A project report on

IoT Based Health Monitoring System

Submitted in complete fulfillment of the requirements of

Bachelor of Engineering

in

Electronics and Telecommunication Engineering

by

Diksha Bhosale (3021159)

Pratik Belamkar (3021107)

Varun Somwanshi (3021151)

Mudabbir Ahmad (3021133)

Under the guidance of

Mrs.Akshata Raut

Department of Electronics and Telecommunication Engineering

Fr. C. Rodrigues Institute of Technology, Vashi

University Of Mumbai

2024-2025

BONAFIDE CERTIFICATE

This is to certify that the project titled **IoT based Health Monitoring System** is a bonafide record of the work done by

Diksha Bhosale

Pratik Belamkar

Varun Somwanshi

Mudabbir Ahmad

submitted to the University of Mumbai in partial fulfillment of the requirement for the award confirming completion of **Major Project (REV- 2019 'C' Scheme) of Final Year, (BE Sem-VIII)** in **Electronics and Telecommunication Engineering** as laid down by **University of Mumbai** , during the **academic year of 2024-2025**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

(Akshta Raut) (Dr.Megha Kolhekar)

DECLARATION

We thus certify that the ideas included in this written contribution are entirely original with proper citations and references to any original sources used. Additionally, we affirm that we have followed all guidelines for academic honesty and integrity and that none of the ideas, data, facts, or sources in my work have been misrepresented, created, or falsified. We are aware that any infraction of the aforementioned will result in disciplinary action from the institute, which may even take legal action against sources that have not been properly referenced or from whom appropriate authorization has not been obtained where necessary.

Diksha Bhosale (3021159) Signature:_______________

Pratik Belamkar (3021107) Signature:_______________

Varun Somwanshi (3021151) Signature:_______________

Mudabbir Ahmad (3021133) Signature:_______________

Date:

ABSTRACT

The work that has been proposed and completed on the ”IoT-Based Patient Health Monitoring System” project is described in this paper. The primary objective of this project is to create a dependable, real-time health monitoring system that measures and analyzes critical health indicators, such as heart rate and blood oxygen saturation (SpO_2), using Internet of Things (IoT) technologies. The device detects changes in blood volume using visible and infrared light LEDs in combination with non-invasive photoplethysmography (PPG) techniques. The collected data is processed by a microcontroller before being wirelessly transmitted to a cloud-based platform for remote monitoring and analysis.

The implemented system consists of Internet of Things networking modules, sensors, and microcontroller-based processing. The hardware architecture includes a photodiode sensor, ordinary and infrared LEDs, and a signal processing unit to gather and clean the data. The software design ensures that health metrics are securely uploaded to the cloud, allowing healthcare professionals to access them from any place. The technology can generate warnings in the event of abnormal readings, ensuring that medical emergencies are handled quickly. This innovation would be especially helpful to patients who are elderly, have chronic conditions, or reside in remote areas with little access to hospitals.

The results of the system’s simulation demonstrate the effectiveness of PPG signals for real-time surveillance. SpO_2 and heart rate estimates were transmitted over the network with little latency. The proposed system showed reliable accuracy, cost, and portability when compared to traditional monitoring equipment. The cloud-based dashboard provided an intuitive user interface for real-time patient data display, and data logs were stored for future analysis and review.

When everything is taken into account, the ”IoT-Based Patient Health Monitoring System” offers a practical and expandable method of ongoing patient monitoring. The technology helps bridge the gap between patients and healthcare providers by enabling proactive treatment and decision-making based on real-time health data. It lays the foundation for future advancements such as the incorporation of electrocardiogram (ECG) or body temperature sensors and the use of machine learning to predictive healthcare analytics. The proposed paradigm is a significant step toward developing intelligent healthcare solutions in the age of digital transformation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ADC Analog To Digital Converter

DDR Double Data Rate

DSP Digital Signal Processor

FPGA Field Programming Gate Array

HDL Hardware Description Language

JTAG Joint Test Action Group

LED Light Emitting Diode

PCLK Pixel Clock

PLL Phase Lock Loop

PWM Pulse Width Modulation

SCCB Serial Camera Control Bus

SFM Structure From Motion

UART Universal Asynchronous Receiver Transmitter

VGA Video Graphics Array

XCLK External Clock

XDC Xilinx Design Constraint file

WNS Worst Negative Slack

CHAPTER 1
Introduction

In this study, the ”IoT-Based Patient Health Monitoring System” project is thoroughly and methodically reviewed. This includes the justification of the study, a review of prior similar work, the suggested technique, system implementation, simulation results, comparative analysis, and all other important components of the research. For scholars and students researching related subjects in the fields of healthcare technology and the Internet of Things (IoT), the study is a useful resource because it also describes the system's possible uses and looks into future advancements.

1.1 Problem overview

Remote patient monitoring is being used more and more by the healthcare sector to enhance medical services, especially in cases where 24-hour care is necessary. Traditional health monitoring methods sometimes need large, cumbersome equipment and in-person evaluations, which are inappropriate for long-term or home-based monitoring. This causes a significant delay in the prompt delivery of healthcare, particularly for older patients, patients with chronic conditions, and people residing in remote or impoverished locations. Intelligent health monitoring systems that can continuously and non-invasively track vital parameters like blood oxygen saturation (SpO_2), body temperature, sodium, potassium, kidney creatine, blood pressure, sugar, heart rate, and blood pressure have been made possible by recent developments in the Internet of Things (IoT). These technologies give caretakers real-time notifications and facilitate cloud-based data sharing with medical specialists, allowing for remote diagnosis and prompt intervention.

Studies like [1] have demonstrated that when ESP-series microcontrollers (such the ESP8266) are interfaced with sensors like the MAX30100 and DHT11, real-time Wi-Fi transmission of health data is made possible. This makes healthcare more responsive and accessible by guaranteeing usability and accessibility across several sites.

There are still problems despite these technological developments, especially with data security, sensor calibration, power efficiency, and the use of reliable communication protocols.

With their portable and affordable designs, the two technologies in question aim to bridge the gap between patients and caregivers. Through system enhancement, real-time monitoring, and the addition of features like LabVIEW-based analysis and validation, this project seeks to answer the urgent demand for an IoT-based health monitoring platform that is fully integrated, secure, and clinically reliable..

1.2 Project Objectives

The main objectives of this project are:

1. Create a portable, affordable health monitoring gadget that tracks heart rate and SpO_2 using sensors like the MAX30100.
2. Connect the ESP8266 microcontroller for effective wireless communication, data processing, and collection. Send health information safely to a cloud-based platform using Internet of Things protocols such as Wi-Fi, MQTT, or HTTP.
3. Provide real-time patient data access to medical professionals and caregivers via an easy-to-use user interface (web or mobile application).
4. Install alarm systems to inform physicians or patients of any unusual health data.
5. Encourage remote health monitoring to help older or chronically ill people in rural areas become less dependent on hospitals.
6. Include LabVIEW as an extra platform for waveform visualization, offline signal processing, and real-time system verification.

1.3 Report outline

From conception to implementation and validation, each of the six comprehensive chapters that comprise the framework of this report covers a critical phase of the project.

Chapter 2: Literature Review Current studies on Internet of Things-based patient health monitoring systems are included in this chapter. With an emphasis on sensor integration, cloud connectivity, and data visualization, it condenses the design concepts and implementations of earlier models [1], [2]. Notable drawbacks include insufficient alert systems, restricted movement, and restricted parameter monitoring. The current project has been improved as a result of these deficiencies.

Chapter 3: Sensor Integration and Microcontroller System Design Key hardware elements including the ESP8266 microcontroller, the MAX30100 sensor, and more analog circuits are integrated in this chapter. The architecture facilitates PPG signal recording, amplification, digital processing, and wireless data transmission. There are discussions on design improvements over earlier efforts [3] along with schematics and diagrams.

Chapter 4: Interface for Cloud Monitoring and Communication The communication model utilized for safe MQTT or HTTP transmission of health data is covered in this chapter. The development of an online and mobile interface allows for the display of patient parameters in real time. In order to ensure system accuracy throughout the testing process, LabVIEW is also used to simulate various health conditions and verify signal integrity to guarantee system accuracy during testing. The system is compared with other platforms, such as the one proposed in [4].

Chapter 5: Analysis and Findings This chapter presents the system's accuracy, signal fidelity, latency, important parameters, and performance metrics. The readings are compared using standard clinical tools. LabVIEW is used for parameter display and offline waveform analysis to facilitate the evaluation. Comparative performance metrics from research [5] and [6] are used to benchmark the system.

Chapter 6: Overview and Future Directions This last chapter compiles the project's contributions, system effectiveness, and challenges encountered. Suggestions for future improvements include adding more sensors, improving energy efficiency, and using AI to diagnose patients more accurately. The potential contribution of LabVIEW to future research is also highlighted, especially in the domains of clinical-grade system development and testing [8].

CHAPTER 2

Literature Review

2.1 Introduction to Literature and Its Role in Research

The literature evaluation has a significant impact on the path of any research endeavor. It helps academics understand the past, look into technical advancements, and identify any gaps in the body of previous work. A comprehensive analysis of prior research offers valuable insights into the integration of sensors, communication technologies, real-time monitoring techniques, and cloud platforms in patient health monitoring for this IoT-based health monitoring system project. The evaluation also lays the groundwork for enhancing and creating a system that is more dependable, economical, and successful.

2.2 The Role of IoT in Environmental and Health Monitoring

By allowing objects to connect and operate independently, the Internet of Things (IoT) has greatly enhanced environmental and medical monitoring. In the context of health monitoring, IoT enables remote surveillance of vital indicators such as heart rate, oxygen saturation, temperature, and movement.Studies have demonstrated that wearable and embedded sensor systems, in conjunction with microcontrollers that enable Bluetooth or Wi-Fi, can help doctors and caregivers monitor patients' health in real time [1][2]. This approach is especially beneficial for older patients and those who live in remote areas with little access to conventional care.

2.3 Sensor Integration and Challenges

Sensor integration is a crucial part of IoT-based health systems. Sensors like the DHT11 (for temperature) and MAX30100 (for SpO_2 and heart rate) are often used in research prototypes [3][4]. Despite providing real-time data, these sensors often have problems, including inaccurate readings caused by ambient light or motion artifacts (especially in optical sensors), power consumption problems during continuous monitoring, and a lack of calibration against medical-grade equipment. Even though research has proposed methods such signal filtering and amplifier circuits [5], it is still challenging to integrate several sensors without increasing delay or signal loss.

2.4 Cloud Computing for IoT Data Management

Cloud computing enables the safe storage, remote access, and visualization of health data. Several studies [6], [7] use Firebase, ThingSpeak, and AWS IoT to send patient vitals from microcontrollers to the cloud. There have been applications for protocols such as HTTP and MQTT; MQTT has the benefit of low-latency and low-power communication [8]. However, problems with data security and privacy still exist. Many of the systems under review have insufficient authentication or encryption, putting sensitive health information at risk. Reliable internet connectivity is another crucial factor affecting system performance, especially in developing nations.

2.5 Real-Time Alert Mechanisms and Their Importance

One essential element of modern health monitoring systems is the inclusion of real-time alert mechanisms.These systems provide alerts via email, SMS, or mobile applications when a patient's vital signs meet certain standards. Studies have demonstrated that these alerts enhance patient outcomes by ensuring timely medical reactions [9]. However, certain systems may generate false alarms due to inaccurate sensor data or communication issues. Very few of the systems in use today employ complex filtering or decision-making algorithms before issuing alerts. Consequently, for more precise and dependable operation, smarter alert Systems that take a variety of health factors into account are necessary.

2.6 Identified Gaps and Future Directions

There are some gaps in the literature review. Many systems only monitor one or two health indicators, and few offer clinical-level accuracy or validation. Often, energy economy is overlooked while building portable gadgets. Most prototypes lack advanced signal processing, and alert systems are often very basic and prone to false positives.

2.6.1 Role of LabVIEW in Future Developments

LabVIEW is a tool for offline signal visualization and validation that has been used into some current research [10] to help close the gap between signal analysis and debugging. LabVIEW can: Display temperature readings and waveforms in real time; acquire real-time data from sensors via serial connectivity. Apply logic analysis and signal filtering techniques to improve the quality of your data.

CHAPTER 3

System Design

The Internet of Things-based patient health monitoring system's objective is to routinely assess vital signs including heart rate and blood oxygen saturation (SpO_2). from Figure 3.1 By combining sensors, microcontrollers, and cloud platforms, the system provides real-time monitoring and early identification of possible health problems, reducing delays in medical intervention[1],[4]. Data collected by the system is transmitted to cloud platforms such as AWS via the ESP8266 microcontroller, where it is accessible through web dashboards or mobile devices [3],[6]. Patients with chronic illnesses and the elderly benefit most from this technique. This arrangement facilitates better patient outcomes, ensures timely response, and reduces the burden on the healthcare system [8].

3.1 Wifi Connection

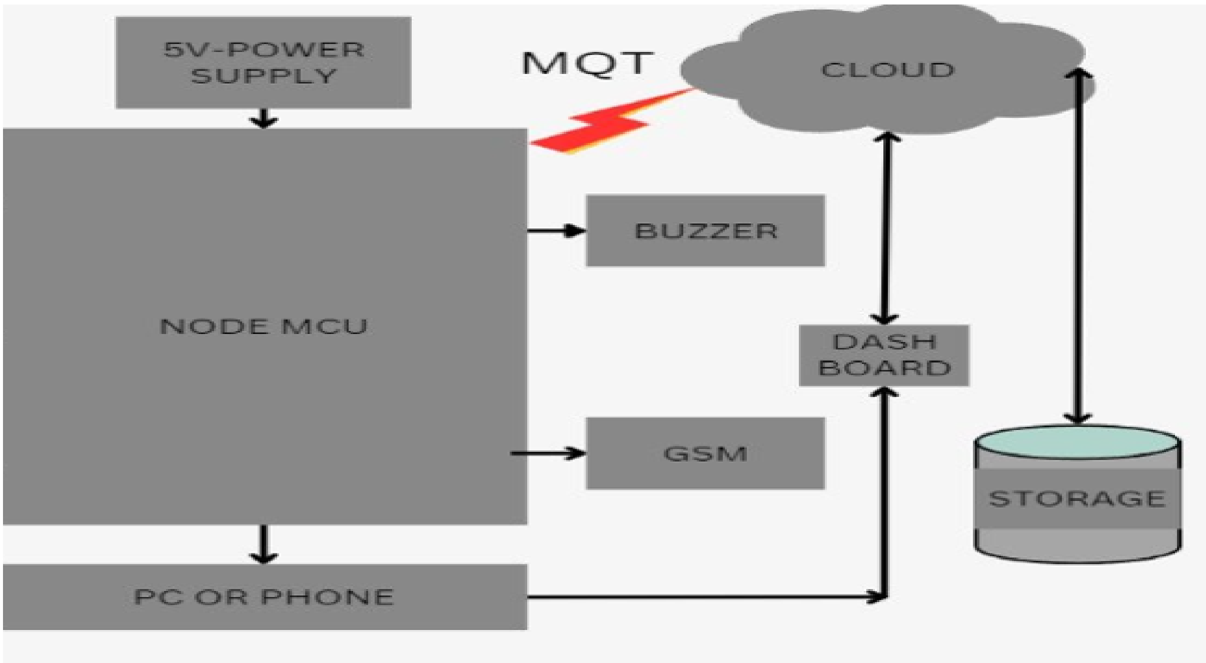

Figure 3.1: WiFi Connection

- **5V-POWER SUPPLY**: This block indicates the power source for the system, providing a stable 5V DC to the NodeMCU.

- **ESP8266**: The ESP8266 is a Wi-Fi-enabled microcontroller used to connect all sensors to the internet. It collects data from the sensors and sends it securely to AWS IoT Core for processing and visualization,

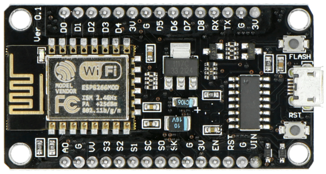

Figure 3.2: ESP8266

making it the central hub of communication and control in the system.

- **DHT11**: The DHT11 is a digital temperature and humidity sensor. It continuously monitors environmental conditions and provides real-time data to the ESP8266.

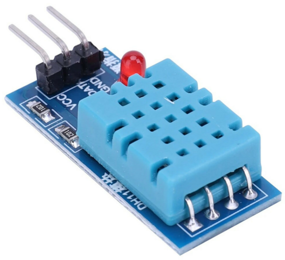

Figure 3.3: DHT11

Its role is crucial in health and environment-related monitoring applications due to its reliability and simplicity in integration.

- **MAX30100**: The MAX30100 is a pulse oximeter and heart-rate sensor. It detects heartbeats and measures blood oxygen saturation (SpO) levels. Information is transmitted to the

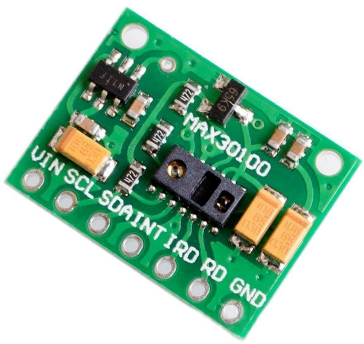

Figure 3.4: MAX30100

microcontroller, helping track vital health signs in real-time, making it ideal for patient or wellness monitoring.

- **Ultrasonic Sensor**: This sensor measures distance using ultrasonic waves. In this project, it checks the level of

Figure 3.5: Ultrasonic Sensor

saline or liquid, helping determine whether the container is full, normal, or near empty, which then triggers alerts as needed."

- **AWS IoT Core**: AWS IoT Core allows the ESP8266 to securely connect to cloud services. It facilitates the real-time transfer of data from the device to AWS, where it may be stored, shown, or analyzed further using other AWS services.

- SNS is used to send email alert alerts using the Simple Notification Service (Amazon SNS). SNS automatically alerts users or caregivers when sensor thresholds are surpassed (e.g., saline empty or aberrant vitals) to ensure timely response and action.

- The Arduino IDE is the development environment used to create, compile, and upload code to the ESP8266. It supports all required libraries and provides serial monitoring for debugging and confirming sensor data and connection.

- **Amazon DynamoDB** Amazon DynamoDB is a fast and scalable NoSQL database solution from AWS. In this project, real-time health and sensor data, including temperature and humidity, heart rate, SpO_2, and liquid levels, are obtained from the ESP8266 via AWS IoT Core and saved in DynamoDB. It ensures efficient data logging, easy retrieval for analysis, and a seamless integration with other AWS services. Its serverless architecture and low latency make it ideal for handling frequent updates from IoT devices.

3.2 Architecture of Project

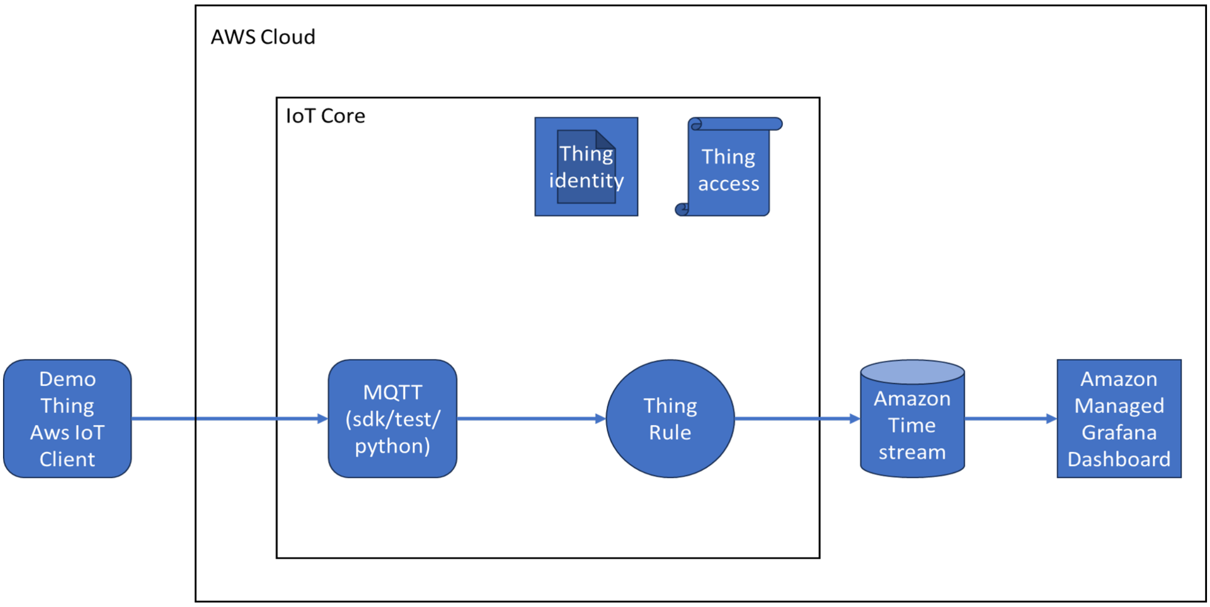

Figure 3.6: Architecture

,**Sample AWS IoT Client** This might be any Internet of Things device or emulator, such as a Raspberry Pi, virtual device, or Python script. It sends messages or sensor data by connecting to AWS IoT Core.

,**MQTT (SDK)** Beginning with MQTT is a lightweight communications protocol used by IoT devices. The client (your device) communicates with AWS IoT Core via an SDK (like the AWS IoT Device SDK), a test console, or a Python MQTT module.The IoT Rules Engine processes incoming MQTT messages. It filters, modifies, and routes data according to rules you provide. In this case, the rule transmits data to Amazon Timestream.

, **Amazon Timestream** A time series database allows for the long-term storage of temperature, humidity, and other data. created with consideration for time-stamped data. retains the sensor data obtained via IoT Rules.

,**Grafana Dashboard for Amazon Managed** Grafana is used to view the data. Amazon Managed Grafana, which displays dashboards in real time, is linked to Timestream. Top Section: Identity Access Identity of Thing: X.509 certificates or keys offer a unique identify for the device. Thing Access: IAM or IoT policies that outline what the Thing can do (such publishing to a topic).

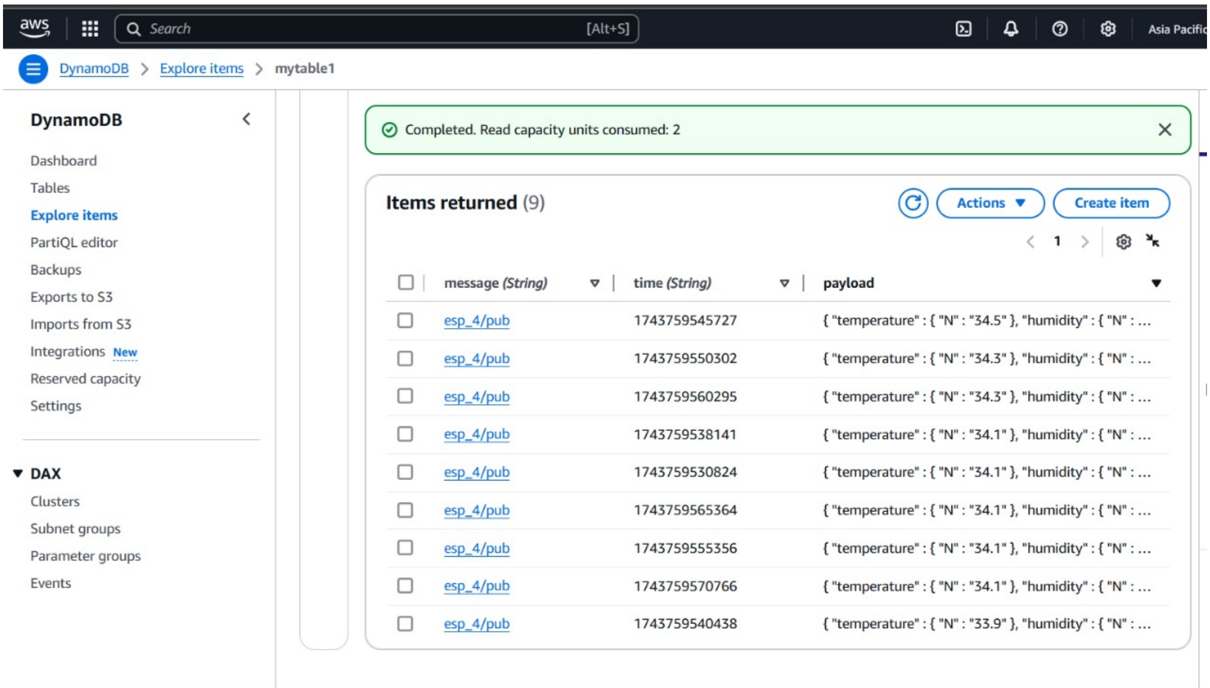

Figure 3.7: Dynamo DB

3.3 AWS DataBase

Step 1: Circuit Configuration and Hardware Selection The ESP8266 NodeMCU microcontroller was selected due of its low power consumption and Wi-Fi functionality. Sensors in the system measure important variables, such as: Digital temperature and humidity sensor DHT11; SpO_2 and heart rate sensor MAX30100

Step 2: Calibration and Testing of Sensors The Arduino IDE was used to test the sensors. Water-based simulation was used to evaluate sensors based on temperature and fluid level. Appropriate calibration and noise filtering were employed to provide reliable results.

Step 3: Firmware Development Code was written using the Arduino IDE to collect sensor data, process it, and normalize the results. Connect to Wi-Fi and transfer data to the cloud. Step 4: Integration with Clouds in Real Time Data was sent to the ThingSpeak cloud platform using the HTTP protocol. Visual dashboards were used to set up data recording and alerts.

Step 5: Monitoring and Testing Several factors were used to test the entire system. Vital sign thresholds were set up to sound an alarm. Known values were used to validate test findings.

3.4 Labview

Step 1: Sensor Integration and Component Selection Important components include oxygen saturation, or SpO_2, a blood test indication. Temperature, blood pressure (BP), potassium, and salt levels The quantity of sugar The objective was to measure kidney function (creatinine). The proper sensors for each parameter were connected to a micro-controller (ESP32-CAM sans camera) on a PCB.

Step 2: Configuring Hardware Numerous analog and digital sensors are linked to the ESP32 microcontroller via jumper wires on a breadboard. Prior to being transmitted to a local interface or the cloud, real-time sensor data is gathered and processed.

Step 3: Calibration and Testing of Sensors Each sensor was inspected to guarantee accuracy and functionality. under carefully monitored conditions (for example, a temperature sensor immersed in water, as seen in Figure 3.3).

Step 4: LabVIEW Data Visualization LabVIEW was used to create a visual dashboard for monitoring sensor data in real time. The serial data from the ESP8266 was fed into LabVIEW using either the VISA protocol or serial communication blocks. Graphical indicators including plots, dials, and warnings were used to provide visual feedback.

Step 5: Alerts and Real-Time Monitoring The system continually tracks the patient's health indicators. If there are abnormal readings, real-time alerts are produced. This functionality may be expanded to include cloud-based logging or mobile alerts.

CHAPTER 4

Cloud Communication and Monitoring Interface

4.1 AWS IoT Core Device Messaging and Connectivity

AWS IoT Core provides the communication interface between your IoT sensors and the cloud. Devices like the ESP8266 broadcast sensor data (temperature, humidity, SpO, etc.) to AWS IoT Core via MQTT. Key attributes used: Device shadows for the virtual depiction of linked sensors.

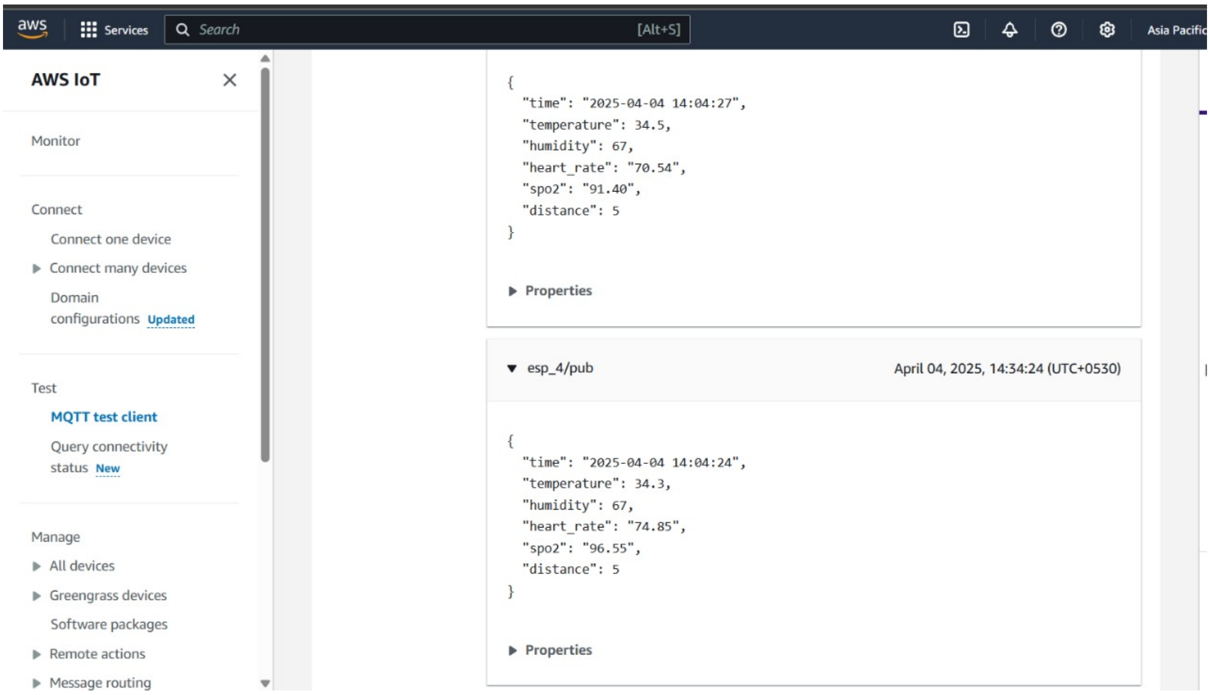

Figure 4.1: AWS IoT Core

MQTT topics for real-time publishing and subscription to sensor values.Secure communication authentication uses X.509 certificates. The ESP8266 sends temperature data (like 33.9°C) to AWS IoT via MQTT every few seconds.

4.2 AWS DynamoDB

Sensor Data Storage DynamoDB, a NoSQL database, stores incoming sensor data in JSON format with timestamps. High-speed, serverless storage options for structured data include: Dynamic database tables provide for quick searches and retrievals for analytics or notifications. Use case: Every field reading is captured for further examination and presentation.

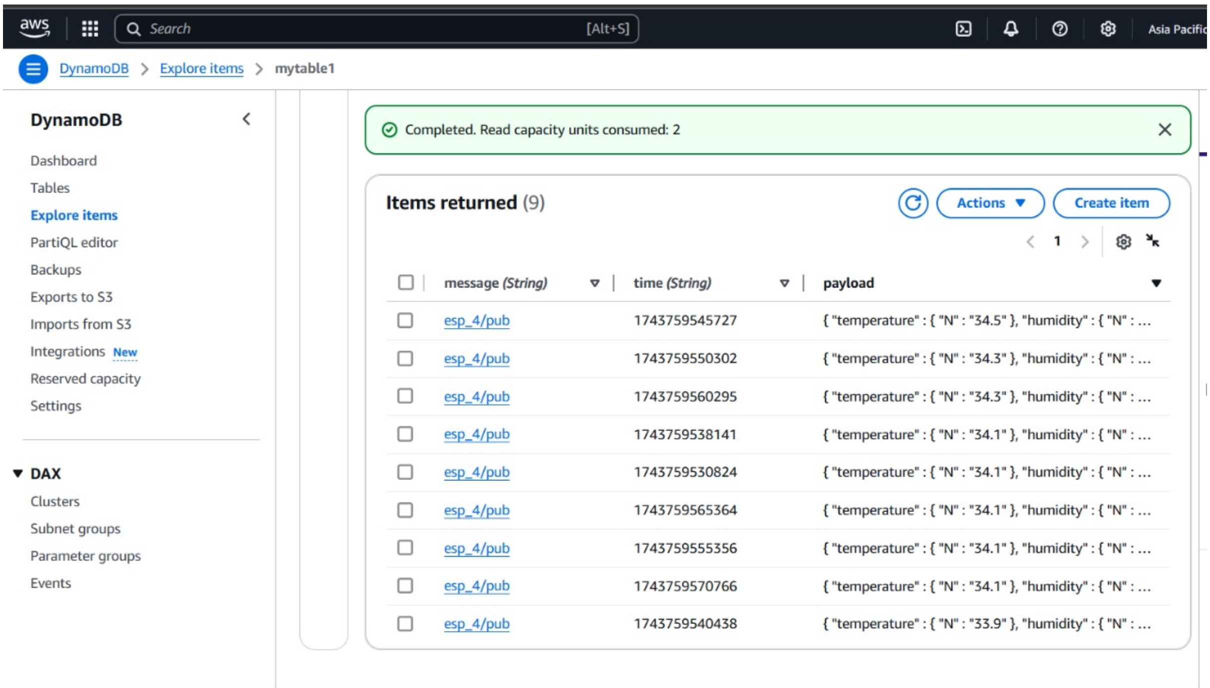

Figure 4.2: Dynamo DataBase

Use case: All field readings are recorded for further analysis and display.

4.3 AWS Lambda

AWS Data Processing and Filtering For example, new MQTT messages might activate the Lambda serverless computing service. Adding information to DynamoDB. functions created using Python or Node.js. JavaScript is used to check sensor data, for as by removing incorrect values like SpO = 0. Employ threshold logic to trigger an alert if the temperature> 38°C. Lambda guarantees lightweight, real-time decision-making. AWS SNS (Simple Notification Service)

4.4 AWS SNS (Simple Notification Service)

System of Warning SNS is used to send email or SMS alerts when sensor readings exceed danger thresholds. Lambda was added for alert delivery. enables multiple alert subscribers, such as a user, doctor, or emergency contact. If the SpO is abnormal or the heart rate drops, an automatic email notice is sent out immediately.

4.5 AWS CloudWatch (Useful Although Optional)

Monitoring and Logging: AWS services, including MQTT traffic and Lambda invocations, are monitored using CloudWatch. capturing error traces in order to debug Lambda functions. ensures the system's stability and lets you keep an eye on data mistakes or abnormalities.

4.6 Identity and Access Management (AWS IAM)

Security IAM roles and policies are configured to: Grant least-privilege access to every component. Secure communication using ESP8266, Lambda, DynamoDB, and SNS. upholds best practices and the security of the system.

CHAPTER 5

Results and Analysis

5.1 Results

5.1.1 Real-time Data Monitoring

The system successfully sent vital health parameters in real time to AWS IoT Core, including temperature, humidity, heart rate, SpO, and distance. The data on the esp$_4/pubtopicwaspublishedu$

5.2 Rule Triggering and Filtering

An SQL-based rule was applied to monitor abnormal values such as:

Parameter	**Expected/Simulated Range**
SpO_2	95–100%
Blood Pressure	80–120 mmHg (simulated)
Sugar Level	80–150 mg/dL (scaled)
Temperature	≈ 37°C (after scaling)

Table 5.1: Observed Biomedical Parameters

5.3 Email Notification Alerts

When the system detected unusual readings, it sent an alarm to a subject on social networking sites, and the topic sent the notice via email. The actual health information was included in the alert email, as can be seen. The alert email contained the actual health data, as seen in

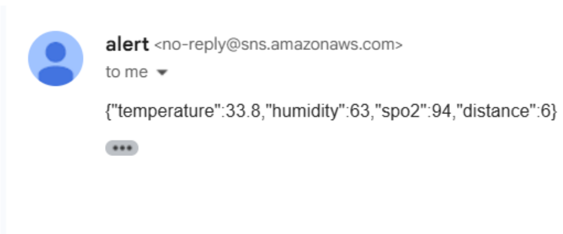

Figure 5.1: Email Notification Alert

5.4 Data Logging in DynamoDB

Every MQTT message was concurrently recorded in DynamoDB for additional analysis; each record contains a date, subject name, and sensor readings.

5.5 LabVIEW: Signal Acquisition and Processing

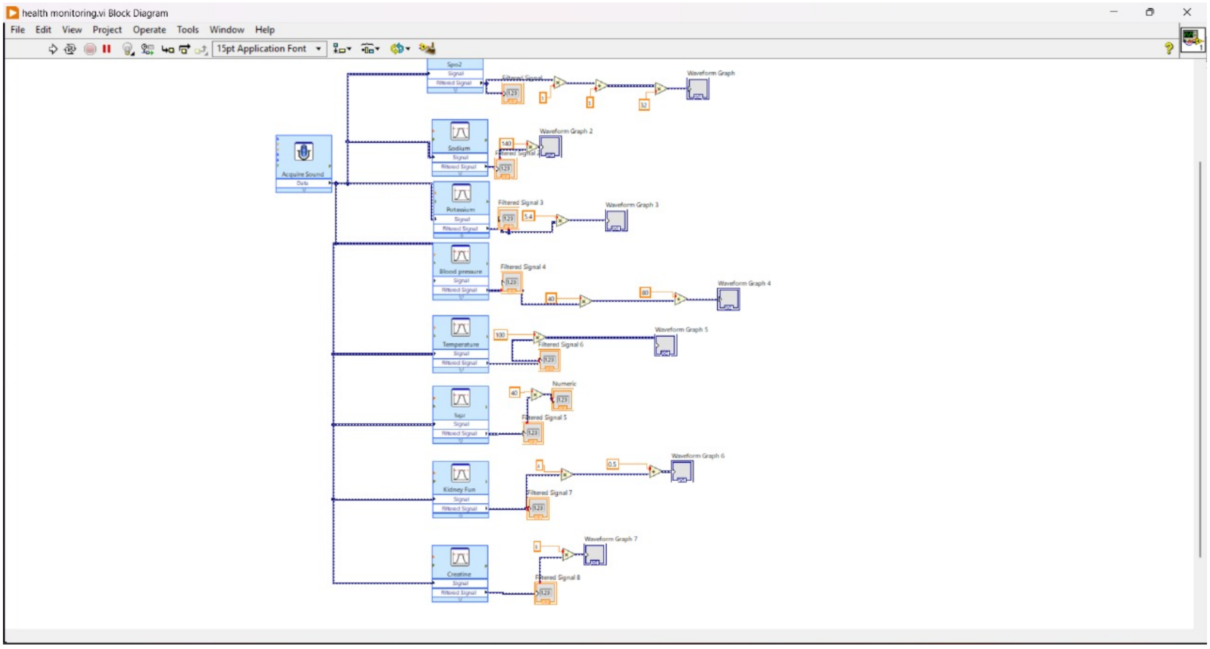

Figure 5.2: Labview Simulation

The LabVIEW VI was developed to gather and filter data from several biological sensors.From Figure 5.2 The input signal is acquired using the ”Acquire Sound” block, which represents analog signal input from many sensor modules. The following is how vital signals are processed: SpO_2 Potassium and Sodium Temperature of Blood Pressure Level of Sugar Function of the Kidneys Level of Creatinine A filter block is applied to each signal in order to eliminate noise and extract pertinent biological information.

5.6 Analysis

The system has the ability to capture and transmit real-time health data at regular intervals. AWS IoT rules successfully screened unusual values in real time. Alerts were triggered within seconds of detecting irregularities. Data that has been preserved can be used for trend analysis and long-term patient monitoring. This cloud-integrated approach provides a scalable, automated, and real-time framework for health monitoring that might be extended to hospitals or remote health systems. for Labview A Waveform Graph that shows data in real time is connected to each parameter. The user may monitor changes and patterns in health parameters over time thanks to this. This real-time graphic helps customers or clinicians keep an eye on developments in real time.

CHAPTER 6

Conclusion and Future Scope

6.1 Conclusion

When cloud computing and the Internet of Things are combined, an efficient platform for real-time environmental and health monitoring is offered. This project successfully demonstrates a full-stack system that collects, processes, and analyzes data remotely using sensors, Amazon Web Services (AWS), and microcontrollers (ESP8266). With the help of AWS IoT Core, DynamoDB, Lambda, and SNS, the solution ensures accurate and timely monitoring by offering scalable data management and quick warning generating. One of the system's main selling points is its ability to effectively handle interference and anomalous data using cloud-side validation and smart rule triggers. By addressing typical problems like signal noise and uneven sensor output, the research increases the reliability of remote monitoring systems. The proposed paradigm, which includes edge computing for offline operation, mobile app compatibility, and AI-based predictions, offers a solid foundation for further advancement. Future developments in this technology might have a big influence on environmental management, smart healthcare, and emergency alarm systems, making homes safer and more connected.

6.2 Scope of the Future

6.2.1 Predictive Monitoring with AI/ML Integration

Utilize AI/ML models in conjunction with services like AWS SageMaker to predict health anomalies or environmental threats. enables the early detection of issues such as elevated temperatures, oxygen deprivation, or rising pollution levels. The process of creating mobile applications Using AWS Amplify or React Native, develop a user-friendly mobile application that offers real-time data access, configurable thresholds, and notification alerts. increases user engagement and mobility.

6.2.2 Data Visualization and Improved Reporting

Make periodic reports and dynamic dashboards with technologies like Amazon QuickSight or Grafana. helps consumers and medical professionals make data-driven decisions and analyze trends.

6.2.3 Offline Processing using Edge Computing

Use ESP32/Raspberry Pi local processing or AWS Greengrass for offline alerting and local decision-making. reduces dependency on the internet and improves reaction time in remote areas. Multiple Users with Role-Based Access Implement secure access control using AWS Cognito or IAM to allow different users (such a doctor, patient, or administrator) to see and manage data in accordance with their responsibilities. enhances data privacy, scalability, and usability in bigger systems.

Learning Outcomes

(PO-1): Apply the knowledge based on curricular and co-curricular activities to solve Electronics and Telecommunication-based project work: The knowledge acquired from subjects like Python, excel, machine learning, etc. that are part of the curriculum and knowledge gained by studying these subjects helped us to understand our project better.

(PO-2): Systematically analyze electronics and telecommunication-related project based on the literature review: We referred to various paper where various applications of SVM, ANN, and dynamic weightage assignment were discussed. These papers helped us to understand and design systems.

(PO-3): Design and develop software code based on the problem specifications of the project: Our aim of the project is to implement a prototype to detect and analyses different parameters to assign weightage and to indicate whether to Buy/Sell a particular signal.

(PO-4): Carry out different experiments to generate data, analyze and interpret the data, and draw valid conclusions related to the project work: Use Excel to study and plan the different parameters of the stock market.

(PO-5): Select and apply appropriate modern tools for the solution of their project problem: We have used various software such as Excel, and Python and integrated the use of libraries as well.

(PO-6): Know the responsibility of the engineer towards society concerning their project work: As Electronics and Telecommunication Engineers, it is our responsibility to analyze the current needs and problems faced by society and if possible, plan, design develop effective solutions for the same.
(PO-7): Understand the impact of engineering solutions related to their project work in societal context for sustainable development: Engineers are increasingly required to play a leadership role in sustainable development, overcoming global challenges, such as depletion of resources, environmental pollution, rapid population growth and damage to ecosystems.

(PO-8): Apply professional ethical principles while project implementation, report writing, and publication. An engineer must always consider the safety of citizens, moral values, and the general well-being of society. We have therefore given proper references for the materials we have referred.

(PO-9): Work effectively as an individual and as a member of the team while the project work is carried out: While working on various aspects of our project, we realized the importance of teamwork and determination needed for the successful completion of the project in the context of helping each other.

(PO-10): Communicate effectively while project report writing and oral/visual presentations: Communication is an excellent skill that is helpful to express and understand ideas and interests, and the presentations have helped us achieve this skill.

(PO-11): Gain knowledge of engineering and management aspects while the project is being implemented: The knowledge of various engineering and management aspects such as performance, feasibility, accuracy, cost-effectiveness, etc. was gained while doing this project.

(PO-12): Engage themselves in independent and lifelong learning: Technology never remains stagnant. It keeps evolving over the years, and as engineers, we realize that we must keep ourselves updated throughout to improve our technical knowledge.

CO PO Mapping

Table 6.1: CO-PO Mapping

Course Outcome	PO1	PO2	P03	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	-
CO2	-	-	3	3	-	-	-	-	-	-	-	-
CO3	-	-	-	-	3	3	-	-	-	-	-	-
CO4	-	-	-	-	-	-	3	3	-	-	-	-
CO5	-	-	-	-	-	-	-	-	3	3	-	-
CO6	-	-	-	-	-	3	-	-	-	-	3	3

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and appreciation to all the people and organizations who have been instrumental in the accomplishment of our significant endeavor. Our profound gratitude goes out to our Head of the Department, **Dr. Megha Kolhekar**, for giving us the chance to work with **Dr. Umesh M** and for the support offered during this project. This initiative has been greatly influenced by your leadership and priceless thoughts.

Our sincere appreciation is extended to our project mentors **Akshata Raut** for their unwavering guidance, tolerance, and commitment. Your knowledge, support, and insightful criticism have been crucial to the success of my project. It gives us great pleasure to thank **Dr.Nilashree W** for supporting and being there for the team in any circumstance.

We would like to express our gratitude to the whole faculty of FCRIT's Electronics and Telecommunication Engineering Department for their invaluable instruction, which served as the basis for this project. Your dedication to sharing information and your readiness to answer our questions have been quite helpful. We appreciate the non-teaching staff's efforts in creating a supportive atmosphere for learning and research. Although their behind-the-scenes labor is sometimes overlooked, it is much valued. We appreciate the support and expertise that our classmates and fellow students have given us. We really appreciate your constant support and direction. I appreciate you playing a crucial role in this important turning point in my academic career.

• Diksha Bhosale (3021159)

Pratik Belamkar (3021107)

Varun Somwanshi (3021151)

Mudabbir Ahmad (3021133)