



Faculty of Computer Science and Information Technology

Smart Hospital Room Environmental Monitoring: A Real-Time IoT Solution for Patient Care Enhancement

Izzan Hazirah Binti Harun

(78071)

Bachelor of Computer Science (Network Computing) with Honours

2025

**SMART HOSPITAL ROOM ENVIRONMENTAL MONITORING: A REAL-TIME IOT
SOLUTION FOR PATIENT CARE ENHANCEMENT**

IZZAN HAZIRAH BINTI HARUN

**This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science with Honors (Network Computing)**

**Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2025**

**PEMANTAUAN PERSEKITARAN BILIK HOSPITAL PINTAR: PENYELESAIAN IOT
MASA NYATA UNTUK PENINGKATAN PENJAGAAN PESAKIT**

IZZAN HAZIRAH BINTI HARUN

Projek ini merupakan salah satu keperluan untuk

Ijazah Sarjana Muda Sains Komputer dengan Kepujian

(Pengkomputeran Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK
2025

DECLARATION

I hereby declare that this research together with all of its content is of my own work except for the research-based information and relative materials that were adapted and extracted from other resources, which have been quoted or stated respectively.

Signed,



.....
IZZAN HAZIRAH BINTI HARUN

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

23rd June 2025

ACKNOWLEDGEMENT

This Final Year Project made it this far with the help and support from everyone. I would like to express my sincere gratitude towards my supervisor, Dr. Kartinah binti Zen for her guidance, advice, patience, and encouragements throughout the process of the project. Without her guidance, this project will not made it this far. I would also like to thank my examiner Dr. Dr. Halikul bin Lenando for his feedback and comments on my report for my own improvements. And not to forget, our Final Year Project coordinator, Professor Wang Yin Chai for his guidelines towards the progress of the project. Besides, I would also like to thank my family and friends for their supports for me to complete this project successfully. In conclusion, I hope that this project will contribute to the development of smarter, more responsive healthcare environments that prioritize patient well-being and safety.

ABSTRACT

In today's healthcare environment, ensuring patient comfort and safety goes beyond medical treatment, it also involves maintaining a healthy and stable room environment. This project presents the development of a Smart Hospital Room Environmental Monitoring System that uses IoT (Internet of Things) technology to monitor key environmental parameters in real-time. The system is built using an Arduino Nano 33 IoT microcontroller and integrates several sensors including the DHT11 (for temperature and humidity), MLX90614 (for non-contact body temperature), MQ-135 (for air quality), and MQ-9 (for gas and carbon monoxide detection). All collected data is transmitted to the Blynk IoT cloud platform, where healthcare professionals can monitor room conditions remotely through a mobile or web dashboard. The system is also programmed with alert thresholds based on WHO and medical guidelines, so if any reading goes beyond safe limits, immediate notifications are sent to the users. This helps in taking prompt action, especially in critical care environments where patient health can be affected by temperature, air quality, or harmful gases. By combining low-cost hardware and a user-friendly IoT platform, this project aims to demonstrate a practical solution for improving patient care and hospital safety. It also provides a foundation for future upgrades, such as automated ventilation control or AI-driven health prediction. Overall, this system shows how smart technology can support better healthcare environments, especially in resource-constrained settings.

ABSTRAK

Dalam penjagaan kesihatan hari ini, memastikan keselesaan dan keselamatan pesakit bukan sahaja melibatkan rawatan perubatan, tetapi juga melibatkan penyelenggaraan persekitaran bilik yang sihat dan stabil. Projek ini membentangkan pembangunan Sistem Pemantauan Persekitaran Bilik Hospital Pintar yang menggunakan teknologi IoT (Internet of Things) untuk memantau parameter persekitaran secara masa nyata. Sistem ini dibina menggunakan mikropengawal Arduino Nano 33 IoT dan mengintegrasikan beberapa alat pengesan termasuk DHT11 (untuk suhu dan kelembapan), MLX90614 (untuk suhu badan tanpa sentuhan), MQ-135 (untuk kualiti udara), dan MQ-9 (untuk pengesanan gas dan karbon monoksida). Semua data yang dikumpul dihantar ke platform Cloud IoT Blynk, di mana para kakitangan kesihatan boleh memantau keadaan bilik dari jauh melalui Aplikasi telefon mudah alih atau web. Sistem ini juga diprogramkan dengan amaran berdasarkan garis panduan WHO dan perubatan, oleh itu jika sebarang bacaan melebihi had selamat, pemberitahuan segera akan dihantar kepada pengguna. Ini membantu dalam membuat keputusan bagi tindakan segera, terutamanya dalam persekitaran penjagaan kritikal di mana kesihatan pesakit boleh terjejas oleh suhu, kualiti udara, atau gas berbahaya. Dengan menggabungkan sensor berkos rendah dan platform IoT yang mesra pengguna, projek ini bertujuan untuk menunjukkan penyelesaian praktikal bagi meningkatkan penjagaan pesakit dan keselamatan hospital. Ia juga boleh menjadi asas kepada inovatif masa depan, seperti kawalan pengudaraan automatik atau ramalan kesihatan yang dipacu AI. Secara keseluruhannya, sistem ini menunjukkan bagaimana teknologi pintar boleh menyokong persekitaran penjagaan kesihatan yang lebih baik, terutamanya dalam persekitaran yang mempunyai sumber yang terhad.

Table of Contents

DECLARATION.....	i
ACKNOWLEDGEMENT	ii
ABSTRACT.....	iii
ABSTRAK.....	iv
CHAPTER 1: INTRODUCTION	1
1.1 Introduction.....	1
1.2 Problem Statement	3
1.3 Scope.....	4
1.4 Objectives	5
1.5 Brief Methodology.....	5
1.6 Significance of Project.....	8
1.7 Project Schedule.....	9
1.8 Expected outcomes	12
1.9 Summary	13
CHAPTER 2: LITERATURE REVIEW	14
2.1 Introduction.....	14
2.2 Review of Existing Systems	14

2.2.1 Development of Smart Healthcare Monitoring System in IoT Environment	14
2.2.2 Innovative IoT Solutions for Enhanced Patient	16
2.2.3 Smart room for patient monitoring based on IoT technologies	18
2.3 Comparison on Existing Systems and Proposed System.....	21
2.4 Brief Overview of The Proposed System	23
2.5 Summary	25
CHAPTER 3: METHODOLOGY/REQUIREMENT ANALYSIS AND DESIGN	26
3.1 Introduction.....	26
3.2 Project Methodology.....	27
3.3 Requirements Analysis.....	29
3.3.1 Identify the User	29
3.3.2 Functional Requirements	30
3.3.3 Hardware Requirements.....	31
3.3.4 Software Requirements	35
3.4 System Design and general Architecture	36
3.4.1 System Architecture	36
3.4.2 System Flowchart Diagram.....	39
3.4.3 User Interface Design	41
3.5 Summary	44

CHAPTER 4: DEVELOPMENT AND IMPLEMENTATION	45
4.1 Introduction.....	45
4.2 Software Installation and Configuration	46
4.2.1 Arduino IDE and Library Setup	46
4.2.1 WiFi Configuration and Security	47
4.3 Hardware Setup and Sensor Integration	48
4.3.1 Pin Assignments and Power Configuration	48
4.3.2 DHT11 Temperature and Humidity Sensor Integration.....	50
4.3.3 MQ-135 Air Quality Sensor Integration	50
4.3.4 MQ-9 Carbon Monoxide and Gas Sensor Integration	51
4.3.5 MLX90614 Infrared Temperature Sensor Integration	51
4.4 Programming and Embedded Logic	52
4.4.1 System Initialization and Setup	52
4.4.2 Continuous Data Monitoring Loop.....	53
4.4.3 Data Filtering and Noise Reduction.....	55
4.4.4 Threshold-Based Alert System	55
4.4.5 Smart Alert Management	56
4.4.6 System Control and Remote Management	57
4.4.7 Robust Communication Protocol.....	57

4.4.8 Event-Driven Alert Notifications	58
4.5 Cloud Integration and Blynk Dashboard	59
4.5.1 Device Configuration and Authentication	59
4.5.2 Virtual Pin Architecture and Data Mapping	60
4.5.3 Dashboard Visualization and Widget Configuration	61
4.5.4 Color-Coded Status System	63
4.5.5 Data Transmission Protocol and Real-Time Monitoring	64
4.5.6 Error Handling and System Monitoring	65
4.5.7 Intelligent Alert System and Notification Management	66
4.5.8 Remote System Control and Administrative Features	67
4.6 System Testing and Validation	68
4.7 Summary	68
CHAPTER 5: TESTING AND EVALUATION	69
5.1 Introduction	69
5.2 Functional Testing	70
5.2.1 Test Cases	70
5.3 Non-Functional Testing	78
5.3.1 Pre-Test and Post-Test Setup	78
5.3.2 Analysis of Test Results	79

5.3.3 Power Stability Testing	81
5.4 Summary	81
CHAPTER 6: CONCLUSION AND FUTURE WORK	82
6.1 Introduction.....	82
6.2 Objectives Achieved	82
6.3 Limitations and Constraints	83
6.4 Future Work	84
6.5 Conclusion	85
REFERENCES	86
APPENDIX.....	89

Table of Table

Table 1 Shows details task for FYP1 & FYP2.....	9
Table 2 Shows comparison of features between existing systems and proposed systems.....	21
Table 3 Shows list of hardware requirement.....	31
Table 4 Shows list of the software requirements	35
Table 5 shows detail for Arduino IDE and Library Setup	46
Table 6 Details of components integrated.....	49
Table 7 Pin wiring for DT11	50
Table 8 Pin wiring for MQ-9	51
Table 9 Pin wiring for MLX90614	52
Table 10 Virtual Pin Assignment	57
Table 11 Virtual Pin Architecture.....	60
Table 12 Test System Startup.....	71
Table 13 Test Case for High Temperature Alert.....	72
Table 14 Humidity Below Safe Range.....	73
Table 15 Test Case for Poor Air Quality Detection	74
Table 16 Test case for detection of harmful gas levels	75
Table 17 Test Case for Auto Wi-Fi Reconnection.....	76
Table 18 Real-Time Sensor Data Display	77
Table 19 Pre-test setup	78
Table 20 Post-test setup	79
Table 21 Shows analysis of test results.....	79

Table of Figure

Figure 1 Gantt Chart of FYP1 Task	11
Figure 2 Gantt Chart of FYP2 Task	11
Figure 3 The overall system architecture of the healthcare monitoring system (Islam et al., 2020)	15
Figure 4 Overview of the Proposed System (Yuvaraj, 2024)	17
Figure 5 Shows diagrams and actual models of operation of a smart room (Zhukova et al., 2023)	18
Figure 6 Shows general structure of the information processing method for smart rooms (Zhukova et al., 2023)	19
Figure 7 Detection poses and skeleton process, (a) Input image from the public camera, (b) skeleton human identification (Zhukova et al., 2023)	19
Figure 8 The main screen of the smartphone application (Zhukova et al., 2023)	20
Figure 9 Picture of Arduino Nano 33 Microcontroller	32
Figure 10 Picture of CO2 Sensor	33
Figure 11 show picture of CO Sensor	33
Figure 12 Show picture of room temperature sensor	34
Figure 13 Picture of Infrared Thermometer	34
Figure 14 System Architecture of the proposed system.....	36
Figure 15 Show the data flowchart of the system.....	39
Figure 16 Show the user interface	41
Figure 17 Show the mobile application user interface.....	43

Figure 18 show Arduino IDE for configuration.....	47
Figure 19 Show wiring and pin assignment.....	48
Figure 20 Shows the System Initialization snippet code	53
Figure 21 Shows the continuous loop snippet code.....	54
Figure 22 Shows snippet code for threshold set	56
Figure 23 shows snippet code for alert management.....	56
Figure 24 Shows snippet code for system control	57
Figure 25 Shows snippet code for HTTP response.....	58
Figure 26 Shows snippet to get authentication	59
Figure 27 Shows snippet virtual pin setup.....	61
Figure 28 Snippet of Gauge Design in App	62
Figure 29 Snippet of color-coded UI in Web	63

CHAPTER 1: INTRODUCTION

1.1 Introduction

The rapid advancement of IoT technology has revolutionized multiple sectors, including healthcare, by enabling real-time, data-driven solutions for complex challenges. Hospitals are increasingly adopting IoT solutions to monitor and improve patient care through efficient data collection and automation. Specifically, maintaining optimal environmental conditions in hospital rooms, especially in critical care units is essential for patient safety and recovery. Environmental factors such as temperature, humidity, and air quality significantly impact patient outcomes, with irregularities potentially causing respiratory issues or infections that delay recovery. However, many hospitals still rely on manual monitoring methods, which are prone to human error and can lead to unnoticed changes in these critical conditions (Abdulmalek et al., 2022; Carling, 2021).

Currently, most hospitals face limitations in their environmental monitoring systems, especially in ICU settings where patients are most vulnerable. The manual methods used are labor-intensive and unable to provide continuous, real-time data, which can result in delayed responses to critical environmental changes. This delay can have severe consequences for patients with compromised immune systems or respiratory challenges, whose conditions may worsen due to suboptimal environmental factors (Bockholt et al., 2022). This project aims to bridge this gap by implementing an IoT-based monitoring system that not only continuously tracks but also instantly alerts staff about any deviations from optimal environmental standards by push notification in Blynk, enhancing patient safety and recovery outcomes.

To address this critical gap, this project will develop an IoT-based real-time monitoring system using a sensor for temperature, humidity, air quality and non-contact infrared temperature sensor.

The system will be integrated with the cloud IoT to enable healthcare staff to access real-time data remotely and receive alerts when environmental factors or any movements deviate from set thresholds. The proposed solution will be web-based, allowing data to be continuously updated and accessible through mobile devices, thus providing flexibility and immediacy for healthcare professionals. This enhanced accessibility will support a proactive approach to environmental management, allowing for swift corrective action that may prevent complications in patient health.

Upon successful implementation, this project is expected to improve hospital room environmental management by providing healthcare staff with continuous, reliable data on critical conditions. This real-time monitoring system has the potential to significantly improve patient outcomes, particularly for those in intensive care, by minimizing risks associated with poor environmental quality and reducing manual monitoring burdens on healthcare staff.

1.2 Problem Statement

Effective environmental control is crucial for patient recovery, particularly in ICUs where minor fluctuations in temperature, humidity, and air quality can have significant health implications. Studies indicate that maintaining stable environmental conditions is essential to reduce infection risks, prevent respiratory complications, and support immune-compromised patients (Carling, 2021). However, due to the reliance on manual methods, current monitoring practices are inconsistent and prone to human error, often leading to unnoticed changes in room conditions. This lack of a reliable and continuous monitoring system poses a serious threat to patient safety, especially for those who are most vulnerable to environmental fluctuations (Bockholt et al., 2022).

The absence of real-time monitoring in critical care units hinders healthcare professionals' ability to respond promptly to changes that may compromise patient recovery. Manual monitoring requires time and attention from already overburdened healthcare staff, leading to situations where critical deviations go unnoticed until they visibly impact the patient. Consequently, patients may suffer from prolonged recovery periods, increased risk of infections, and overall diminished quality of care. This problem underscores the urgent need for a system that automates environmental monitoring and alerts staff in real-time, ensuring optimal room conditions without additional strain on healthcare providers.

This IoT-based solution will fill these gaps by providing continuous, automated monitoring, reducing reliance on manual processes and enabling immediate interventions when necessary. Such a system could not only enhance patient care quality but also support healthcare staff by reducing their workload related to environmental monitoring, thus promoting a more efficient, responsive hospital environment.

1.3 Scope

The scope of this project encompasses the development of an IoT-based system utilizing IoT cloud to monitor patient health and environmental parameters in hospital rooms. This includes the integration of sensors that track temperature, humidity, air quality and patient temperature monitoring in real time, ensuring that the conditions remain optimal for patient care. To facilitate user-friendly monitoring and alerting, an interface will be developed through the IoT cloud, allowing healthcare professionals to receive timely notifications regarding any changes in the monitored parameters. Furthermore, the project will involve rigorous testing and validation of the system's reliability under various conditions within a hospital environment, ensuring that it meets the necessary standards for effective patient care enhancement.

1.4 Objectives

The objectives of this Hospital Room Environmental Monitoring are:

- I. Create an IoT system to monitor patient health, temperature, humidity, and air quality in hospital ICUs.
- II. Integrating and configuring sensors to provide data to a central hub.
- III. Creating an intuitive user interface so hospital employees can see data and get alerts.

1.5 Brief Methodology

The project will follow an Agile methodology. This is because Agile methods save time and money by reducing documentation, focusing on quick development, and getting regular feedback (Shankarmani et al., 2012). Agile methodologies represent a newer class of development methodologies that seek to overcome the inherent limitations of traditional, plan-driven Software Development (SD) approaches (Chan & Thong, 2008). Hence, Agile makes the perfect method for this project divided into these key phases:

- I. **Planning Phase.** In this initial phase, we will meticulously identify and document all project requirements. This includes the selection of essential hardware components such as the temperature, humidity, and air quality sensors. Additionally, we will finalize the choice of the IoT cloud for real-time data visualization and remote monitoring. A thorough analysis of the project scope and timeline will be conducted to ensure efficient resource allocation and timely delivery.

II. **Design Phase.** This crucial phase involves creating detailed design plans, user interfaces, and system architecture blueprints. Key decisions include hardware and software component selection, sensor choices, communication protocols, data processing platforms, and sensor integration strategies. Additionally, defining communication protocols and data formats for sensor data transmission is essential.

III. **Development Phase.** The development phase will involve the key activities like system setup where the microcontroller will be carefully configured, and the necessary sensors will be meticulously installed, ensuring optimal performance and reliability. Interface development where the IoT cloud platform interface will be meticulously configured to provide a user-friendly interface for real-time monitoring and alert notifications. The interface will be optimized for intuitive navigation and easy interpretation of data. Data processing which robust data processing algorithms will be implemented to efficiently collect, analyze, and store sensor data. Alert thresholds will be meticulously defined to trigger timely notifications in case of any environmental anomalies. Detailed logs will be maintained to facilitate troubleshooting, performance analysis, and system optimization.

- IV. **Testing Phase.** Rigorous testing will be conducted in simulated environments to validate the system's accuracy, reliability, and responsiveness. We will meticulously record and analyze the system's performance metrics, including accuracy, precision, and response time. The system's ability to detect and alert deviations from optimal conditions will be rigorously tested under various scenarios. Any identified issues or shortcomings will be promptly addressed through iterative refinement and optimization.
- V. **Deployment Phase.** The fully tested and optimized IoT system will be carefully deployed in a controlled environment following WHO (World Health Organization) guidelines and standards for healthcare monitoring systems. The system will be configured to automatically maintain optimal conditions based on WHO recommendations for temperature, humidity, and air quality parameters. The system's performance and reliability will be closely monitored during the deployment phase to ensure compliance with international healthcare standards and to identify and rectify any potential issues.

VI. **Review Phase.** A comprehensive post-deployment evaluation will be conducted to assess the system's overall performance and compliance with WHO standards. System performance data, including temperature stability, humidity control, air quality maintenance, and sensor accuracy, will be meticulously collected and analyzed to identify areas for improvement. The insights gained from this review process will be used to refine the system's functionality, enhance automated responses, and optimize future iterations to better align with international healthcare monitoring standards.

1.6 Significance of Project

This project will support healthcare staff in maintaining an optimal environment for patient recovery with minimal manual intervention. The automated system reduces labor costs, minimizes human error, and enhances response times to environmental changes, potentially reducing patient recovery times and improving overall hospital efficiency.

1.7 Project Schedule

This first final year project starts on 7th of October 2024 and is expected to be completed on 17 of February 2024 which takes approximately 4 months. Meanwhile FYP2 is set to start on April 1st, 2025, and end on 28th July 2025. The details of the project are described below in the table represented in the Gantt Chart.

Table 1 Shows details task for FYP1 & FYP2

Task	Start Date	End Date
PROJECT PROPOSAL	7 Oct 2024	14 Nov 2024
Brief Proposal Preparation	7 Oct 2024	20 Oct 2024
Submission of Brief Proposal	20 Oct 2024	20 Oct 2024
Approval and Comments by Reviewer	28 Oct 2024	28 Oct 2024
Full Proposal Preparation	20 Oct 2024	14 Nov 2024
Submission of Full Proposal	14 Nov 2024	14 Nov 2024
FYP1	7 Oct 2024	12 Jan 2025
Chapter 1: Introduction	14 Nov 2024	21 Nov 2024
Submission of Chapter 1	21 Nov 2024	21 Nov 2024
Chapter 2: Literature Review	21 Nov 2024	13 Dec 2024
Submission of Chapter 2	13 Dec 2024	13 Dec 2024
Chapter 3: Requirement Analysis and Design	13 Dec 2024	5 Jan 2025
Submission of Chapter 3	5 Jan 2025	5 Jan 2025
Submission of FYP 1 Final Report	17 Jan 2025	17 Jan 2025
Submission of Paper and video presentation assessment	19 Jan 2025	19 Jan 2025

Evaluation by Supervisor and Examiner	19 Jan 2025	25 Jan 2025
Amendment and Modification Period for FYP	17 Jan 2025	10 Feb 2025
Submission of Final Report	19 Jan 2025	19 Jan 2025
FYP 2	1 April 2025	28 July 2025
Submission of the proposed/revised structure of FYP report, Title and Gantt Chart	1 April 2025	1 April 2025
Chapter 4: Development and Implementation	1 April 2025	14 May 2025
Submission of first draft Chapter 4	15 May 2025	15 May 2025
Chapter 5: Testing and Evaluation Chapter 6: Conclusion and Future Work	15 May 2025	30 May 2025
Submission of first draft Chapter 5, Chapter 6 & Abstract for paper	31 May 2025	31 May 2025
Submission of First Draft for FYP 2 Full Report & Paper	10 Jun 2025	10 Jun 2025
Submission of Final Report (word and Pdf Format), source code, installation kits, user manual and Paper for Assessment	23 Jun 2025	23 Jun 2025
FYP Symposium	1 July 2025	2 July 2025
Final Date for Examiners to give their comments & feedback (if any)	4 July 2025	4 July 2025
Final Date for Submission of assessment mark to FYP Management System (Online)	15 July 2025	15 July 2025
Amendment and Modification Period for FYP (if any)	1 July 2025	18 July 2025

Submission of Final Report	28 July 2025	28 July 2025
----------------------------	--------------	--------------

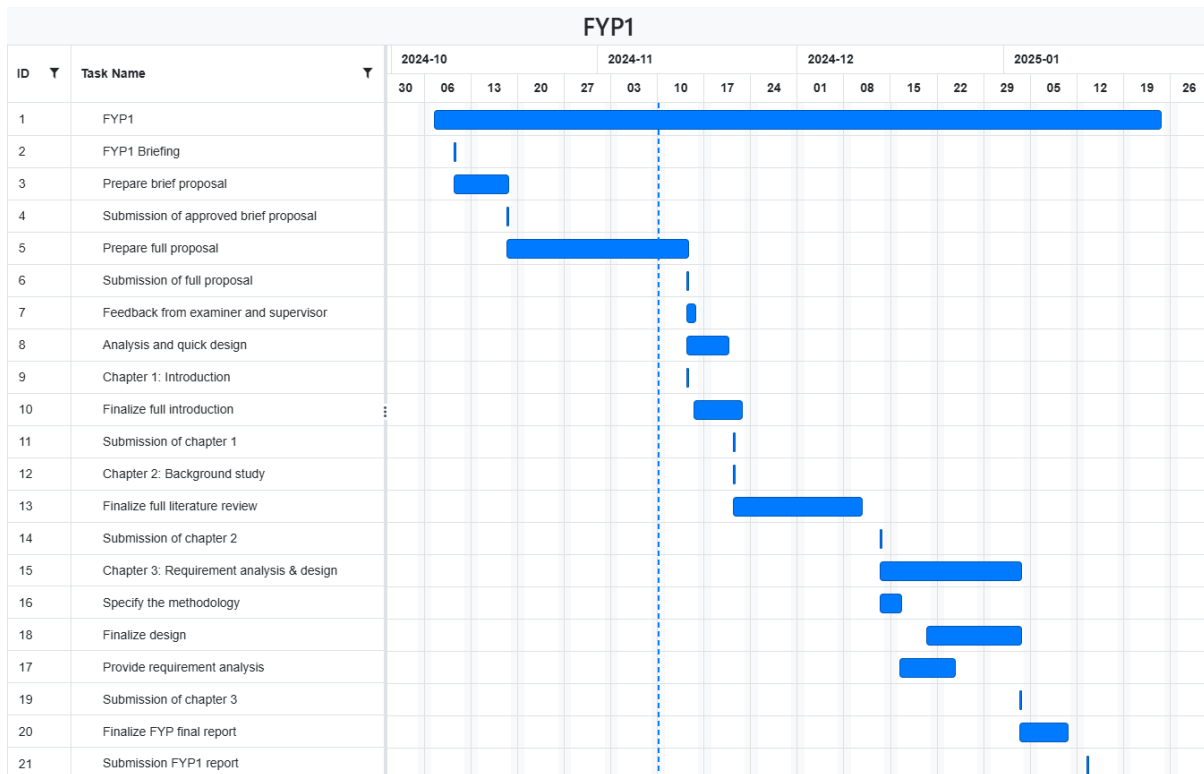


Figure 1 Gantt Chart of FYP1 Task

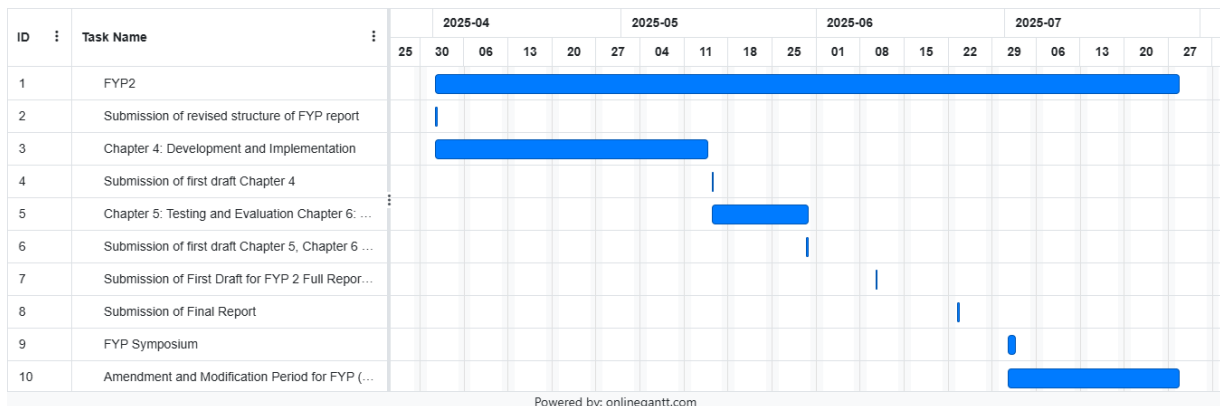


Figure 2 Gantt Chart of FYP2 Task

1.8 Expected outcomes

This project will benefit healthcare providers by automating room condition monitoring, ensuring optimal patient environments by providing the following:

- I. **Reliable IoT System:** Enables continuous monitoring of critical environmental parameters within hospital rooms and patient temperature.
- II. **Intuitive User Interface:** An IoT cloud-powered interface designed for seamless monitoring and prompt alert notifications.
- III. **Enhanced Patient Care:** Optimizes hospital room environmental management through real-time alerts, supporting timely interventions to maintain optimal conditions for patient recovery.

1.9 Summary

In this chapter, an overview of the project has been outlined and discussed. The project aims to provide a real-time IoT-based monitoring system for hospital room environments, focusing on factors like temperature, humidity, and air quality to improve patient care, especially in critical units like ICUs. The objectives include addressing the challenges of manual monitoring and ensuring timely alerts to healthcare providers. The methodology for this project has been identified, incorporating tools like IoT cloud platforms, while using C/C++ programming language for development. Additionally, a project schedule has been developed to ensure milestones are achieved. The significance of the project lies in its potential to reduce human error, optimize healthcare conditions, and enhance patient recovery outcomes. Therefore, the next chapter will elaborate on the detailed methodology employed for achieving these objectives.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

To achieve the objectives of the project outlined in the previous chapter, background research of similar systems has been conducted and analyzed. This chapter reviews three existing systems relevant to the topic and provides a comparative analysis with the proposed system to identify gaps and opportunities. Furthermore, a brief overview of the key features of the proposed system is included. The comparison is based on the following three journals articles:

- I. Development of Smart Healthcare Monitoring System in IoT Environment
(Islam et al.,2020)
- II. Innovative IoT Solutions for Enhanced Patient Monitoring (Yuvaraj, 2024).
- III. Smart Room for Patient Monitoring Based on IoT Technologies (Zhukova et al., 2023).

2.2 Review of Existing Systems

2.2.1 Development of Smart Healthcare Monitoring System in IoT Environment

The study conducted by Islam et al. (2020) focuses on addressing significant challenges in healthcare systems, particularly the gaps in services between rural and urban areas. The researchers highlight the limitations of traditional healthcare systems, such as the lack of timely emergency responses and insufficient availability of medical professionals, especially during critical situations. For example, according to the Agency for Healthcare Research and Quality (US), rural areas often experience a doctor-to-patient

ratio that is 50% lower than in urban areas. To address these issues, the authors proposed a smart healthcare monitoring system leveraging Internet of Things (IoT) technologies.

The primary objectives of the system were to enable real-time monitoring of patients' vital health parameters, monitor environmental conditions in patient rooms, and provide a secure and accessible web-based platform for healthcare professionals. These goals aim to improve healthcare delivery by reducing response times and increasing the efficiency of patient monitoring, particularly in underserved regions (Islam et al., 2020).

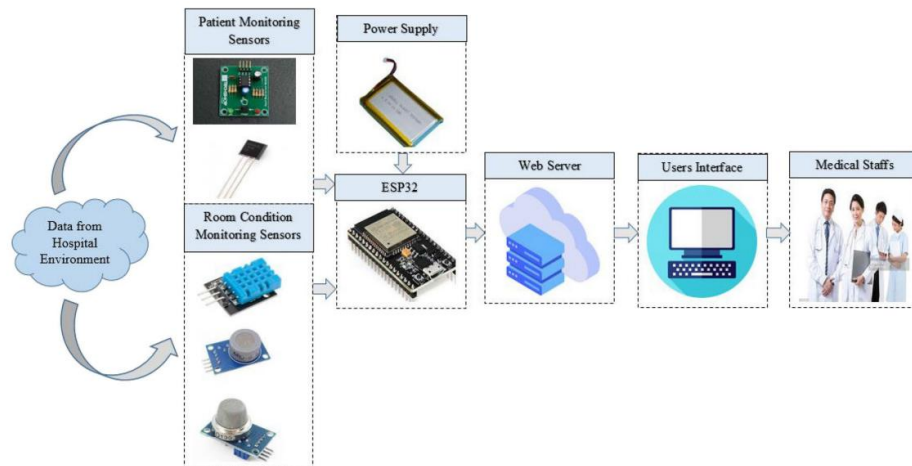


Figure 3 The overall system architecture of the healthcare monitoring system (Islam et al., 2020)

The system's hardware design included multiple sensors such as a heartbeat sensor for tracking pulse rates, a body temperature sensor (LM35) for accurate temperature readings, a room temperature sensor (DHT11) for monitoring humidity and temperature, and gas sensors (MQ-9 and MQ-135) for detecting harmful gases like CO₂. The data collected by these sensors was processed using an ESP32 microcontroller, which transmitted the data to a cloud platform via Wi-Fi. The ThingSpeak platform was used for

visualizing the collected data, offering medical personnel an intuitive graphical interface updated every 15 seconds. The system also employed HTTP protocols for secure data transmission and password authentication for user access.

Experimental results demonstrated the system's high accuracy, with error rates consistently below 5% (Islam et al., 2020). However, certain challenges were noted, such as motion artifacts, sensor displacement, and environmental interferences, which occasionally affected performance. Despite these limitations, the system's innovative approach provided a significant improvement in healthcare monitoring, especially in remote and underserved areas. The researchers recommended future improvements, including miniaturization of the device and integration of additional health metrics such as glucose levels and respiratory indicators.

2.2.2 Innovative IoT Solutions for Enhanced Patient

Yuvaraj (2024) explored the transformative potential of IoT in healthcare, focusing on how it has redefined patient monitoring and improved healthcare resource management. This study was conducted in the context of Industry 4.0, where IoT technologies have become integral to enhancing operational efficiency and decision-making processes in healthcare institutions.

The system proposed in this study was designed to monitor multiple vital signs such as heart rate, blood pressure, body temperature, and respiration rate. By using IoT-enabled devices, continuous data collection was achieved, allowing healthcare providers to gain real-time insights into patient conditions. The system incorporated diverse communication protocols, including Wi-Fi, ZigBee, and RFID, ensuring seamless data transfer across

devices. Anomaly detection features were also integrated, enabling early intervention in case of medical emergencies, which helped reduce the risk of complications or fatalities.

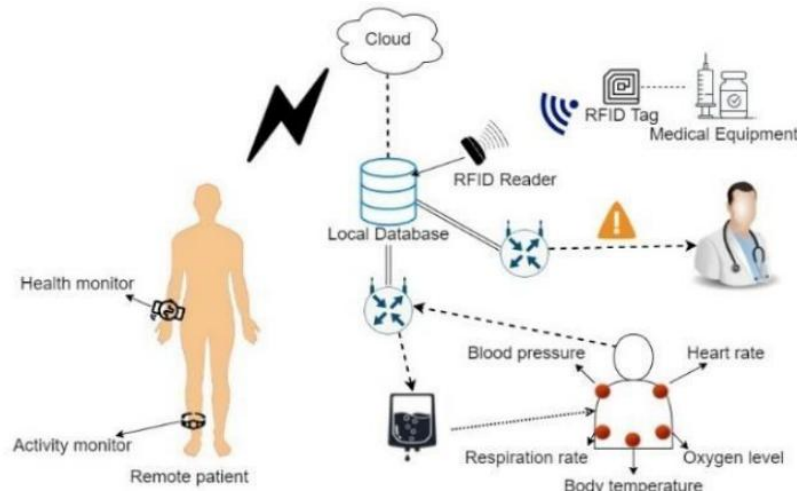


Figure 4 Overview of the Proposed System (Yuvaraj, 2024)

While the system demonstrated significant advancements in real-time monitoring, it faced challenges related to data security and privacy. The interconnected nature of IoT devices made the system vulnerable to cyberattacks, raising concerns about the protection of sensitive patient data (Yuvaraj, 2024). Moreover, interoperability issues between devices and platforms were highlighted, emphasizing the need for standardization in IoT healthcare systems. Despite these limitations, the study underscored the value of IoT in improving patient outcomes, particularly during the COVID-19 pandemic, where remote monitoring played a critical role in minimizing hospital overcrowding and reducing the risk of virus transmission.

2.2.3 Smart room for patient monitoring based on IoT technologies

The research by Zhukova et al. (2023) proposed a novel approach to patient monitoring through the development of a smart room solution leveraging IoT technologies. This study aimed to address the high costs and inefficiencies associated with traditional monitoring systems, particularly in resource-constrained settings. By integrating IoT with machine learning, the system provided a scalable and cost-effective solution suitable for both hospital and home care environments.

The smart room utilized readily available hardware, such as existing cameras and smartphones, to reduce costs while maintaining functionality. Machine learning algorithms were employed for advanced features such as posture recognition, emotion detection, and event classification. These capabilities allowed the system to provide non-intrusive monitoring while ensuring patient safety and privacy. Distributed processing via fog and cloud computing enhanced the scalability and responsiveness of the system, making it adaptable to diverse healthcare scenarios.

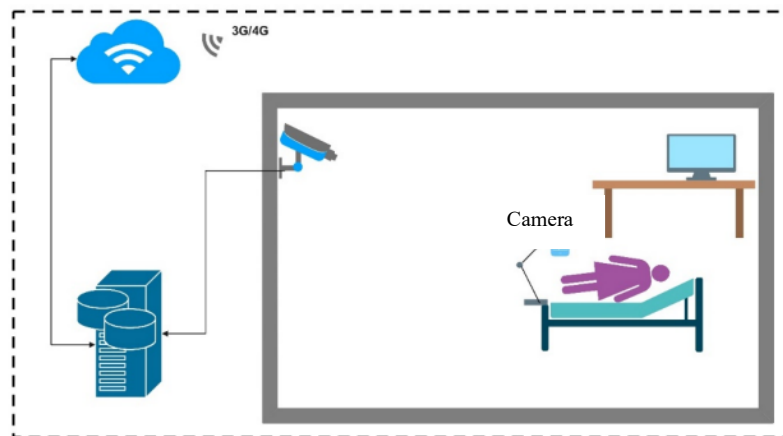


Figure 5 Shows diagrams and actual models of operation of a smart room (Zhukova et al., 2023)

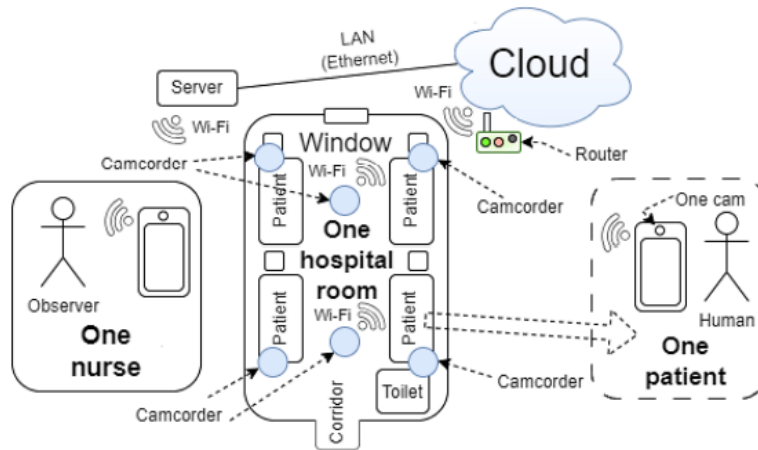


Figure 6 Shows general structure of the information processing method for smart rooms (Zhukova et al., 2023)



(a)



(b)

Figure 7 Detection poses and skeleton process, (a) Input image from the public camera, (b) skeleton human identification (Zhukova et al., 2023)



*Figure 8 The main screen of the smartphone application
(Zhukova et al., 2023)*

Experimental validation of the system demonstrated its effectiveness, with an average improvement of 25.85% in assistance speed (Zhukova et al., 2023). However, the study also identified several areas for future improvement, including the development of fall warning systems, depression recognition, and gesture monitoring. By addressing these challenges, the smart room solution aimed to provide a comprehensive and adaptable framework for patient monitoring, significantly contributing to the evolution of IoT-based healthcare technologies.

2.3 Comparison on Existing Systems and Proposed System

Table 2 Shows comparison of features between existing systems and proposed systems

Feature	Development of Smart Healthcare Monitoring System in IoT Environment	Innovative IoT Solutions for Enhanced Patient	Smart room for patient monitoring based on IoT technologies	Proposed System
Real-time monitoring	Yes	Yes	Yes	Yes
Environmental monitoring	Yes	No	Partial	Comprehensive
Sensor types	Health and environmental sensors	Health sensors only	Cameras for posture and emotion	Health and environmental sensors
Data platform	ThingSpeak	Custom platform	Smartphone app	IoT cloud
Scalability	Limited	Moderate	High	High
Key innovation	Comprehensive sensor integration	Diverse protocols	Cost-efficiency and machine learning	IoT-driven alerts and advanced data

Based on Table 1, the comparison of features between existing systems and proposed system are as follows:

- I. The first system, developed by Islam et al. (2020), provides a strong foundation for healthcare monitoring by integrating both health and environmental sensors. The inclusion of CO₂ and CO sensors for air quality monitoring, alongside patient vital sensors, makes it comprehensive. However, its scalability is limited due to the use of a single platform (ThingSpeak) and a focus on individual room monitoring rather than system-wide adaptability.
- II. The second system, reviewed by Yuvaraj (2024), stands out for its emphasis on communication protocol diversity, allowing seamless data transmission across multiple devices. While it excels in real-time health monitoring and anomaly detection, the lack of environmental monitoring features limits its applicability in hospital environments where air quality and temperature are critical for patient care.
- III. The third system, proposed by Zhukova et al. (2023), demonstrates innovative use of machine learning for posture recognition and emotion detection. Its reliance on readily available hardware like cameras and smartphones enhances cost-efficiency and scalability. However, monitoring capabilities are restricted to physical behavior and do not include direct health or environmental parameters.

- IV. The proposed system aims to address the gaps identified in the existing systems by offering a more holistic approach. By combining health and environmental monitoring with IoT-driven alerts, the system ensures comprehensive patient care. Its use of advanced platforms like IoT Cloud enhance scalability and adaptability, making it suitable for both small-scale and large-scale hospital implementations.

2.4 Brief Overview of The Proposed System

After reviewing the existing systems, it is evident that while each solution provides significant advancements, none comprehensively addresses the dual needs of health and environmental monitoring in hospital rooms. The proposed system uniquely integrates health and environmental metrics into a unified platform, ensuring simultaneous tracking of patient vitals and room conditions to optimize care and recovery environments. The proposed system aims to bridge these gaps by integrating IoT technology to monitor critical environmental factors such as temperature, humidity, and air quality, alongside contactless patient monitoring. This dual monitoring approach ensures that both patient safety and comfort are maintained at optimal levels, particularly in critical care settings where environmental conditions can directly impact patient recovery. For instance, studies, such as the 2003 report by the Centers for Disease Control and Prevention (CDC), highlight that maintaining optimal temperature and air quality reduces the risk of hospital-acquired infections by minimizing pathogen proliferation and improving patient recovery environments. and improves recovery rates for patients in intensive care units (ICUs). Unlike existing systems, which primarily focus on health metrics or limited environmental parameters, the proposed system combines these aspects into a single, scalable solution. It

leverages the capabilities of IoT Cloud platform to provide real-time data visualization and automated alerts, enabling swift responses to changing conditions.

This proactive approach not only enhances patient care but also reduces the workload of healthcare professionals by streamlining monitoring processes. Furthermore, the system's cloud-based data storage enables long-term trend analysis, which can support preventive care strategies and improve overall hospital efficiency. For example, by identifying patterns in air quality and temperature fluctuations, hospitals can preemptively address maintenance issues or adjust care protocols to optimize patient outcomes.

2.5 Summary

This chapter reviewed three existing IoT-based healthcare monitoring systems and identified their features, methodologies, and limitations. A comparative analysis highlighted the advantages of the proposed system, such as its comprehensive monitoring capabilities and scalability. This evaluation establishes a strong foundation for the development of a cutting-edge solution to improve patient care through real-time environmental monitoring

CHAPTER 3: METHODOLOGY/REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

The development of an effective environmental monitoring system for hospital rooms requires a carefully structured methodology that addresses both technical capabilities and healthcare-specific requirements. This chapter presents a comprehensive framework for developing the Smart Hospital Room Environmental Monitoring system, encompassing the systematic approaches used for requirement gathering, system design, and implementation planning.

Environmental parameters in hospital settings directly impact patient recovery, staff efficiency, and medical equipment operation. Temperature fluctuations can affect patient comfort and recovery rates, while humidity levels influence infection control and equipment functionality. Air quality parameters, including particulate matter and CO₂ levels, play crucial roles in preventing hospital-acquired infections and ensuring optimal healing conditions. This understanding of the healthcare environment's complexity has informed our methodological approach to developing an IoT-based monitoring solution.

3.2 Project Methodology

Agile methodology is implemented in developing the proposed system for Smart Hospital Room Environmental Monitoring as stated in chapter 1. This methodology adapted the sprint which enables the processes to be divided into length of time. Therefore, the time management and process development of the system will be more manageable and allow minimization of the system's cost.

The phases that are included in this method are the planning phase, designing phase, development phase, testing phase, execution phase and feedback phase. In the planning phase, all the data regarding environment monitoring will be analyzed and gathered to fulfill the requirement of the system. The problem statement of this project is identified, and the goals are defined to solve the problem statement.

Next in the designing phase, the design of the monitoring system was created following the requirement that have gathered previously in the planning phase. The flowchart and the system design are developed in this process to provide a better understanding of the flow of the system.

The development and iteration took place iteratively to make sure the elements of the system adhered to the planning and design phase. The hospital room environment monitoring will be developed using Arduino Nano 33 IoT processor, CO2 Sensor (MQ-135), Room Temperature Sensor (DHT11), CO Sensor (MQ-9), and Infrared Temperature Non-contact (MLX90614)

The Arduino Nano 33 IoT is not just a microcontroller; it's a compact yet powerful IoT-enabling platform that aligns perfectly with the needs of a Smart Hospital Room Environmental Monitoring system. Its combination of built-in WiFi and Bluetooth connectivity, ARM Cortex-M0+ processor efficiency, secure communication capabilities, and compact form factors make it indispensable for delivering reliable, real-time environmental monitoring in modern healthcare settings.

After the development phase, the testing of the system will take place to observe the functionality of the system. The iteration will take place in testing phase to improve the functionality of the system to be better than prior development through trials and errors. In this phase, the fully development hospital room environment monitoring system will be tested and executed in real-world environments. Data collection from sensors, transmission, and analysis will be carried out in accordance with the project's plan. The system will perform its intended function in this phase.

In the evaluation phase, automated system performance monitoring and compliance verification with WHO standards are very important to keep track of the requirements that have been listed in the planning phase. Therefore, if there are any deviations or performance issues that must be addressed, the problems need to be identified and resolved through system optimization so that the monitoring capabilities can be improved.

In conclusion, these phases will be in the form of a cycle starting from the planning until the evaluation phase and will be in iterative mode to fulfill the optimal performance conditions needed for the healthcare environmental monitoring system.

3.3 Requirements Analysis

3.3.1 Identify the User

The primary users of the Smart Hospital Room Environmental Monitoring system include healthcare professionals, such as nurses and doctors, who monitor environmental conditions to ensure patient safety and comfort. These users rely on real-time data and alerts to make immediate decisions regarding patient care. Secondary users include hospital administrators who utilize the system for analyzing trends and planning operational strategies to optimize hospital room conditions. Additionally, technical users, such as IT and maintenance staff, are responsible for ensuring the system functions smoothly by performing regular maintenance, troubleshooting, and updates. The system is designed to cater to the specific needs of these user groups, ensuring usability and efficiency.

3.3.2 Functional Requirements

The system's functional requirements have been determined through a review of relevant literature, including WHO articles and other sources that address healthcare and environmental needs:

- Environmental Monitoring Capabilities:

The system must continuously monitor room temperature with an accuracy of $\pm 1.5^{\circ}\text{C}$ within the range of $18\text{-}25^{\circ}\text{C}$, as specified by WHO guidelines for healthcare facilities. Humidity monitoring must maintain accuracy within $\pm 5\%$ for the critical range of $40\text{-}60\%$ relative humidity (Heseltine & Rosen, 2009). Air quality measurements must track CO_2 levels, particulate matter ($\text{PM}_{2.5}$ and PM_{10}), and volatile organic compounds (VOCs) (World Health Organization, 2021).

- Alert Management:

Based on research from articles by Iqbal et al. (2021), the system must generate immediate notifications when environmental parameters exceed predetermined thresholds. Alerts should be prioritized based on severity and delivered through multiple channels (mobile notifications, email, and dashboard alerts). Healthcare staff must be able to acknowledge and respond to alerts through the system interface.

- Data Management:

The system must maintain a continuous log of environmental data with a sampling rate of at least once per minute. Historical data must be stored for a minimum of one year and be easily accessible for trend analysis and reporting. The system should generate automated daily, weekly, and monthly reports of environmental conditions.

3.3.3 Hardware Requirements

The proposed system will require this hardware to build the prototype of the system which consists of microcontroller, sensor and other tools. The table shows details of the hardware requirement for the proposed system.

Table 3 Shows list of hardware requirement

Hardware	Specification
Microcontroller	Arduino Nano 33 IoT
Sensors	i. CO ₂ Sensor (MQ-135) ii. Room Temperature Sensor (DHT11) iii. CO Sensor (MQ-9) iv. Infrared Temperature Non-contact (MLX90614)
Other tools	i. Micro USB cable ii. Power Supply

Based on Table 1, the Arduino Nano 33 IoT is the microcontroller in this proposed system. This microcontroller is the central processing unit of the system, chosen for its built-in Wi-Fi and Bluetooth connectivity, compact form factor, and ARM Cortex-M0+ processing capabilities. It enables seamless communication with sensors and the IoT cloud for real-time data monitoring and control.

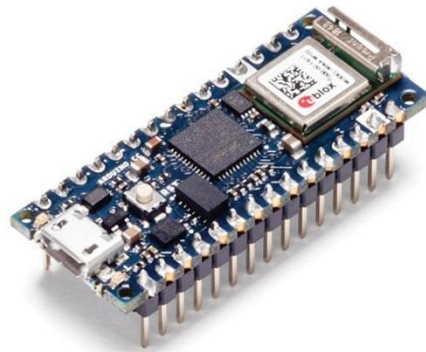


*Figure 9 Picture of Arduino Nano 33
Microcontroller*

This system includes a few sensors, and the first one is, CO2 Sensor (MQ-135). This sensor detects carbon dioxide (CO2) levels in the hospital room, a critical factor in maintaining indoor air quality and ensuring patient safety. Elevated CO2 levels can indicate inadequate ventilation, prompting necessary actions.



Figure 10 Picture of CO2 Sensor

Alongside the MQ-135 we have CO Sensor (MQ-9) The MQ-9 sensor measures carbon monoxide (CO) levels, a hazardous gas that can compromise patient health if undetected. This sensor adds an extra layer of safety by identifying potential gas leaks or air quality issues.



Figure 11 show picture of CO Sensor

Next is the Room Temperature Sensor (DHT11) The DHT11 monitors room temperature and humidity, providing essential data for maintaining a comfortable and safe environment for patients. It is known for its accuracy and cost-effectiveness.

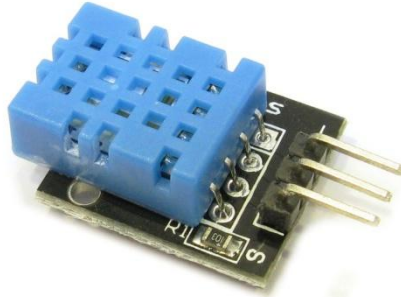


Figure 12 Show picture of room temperature sensor

Another sensor is MLX90614 Infrared Thermometer for non-contact patient body temperature monitoring. This sensor detects body temperature without physical contact, ensuring hygiene and reducing the risk of cross-contamination. It is particularly effective in scenarios where immediate alerts are necessary if a patient's temperature deviates from the normal range.



Figure 13 Picture of Infrared Thermometer

As for the Type-C cable, it is used for powering and programming the ESP32. Its universal compatibility and efficient power delivery make it a reliable choice for maintaining continuous system operation.

3.3.4 Software Requirements

These software requirements are essential for successful development to be fulfilled to develop the proposed system. The software is required for the prototype of the proposed system is outlined in Table

Table 4 Shows list of the software requirements

Software	Specification
Integrated Development Environment (IDE)	Arduino IDE Software
Programming Language	C/C++ language
IoT platform	Blynk cloud

3.4 System Design and general Architecture

3.4.1 System Architecture

The System Architecture for the Smart Hospital Room Environmental Monitoring system is designed to ensure seamless, real-time monitoring of environmental parameters within hospital rooms. At the core of this architecture is the Arduino Nano 33 IoT microcontroller, which serves as the central processing unit. This microcontroller is chosen for its robust features, including built-in Wi-Fi and Bluetooth connectivity, allowing it to collect data from various sensors and communicate with the Blynk cloud platform for data storage, analysis, and visualization.

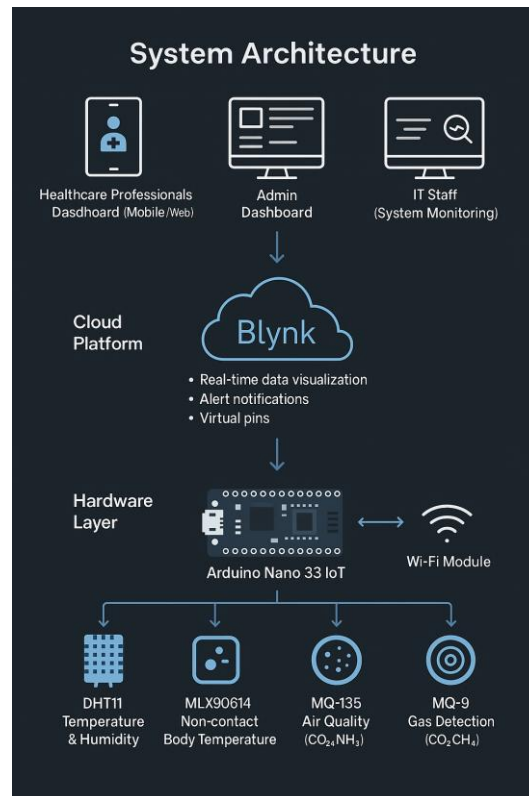


Figure 14 System Architecture of the proposed system

The system integrates several environmental sensors including the CO2 Sensor (MQ-135), Room Temperature Sensor (DHT11), CO Sensor (MQ-9), and Infrared Temperature Non-contact (MLX90614). These sensors are responsible for continuously monitoring critical parameters such as carbon dioxide levels, ambient temperature, humidity, carbon monoxide levels, and non-contact temperature measurements. Each sensor is connected to the Arduino Nano 33 IoT, which processes the sensor data locally using its integrated microcontroller. The board then transmits the data via its built-in Wi-Fi connectivity to the cloud platform, specifically the Blynk Cloud, where the data is analyzed and visualized in real-time.

The Blynk Cloud platform serves as the central hub of the system, storing incoming sensor data and providing comprehensive IoT analytics. It enables real-time monitoring through intuitive mobile and web dashboards that display the environmental conditions of the hospital room with customizable widgets and graphs. Historical data is automatically stored in the cloud for trend analysis and compliance reporting, helping hospital administrators and healthcare professionals track environmental patterns over time. The platform's robust notification system triggers immediate alerts when environmental parameters exceed predefined thresholds, delivering notifications through multiple channels including push notifications to mobile devices, emails, and in-app alerts. These notifications can be customized based on severity levels and can be configured by healthcare staff to ensure critical environmental issues are addressed promptly.

The user interface, accessible through Blynk's mobile app and web console, provides healthcare professionals with an intuitive overview of the hospital room's environmental status. The interface features real-time data visualization, customizable alert thresholds, and historical data charts that enable proactive environmental management to ensure optimal patient comfort and safety. The system also supports remote monitoring capabilities, allowing healthcare staff to monitor multiple rooms simultaneously and receive instant notifications regardless of their location within the facility.

In summary, the architecture leverages the Arduino Nano 33 IoT's integrated connectivity with multiple environmental sensors and the Blynk Cloud platform to create a robust, cost-effective, and user-friendly solution for real-time environmental monitoring in hospital rooms. This integrated system empowers healthcare professionals to actively manage room conditions, ensuring a safe and comfortable environment for patients while improving overall healthcare delivery and operational efficiency.

3.4.2 System Flowchart Diagram

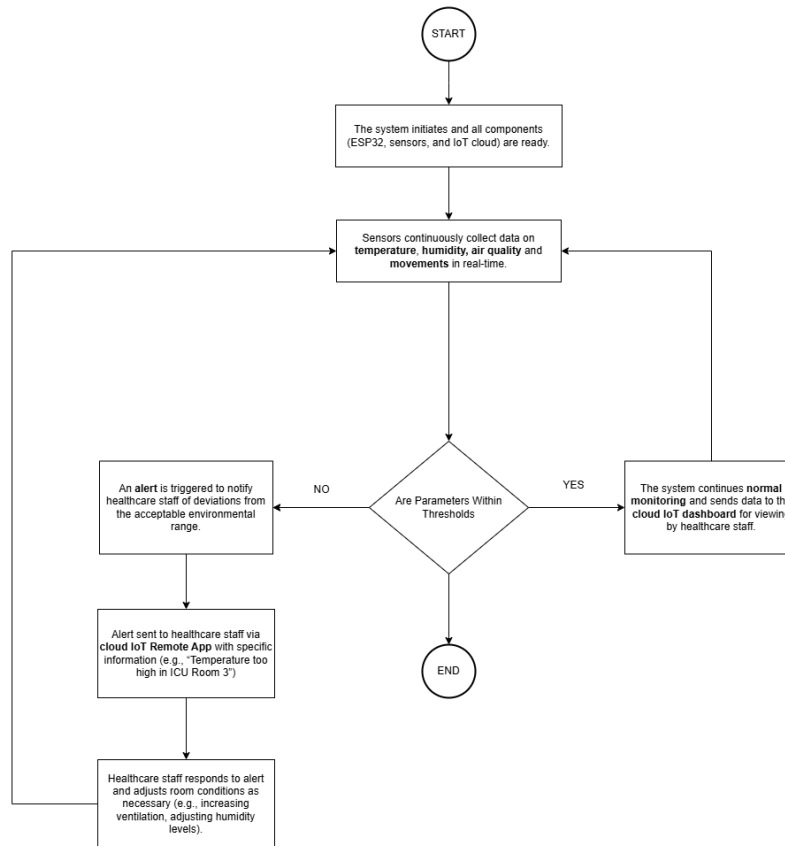


Figure 15 Show the data flowchart of the system

The Data Flow Diagram illustrates the flow of information and actions within the Smart Hospital Room Environmental Monitoring system. The process begins with the initialization of the system, where all components, including the Arduino Nano 33 IoT microcontroller, sensors, and Blynk Cloud platform, are set up and ready to operate. Sensors continuously monitor environmental parameters such as temperature, humidity, air quality, and infrared temperature readings, relaying this data in real-time to the Arduino Nano 33 IoT for processing.

The Arduino Nano 33 IoT microcontroller evaluates the collected data and determines whether the parameters fall within acceptable thresholds. If the parameters remain within the safe range, the system continues its regular monitoring operations while uploading the data to the Blynk Cloud for analysis and dashboard visualization. In cases where the parameters deviate from the predefined thresholds, alerts are triggered to notify healthcare staff via the Blynk mobile application or web console. These alerts are accompanied by actionable information, such as room-specific details and recommended corrective measures like adjusting ventilation or temperature settings.

Healthcare staff can then respond promptly to ensure optimal room conditions, with their actions logged and monitored via the Blynk platform for future evaluations. This iterative and automated data flow ensures efficient, real-time decision-making to enhance patient care and maintain a safe hospital environment.

3.4.3 User Interface Design

The user interface for the system is designed using Blynk Cloud to ensure intuitive and efficient monitoring experience. The dashboard displays real-time data from sensors, including temperature, humidity, CO2, and CO levels, in a visually appealing format with customizable widgets, graphs, and color-coded alerts. The interface also includes options to configure thresholds and view historical data logs through interactive charts and data tables. Alerts are prominently displayed with push notifications and in-app warnings, allowing healthcare staff to respond quickly to deviations. Blynk's cross-platform compatibility ensures that users can access the system from any device through the mobile app or web console, providing convenience and flexibility in managing hospital room conditions from anywhere within the facility.

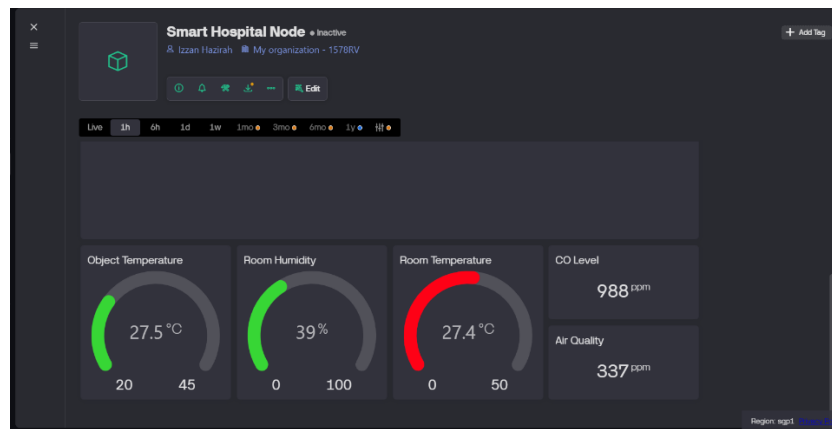


Figure 16 Show the user interface

Figure 16 shows the example of a user interface for the Smart Hospital Room Environmental Monitoring system that is implemented using the Blynk Cloud platform, offering a comprehensive and intuitive dashboard design that enables healthcare staff to efficiently monitor and control room environmental conditions.

The interface features various widget components strategically arranged for optimal visualization and interaction, including real-time value displays showing precise numerical readings of environmental parameters, circular gauges providing quick visual references, and percentage indicators for relative values such as humidity levels. The dashboard incorporates interactive control elements such as toggle switches, push buttons, and sliders that allow staff to adjust threshold values and settings, while live charts and historical data visualizations help track parameter trends over time.

For enhanced usability, the interface supports both desktop and mobile views with flexible widget sizing and customization options, ensuring accessibility across different devices and user preferences. The clean, modern design employs color coding and intuitive navigation to reduce cognitive load, enabling healthcare staff to quickly identify and respond to environmental changes or alerts while maintaining focus on patient care. Essential monitoring capabilities include real-time data updates, push notifications, and integrated messaging features for communication, all integrated into a responsive layout that automatically adapts to different screen sizes. This thoughtfully designed interface successfully balances the complexity of environmental monitoring with the need for straightforward, efficient operation in a healthcare setting.

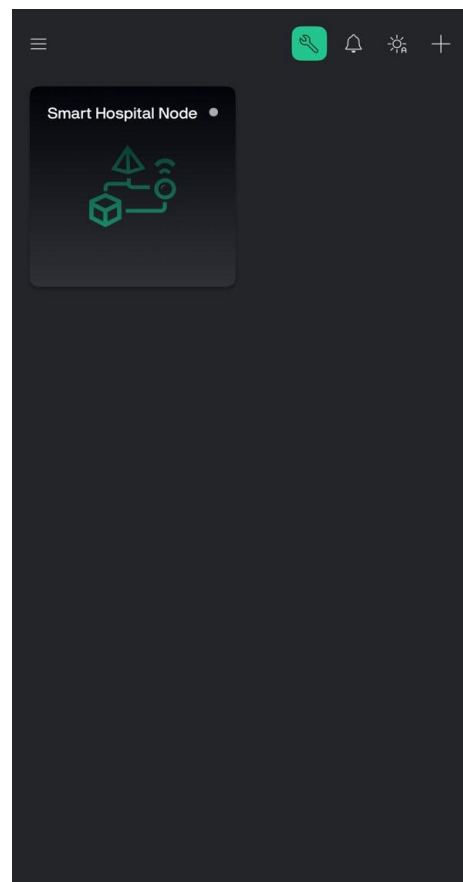


Figure 17 Show the mobile application user interface

3.5 Summary

This chapter has outlined the methodology, requirements analysis, and system design for the Smart Hospital Room Environmental Monitoring system. The Agile development methodology will be employed to ensure flexibility and iterative development. The system will be designed to meet the specific needs of patients, healthcare professionals, and hospital administration, while adhering to rigorous functional and non-functional requirements. The system architecture will incorporate a combination of hardware and software components to provide real-time monitoring, data analysis, and control capabilities. The successful implementation of this system has the potential to significantly enhance patient care, improve operational efficiency, and contribute to a safer and more comfortable hospital environment.

CHAPTER 4: DEVELOPMENT AND IMPLEMENTATION

4.1 Introduction

This chapter presents the development and implementation of the Smart Hospital Room Environmental Monitoring System. The system monitors temperature, humidity, air quality, and harmful gases in hospital rooms using an Arduino Nano 33 IoT microcontroller.

The development process involved hardware configuration, sensor integration, embedded software development, cloud connectivity setup, dashboard implementation, and system testing. The Arduino Nano 33 IoT was selected for its integrated Wi-Fi capabilities and sensor compatibility.

The implementation followed a systematic approach which included development environment setup, hardware configuration, sensor calibration, embedded programming for data acquisition, cloud platform configuration, and performance testing. The system integrates multiple environmental sensors with real-time data transmission to a web-based dashboard for remote monitoring by healthcare personnel.

4.2 Software Installation and Configuration

4.2.1 Arduino IDE and Library Setup

The Arduino Integrated Development Environment (IDE) was used as the main platform for coding and uploading the microcontroller firmware. Before beginning development, necessary libraries were installed, including the DHT sensor library, Adafruit MLX90614 library, and Wi-FiNINA for network connectivity. The board support for the Arduino Nano 33 IoT was added via the Board Manager. Additionally, the Blynk IoT platform was selected to serve as a cloud interface, providing remote data monitoring and visualization. The Blynk library was installed using the Library Manager, and a new project was created in the Blynk web console for dashboard configuration. Below are the details and figures to show the installation. The board manager was also updated to include the Arduino SAMD boards package to enable support for the Nano 33 IoT.

Table 5 shows detail for Arduino IDE and Library Setup

Main Libraries	Description
DHT sensor library	Arduino library for DHT11, DHT22
Adafruit MLX90614	Arduino library for MLX90614 sensors in the Adafruit shop
Wi-FiNINA	Enables network connection (local and Internet) with the Nano 33 IoT.
Blynk library	For cloud dashboard integration

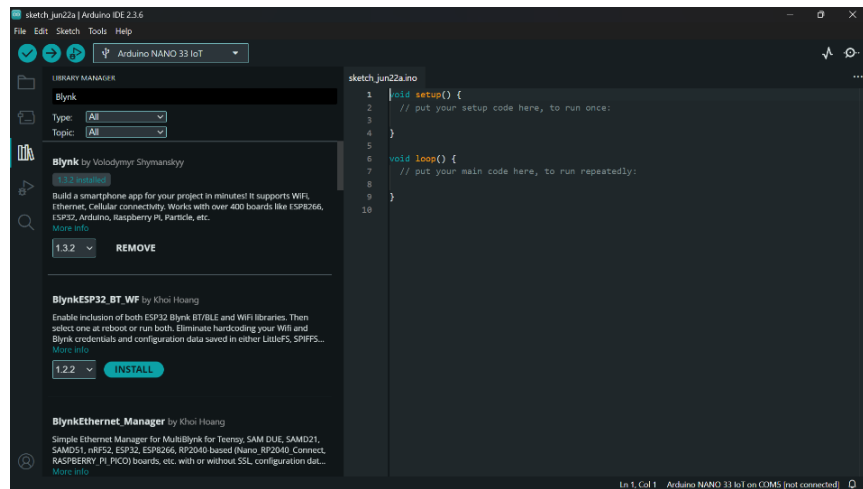


Figure 18 show Arduino IDE for configuration

4.2.1 WiFi Configuration and Security

WiFi credentials were stored in a separate header file (secrets.h) for security purposes. This approach ensures sensitive network information remains protected and separate from the main code. The credentials were configured to establish internet connectivity, enabling data transmission to the Blynk cloud platform.

4.3 Hardware Setup and Sensor Integration

4.3.1 Pin Assignments and Power Configuration

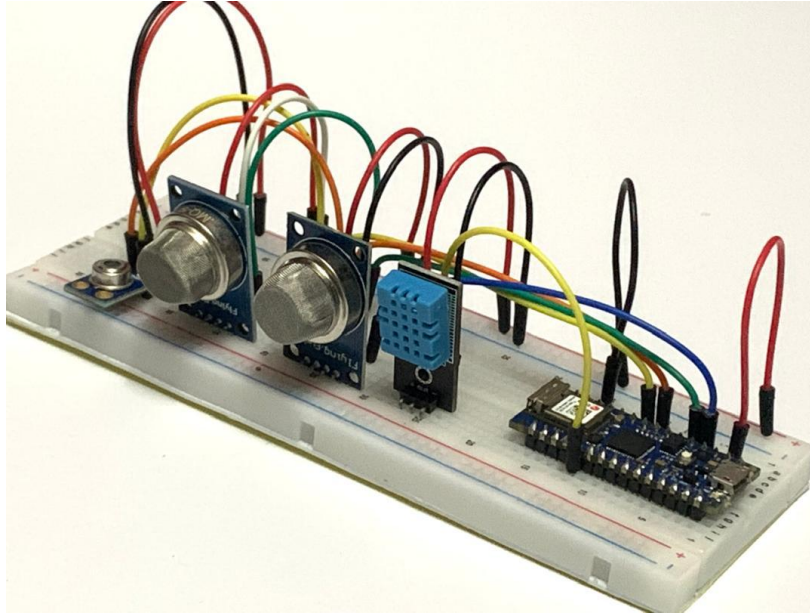


Figure 19 Show wiring and pin assignment

The smart hospital monitoring system was built using the Arduino Nano 33 IoT microcontroller as the central processing unit. This board was selected for its built-in Wi-Fi capabilities and compatibility with IoT applications. The main hardware components integrated into the prototype include:

Table 6 Details of components integrated

Sensors	Functionality
DHT11	For temperature and humidity sensing.
MQ-9	for sensing carbon monoxide and flammable gases
MQ-135	for detecting air quality levels such as ammonia and carbon dioxide
MLX90614	A non-contact infrared sensor for measuring body or object temperature

All sensors were connected via jumper wires to a solderless breadboard and powered using the Arduino's 3.3V or 5V output depending on the sensor specifications. The analog outputs of the MQ-135 and MQ-9 sensors were connected to analog pins A0 and A1, respectively. The DHT11 digital output was connected to digital pin D2, while the MLX90614 used I2C communication on pins A4 (SDA) and A5 (SCL).

Power distribution was managed according to manufacturer specifications, with sensors powered using either 3.3V or 5V pins as required. A breadboard and jumper wires were utilized for prototyping and connection management.

4.3.2 DHT11 Temperature and Humidity Sensor Integration

The DHT11 sensor was integrated to measure ambient temperature and humidity conditions. The sensor was configured to provide readings every 2 seconds, ensuring continuous environmental monitoring. Data calibration was performed by comparing readings with standard reference instruments to validate measurement accuracy.

Table 7 Pin wiring for DT11

Wiring	
VCC	3.3V
GND	GND
Data	Digital Pin D2

4.3.3 MQ-135 Air Quality Sensor Integration

The MQ-135 sensor was implemented to detect various gases including CO₂, ammonia, and benzene. Connected to analog pin A0, the sensor outputs analog voltage values corresponding to gas concentration levels. A preheating period was incorporated into the setup to ensure accurate measurements, as the sensor requires thermal stabilization for optimal performance.

Wiring	
VCC	3.3V
GND	GND
AOUT	Analog Pin A0

4.3.4 MQ-9 Carbon Monoxide and Gas Sensor Integration

The MQ-9 sensor was integrated to detect carbon monoxide and other combustible gases, connected to analog pin A1. To achieve optimal accuracy, a warm-up period of at least 24 hours was observed before taking measurements, allowing the sensor's heating element to stabilize and provide reliable readings.

Table 8 Pin wiring for MQ-9

Wiring	
VCC	3.3V
GND	GND
AOUT	Analog Pin A1

4.3.5 MLX90614 Infrared Temperature Sensor Integration

The MLX90614 non-contact infrared sensor was implemented for measuring object temperatures, particularly suitable for medical applications such as forehead temperature measurement. The sensor utilizes I2C communication protocol and was integrated using the Adafruit MLX90614 library, providing accurate temperature readings without physical contact.

Table 9 Pin wiring for MLX90614

Wiring (I2C interface)	
VCC	3.3V
GND	GND
SDA	A4
SCL	A5

4.4 Programming and Embedded Logic

The embedded program was developed in C++ using the Arduino IDE, implementing a comprehensive environmental monitoring system for hospital rooms. The system architecture follows a modular approach with distinct initialization, data acquisition, processing, and communication phases.

4.4.1 System Initialization and Setup

The system initialization occurs in the setup () function, where all sensors are properly configured and communication protocols are established. The DHT11 temperature and humidity sensor is initialized using the DHT library, while the MLX90614 infrared temperature sensor is configured through the Adafruit MLX90614 library. The MQ-135 air quality sensor and MQ-9 carbon monoxide sensor are configured as analog inputs on pins A0 and A1 respectively.

WiFi connectivity is established using the WiFiNINA library with predefined network credentials, followed by Blynk cloud platform initialization using the authentication token. This ensures reliable data transmission to the cloud dashboard for remote monitoring capabilities.

```
46 void setup() {  
47   Serial.begin(9600);  
48   delay(1000);  
49  
50   dht.begin();  
51   mlx.begin();  
52  
53   Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);  
54   Serial.println("Smart Hospital Room Monitor Started");  
55 }  
56
```

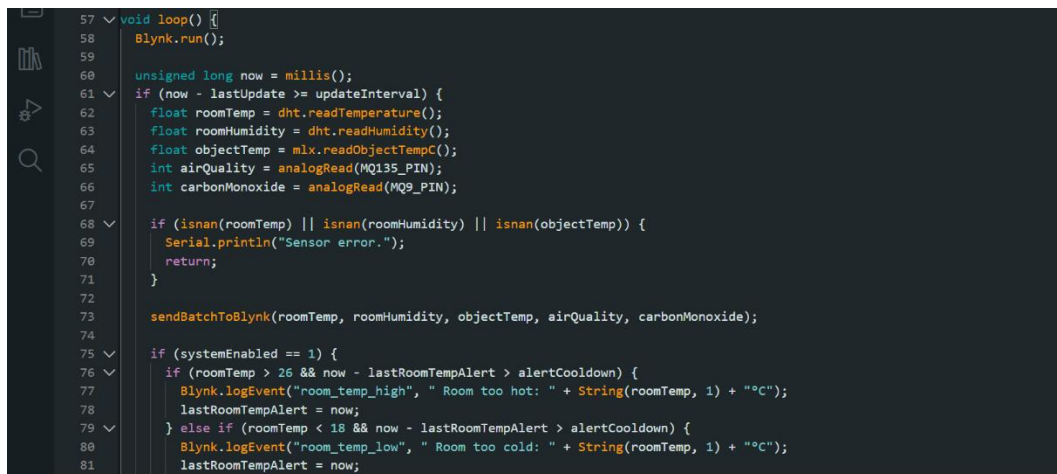
Figure 20 Shows the System Initialization snippet code

4.4.2 Continuous Data Monitoring Loop

The main monitoring logic is implemented in the loop() function, which operates on a time-based scheduling system with a 60-second update interval. This approach prevents sensor oversampling and reduces power consumption while maintaining adequate monitoring frequency for hospital environmental conditions.

The monitoring process follows this sequence:

- a) Data Acquisition: Sensor readings are collected simultaneously from all monitoring devices
- b) Data Validation: Built-in error checking using `isnan()` functions to detect sensor failures
- c) Data Transmission: Batch upload to Blynk cloud platform using HTTP GET requests
- d) Alert Processing: Conditional logic evaluation against predefined thresholds
- e)
- f) Timing Control: Update interval management to ensure consistent monitoring cycles



```
57 void loop() {  
58   Blynk.run();  
59  
60   unsigned long now = millis();  
61   if (now - lastUpdate >= updateInterval) {  
62     float roomTemp = dht.readTemperature();  
63     float roomHumidity = dht.readHumidity();  
64     float objectTemp = mlx.readObjectTempC();  
65     int airQuality = analogRead(MQ135_PIN);  
66     int carbonMonoxide = analogRead(MQ9_PIN);  
67  
68     if (isnan(roomTemp) || isnan(roomHumidity) || isnan(objectTemp)) {  
69       Serial.println("Sensor error.");  
70       return;  
71     }  
72  
73     sendBatchToBlynk(roomTemp, roomHumidity, objectTemp, airQuality, carbonMonoxide);  
74  
75     if (systemEnabled == 1) {  
76       if (roomTemp > 26 && now - lastRoomTempAlert > alertCooldown) {  
77         Blynk.logEvent("room_temp_high", " Room too hot: " + String(roomTemp, 1) + "°C");  
78         lastRoomTempAlert = now;  
79       } else if (roomTemp < 18 && now - lastRoomTempAlert > alertCooldown) {  
80         Blynk.logEvent("room_temp_low", " Room too cold: " + String(roomTemp, 1) + "°C");  
81         lastRoomTempAlert = now;  
82       }  
83     }  
84   }  
85 }
```

Figure 21 Shows the continous loop snippet code

4.4.3 Data Filtering and Noise Reduction

Although the current implementation uses direct sensor readings, the system architecture supports advanced data filtering techniques. The modular design allows for easy integration of averaging algorithms, moving window filters, or Kalman filtering to reduce sensor noise and improve measurement accuracy. The 60-second sampling interval naturally provides some filtering by avoiding high-frequency noise typical in analog sensors.

4.4.4 Threshold-Based Alert System

The system implements a sophisticated alert mechanism based on WHO and healthcare facility standards. Critical thresholds are defined as follows:

- a) Room Temperature: Alerts triggered below 18°C (too cold) or above 26°C (too hot)
- b) Humidity: Alert activated when levels exceed 70% (promoting bacterial growth)
- c) Body Temperature: Fever detection at 38°C using contactless infrared measurement
- d) Carbon Monoxide: Critical alert at MQ-9 sensor value above 1000
- e) Air Quality: Monitored through MQ-135 sensor for general air contamination

```

73 sendBatchToBlynk(roomTemp, roomHumidity, objectTemp, airQuality, carbonMonoxide);
74
75 if (systemEnabled == 1) {
76     if (roomTemp > 26 && now - lastRoomTempAlert > alertCooldown) {
77         Blynk.logEvent("room_temp_high", " Room too hot: " + String(roomTemp, 1) + "°C");
78         lastRoomTempAlert = now;
79     } else if (roomTemp < 18 && now - lastRoomTempAlert > alertCooldown) {
80         Blynk.logEvent("room_temp_low", " Room too cold: " + String(roomTemp, 1) + "°C");
81         lastRoomTempAlert = now;
82     }
83
84     if (roomHumidity > 70 && now - lastHumidityAlert > alertCooldown) {
85         Blynk.logEvent("humidity_high", " Humidity too high: " + String(roomHumidity, 1) + "%");
86         lastHumidityAlert = now;
87     }
88
89     if (objectTemp > 38 && now - lastBodyTempAlert > alertCooldown) {
90         Blynk.logEvent("body_temp_high", " Fever alert: " + String(objectTemp, 1) + "°C");
91         lastBodyTempAlert = now;
92     }
93
94     if (carbonMonoxide > 1000 && now - lastCOAlert > alertCooldown) {
95         Blynk.logEvent("co_high", " CO Level Critical: MQ-9 = " + String(carbonMonoxide));
96         lastCOAlert = now;
97     }
98 }
99

```

Figure 22 Shows snippet code for threshold set

4.4.5 Smart Alert Management

To prevent alert fatigue and system spam, a cooldown mechanism is implemented with a 5-minute interval between repeated alerts for the same parameter. This ensures that healthcare staff receive important notifications without being overwhelmed by continuous alerts for persistent conditions. Each alert type maintains its own timestamp tracking using variables like lastRoomTempAlert, lastHumidityAlert, and lastBodyTempAlert.

```

39
40 // Last alert times
41 unsigned long lastRoomTempAlert = 0;
42 unsigned long lastHumidityAlert = 0;
43 unsigned long lastBodyTempAlert = 0;
44 unsigned long lastCOAlert = 0;
45

```

Figure 23 shows snippet code for alert management

4.4.6 System Control and Remote Management

The system incorporates remote enable/disable functionality through Blynk virtual pin V5, allowing healthcare administrators to control monitoring operations remotely. When systemEnabled is set to 0, the system continues data collection and transmission but suppresses alert generation, useful during maintenance or non-critical periods.

```
15 // Blynk ON/OFF control
16 int systemEnabled = 0;
17 BLYNK_WRITE(V5) {
18   systemEnabled = param.asInt();
19 }
```

Figure 24 Shows snippet code for system control

4.4.7 Robust Communication Protocol

Data transmission utilizes Blynk's batch update API, combining all sensor readings into a single HTTP request to minimize network overhead and improve reliability. The batch update format efficiently maps sensor data to virtual pins:

Table 10 Virtual Pin Assignment

Virtual Pin	Name
V0	Room temperature (°C)
V1	Humidity Room (%)
V2	Body/object temperature (°C)
V3	Air quality index (MQ-135 reading)
V4	Carbon monoxide level (MQ-9 reading)

Error handling includes HTTP response code checking and serial output for debugging purposes, ensuring system administrators can monitor communication status and diagnose connectivity issues.

```
104 void sendBatchToBlynk(float t, float h, float bodyT, int co2, int co) {
105     String url = "/external/api/batch/update?token=" + String(BLYNK_AUTH_TOKEN);
106     url += "&v0=" + String(t, 1); // room temperature
107     url += "&v1=" + String(h, 1); // humidity
108     url += "&v2=" + String(bodyT, 1); // body temp
109     url += "&v3=" + String(co2); // air quality
110     url += "&v4=" + String(co); // CO level
111
112     client.get(url);
113     int statusCode = client.responseStatusCode();
114     String response = client.responseBody();
115
116     Serial.println("=== Blynk Batch Update ===");
117     Serial.print("Status Code: "); Serial.println(statusCode);
118     Serial.print("Response: "); Serial.println(response);
119 }
120
```

Figure 25 Shows snippet code for HTTP response

4.4.8 Event-Driven Alert Notifications

The system uses Blynk's `logEvent()` function to generate structured alert messages with descriptive text and emoji indicators for quick visual identification. Each alert includes the specific sensor reading that triggered the condition, enabling rapid response by healthcare personnel. Alert messages are categorized by severity using visual indicators:

- a) for temperature warnings
- b) for humidity alerts
- c) for fever detection
- d) for critical CO levels

This comprehensive embedded logic ensures reliable, continuous monitoring of hospital environmental conditions while providing intelligent alert management and robust communication capabilities essential for healthcare applications.

4.5 Cloud Integration and Blynk Dashboard

The Blynk IoT Cloud platform serves as the central hub for remote monitoring and user interaction in the smart hospital environmental monitoring system. This cloud-based approach enables healthcare professionals to access real-time environmental data from anywhere, ensuring continuous oversight of critical hospital room conditions.

4.5.1 Device Configuration and Authentication

In the Blynk console, a dedicated device template was created specifically for the hospital monitoring application using a template ID with the descriptive name "Smart Hospital Environmental Monitoring". The Arduino board was securely linked to the cloud platform using a unique authentication token (BLYNK_AUTH_TOKEN), ensuring encrypted communication and preventing unauthorized access to sensitive healthcare data.

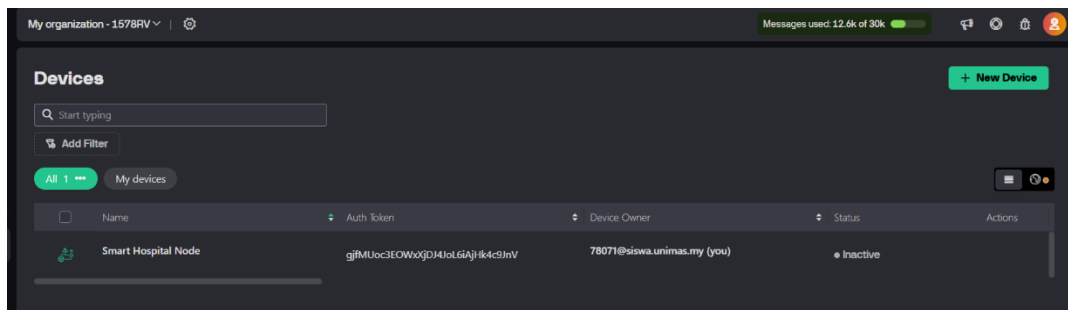


Figure 26 Shows snippet to get authentication

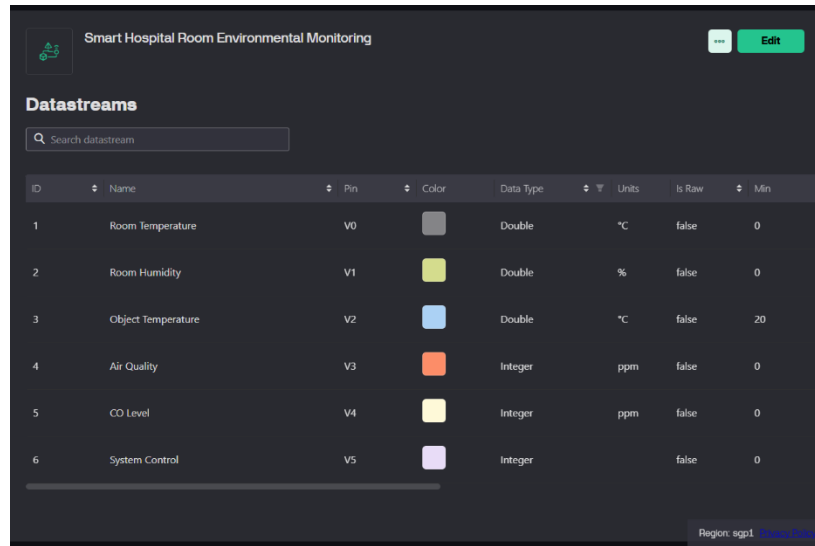
The device configuration process involved defining the hardware specifications, communication protocol (WiFi), and data update frequencies. This template-based approach allows for easy replication and deployment across multiple hospital rooms while maintaining consistent monitoring standards.

4.5.2 Virtual Pin Architecture and Data Mapping

The system utilizes Blynk's virtual pin architecture to create a standardized data communication interface. Five virtual pins (V0 to V4) were strategically assigned to different environmental parameters:

Table 11 Virtual Pin Architecture

Virtual Pin	Name	Sensor
V0	Room Temperature (°C)	DHT11 sensor data
V1	Room Humidity (%)	DHT11 sensor data
V2	Body/Object Temperature (°C)	MLX90614 infrared sensor data
V3	Air Quality Index	MQ-135 sensor readings
V4	Carbon Monoxide Level	MQ-9 sensor readings



The screenshot shows a Blynk dashboard titled "Smart Hospital Room Environmental Monitoring". It features a "Datastreams" section with a search bar and a table listing six datastreams. Each row includes an ID, Name, Pin, Color, Data Type, Units, Is Raw, and Min value.

ID	Name	Pin	Color	Data Type	Units	Is Raw	Min
1	Room Temperature	V0	Grey	Double	°C	false	0
2	Room Humidity	V1	Yellow	Double	%	false	0
3	Object Temperature	V2	Blue	Double	°C	false	20
4	Air Quality	V3	Orange	Integer	ppm	false	0
5	CO Level	V4	Yellow	Integer	ppm	false	0
6	System Control	V5	Purple	Integer		false	0

At the bottom right, it says "Region: esp1" and "Blynk Cloud".

Figure 27 Shows snippet virtual pin setup

Additionally, virtual pin V5 serves as a remote-control interface, allowing authorized personnel to enable or disable the alert system remotely. This architecture provides flexibility for future expansion and ensures organized data management.

4.5.3 Dashboard Visualization and Widget Configuration

The Blynk dashboard incorporates multiple visualization widgets optimized for healthcare monitoring applications. Real-time sensor values are displayed through:

- I. Gauge Widgets: Providing analog-style displays for temperature and humidity with clear numerical readings and color-coded ranges
- II. Labeled Value Displays: Showing precise digital readouts for all sensor parameters
- III. Time-Series Charts: Enabling trend analysis and historical data visualization

IV. LED Indicators: Providing instant visual status confirmation for system operation and alert conditions

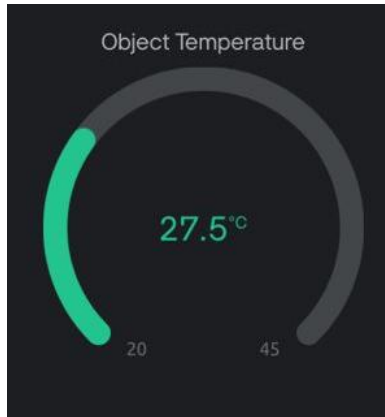


Figure 28 Snippet of Gauge Design in App

The dashboard interface is accessible through both the Blynk mobile application and web browser, ensuring healthcare professionals can monitor conditions using their preferred platform. The responsive design adapts to different screen sizes while maintaining readability and functionality.

4.5.4 Color-Coded Status System

To ensure rapid interpretation by healthcare professionals, the dashboard implements a comprehensive color-coding system based on medical facility standards:

Green Indicators: Normal operating conditions within safe ranges :

- I. Temperature: 18-26°C
- II. Humidity: Below 70%
- III. Body Temperature: Below 38°C

Yellow Indicators: Warning conditions approaching threshold limits:

- I. Red Indicators: Critical conditions requiring immediate attention
- II. High fever detection ($>38^{\circ}\text{C}$)
- III. Dangerous CO levels
- IV. Extreme temperatures or humidity conditions

This intuitive visual system allows for quick assessment of room conditions, even during high-stress emergency situations.



Figure 29 Snippet of color-coded UI in Web

4.5.5 Data Transmission Protocol and Real-Time Monitoring

The system employs Blynk's batch update API to optimize data transmission efficiency. Instead of individual pin updates, all sensor readings are consolidated into a single HTTP GET request sent every 60 seconds. This approach reduces network overhead, minimizes power consumption, and ensures synchronized data updates across all dashboard widgets.

The batch transmission format follows the structure:

```
/external/api/batch/update?token=AUTH_TOKEN&v0=temp&v1=humidity&v2=bodytemp&v3=airquality&v4=co
```

This update frequency provides near real-time monitoring while balancing accuracy with system resources. The 60-second interval is optimal for environmental monitoring as it captures meaningful changes while avoiding unnecessary data transmission that could overwhelm healthcare networks.

The system incorporates robust error handling mechanisms to ensure reliable data transmission including HTTP response monitoring, comprehensive serial debug output for system administrators, and automatic retry logic with built-in reconnection capabilities for temporary network disruptions.

4.5.6 Error Handling and System Monitoring

The system implements essential error handling mechanisms to ensure data integrity and system reliability:

- I. Sensor Validation: Built-in error checking using `isnan()` functions to detect invalid sensor readings from the DHT11 temperature/humidity sensor and MLX90614 infrared temperature sensor

- II. Communication Monitoring: HTTP response status codes are captured and logged via serial output for debugging and system health monitoring purposes

- III. Graceful Error Recovery: When sensor errors are detected, the system skips the current update cycle and continues monitoring in the next scheduled interval, preventing invalid data transmission

The error handling process follows a fail-safe approach where sensor malfunctions are detected before data transmission, ensuring that only valid environmental readings are sent to the cloud dashboard. This prevents false alerts and maintains data quality for healthcare decision-making.

4.5.7 Intelligent Alert System and Notification Management

The Blynk platform's event logging system (`Blynk.logEvent()`) creates structured, categorized alerts with descriptive event names for clear identification of alert conditions (e.g., "room_temp_high", "body_temp_high"). Each alert includes contextual messages with specific sensor readings and visual indicators using emoji-based quick recognition systems for rapid response.

To prevent alert fatigue while ensuring critical notifications reach user, the system implements a sophisticated 5-minute cooldown period for each alert type. This mechanism prevents repetitive notifications for persistent conditions while maintaining alert effectiveness for new threshold breaches, allowing independent timing for different sensor parameters.

The integrated alert system delivers notifications through multiple channels:

- I. Mobile Push Notifications: Instant alerts sent directly to healthcare staff smartphones via the Blynk mobile app
- II. Web Dashboard Alerts: Real-time notification display for staff monitoring web-based interfaces
- III. Historical Alert Logging: Complete audit trail of all alert events for compliance and analysis purposes

4.5.8 Remote System Control and Administrative Features

The dashboard includes comprehensive administrative controls accessible to authorized personnel through virtual pin V5, enabling remote control of alert generation while maintaining continuous data collection. This feature allows healthcare administrators to disable notifications during maintenance periods or non-critical operational phases without interrupting environmental monitoring.

The system provides real-time status monitoring with live system health indicators showing sensor status, connectivity metrics, and operational performance. Administrative personnel can monitor communication health through detailed logging systems that track HTTP response codes, data transmission success rates, and system uptime statistics.

This comprehensive cloud integration approach provides healthcare facilities with a professional-grade environmental monitoring solution that meets the demanding requirements of medical environments. The system ensures reliable operation, intuitive user interfaces, and robust alert mechanisms while maintaining the flexibility needed for diverse hospital room configurations and operational requirements.

4.6 System Testing and Validation

Before final integration, each sensor was tested independently. DHT11 was validated using a calibrated thermometer in a controlled environment. The MQ sensors were tested by exposing them to gas sources such as incense and alcohol vapors. MLX90614 was validated using both ambient room temperature and body temperature measured at different distances.

After confirming individual functionality, all components were integrated and tested together. Observations were made regarding accuracy, stability, and reliability. The WiFi connectivity was also evaluated by disconnecting and reconnecting the router to test the reconnection logic, which performed reliably.

4.7 Summary

This chapter has detailed the systematic development and implementation of the Smart Hospital Room Environmental Monitoring System. It described the installation of software tools, configuration of the microcontroller and sensors, development of embedded logic, cloud integration with Blynk IoT, and testing procedures. The successful combination of sensors and cloud services resulted in a functional prototype capable of continuously monitoring and reporting environmental conditions in real-time. The next chapter will evaluate the system's performance in more detail and discuss its effectiveness and areas for future improvement.

CHAPTER 5: TESTING AND EVALUATION

5.1 Introduction

After the successful development and implementation of the Smart Hospital Room Environmental Monitoring system, it was essential to perform a series of comprehensive tests to verify that the system operates effectively under both normal and abnormal conditions. This chapter outlines the various testing phases conducted to assess the functionality, accuracy, responsiveness, and reliability of the system.

The system integrates multiple sensors and communicates data in real time to a cloud-based IoT platform (Blynk). Since it is designed to operate in a medical environment where patient safety and comfort are critical, ensuring that all components work correctly was a top priority. Therefore, both functional testing to verify the core features and non-functional testing to measure performance, response time, and reliability were conducted. The results and analysis presented in this chapter provide evidence that the system performs reliably and can be considered for real-world applications in hospital or healthcare settings.

5.2 Functional Testing

Functional testing is conducted to determine whether the software and hardware components of the system function as specified in the design. In this project, functional testing involved verifying whether the sensors (DHT11, MLX90614, MQ-135, MQ-9) correctly measured environmental conditions and whether this data was successfully transmitted to the Blynk dashboard in real-time. It also included verifying whether the system triggers the appropriate alert notifications when certain thresholds are exceeded.

The testing was done by simulating various real-world conditions inside a controlled room environment. Temperature was raised using a hair dryer, gas and air quality sensors were tested using a gas lighter and perfume spray, and humidity changes were created using an air humidifier and dry air from a fan. This process helped evaluate how accurately and quickly the system responded to sudden changes.

5.2.1 Test Cases

This section outlines the key test cases conducted to verify the system's main functionalities. Each test targets a critical aspect of the Smart Hospital Room Environmental Monitoring system including sensor readings, cloud connectivity, real-time alert triggering, and user interface updates.

Table 12 Test System Startup

Test Case ID		TC1			
Module Name		System Startup			
Test Title		Sensor Initialization and Blynk Connection			
Descriptions		Test if all sensors initialize correctly and system connects to Blynk cloud on boot.			
Pre-Condition		Arduino Nano 33 IoT is powered on.			
Dependencies		Arduino Nano 33 IoT has a or connected to a power source			
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Power on the device		Sensors initialize, and Blynk shows online status.	Sensors and Blynk connected successfully.	PASS
2	Waiting for system initialization		Sensors initialize, and Blynk shows online status.	Sensors and Blynk connected successfully.	PASS

Table 13 Test Case for High Temperature Alert

Test Case ID		TC2			
Module Name		MLX90614 Sensor			
Test Title		High Temperature Alert			
Descriptions		Test if an alert is triggered when temperature exceeds 37.5°C.			
Pre-Condition		Sensor is active and room temp < 37.5°C.			
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Simulate temp over threshold using a heat source		Blynk dashboard displays alert and notification is sent.	Alert triggered and visible on Blynk.	PASS
Post-Condition		Step 2. A Unique ID is automatically assigned and the Display Name the user entered is registered into the database. Step 3. The user enters the gameplay mode.			

Table 14 Humidity Below Safe Range

Test Case ID		TC3			
Module Name		DHT11 Sensor			
Test Title		Humidity Below Safe Range			
Descriptions		Test alert when humidity drops below 40%.			
Pre-Condition		Humidity in safe range.			
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Simulate dry air using fan or dehumidifier.		Blynk shows alert for low humidity.	Alert triggered correctly	PASS
Post-Condition		Step 2. A Unique ID is automatically assigned and the Display Name the user entered is registered into the database. Step 3. The user enters the gameplay mode.			

Table 15 Test Case for Poor Air Quality Detection

Test Case ID	TC4				
Module Name	MQ-135 Sensor				
Test Title	Poor Air Quality Detection				
Descriptions	Test if system detects poor air quality and sends alert.				
Pre-Condition	Normal indoor air conditions.				
Dependencies	The name entered is not yet registered in the database				
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Spray perfume or smoke near sensor.		The alert displayed on Blynk for poor air.	Warning triggered and visible on dashboard.	PASS

Table 16 Test case for detection of harmful gas levels

Test Case ID	TC5				
Module Name	MQ-9 Sensor				
Test Title	Gas Leakage Alert				
Descriptions	Test detection of harmful gas levels				
Pre-Condition	Safe gas conditions.				
Dependencies	The name entered is not yet registered in the database				
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Place gas lighter near sensor.		CO/gas alert shown on dashboard.	Alert triggered correctly	PASS

Table 17 Test Case for Auto Wi-Fi Reconnection

Test Case ID	TC6				
Module Name	Connectivity				
Test Title	Auto Wi-Fi Reconnection				
Descriptions	Test if system reconnects after Wi-Fi disconnection.				
Pre-Condition	The system is connected to Wi-Fi.				
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Disconnect Wi-Fi temporarily		System reconnects automatically.	Reconnection successful within ~10s.	PASS
2	Wait and monitor		System reconnects automatically.	Reconnection successful within ~10s.	PASS

Table 18 Real-Time Sensor Data Display

Test Case ID	TC7				
Module Name	Blynk IoT Platform				
Test Title	Real-Time Sensor Data Display				
Descriptions	Test if dashboard updates sensor values in real-time.				
Pre-Condition	All sensors online.				
Step	Test Procedures	Test Data	Expected Result	Actual Result	Status
1	Observe dashboard while changing environment		Values update in <2 seconds delay	Real-time updates observed	PASS

5.3 Non-Functional Testing

While functional testing ensures that the system works as expected, non-functional testing evaluates aspects like system performance, accuracy of sensor readings, responsiveness of the interface, power reliability, and overall user experience. These factors are critical in a hospital setting where time-sensitive information and user-friendly interfaces are essential for healthcare professionals.

5.3.1 Pre-Test and Post-Test Setup

To simulate realistic conditions, a small room was set up with the prototype running on a stable 5V power supply. Initial readings were recorded to establish baseline values under normal conditions. Several test cycles were conducted where each parameter was altered one at a time (temperature, humidity, gas presence), and the response of the system was monitored.

Pre-test setup included:

Table 19 Pre-test setup

Pre-test setup
Sensor calibration and confirmation of correct pin connections
Stable Wi-Fi connection and Blynk project initialized
Reference devices used: thermometer, hygrometer, and portable air quality detector

Post-test evaluation included:

Table 20 Post-test setup

Post-test setup
Logging sensor data.
Monitoring Blynk response time.
Measuring deviation from reference instruments.

5.3.2 Analysis of Test Results

The analysis revealed that the system performed well under stress conditions. The table and charts below summarize the accuracy and responsiveness of the sensors.

Table 21 Shows analysis of test results

Parameter	Reference Device Reading	System Reading	Deviation (%)
Room Temperature	29.8°C	30.2°C	1.34%
Humidity	42%	40.5%	3.57%
Air Quality (Clean)	Good	Good	0%
Air Quality (Spray)	Poor	Poor	0%
Gas Detection (Butane)	Present	Alert triggered	0%

Key Observations:

The testing demonstrated excellent sensor responsiveness across multiple components. Both the MLX90614 temperature sensor and MQ gas sensors exhibited rapid response times with delays under 2 seconds, making them well-suited for real-time monitoring applications. The DHT11 humidity sensor showed minor fluctuations in readings, which falls within expected parameters for this cost-effective sensor model. These variations do not compromise the overall system reliability and remain acceptable for the intended monitoring purposes. Network connectivity proved robust through comprehensive testing of the Wi-Fi reconnection system. The automated reconnection mechanism successfully restored network connections after interruptions without requiring manual intervention, ensuring continuous operation in critical monitoring scenarios. The Blynk platform interface delivered responsive performance throughout testing, confirming its suitability for hospital-grade monitoring applications where real-time data visualization and alerts are essential for patient safety and operational efficiency. These results indicate the system meets the technical requirements for reliable, continuous monitoring with appropriate sensor accuracy and network resilience for healthcare environments.

5.3.3 Power Stability Testing

The system was tested on both USB laptop power and external power banks. Results showed better stability and fewer resets when powered by a high-quality power bank. Continuous operation was maintained for over 6 hours without crashes.

5.4 Summary

In this chapter, the Smart Hospital Room Environmental Monitoring system was rigorously tested under various conditions to ensure it meets its intended functionality and performance standards. Both functional and non-functional testing confirmed that the system can monitor room conditions such as temperature, humidity, air quality, and gas levels reliably and in real-time. The Blynk cloud platform effectively displayed sensor data and triggered alerts without significant delay.

Although the system performed well overall, some areas were identified for future improvement, including sensor accuracy, for example replacing DHT11 with DHT22 or SHT31, improving the resolution of Blynk widgets, and adding local data logging for offline scenarios. Nevertheless, the testing phase has demonstrated that this IoT-based system is practical, scalable, and ready for deployment in a medical or healthcare environment where patient comfort and safety are priorities.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Introduction

The Smart Hospital Room Environmental Monitoring system represents a significant step forward in integrating IoT technology into healthcare infrastructure. The system was developed with the aim of creating a safer, more responsive hospital environment by monitoring environmental variables that directly impact patient health and safety. This chapter provides a detailed reflection on the project's outcomes, discusses the extent to which the original objectives were met, acknowledges the limitations encountered during development, and outlines several promising directions for future enhancement.

6.2 Objectives Achieved

The project began with clearly defined objectives: to monitor environmental parameters in a hospital room using IoT technology, provide real-time data to healthcare staff, trigger alerts for unsafe conditions, and deliver this information through a user-friendly interface. These objectives were successfully achieved through a well-integrated system combining hardware, software, and cloud-based tools.

The Arduino Nano 33 IoT served as the microcontroller, interfacing seamlessly with the DHT11 for temperature and humidity monitoring, the MLX90614 for non-contact body temperature measurement, and the MQ-135 and MQ-9 sensors for detecting harmful gases. Each of these sensors played a critical role in building a comprehensive view of the hospital room environment. The Blynk IoT platform was used as the primary interface for displaying sensor

values and managing user interaction, and it performed reliably throughout the project. Alerts were effectively triggered when air quality levels exceeded WHO-recommended thresholds, and the system's real-time nature allowed for immediate visibility into potentially dangerous situations.

Furthermore, technical enhancements such as automatic Wi-Fi reconnection and the integration of a watchdog timer ensured the system's reliability. These features, while often overlooked, are vital in a hospital setting where any downtime can have serious consequences. The outcome was a fully functional prototype that not only achieved its goals but also laid the groundwork for a scalable, hospital-wide implementation.

6.3 Limitations and Constraints

Despite the system's success, several limitations were observed. One major challenge was related to sensor calibration. While low-cost sensors like MQ-135 and MQ-9 are widely accessible and easy to use, they require periodic calibration to maintain accuracy. Over time, environmental exposure and sensor drift may lead to less reliable readings if not properly managed.

Another constraint was the system's dependence on stable Wi-Fi connectivity. While the reconnection logic was robust, the system's real-time capabilities would still be compromised during extended network outages. In a real hospital environment, network redundancy would need to be addressed to ensure uninterrupted monitoring. Additionally, the system was developed as a prototype for a single room. Scaling it to multiple rooms or even entire hospital floors would require the design of a centralized backend system capable of handling larger data volumes and providing a unified interface for multiple users.

The Blynk platform, although ideal for prototyping, has limitations regarding historical data access and multi-user management on its free tier. Advanced features such as long-term data logging and analytics were not implemented due to these constraints, but they remain highly desirable for future enhancements.

6.4 Future Work

Looking ahead, several opportunities exist to build upon the foundation established by this project. One immediate improvement would be to integrate the system with a hospital's existing digital infrastructure, such as a Hospital Information System (HIS). This integration would allow sensor data to be linked with patient records, enabling a more holistic view of patient health and room conditions.

Another area for future development is scalability. A centralized cloud-based server or dashboard could be introduced to manage data from multiple hospital rooms, allowing administrators to monitor the entire facility from one interface. This would involve implementing secure data communication, user authentication, and real-time alert dashboards accessible from any device.

Additionally, incorporating artificial intelligence and machine learning could enhance the system's ability to identify trends and predict hazardous conditions before they occur. For example, a pattern of increasing CO₂ levels could be used to predict insufficient ventilation or potential equipment malfunction.

Creating a standalone mobile application with role-based access for different users such as nurses, doctors, and IT staff would improve usability and system adoption in clinical settings. Finally, ensuring that the system adheres to medical safety and quality standards, such as ISO 13485 for medical devices, would be essential for future real-world deployment.

6.5 Conclusion

In conclusion, the Smart Hospital Room Environmental Monitoring system demonstrates the powerful potential of IoT technology in advancing healthcare outcomes. By providing real-time insights into environmental conditions, the system supports proactive decision-making and helps mitigate risks associated with poor air quality, temperature imbalances, and undetected gas leaks. Although certain limitations were encountered, the project successfully achieved its core objectives and opened the door for further innovations. With additional enhancements in scalability, analytics, and integration, the system holds great promise as a valuable tool in modern smart healthcare environments.

REFERENCES

- Abdulmalek, S., Nasir, A., Jabbar, W. A., Almuahaya, M. a. M., Bairagi, A. K., Khan, M. A., & Kee, S. (2022). *IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review*. *Healthcare*, 10(10), 1993. <https://doi.org/10.3390/healthcare10101993>
- Bockholt, R., Paschke, S., Heubner, L., Ibarlucea, B., Laupp, A., Janićijević, Ž., Klinghammer, S., Balakin, S., Maitz, M. F., Werner, C., Cuniberti, G., Baraban, L., & Spieth, P. M. (2022). *Real-Time monitoring of blood parameters in the intensive care unit: State-of-the-Art and Perspectives*. *Journal of Clinical Medicine*, 11(9), 2408. <https://doi.org/10.3390/jcm11092408>
- Carling, P. C. (2021). *Health care environmental hygiene*. *Infectious Disease Clinics of North America*, 35(3), 609–629. <https://doi.org/10.1016/j.idc.2021.04.005>
- Chan, F. K., & Thong, J. Y. (2008). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 46(4), 803–814. <https://doi.org/10.1016/j.dss.2008.11.009>
- Docs.arduino.cc. (n.d.-a). <https://docs.arduino.cc/arduino-cloud/cloud-interface/dashboard-widgets/>
- Heseltine, E., & Rosen, J. (2009). *WHO guidelines for indoor air quality: Dampness and Mould*. WHO Regional Office Europe.
- Iqbal, F., Lam, K., Joshi, M., Khan, S., Ashrafian, H., & Darzi, A. (2021). *Clinical outcomes of digital sensor alerting systems in remote monitoring: a systematic review and meta-analysis*. *NPJ Digital Medicine*, 4. <https://doi.org/10.1038/s41746-020-00378-0>.

- Islam, M. M., Rahaman, A., & Islam, M. R. (2020). Development of smart healthcare monitoring system in IoT environment. *SN Computer Science*, 1(3). <https://doi.org/10.1007/s42979-020-00195-y>
- Reddy, N., J.S, M., Kumar, V., & Kampa, M. (2023). *Smart Healthcare Empowerment: Integrating IoT, Cloud, And Healthcare Standards with Raspberry Pi for Enhanced Patientdata Collection. International Journal of Engineering Applied Sciences and Technology*. <https://doi.org/10.33564/ijeast.2023.v08i06.016>.
- Sattar, H., Bajwa, I. S., Ul-Amin, R., Mahmood, A., Anwar, W., Kasi, B., Kazmi, R., & Farooq, U. (2019). *An intelligent and smart environment monitoring system for healthcare. Applied Sciences*, 9(19), 4172. <https://doi.org/10.3390/app9194172>
- Shankarmani, R., Pawar, R., Mantha, S. S., & Babu, V. (2012). Agile methodology adoption: benefits and constraints. *International Journal of Computer Applications*, 58(15), 31–37. <https://doi.org/10.5120/9361-3698>
- Vappangi, S., Penjarla, N. K., Mathe, S. E., & Kondaveeti, H. K. (2022). *Applications of Raspberry Pi in Bio-Technology: A review*. 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP). <https://doi.org/10.1109/aisp53593.2022.9760691>
- World Health Organization. (2021). *WHO global air quality guidelines*. <https://iris.who.int/bitstream/handle/10665/345329/9789240034228-eng.pdf>

- Yuvaraj, V. (2024). Innovative IoT solutions for enhanced patient health monitoring and healthcare efficiency in hospitals. *International Journal for Research in Applied Science and Engineering Technology*, 12(8), 476–483. <https://doi.org/10.22214/ijraset.2024.63948>
- Zhukova, N., Motienko, A., Levonevskiy, D., Subbotin, A., & Anh, P. T. (2023). Smart room for patient monitoring based on IoT technologies. *AICCC '23: Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*, 151–158. <https://doi.org/10.1145/3639592.3639630>

APPENDIX

Appendix A: Similarity check percentages

The screenshot displays the Feedback Studio interface. The top header includes the 'feedback studio' logo, the user name 'IZZAN HAZIRAH BINTI HARUN', and the course 'FYP'. The main content area shows a document with the logo of 'UNIVERSITI MALAYSIA SARAWAK' and the text 'Faculty of Computer Science and Information Technology'. On the right, a 'Match Overview' panel shows a similarity percentage of '1%'. Below this, a table lists the matches:

Match	Source	Similarity
1	ir.unimas.my Internet Source	1%

The bottom status bar indicates 'Page: 1 of 99', 'Word Count: 12397', and options for 'Text-Only Report' and 'High Resolution' (set to 'On').