Internet of Things and Cloud Computing
2025; 13(2): 28-37
http://www.sciencepublishinggroup.com/j/iotcc
doi: 10.11648/j.iotcc.20251302.11
ISSN: 2376-7715 (Print); ISSN: 2376-7731 (Online)

Development of a IoT Based Thermologger for Real-Time Temperature Monitoring

Md Hafizur Rahman[1, 2, *], Md Masud Reza[3], Rebina Ferdous[4], Naderuzzaman[5], Mohammod Abul Kashem[5]

[1]ICT Cell, Jagannath University, Dhaka, Bangladesh
[2]BhuiyanSoft, Dhaka, Bangladesh
[3]Genome Research Center, Bangladesh Jute Research Institute, Dhaka, Bangladesh
[4]Department of Computer Science and Engineering, The Peoples University of Bangladesh, Dhaka, Bangladesh
[5]Department of Computer Science and Engineering, Dhaka University of Engineering and Technology, Gazipur, Bangladesh

Email address:

hafizurfpbd@gmail.com (Md Hafizur Rahman)
[*]Corresponding author

To cite this article:

Md Hafizur Rahman, Md Masud Reza, Rebina Ferdous, Naderuzzaman, Mohammod Abul Kashem. (2025). Development of a IoT Based Thermologger for Real-Time Temperature Monitoring. *Internet of Things and Cloud Computing, 13*(2), 28-37. https://doi.org/10.11648/j.iotcc.20251302.11

Received: 13 May 2025; **Accepted:** 30 May 2025; **Published:** 21 June 2025

Abstract: The temperature-sensitive industries including healthcare, agriculture and cold chain logistics the Internet of Things (IoT) has greatly increased monitoring and management of environmental conditions. Minor temperature fluctuations can lead to the deterioration of products, diminished effectiveness of pharmaceuticals, or suboptimal agricultural results. This work presents the design and development of a thermologger system for real-time temperature monitoring and data recording system based on the Internet of Things (IoT). The Internet of Things (IoT) has emerged as a revolutionary solution, facilitating real-time monitoring, sophisticated analysis and automated decision-making across several sectors. The system incorporates low-power digital temperature sensors, a microcontroller unit (MCU) and a wireless communication module based on the ESP-8266. These components work together to collect temperature data and transmit it to a cloud platform for storage and analysis. A key feature of the system is its user-friendly interface, available through a mobile app and a web dashboard. These platforms enable users to view temperature trends, receive alerts when temperatures fall outside of safe ranges and generate reports for further analysis. The alert mechanism is especially useful in high-risk areas such as vaccine storage, greenhouse operations, large industrial cold storage, datacenter and food transportation where timely intervention can prevent significant loss. Real-world testing demonstrates the system's accuracy, responsiveness, and dependability. These findings validate the system's efficacy and affordability as a continuous temperature monitoring solution for critical applications. All things considered the proposed thermologger system may enhance operational decision-making, boost safety and optimize resource use across a range of industries.

Keywords: Thermologger, IoT, Cloud, MVC, Database, Analysis, Temperature

1. Introduction

The Internet of Things (IoT) has revolutionized the monitoring and management of environmental parameters, particularly in temperature sensitive domains such as healthcare, agricultureand cold chain logistics [1]. This paper presents the design and development of an IoT-based thermologger system aimed at real-time temperature monitoring and data logging. The proposed system integrates low-power sensors with a Microcontroller unit and wireless communication modules to capture and transmit temperature data to a cloud platform for storage and analysis [1].

1.1. Internet of Things

The Internet of Things (IoT) refers to a vast network of interconnected physical devices embedded with sensors, software, and communication technologies that collect, exchange, and analyze data over the internet. These smart devices¡ªranging from household appliances and wearable gadgets to industrial machines and environmental sensors¡ªenable seamless automation, real-time monitoring, and data-driven decision-making [2]. IoT relies on key technologies such as wireless connectivity (Wi-Fi, Bluetooth, 5G), cloud computing, and artificial intelligence to process and act on the gathered information. By bridging the physical and digital worlds, IoT enhances efficiency, reduces costs, and improves quality of life in applications like smart homes, healthcare, agriculture, and smart cities. However, challenges such as data security, privacy concerns, and interoperability must be addressed to fully realize IoT's potential in transforming industries and everyday life [1–3].

1.2. 5-Tier Architecture of IoT

The 5-tier IoT architecture provides a structured framework for designing and implementing IoT systems, ensuring efficient data flow from physical devices to end-user applications [4].

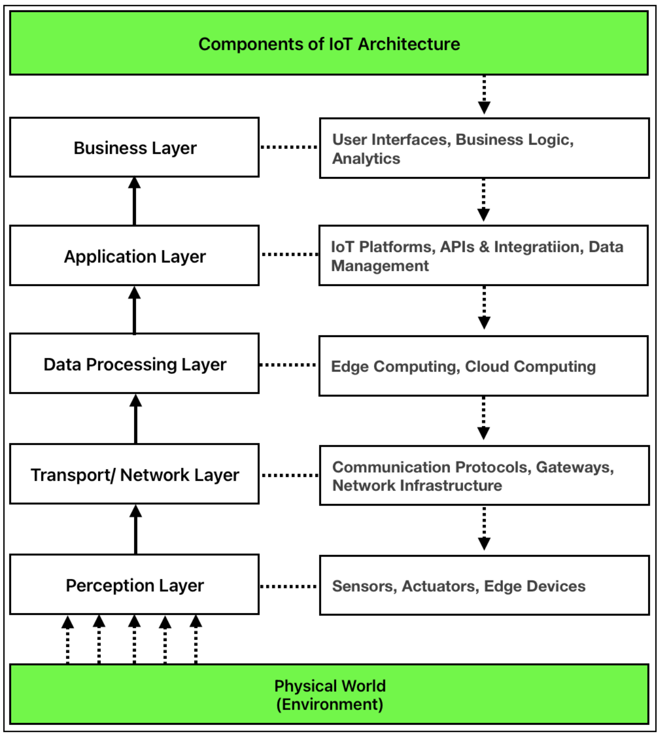

Figure 1. *5-Tier Architecture of IoT.*

1.2.1. Perception Layer(Sensors/Actuators)

The Perception Layer is the lowest and most physical layer in IoT architecture, acting as the bridge between the digital and physical worlds. It consists of sensors that collect real-world data and actuators that perform physical actions based on digital commands.

1.2.2. Transport/Network Layer

The Transport/Network Layer is the backbone of IoT communication, responsible for secure and reliable data transmission between devices (Perception Layer) and cloud/processing systems.

1.2.3. Data Processing Layer (Edge/Cloud Computing)

The Data Processing Layer is where raw sensor data is transformed into actionable insights. It spans edge devices (local processing) and cloud platforms (centralized processing), balancing speed, cost, and scalability.

1.2.4. Application Layer (Middleware)

the Application Layer and the Middleware Layer serve two distinct yet complementary functions. The Middleware Layer, often referred to as the Data Processing Layer, acts as the intelligent bridge between the lower layers (perception and network) and the upper layers. It is responsible for data filtering, aggregation, storage, and real-time analytics. Technologies such as cloud computing, edge/fog computing, and database systems operate at this layer to ensure that only relevant and actionable data progresses further up the stack. Above it, the Application Layer translates processed data into meaningful services for end-users. It serves as the interface between the IoT system and human users, providing domain-specific applications in areas like smart homes, healthcare, industrial monitoring, and agriculture. This layer ensures that insights derived from sensor data are accessible, understandable, and useful for decision-making, often through mobile apps, web dashboards or integrated systems [5].

1.2.5. Business Layer (User/Decision Layer)

The Business Layer, also known as the User or Decision Layer, is the topmost tier in the five-tier IoT architecture and plays a critical role in strategic decision-making and policy enforcement. This layer interprets the outcomes and insights generated by the application and data processing layers to support business intelligence, long-term planning, and user-specific actions. It integrates analytics with business goals, enabling organizations to derive value from IoT data through actionable insights. Functions at this layer include trend analysis, KPI monitoring, automated decision support, and the implementation of rules or workflows based on user preferences or enterprise policies. In user-centric systems, this layer may provide feedback to users, allow configuration of services, or adapt system behavior based on usage patterns. By bridging technological capabilities with real-world business or user objectives, the Business Layer ensures that IoT systems are not only functional but also contextually intelligent and goal-oriented.

1.3. Thermologger

A Thermologger is a specialized device used to monitor and record temperature data over time, commonly employed in industries such as food storage, pharmaceuticals, healthcare, and environmental research. Equipped with high-precision sensors, it captures temperature readings at set intervals and stores the data for analysis, ensuring compliance with safety standards and quality control. Many Thermologgers feature real-time monitoring, alarms for temperature deviations, and wireless connectivity for remote access. They are essential for maintaining optimal conditions in sensitive environments, preventing spoilage, and ensuring regulatory compliance. With durable designs and long battery life, Thermologgers provide reliable, accurate, and continuous temperature tracking for various applications.

1.4. Temperature Monitoring

Temperature Monitoring is the process of measuring and tracking temperature variations in various environments to ensure optimal conditions, safety, and compliance with industry standards. It is widely used in industries such as healthcare (for vaccine storage), food and beverage (to prevent spoilage), pharmaceuticals (for drug stability), and manufacturing (for equipment efficiency). Advanced temperature monitoring systems utilize digital sensors, data loggers, and IoT-enabled devices to provide real-time alerts, historical data analysis, and remote access via cloud platforms. These systems help prevent product degradation, reduce energy consumption, and maintain regulatory compliance. With the integration of AI and automation, modern temperature monitoring solutions offer greater accuracy, predictive analytics, and seamless integration with other smart systems, making them indispensable for quality control and operational efficiency [8].

1.5. Temparature Analysis

Temperature Analysis involves the systematic examination of temperature data to identify patterns, trends, and anomalies for informed decision-making. Using sensors, data loggers, and IoT devices, temperature measurements are collected over time and analyzed through statistical methods, machine learning, or visualization tools like graphs and heatmaps. This process is critical in industries such as healthcare (monitoring patient conditions or vaccine storage), food safety (ensuring proper refrigeration), and manufacturing (optimizing equipment performance). Advanced temperature analysis can predict potential failures, optimize energy usage, and ensure compliance with regulatory standards. By leveraging real-time analytics and AI-driven insights, businesses and researchers can enhance efficiency, reduce risks and maintain optimal environmental conditions [8].

1.6. Cloud Applications

Cloud Applications for Temperature Monitoring leverage cloud computing to provide real-time, scalable, and remote temperature tracking across various industries. These applications collect data from IoT-enabled sensors, wireless data loggers, and smart devices, then securely transmit it to cloud platforms for storage and analysis. Users can access dashboards from anywhere, visualize trends through graphs and alerts, and receive instant notifications if temperatures deviate from predefined thresholds. Industries like pharmaceuticals (for cold chain compliance), food storage (to prevent spoilage), and HVAC systems (for energy efficiency) benefit from cloud-based solutions, which eliminate the need for manual checks and ensure data integrity. Advanced features like AI-driven predictive analytics, automated reporting, and integration with other enterprise systems further enhance operational efficiency and regulatory compliance. By centralizing temperature data in the cloud, these applications enable proactive decision-making, reduce risks, and optimize resource management [9].

1.7. Applications

1.7.1. Healthcare and Hospitals

Thermologgers are essential IoT-enabled devices used in healthcare and hospital environments to continuously monitor and record temperature data with high precision. In clinical settings, they play a critical role in maintaining the integrity of temperature-sensitive assets such as vaccines, blood bags, medications, and organ transplants, where even minor deviations can compromise safety and efficacy. These devices are typically equipped with sensors and wireless communication modules that enable real-time data transmission to centralized monitoring systems. Integrated with cloud platforms or hospital management software, thermologgers provide automated alerts when temperature thresholds are breached, allowing for immediate corrective actions. Their use not only enhances regulatory compliance and patient safety but also reduces manual monitoring efforts, streamlines operations, and ensures the quality of stored medical supplies. In the broader IoT framework, thermologgers represent the synergy between environmental sensing and intelligent healthcare infrastructure.

1.7.2. Smart Homes/Buildings

Thermologgers serve as crucial components for enhancing comfort, energy efficiency, and safety. These IoT-enabled devices continuously monitor ambient temperature across different zones within a structure and can communicate data to a centralized smart home management system. By integrating with HVAC (Heating, Ventilation, and Air Conditioning) systems, thermologgers allow for real-time temperature adjustments based on user preferences, occupancy, and environmental conditions, thus optimizing energy consumption. Additionally, they can detect unusual thermal patterns-such as sudden spikes due to appliance malfunctions or fire risks¡ªand trigger alerts for preventive action. When paired with cloud services and mobile apps, residents gain remote visibility and control over home temperature conditions. As part of the broader smart infrastructure, thermologgers contribute to sustainability goals, personalized living environments, and automated building intelligence.

1.7.3. Environmental and Weather Monitoring

thermologgers play a vital role in collecting accurate and continuous temperature data from natural surroundings. Deployed in outdoor or remote locations, these IoT-enabled devices are designed to operate under various climatic conditions, often powered by solar energy or low-power technologies for long-term deployment. Thermologgers help monitor temperature fluctuations in ecosystems, forests, oceans, agricultural fields, and urban environments to study climate change, microclimates, and weather patterns. When integrated with wireless communication technologies such as LoRa, NB-IoT, or satellite links, they can transmit real-time data to centralized platforms for analysis and visualization. Environmental scientists and meteorologists use this data to predict weather events, manage disaster response, and conduct climate research. As part of an IoT-based sensing network, thermologgers contribute to a deeper understanding of environmental dynamics and support data-driven policy making for sustainable development.

1.7.4. Agriculture and Greenhouses

In the agricultural sector and greenhouse environments, thermologgers are essential IoT devices used to maintain optimal temperature conditions for crop growth and productivity. These sensors continuously monitor the ambient and soil temperatures, enabling farmers and greenhouse operators to track microclimatic variations that directly impact plant health, germination, and yield. By integrating thermologgers with automated climate control systems¡ªsuch as ventilation, heating, and irrigation¡ªfarmers can ensure stable environmental conditions, reduce the risk of disease, and enhance resource efficiency. In precision agriculture, thermologgers contribute to data-driven decision-making by providing historical and real-time temperature data that help optimize planting schedules, fertilization, and pest control strategies. Furthermore, cloud-based platforms allow remote monitoring and alerting, making it possible to react promptly to sudden temperature changes. As part of a broader smart farming ecosystem, thermologgers support sustainability, improved crop management, and higher agricultural outputs.

1.7.5. Industrial and Manufacturing

In industrial and manufacturing environments, thermologgers are critical IoT devices used to monitor and control temperature-sensitive processes, ensuring operational efficiency, safety, and product quality. These devices are deployed across various stages of production, storage, and transportation to track temperature conditions in real time. In industries such as food processing, pharmaceuticals, chemical manufacturing, and electronics, maintaining specific temperature thresholds is essential to meet regulatory standards and avoid product degradation or system failure. Thermologgers can be integrated into industrial control systems and SCADA (Supervisory Control and Data Acquisition) platforms to provide automated alerts, trigger safety mechanisms, or adjust process parameters dynamically. Their data can also be stored and analyzed for predictive maintenance, helping to detect overheating equipment, prevent breakdowns, and optimize energy consumption. As part of an Industrial IoT (IIoT) framework, thermologgers support data-driven decision-making, process transparency, and smarter factory automation.

1.7.6. Data Centers

In modern data centers, thermologgers are vital IoT tools used to ensure optimal thermal conditions for servers and critical IT infrastructure. As data centers operate continuously with high-density computing equipment, precise temperature monitoring is essential to prevent overheating, reduce energy waste, and maintain system uptime. Thermologgers are strategically deployed across server racks, cooling systems, and air flow zones to detect hot spots, monitor temperature gradients, and validate the performance of HVAC systems. Real-time data collected by thermologgers can be integrated with data center infrastructure management (DCIM) platforms, enabling automated cooling adjustments, load balancing, and predictive maintenance. By providing granular thermal insights, these devices help optimize energy efficiency through intelligent cooling strategies, reduce operational costs, and extend equipment lifespan. In the context of green IT and sustainable infrastructure, thermologgers play a crucial role in minimizing environmental impact while maintaining high-performance computing environments.

2. System Architecture

The system architecture (Figure 2) of the smart IoT thermologger for real-time temperature monitoring is designed to ensure efficient data acquisition, transmission, storage, and visualization.

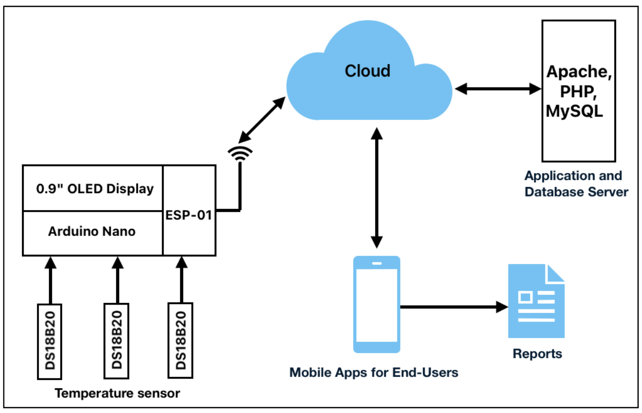

Figure 2. *System Diagram.*

2.1. System Design

At its core, the system comprises temperature sensors (such as DS18B20) connected to a microcontroller unit (Arduino Nano), which acts as the central processing hub. The microcontroller reads temperature data at defined intervals and uses ESP-8266 Wi-Fi module to transmit the data to a cloud server or IoT platform such as customize cloud server. The cloud server is responsible for storing the incoming data and offering APIs for real-time visualization and analysis.

2.2. Hardware Components

ESP-8266: The ESP-8266 (Figure 3) is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability, developed by Espressif Systems. Widely used in Internet of Things (IoT) applications, it enables devices to connect to Wi-Fi networks and transmit or receive data without needing a separate processor.

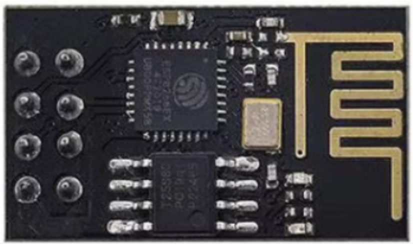

Figure 3. *ESP-8266.*

Arduino Nano: The Arduino Nano (Figure 4) is a compact, breadboard-friendly microcontroller board based on the ATmega328P, widely appreciated for its small size and versatility. Despite its tiny footprint, it offers the same core functionality as the larger Arduino Uno, including 14 digital input/output pins, 8 analog inputs, and support for serial communication.

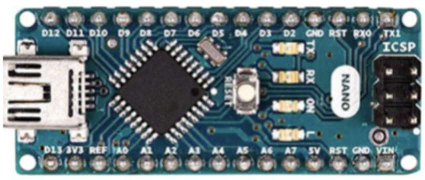

Figure 4. *Arduino Nano.*

OLED Display: The 0.9 inche OLED display (Figure 5) is a small, high-contrast screen commonly used in compact electronics projects for displaying text, graphics, or sensor data. Utilizing organic light-emitting diode (OLED) technology, it delivers crisp visuals with wide viewing angles and excellent brightness, even in low-light conditions. Typically offering a resolution of 128x64 pixels

Figure 5. *0.9 OLED Dispaly.*

Temperature Sensor: The DS1820 (Figure 6) is a digital temperature sensor known for its accuracy, ease of use, and compatibility with a wide range of microcontrollers. It communicates via the 1-Wire protocol, allowing multiple sensors to be connected using just one data pin, which simplifies wiring in complex systems. The sensor can measure temperatures ranging from -55°C to +125°C with a precision of ± 0.5°C in the most commonly used range.

Figure 6. *Temperature Sensor.*

10K ohm Registor: A 10K ohm resistor (Figure 7) is a common passive electronic component that limits current flow and divides voltage in circuits. Typical applications include pull-up or pull-down resistors, sensor circuits, and voltage regulation setups.

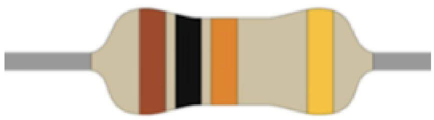

Figure 7. *10K ohm Registor.*

2.3. Software Components

The various software components used in the system are summarized in Table 1. These components represent the essential building blocks of the software architecture, each responsible for a distinct functionality.

Table 1. Software Components.

S.N.	Title	Specification
1	Arduino IDE	1.8.19
2	Hardware Programming Language: C	C++11
3	Web Programming Language: PHP	8.4
4	MVC Framework: Codeigniter	v3.1.12
5	Database: MySQL	8.0
6	Scripting Language: HTML	5
7	Frontend toolkit: Bootstrap	5
8	Webserver: Apache	2.4.X

3. Implementation

The smart IoT thermologger system integrates hardware, embedded programming, web development, API communication, database management, and data analysis.

3.1. Hardware Setup:

The core hardware includes (Figure 8) an Arduino Nano and ESP-8266 microcontroller and a DS18B20 temperature sensor with OLED dispaly.

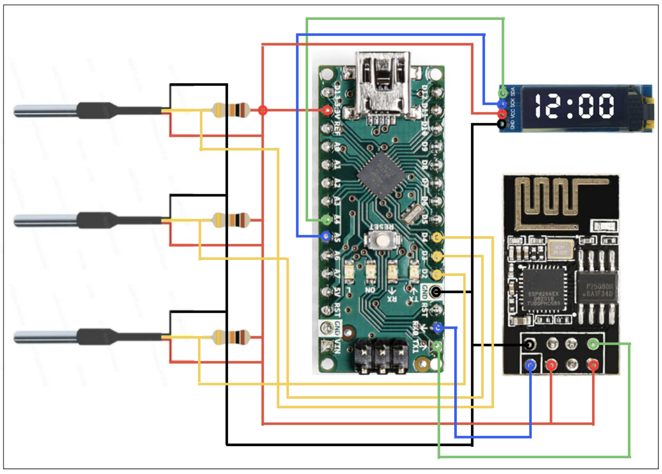

Figure 8. Connection Diagram.

3.2. Coding for Devices

Firmware was developed in C++ using the Arduino IDE to read sensor data and transmit it over Wi-Fi. Data is sent via HTTP to a cloud-based API, which interfaces with a backend server.

```
[fontsize=\small,frame=lines, linenos, breaklines=true,
  breakanywhere=true,  label={C Code for Arduino Nano}]{C}
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 oled_display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

#define ONE_WIRE_BUS 2
#define ONE_WIRE_BUS_two 3
#define ONE_WIRE_BUS_three 4

OneWire oneWire(ONE_WIRE_BUS);
OneWire oneWire2(ONE_WIRE_BUS_two);
OneWire oneWire3(ONE_WIRE_BUS_three);

DallasTemperature sensors(&oneWire);
DallasTemperature sensors2(&oneWire2);
DallasTemperature sensors3(&oneWire3);

void setup(void)
{
  Serial.begin(9600);
  oled_display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  oled_display.clearDisplay();
  oled_display.setTextSize(2);
  oled_display.setTextColor(SSD1306_WHITE);
  oled_display.setCursor(0,1);
```

```
  oled_display.println("IoT Based Thermologger");
  oled_display.display();

  sensors.begin();
  sensors2.begin();
  sensors3.begin();
}

void loop(void){
  sensors.requestTemperatures();
  sensors2.requestTemperatures();
  sensors3.requestTemperatures();
  delay(2000);
  float tempC = sensors.getTempCByIndex(0);
  float tempC2 = sensors2.getTempCByIndex(0);
  float tempC3 = sensors3.getTempCByIndex(0);
  Serial.print(tempC);
  Serial.print(",");
  Serial.print(tempC2);
  Serial.print(",");
  Serial.print(tempC3);
  Serial.print("\n");

  oled_display.clearDisplay();
  oled_display.setCursor(0,1);
  oled_display.print("S-1: ");
  oled_display.print(tempC);
  oled_display.setCursor(0,22);
  oled_display.print("S-2: ");
  oled_display.print(tempC2);
  oled_display.setCursor(0,43);
  oled_display.print("S-3: ");
  oled_display.print(tempC3);
  oled_display.display();
}

[fontsize=\small,frame=lines, linenos, breaklines=true,
  breakanywhere=true,  label={C Code for ESP-8266}]{C}
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <ESP8266HTTPClient.h>

const char* ssid = "xxxxxx";
const char* password = "********";
const char* host = "https://thermal.iot-apps.org/";
const char* api = "********";
unsigned long lasttime = 0;
const unsigned long interval = 10000;

void setup() {
  Serial.begin(9600);
  delay(100);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}
```

```
void loop() {
  if (Serial.available()) {
    String data = Serial.readStringUntil('E');
    unsigned long currenttime = millis();
    if (currenttime - lasttime >= interval) {
      lasttime = currenttime;

      if (WiFi.status() == WL_CONNECTED) {
        WiFiClientSecure client;
        client.setInsecure();
        HTTPClient http;

        http.begin(client, host+"/" + api +"/" + data);
        int httpCode = http.GET();
        if (httpCode > 0) {
          String response = http.getString();
          Serial.println("Response: " + response);
        } else {
          Serial.println("GET failed, error: " + http.errorToString(httpCode));
        }
        http.end();
      } else {
        WiFi.begin(ssid, password);
      }
    }
  }
}
```

3.3. Coding for Cloud Integration

The backend (Cloud Integration), developed micro-service base API using PHP [10] and Codeigniter MVC Framework (Figure 9) [11], data stores in a MySQL database (Table schema: Figure 10).

```
[fontsize=\small,frame=lines, linenos, breaklines=true,
  breakanywhere=true,label={Codeigniter Code for API}]{php}
  <?php
    class Senddata extends CI_Controller {
    public function __construct(){
        parent::__construct();
}
public function home($key=NULL,$data=NULL){
    try{
    $sensor_data=explode("X",$data);
        $this->db->insert('sensors_data', array('device_id'=>$key,
        'thermal_sensor_one'=>$sensor_data[0], 'thermal_sensor_two'=>$sensor_data[1],
        'thermal_sensor_three'=>$sensor_data[2], 'status'=>1,
        'entry_date'=>DATE('Y-m-d H:i:s') ));
        echo "OK";
    }catch(Exception $e){
        echo "ER";
    }
}
}
?>
```

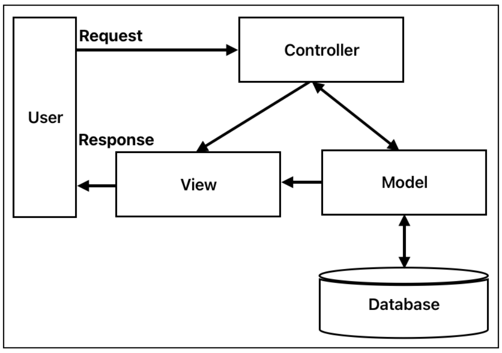

Figure 9. *MVC Framework.*

Field	Type	Null	Key	Default	Extra
slno	bigint(20)	NO	PRI	*NULL*	auto_increment
device_id	varchar(20)	YES		*NULL*	
thermal_sensor_one	double	YES		*NULL*	
thermal_sensor_two	double	YES		*NULL*	
thermal_sensor_three	double	YES		*NULL*	
entry_date	datetime	YES		*NULL*	
status	int(11)	YES		*NULL*	

Figure 10. *Table Schema: sensor data.*

3.4. Web Application Develop for Users

A responsive web dashboard displays real-time and historical temperature data using Python Programming Language for analysis. The system enables remote monitoring, trend analysis, and alert generation.

3.5. Environmental Integration (Deploy)

The entire device is embedded within a living plant setup (Figure 11), likely as a demonstration of indoor environmental monitoring or bio-integrated sensing. This not only shows the system¡¯s aesthetic adaptability but also its application in smart agriculture or home automation contexts.

Figure 11. *Application.*

4. Data Analysis

This descriptive analysis presents the statistical summary (Table 2) of temperature readings collected by a smart IoT thermologger equipped with three thermal sensors (Sensor-1, Sensor-2, Sensor-3). Each sensor recorded 3969 data points.

Time Series Analysis of IoT Thermologger Temperature Data Figure 12 displays the temperature readings from three thermal sensors (Sensor-1, Sensor-2, Sensor-3) over a continuous period from 2025-05-02 09:15:03 to 2025-05-03 07:47:26. Each sensor recorded approximately 3969 data points, as previously summarized.

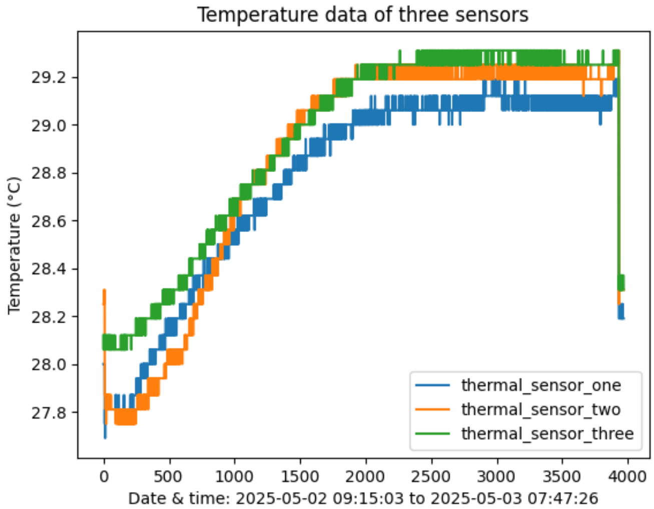

Figure 12. *Time series data.*

Table 2. *Descriptive Statistics of Thermal Sensors.*

S.N.	Status	Sensor-1	Sensor-2	Sensor-3
1	Count	3969	3969	3969
2	Min	27.69	27.75	28.06
3	Max	29.19	29.31	29.31
4	Mean	28.770746	28.859108	28.942046
6	Geometric Mean	28.767671	28.854701	28.939031
8	Median	29.0	29.19	29.19
12	Mode	29.06	29.19	29.25
14	Quantiles(0.25)	28.5	28.56	28.62
15	Quantiles(0.75)	29.06	29.19	29.25
16	Population standard deviation	0.418417	0.501097	0.415917
17	Population variance	0.175073	0.251098	0.172987
18	Sample standard deviation	0.418470	0.501160	0.415970
19	Sample variance	0.175117	0.251162	0.173031
20	Skewness	-1.061725	-1.09937	-0.898367
21	Kurtosis	-0.246811	-0.363386	-0.710792
22	Standard error of the mean	0.006642	0.007954	0.006602
23	Tie correction factor	0.971398	0.955801	0.963242

5. Conclusion

Developing a smart IoT thermologger is a multidisciplinary project involving electronics, embedded systems, wireless communication, cloud computing, and data analytics. By leveraging low-cost hardware and scalable IoT platforms, a highly functional and real-time temperature monitoring solution can be realized. This kind of innovation holds immense potential in automating temperature-sensitive processes, improving compliance, and ensuring product safety across industries.

ORCID

0009-0006-0517-9056 (Md Hafizur Rahman)
0009-0007-0015-7007 (Md. Masud Reza)
0009-0000-5660-6990 (Rebina Ferdous)
0009-0004-0180-8703 (Naderuzzaman)
0009-0009-2055-6050 (Mohammod Abul Kashem)

Abbreviations

IoT	Internet of Things
MCU	Microcontroller Unit
MVC	Model View Controller
API	Application Programming Interface
HVAC	Heating, Ventilation and Air Conditioning
IDCIM	Integrated with Data Center Infrastructure Managemen
SCADA	Supervisory Control and Data Acquisition
TCP/IP	Transmission Control Protocol/Internet Protocol
OLED	Organic Light-Emitting Diode

Conflicts of Interest

This work has no conflicts of interest in terms of financial and interpersonal from any of the authors.

References

[1] Botta, Alessio, Walter De Donato, Valerio Persico, and Antonio Pescapé. "Integration of cloud computing and internet of things: a survey." Future generation computer systems 56 (2016): 684-700. https://doi.org/10.1016/j.future.2015.09.021

[2] Ahmad, Awais, Sadia Din, Anand Paul, Gwanggil Jeon, Moayad Aloqaily, and Mudassar Ahmad. "Real-time route planning and data dissemination for urban scenarios using the Internet of Things." IEEE Wireless Communications 26, no. 6 (2019): 50-55. https://doi.org/10.1109/MWC.001.1900151

[3] Samad, Abdul, and Elisha Blessing. 2025. "Beyond Boundaries: Securing the Iot Ecosystem with Ai-driven Firewalls." OSF Preprints. March 5. https://doi.org/10.31219/osf.io/x38rp_v1

[4] Qureshi, K. N., & Newe, T. (Eds.). (2024). Artificial Intelligence of Things (AIoT): New Standards, Technologies and Communication Systems (1st ed.). CRC Press. https://doi.org/10.1201/9781003430018

[5] Udayakumar, P., Anandan, R. (2024). Get Started with IoT. In: Design and Deploy Microsoft Defender for IoT. Apress, Berkeley, CA. https://doi.org/10.1007/979-8-8688-0239-3_1

[6] Rani, S. (Ed.). (2024). Emerging Technologies and the Application of WSN and IoT: Smart Surveillance, Public Security, and Safety Challenges (1st ed.). CRC Press. https://doi.org/10.1201/9781003438205

[7] Idoko, John Bush, and Rahib Abiyev. "Machine Learning and the Internet of Things in Education." Studies in Computational Intelligence 1115 (2023). https://doi.org/10.1007/978-3-031-42924-8

[8] Subahi, Ahmad F., and Kheir Eddine Bouazza. "An intelligent IoT-based system design for controlling and monitoring greenhouse temperature." IEEE Access 8 (2020): 125488-125500. https://doi.org/10.1109/ACCESS.2020.3007955

[9] Fehling, Christoph, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. Cloud computing patterns: fundamentals to design, build, and manage cloud applications. Vol. 545. Berlin: Springer, 2014. https://doi.org/10.1007/978-3-7091-1568-8

[10] M. H. Rahman, M. Naderuzzaman, M. A. Kashem, B. M. Salahuddin, Z. Mahmud, "Comparative Study: Performance of MVC Frameworks on RDBMS", International Journal of Information Technology and Computer Science(IJITCS), Vol. 16, No. 1, pp. 26-34, 2024. https://doi.org/10.5815/ijitcs.2024.01.03

[11] Rahman, Md Hafizur, Faisal Bin Al Abid, M. Naderuzzaman, Md Arifur Rahman, and Md Masud Reza. "Optimizing and Enhancing Performance of MVC Architecture based on Data Clustering Technique." International Journal of Computer Applications 975: 8887. https://doi.org/10.5120/ijca2016908099