int sensorPin = 0;

void setup()
{
  Serial.begin(9600);
  lcd.begin(16,2);
}

void loop()
{

 int reading = analogRead(sensorPin);

 // measure the 5v with a meter for an accurate value

 float voltage = reading * 4.68;
 voltage /= 1024.0;

 // now print out the temperature

 float temperatureC = (voltage - 0.5) * 100;
 Serial.print(temperatureC);
 Serial.println(" degrees C");

   lcd.setCursor(0,0);
   lcd.print("Temperature ");
   lcd.setCursor(0,1);
   lcd.print("degrees C");
   lcd.setCursor(11,1);
   lcd.print(temperatureC);

 delay(100);
}
```

Fig. 2: A Circuit diagram & code utilised for Virtual simulation.

NB: *To assess the design's functionality within the constraints of available resources, a TMP36 temperature sensor was employed to conduct a virtual evaluation using the Tinkercad platform.*

Hardware Design

An Arduino Uno board, a DHT11 temperature and humidity sensor, a breadboard, a 16x2 LCD, female-to-male DuPont wires, breadboard jumper wires, a potentiometer, and a resistor are among the hardware parts. To achieve the functioning of the IoT system, these components were successfully integrated into the hardware design.

Using a breadboard and jumper wires, the Arduino Uno is linked to the DHT11 sensor, the Temperature and humidity readings are shown in real time through the connection between the Arduino Uno and the 16x2 LCD. For best visibility, the LCD's contrast can be adjusted with the potentiometer, while a resistor was incorporated to shield the DHT11 sensor from possible overvoltage.

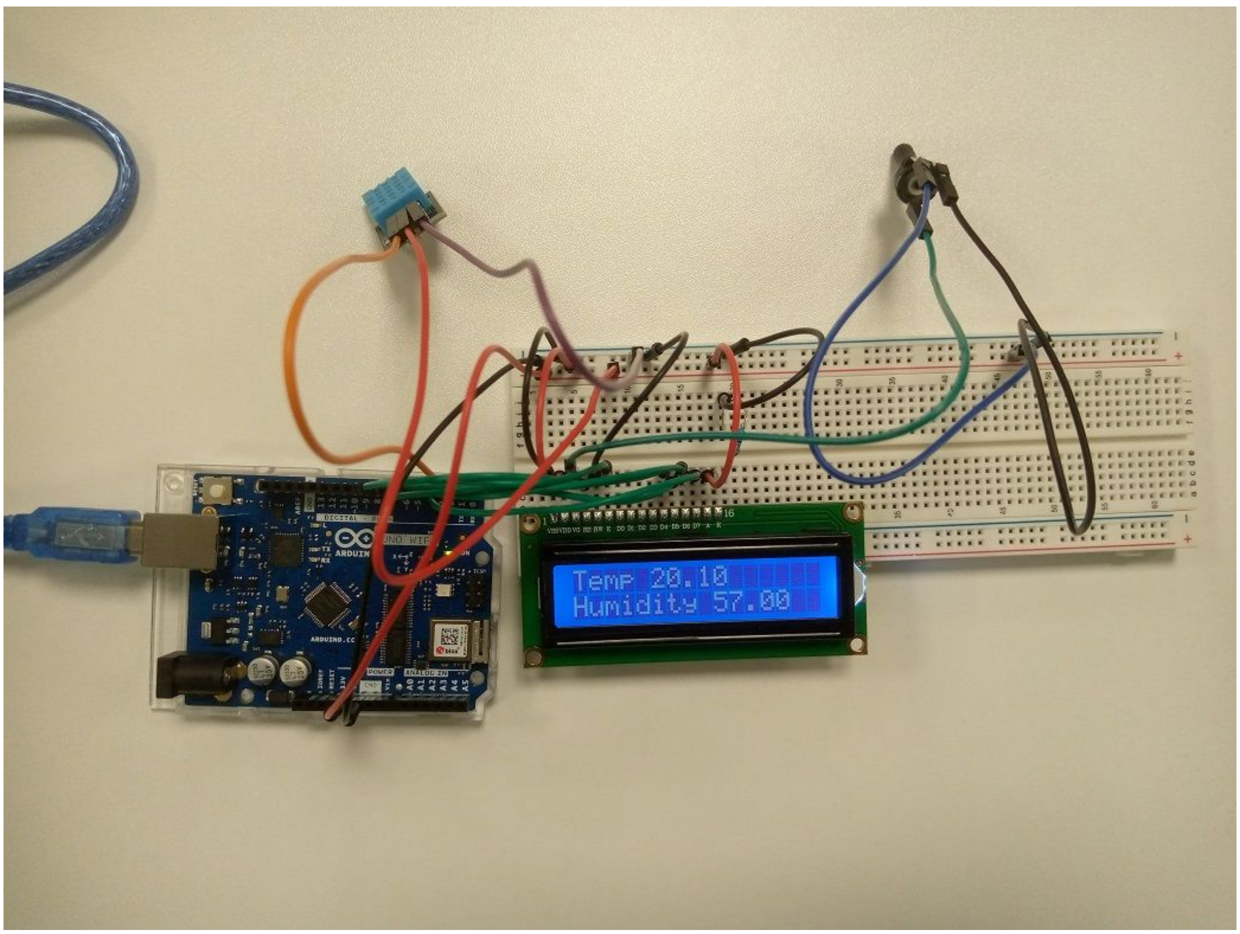


Fig. 3: A Pictorial image of the Prototype device.

Find below the programme code utilised for conducting the physical test.

```
#include <WiFiNINA.h>
#include "DHT.h" //include the DHT11 Library for interfacing with the sensor
#include "LiquidCrystal.h"
#include "ThingSpeak.h" // always include thingspeak header file after otherheader files
LiquidCrystal lcd(8,7,6,5,4,3);

char ssid[] = Infinix HOT 8; // your network SSID (name)
char pass[] = mummy123; // your network password
WiFiClient client;

unsigned long myChannelNumber =  2331344; // Write Channel Number generated from thingspeak website
const char * myWriteAPIKey = WB2G1FTB9U8UP1ZW;  // Write API KEY generated from thingspeak website
int number = 0;

// create an instance of the DHT11 class
// - for Arduino connect the sensor to digital I/0 Pin 2
#define DHTPIN 2

#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);

// Initialize serial communication to allow debugging and data readout
// Using a baud rate of 9600 bps
void setup() {
  Serial.begin(9600);

  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
    }String fv = WiFi.firmwareVersion();
    if (fv != "1.1.0") {Serial.println("Please upgrade the firmware");
    }
    ThingSpeak.begin(client); // Initialize ThingSpeak

  Serial.println(F("DHTxx test!"));
  lcd.begin(16,2);

  dht.begin();
}

void loop() {
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID");
    while(WiFi.status() != WL_CONNECTED){
  WiFi.begin(ssid, pass);
  Serial.print(".");
  delay(5000);

  }
  Serial.println("\nConnected.");
  }
// pieces of information in a channel. Here, we write to field 1 and 2


  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);
```

```
  int x = ThingSpeak.writeField(myChannelNumber, 1, t, myWriteAPIKey);
  int y = ThingSpeak.writeField(myChannelNumber, 2, h, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
    }
    else{
      Serial.println("Problem updating channel. HTTP error code " + String(x));
    }


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

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));

  lcd.setCursor(0,0);
  lcd.print("Temp  ");
  lcd.print(String(t));
  lcd.setCursor(0,1);
  lcd.print("Humidity  ");
  lcd.print(String(h));

}
```

Software Design

ThingSpeak is employed for data visualisation and transmission, while the Arduino Uno IDE is adopted in creating the software design. Data collection from the DHT11 sensor, LCD, and Arduino Uno programming are all done using the Arduino IDE.

The temperature and humidity readings from the DHT11 sensor are read by the Arduino programme. It then uses the ThingSpeak API to deliver this data to ThingSpeak regularly. The ThingSpeak platform allows for the visualisation and analysis of this data, as well as the provision of real-time monitoring features and historical data patterns.

User Interface Design

Healthcare professionals can easily navigate and find it intuitive that the Internet of Things system's user interface is designed with their needs in mind. Personnel on the scene can easily access and understand real-time data through the hardware's 16x2 LCD. The display makes the ambient conditions easily visible at a look by showing both temperature and humidity readings.

The ThingSpeak platform functions not only as an on-site display but also as a remote user interface. Healthcare practitioners can monitor environmental conditions from any location with internet connectivity by using ThingSpeak on their computers or mobile devices. The software lets users see data trends over time using visualisations like graphs and charts, which makes it easier for them to spot patterns and take relevant action.

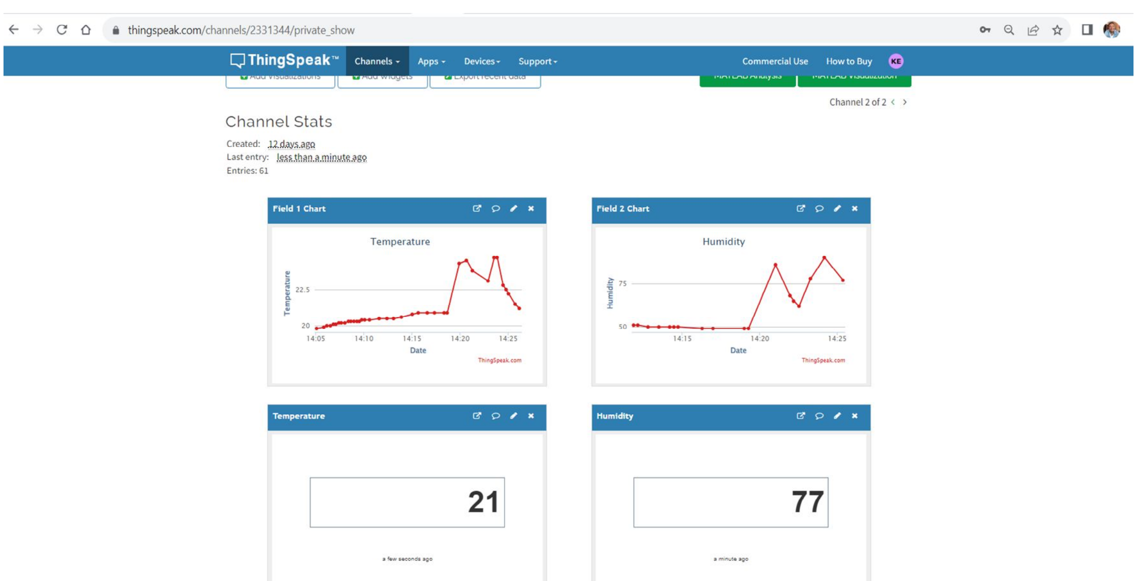


Fig. 4: Graphical user interface on Thingspeak.

Design Justification

To efficiently fulfil the intended aim of enhancing environmental monitoring for sustainable healthcare, the following design decisions were made for this IoT system:

1. Hardware Selection: An affordable and dependable way to monitor temperature and humidity is by using an Arduino Uno board with a DHT11 sensor. The 16x2 LCD panel is easy for on-site staff to use since it provides quick and clear information.

2. Data Visualisation: Comprehensive data analysis and remote access are made possible by the use of ThingSpeak as the data visualisation platform. Healthcare practitioners can make well-informed decisions based on past data trends and real-time condition monitoring.

3. User Interface: The system's accessibility and usability are guaranteed by the combination of an on-site LCD and remote access through ThingSpeak. To maintain a sustainable healthcare environment, healthcare workers can promptly analyse environmental conditions and resolve any difficulties that may arise.

4. Data Accuracy: The resistor that shields the DHT11 sensor from overvoltage guarantees data accuracy and sensor longevity, which are essential for monitoring patient care.

To sum up, the thoughtfully designed hardware and software of the Internet of Things system offers a practical way to enhance environmental monitoring in hospital environments. With the help of prompt remedial actions and temperature and humidity monitoring, it helps healthcare personnel maintain sustainable healthcare environments. The system's accessibility is guaranteed by its user-friendly interface, which benefits patients' general health as well as the long-term viability of healthcare facilities.

3. Testing

As part of the research project Enhancing IoT-Powered Environmental Monitoring for Sustainable Healthcare, extensive testing protocols were used to confirm that the IoT sensors and devices functioned as intended under a variety of challenging circumstances. Verifying that the IoT devices and sensors were operating in line with their specifications was the main goal of these tests. Environmental data collection and transmission were made possible in large part by the hardware components used in this project, which included the Arduino Uno, DHT11 temperature and humidity sensor, breadboard, 16x2 LCD for on-site display, breadboard jumper wires, female-to-male DuPont wires, potentiometer, and resistor.

The testing procedure covered several areas. First, the code that controlled how the IoT devices operated was programmed using the Arduino IDE. The code was meticulously developed and refined to ensure it accurately collected data from the DHT11 sensor and then transmitted to the cloud service. To verify the sensor data collecting and transmission functions, extensive unit testing of the code on the Arduino Uno was performed.

To further guarantee that the IoT sensors and devices functioned dependably in a range of scenarios that mimicked actual healthcare settings, extensive environmental testing was carried out. This involved evaluating the system's operation in a variety of temperature and humidity ranges, as well as in situations where there would be fast changes in the environment and difficulties with data transmission.

Lastly, the ThingSpeak platform was adopted for user interface and data presentation verification, extensive environmental testing was conducted to ensure the system's resilience in a variety of healthcare settings, and code validation and unit testing were part of the testing procedures for the IoT-powered environmental monitoring system. These comprehensive tests guaranteed that the system's operation satisfied the requirements of medical professionals and helped create an environmentally friendly monitoring solution that is both long-lasting and efficient.

4. Conclusion

In general, the IoT system appears to have met some of the requirements. however, there are strengths and weaknesses to take into account.

To begin with, one of the best aspects of this IoT system is its capacity to effectively collect and transmit environmental data, especially temperature and humidity, which are important to applications in the healthcare industry. The DHT11 sensor module and Arduino Uno were affordable and easily obtainable options that could be applied to several healthcare situations. Real-time feedback was supplied with the inclusion of a 16x2 LCD for on-site display, which improved the system's usability. Also, the ThingSpeak platform made data visualisation and remote monitoring possible, enabling stakeholders and healthcare experts to access it.

But there are a few shortcomings to take into account. Although inexpensive, the DHT11 sensor module could not offer the best precision and accuracy in data readings. It is vital to think about switching to a more precise sensor to remedy this. Also, the Arduino Uno is a capable microcontroller but may have limitations in terms of scalability and data processing for more extensive healthcare applications. It is advisable to explore more potent microcontrollers or microprocessors to handle more datasets and more complicated jobs to overcome this limitation. Additionally, while ThingSpeak provides a convenient remote user interface, compared to more specialised IoT systems, it may not offer the same level of data security, privacy, customisation, or advanced data analytics tools. To overcome this limitation, it's important to assess and improve the security measures to safeguard sensitive healthcare data by looking into other secure platforms or creating a unique web interface.

In conclusion, the IoT solution provided real-time data gathering and remote access through ThingSpeak, satisfying the fundamental needs for environmental monitoring in the healthcare industry. To make the system better, it is imperative to think about boosting data security to secure patient information, improving processing power for scalability, and updating the sensor for more accuracy. These enhancements would help create a more durable and dependable system for long-term environmental monitoring of healthcare.