Higher Nationals - Summative Assignment Feedback Form

Student Name/ID	Geshan Malinda Gamage / E138479		
Unit Title	**45: Internet of Things**		
Assignment Number	**1**	**Assessor**	Miss. Wethma Nethmini
Submission Date	14/07/2024	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	

Assessor Feedback:

LO1 Analyse what aspects of IoT are necessary and appropriate when designing software applications

Pass, Merit & Distinction Descripts **P1** ☐ **P2** ☐ **M1** ☐ **M2** ☐ **D1** ☐

LO2 Outline a plan for an appropriate IoT application, using common architecture, frameworks, tools, hardware and APIs

Pass, Merit & Distinction Descripts **P3** ☐ **P4** ☐ **M3** ☐ **M4** ☐

LO3 Develop an IoT application using any combination of hardware, software, data, platforms and services

Pass, Merit & Distinction Descripts **P5** ☐ **P6** ☐ **M5** ☐ **D2** ☐

LO 4 Evaluate your IoT application and the problems it might encounter when integrating into the wider Io ecosystem

Pass, Merit & Distinction Descripts **P7** ☐ **P8** ☐ **M6** ☐ **D3** ☐

Assessor Feedback: *		
Grade:	**Assessor Signature:**	**Date:**
Resubmission Feedback: *Please note resubmission feedback is focussed only on the resubmitted work		
Grade:	**Assessor Signature:**	**Date:**
Internal Verifier's Comments:		

Signature & Date:

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Pearson Higher Nationals in

Computing

45: Internet of Things

Assignment 01 of 01

General Guidelines

1. A Cover page or title page – You should always attach a title page to your assignment. Use the previous page as your cover sheet and make sure all the details are accurately filled.
2. Attach this brief as the first section of your assignment.
3. All the assignments should be prepared using word processing software.
4. All the assignments should be printed on A4-sized papers. Use single-sided printing.
5. Allow 1” for the top, bottom, and right margins and 1.25” for the left margin of each page.

Word Processing Rules

1. The font size should be **12** points and should be in the style of **Time New Roman**.
2. **Use 1.5 line spacing**. Left justify all paragraphs.
3. Ensure that all the headings are consistent in terms of font size and font style.
4. Use the **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each pag**e. This is useful if individual sheets become detached for any reason.
5. Use word processing applications spell check and grammar check functions to help edit your assignment.

Important Points:

1. It is strictly prohibited to use textboxes to add texts to the assignments, except for the compulsory information. eg: Figures, tables of comparison, etc. Adding text boxes in the body except for the before mentioned compulsory information will result in the rejection of your work.
2. Avoid using page borders in your assignment body.
3. Carefully check the hand-in date and the instructions given in the assignment. Late submissions will not be accepted.

4. Ensure that you give yourself enough time to complete the assignment by the due date.
5. Excuses of any nature will not be accepted for failure to hand in the work on time.
6. You must take responsibility for managing your own time effectively.
7. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
8. Failure to achieve at least PASS criteria will result in a REFERRAL grade.
9. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
10. If you use other people's work or ideas in your assignment, reference them properly using the HARVARD referencing system to avoid plagiarism. You have to provide both in-text citations and a reference list.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL, or at worst you could be expelled from the course.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

<table>
<tr><td colspan="2">Student name:

G. Geshan Malinda Gamage</td><td colspan="2">Assessor name:

Miss. Wethma Nethmini</td></tr>
<tr><td>Issue date:

31.05.2024</td><td colspan="2">Submission date:

16.07.2024</td><td>Submitted on:</td></tr>
<tr><td colspan="4">Programme: **Internet of Things**</td></tr>
<tr><td colspan="4">Unit: **45**</td></tr>
<tr><td colspan="4">Assignment number and title: 1, Researching and building an IoT application</td></tr>
</table>

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Guidelines for incorporating AI-generated content into assignments:

The use of AI-generated tools to enhance intellectual development is permitted; nevertheless, submitted work must be original. It is not acceptable to pass off AI-generated work as your own.

Student Declaration

Student declaration
I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.
Student signature: Geshan Date: 16.07.2024

Higher National Diploma in Business

Assignment Brief

Student Name /ID Number	G. Geshan Malinda / E138479
Unit Number and Title	**45: Internet of Things**
Academic Year	2022/2023
Unit Tutor	Miss. Wethma Nethmini
Assignment Title	Researching and building an IoT application
Issue Date	31.05.2024
Submission Date	16.07.2024
IV Name & Date	

Submission format
The assignment submission is in the form of: • **a case study** on a positive example of an Internet of things (IoT) implementation with a recommended word limit of 1,000–1,500 words, written in a concise style, although you will not be penalised for going under, or exceeding, the total word limit • **a project plan and development report** on the development of an appropriate IoT application with a recommended word limit of 2,000–2,500 words, written in a concise style, although you will not be penalised for going under, or exceeding, the total word limit • **a project report** evaluating the success of your final IoT product with a recommended word limit of 1,000–1,500 words, written in a concise style, although you will not be penalised for going under, or exceeding, the total word limit. You are required to make use of headings, paragraphs, and sub-sections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system (or an alternative system). You will also need to provide a bibliography using the Harvard referencing system (or an alternative system). Inaccurate use of referencing may lead to issues of plagiarism if not applied correctly.

	Unit Learning Outcomes:
	LO1 - Analyze what aspects of IoT are necessary and appropriate when designing software applications. LO2 - Outline a plan for an appropriate IoT application, using common architecture, frameworks, tools, hardware, and APIs. LO3 - Develop an IoT application using any combination of hardware, software, data, platforms, and services. LO4 - Evaluate your IoT application and the problems it might encounter when integrating into the wider IoT ecosystem.

Assignment Brief and Guidance:

Scenario

‘***ChannelCert (PVT) Ltd***’ is a leading research and consultancy firm researching new market trends In the IT industry and IoT applications for corporate clients and the consumer market. You currently work as a junior infrastructure architect for ***’ChannelCert (PVT) Ltd***’. As part of your role, your manager has tasked you to conduct research on Internet of Things applications and technologies in order to enlighten a new intake of probationary apprentices in understanding the business, particularly IoT technologies, frameworks, and architecture, as most probationary apprentices have not covered these topics in their studies.

As a part of this assignment, you then have to select an organization of your choice to plan and develop a small IoT application to address an existing problem and to achieve the organization’s goals.

Activity 1 - Case study

Your manager has asked you to help produce a handbook that introduces the fundamental concepts of IoT to new apprentices during their orientation. As part of the handbook, you have been asked to contribute a case study that looks at which aspects of IoT are necessary and appropriate when designing software applications. You should use a range of real-world case studies to demonstrate the different IoT architectures and frameworks. A case study involves an up-close, in-depth, and detailed investigation of a topic and aims to bring the understanding of a complex issue or topic within a given context, so bear this in mind when devising the case study. The case study should provide a positive example of an IoT implementation by giving an overview of the Internet of things concept by evaluating IoT architecture and justifying that it can be used in designing software applications. As part of the case study, you should include:

- An exploration of different types of IoT functionality
- A review of the standard IoT architecture, frameworks, tools, hardware, and APIs (AFTHA) used in development.

• An analysis of the impact that the use of common IoT AFTHA techniques has in the software development life cycle (SDLC)

• An examination of specific forms of AFTHA techniques used for different problem-solving requirements

• An evaluation and justification of the specific forms of IoT architecture used when designing software applications. As this is a case study format, you should use 'real-world' examples from your research to illustrate the IoT architectures you are discussing.

Activity 2 - Project plan and development report

Your manager would like you to outline a plan and then develop an appropriate IoT application to achieve the aims of the chosen organization. Before you plan and develop the product, you will need to identify specifically what problem needs to be solved. You will then use your research to identify a range of IoT AFTHA techniques and determine which are suitable for solving this problem and why. You will employ the SDLC and produce evidentiary documentation in keeping with best practices (software requirements specification, data distribution service, etc.) to iteratively design and develop an IoT application that meets the client's brief. Your project plan should include:

• an investigation of the AFTHA used in the development of IoT applications

• a discussion of the specific problem that the chosen organization wishes to solve using IoT devices

• a plan for the most appropriate AFTHA to include in the solution to the problem

• the application of the selected techniques to the IoT development plan. You should then develop the IoT application, using any combination of hardware, software, data, platforms and services you feel necessary to solve the organization's problem.

Your development report should include:

• A demonstration of the selection of a set of appropriate tools to turn the development plan into an IoT application

• A set of end-user experiments on the application, with feedback

• An examination and evaluation of the end-user feedback

• A determination of the advantages and disadvantages of the chosen IoT techniques based on end-user feedback

• A clear demonstration of how user feedback has been used to create multiple Iterations of the IoT application

• A description of how each Iteration has been modified to provide enhancements based on user feedback and experimentation.

Activity 3 - Project report

Finally, you will write an accompanying report, with your evidentiary documentation appended, for your manager, evaluating the success of your final product. The evaluation report must include:

• A review of the developed IoT application with a detailed description of the problems that it solves

• An investigation of the potential problems that the IoT application might encounter when integrating into the organization's system

• A comparison of the final application with the original plan

• A critical evaluation of the overall success of the application, including the potential impact of the IoT application on people, business and society, the end user.

Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
LO1 Analyse what aspects of IoT are necessary and appropriate when designing software applications		
P1 Explore various forms of IoT functionality. **P2** Review standard architecture, frameworks, tools, hardware and APIs available for use in IoT development.	**M1** Analyse the impact of common IoT architecture, frameworks, tools, hardware and APIs in the software development lifecycle. **M2** Examine specific forms of IoT architecture, frameworks, tools, hardware and APIs for different problem-solving requirements.	**D1** Evaluate specific forms of IoT architecture and justify their use when designing software applications.
LO2 Outline a plan for an appropriate IoT application, using common architecture, frameworks, tools, hardware and APIs		
P3 Investigate architecture, frameworks, tools, hardware and API techniques available to develop IoT applications. **P4** Discuss a specific problem to solve using IoT.	**M3** Plan the most appropriate IoT architecture, frameworks, tools, hardware and API techniques to include in an application to solve a problem. **M4** Apply selected techniques to create an IoT application development plan.	**D2** Make multiple iterations of the IoT application and modify each iteration with enhancements gathered from user feedback and experimentation.
LO3 Develop an IoT application using any combination of hardware, software, data, platforms and services		
P5 Employ an appropriate set of tools to develop a plan into an IoT application. **P6** Run end-user experiments and examine feedback.	**M5** Reconcile end-user feedback and determine advantages and disadvantages of chosen IoT techniques.	

Pass	Merit	Distinction
LO4 Evaluate your IoT application and the problems it might encounter when integrating into the wider IoT ecosystem		
P7 Review the IoT application, detailing the problems it solves. **P8** Investigate the potential problems the IoT application might encounter when integrating into the wider system.	**M6** Compare the final application with the original plan.	**D3** Critically evaluate the overall success of the application including the potential impact of the IoT application on people, business and society, and the end user.

Pearson Higher Nationals in

Computing

45: Internet of Things

Assignment 01 of 01

G. Geshan Malinda Gamage

E138479

1 Acknowledgement

I received many opinions and advice from various people to successfully complete the assessment assigned to us for the subject of **45: Internet of Things, Researching and building an IoT application** in our HND course. I express my gratitude to all of them. First, I would like to thank my mother and father for being there for me and supporting me and encouraging me. Also, I would like to express my gratitude to my lecturer, Miss. Wethma Nethmini, who guided me to complete this work successfully. I would also like to express my gratitude to my other lecturers, who helped me to complete my assessment successfully. And last, I would like to thank who help me with my feedback forms, and who gave their opinions on new technologies that we discussed in my assignment below.

Last without their support and cooperation I would not have completed this assignment in the given time.

2 Table of content

Table of Contents

3 Table of Figures

4 Figures of Tables

5 Introduction to IoT

The Internet of Things (IoT) has indeed come a long way since its early conceptualization. The term itself was coined in 1999 by Kevin Ashton during his work at Procter & Gamble, and it has since grown into a global phenomenon. Early implementations of IoT were focused on RFID tags, but the concept expanded rapidly with the advent of wireless technologies and the internet. Today, IoT encompasses a vast array of devices, from household appliances to industrial machinery, all interconnected and capable of exchanging data to optimize performance, enhance user experiences, and provide valuable insights. The evolution of IoT is marked by the integration of advanced technologies such as machine learning, artificial intelligence, and big data analytics, further enhancing its capabilities. As IoT continues to evolve, it promises to revolutionize how we interact with the physical world, making our environments smarter, more responsive, and increasingly autonomous.

5.1 Importance of IoT

The Internet of Things (IoT) is causing a revolution in how we use technology. It's changing things in many areas. IoT systems let devices talk and work together without people. This makes complex tasks easier and creates new ways to solve everyday problems. In farming IoT gadgets can check how wet the soil is and water plants. This helps plants grow better and saves water. In healthcare IoT devices you wear can keep track of patients' health in real time. This lets doctors help before things get worse. IoT gives us useful information from data. This leads to smarter choices in business better work, and services that fit each person. As IoT keeps growing, it will become a bigger part of our lives. In the future smart devices that work together will make life safer, easier, and more connected.

6 Activity 01

6.1 IoT in Software Application Design

The Internet of Things (IoT) links devices with sensors, software, and tech to gather and share data. This study looks at IoT parts needed to make software apps. It gives a rundown of IoT ideas, structure, tools, gear, and APIs. Real examples show how these setups work.

IoT connects stuff that has sensors and programs. These things talk to each other and swap info. Our case study digs into what IoT bits you need when you're cooking up software. We'll chat about what IoT is all about how it's built, what tools you can use, the hardware side, and ways to connect it all.

We'll keep it real with examples from the wild to help you get your head around these setups. By the end, we will have a solid grip on how IoT ticks and what we need to think about when we are making apps that use it.

6.2 Various Forms of IoT Functionality

The Internet of Things (IoT) is revolutionizing the way we interact with technology and our environment. With the capability to monitor various parameters IoT devices offer real-time data. This data can be crucial for maintaining optimal conditions in both domestic and industrial settings. The control and automation aspect allow for hands-off approach to managing systems. This provides convenience and efficiency.

Optimization through IoT leads to smarter resource management. This is essential in today's world where sustainability is a priority. Predictive maintenance stands out by reducing downtime. It extends the lifespan of equipment, saving considerable costs and preventing potential hazards.

Lastly the data analytics feature of IoT turns raw data into actionable insights. This empowers businesses and individuals to make informed decisions. Collectively, these functionalities of

IoT weave web of interconnected devices. These enhance operational efficiency safety and quality of life.

These are the most popular functionalities of IoT,

- **Monitoring:** IoT devices can continuously monitor environments, such as smart thermostats in homes or temperature sensors in factories.
- **Control and Automation**: Devices can be controlled remotely and automated to perform tasks, such as smart lighting systems or industrial automation robots.
- **Optimization**: IoT systems can optimize processes by analyzing collected data, like smart grid systems optimizing energy consumption.
- **Predictive Maintenance**: Using data from IoT devices, systems can predict and prevent equipment failures, as seen in modern manufacturing plants.
- **Data Analytics**: IoT generates vast amounts of data that can be analyzed for insights, aiding decision-making processes.

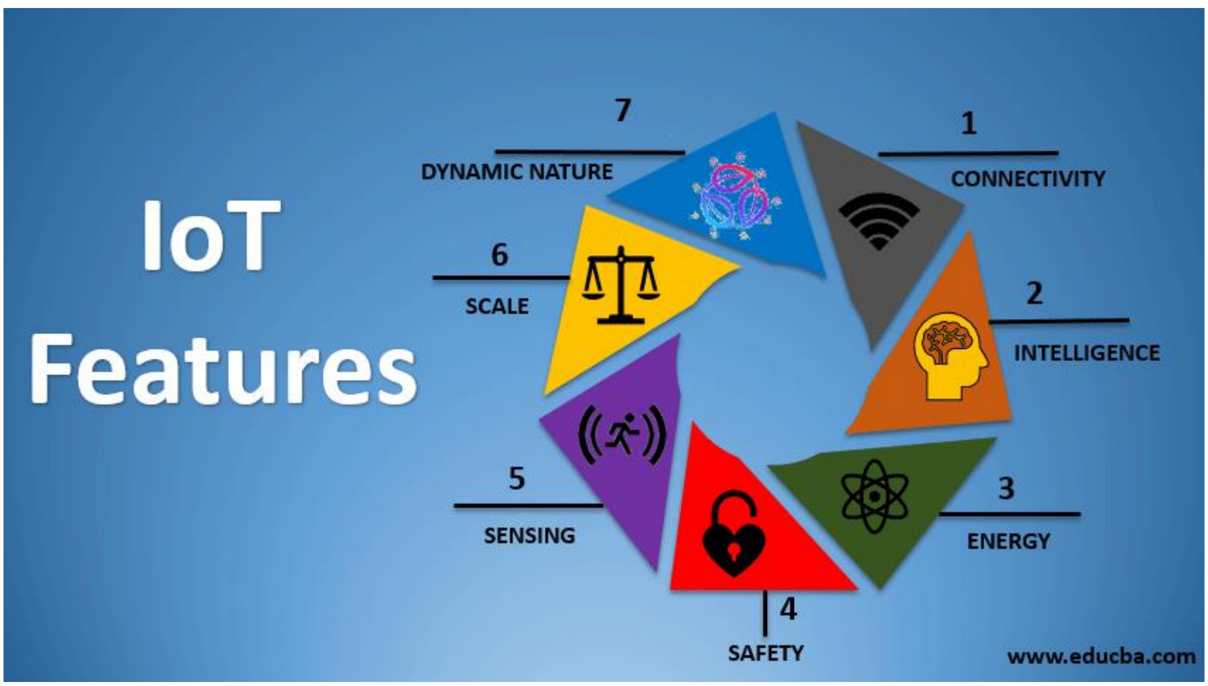


Figure 1 IoT features

6.3 IoT Architecture, Frameworks, Tools, Hardware, and APIs

The IoT world is changing fast and It has many different parts like designs, structures, gadgets, and ways for things to talk to each other. The main idea of IoT is to connect sensors and devices. These things share info with each other and big computer systems through the internet. Structures help people make and use IoT stuff. They focus on making parts that work well together and can grow bigger. There are lots of tools for IoT. Some help make new things, while others test if they work right. The physical parts of IoT, like tiny computers and connection bits, grab info and send it out. APIs have a big job in IoT. They make different systems work together and help share info. All these pieces team up to make IoT work. This tech helps in many areas, from making homes smarter to running factories better.

6.3.1 IoT Architecture

The architecture of the Internet of Things (IoT) is an advanced framework created to manage the massive volumes of data produced by networked devices. The fundamental components of the Device Layer are sensors and actuators, which serve as the main sources of information and tangible interaction. The Network Layer uses effective protocols like MQTT, CoAP, and HTTP to transfer information, ensuring smooth connectivity across the system. The Cloud Layer is the main center for processing and storing data, utilizing machine learning and analytics to extract insightful information. Lastly, the Application Layer provides end users with an interface to interact with the Internet of Things environment. This interface is often provided via user-friendly web interfaces or mobile apps. This multi-layered approach enables a robust and scalable infrastructure, capable of supporting a wide range of IoT applications and services.

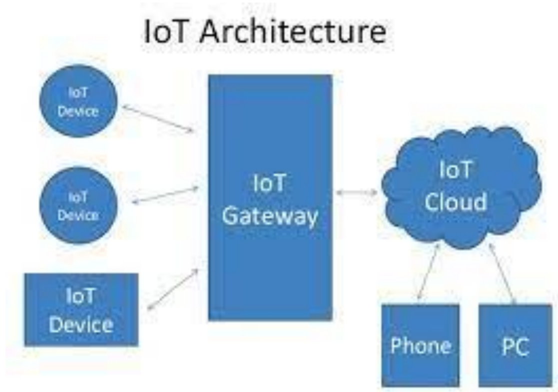


Figure 2 IoT architecture

4 stages of IoT architecture,

1. **Device Layer:** Consists of sensors and actuators that collect data and perform actions.
2. **Network Layer:** Facilitates communication between devices and servers, using protocols like MQTT, CoAP, and HTTP.
3. **Cloud Layer:** Processes and stores data, providing analytics and machine learning capabilities.
4. **Application Layer:** End-user applications that interact with the IoT system, such as mobile apps or web interfaces.

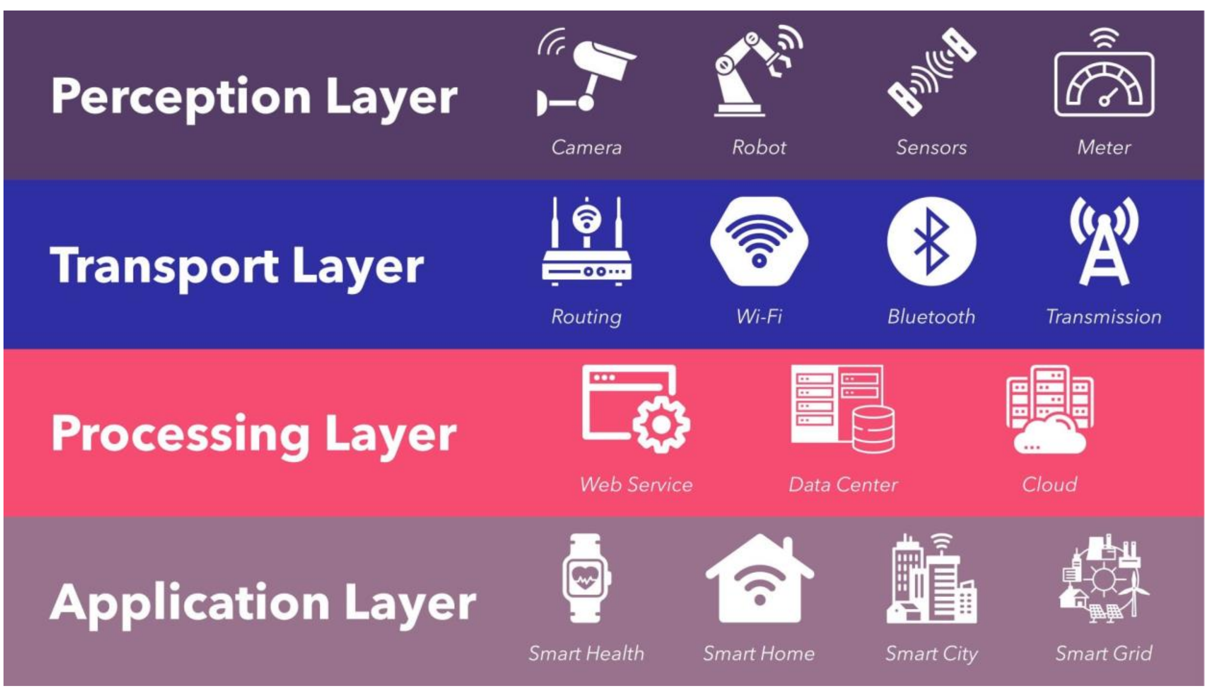


Figure 3 4 stages of IoT architecture.

6.3.2 Frameworks

The Internet of Things (IoT) has sparked a revolution in how we use devices and data. Several big companies offer strong systems to support IoT. Google Cloud IoT shines with its full device control and powerful tools for data intake and study. These tools let users connect and understand data. Amazon Web Services (AWS) IoT is another big player. It gives a wide range of IoT services. These include advanced device control, data handling, and smart analysis to help businesses use their IoT data well. Microsoft Azure IoT Suite rounds out the top three. It has a special IoT hub that helps lots of devices talk to each other. It also has advanced analysis and machine learning to turn data into useful ideas. Each system has its own special tools. They meet different IoT needs and help push new ideas in this changing field.

The Internet of Things (IoT) is revolutionizing the way industries operate by enabling enhanced data-driven decision-making. Google Cloud IoT, AWS IoT, and Microsoft Azure IoT Suite are at the forefront of this transformation, offering robust frameworks for a wide range of applications. From asset tracking in retail to predictive maintenance in manufacturing, these platforms provide the tools necessary for businesses to innovate and optimize their operations. The integration of IoT technology across various sectors not only streamlines processes but also opens up new opportunities for growth and efficiency (ip_admin, 2023).

- Google Cloud IoT: Provides device management, data ingestion, and analysis tools.
- Amazon Web Services (AWS) IoT: Offers a suite of IoT services, including device management, data processing, and analytics.
- Microsoft Azure IoT Suite: Provides IoT hub, analytics, and machine learning services.

6.3.3 Tools use in IoT

Node-RED and ThingSpeak play a key role in making IoT processes simpler. Node-RED offers a user-friendly system for programming that allows quick creation and launch of IoT apps without needing to code in the usual way. This makes it easy for more people to use. ThingSpeak gives users powerful tools to collect, show and study live data as it comes in. This has a big impact on tasks like keeping an eye on the environment running smart gadgets or even making factory work better. These tools give both pros and hobbyists the chance to get the most out of IoT tech and push for new ideas in our world where everything is getting more connected.

- **Node-RED:** A flow-based development tool for visual programming of IoT applications.
- **ThingSpeak:** An IoT analytics platform that allows aggregation, visualization, and analysis of live data streams.

6.3.4 Hardware

The Raspberry Pi, Arduino, and ESP8266/ESP32 play a crucial role in today's maker movement. These tools give hobbyists and pros the power to create and innovate. The Raspberry Pi works as a small but mighty computer. People use it to learn, code, and build complex IoT projects. Arduino opens the door to electronics for many. It has an easy-to-use coding setup and lots of community help. The ESP8266 and ESP32 chips offer cheap ways to add wireless features. They come with full internet connectivity and can control things making them great for smart homes and wearable gadgets. These platforms give anyone interested in electronics and smart tech a flexible set of tools to work with. Together, they have an influence on how people learn about and make new electronic devices. They're causing a revolution in DIY tech projects allowing makers to turn their ideas into reality without needing expensive equipment. From simple LED blink projects to advanced AI-powered robots, these platforms help to bridge the gap between imagination and real-world applications.

- **Raspberry Pi**: A versatile, low-cost computer used for IoT prototyping and development.
- **Arduino**: An open-source electronics platform based on easy-to-use hardware and software.
- **ESP8266/ESP32**: Low-cost Wi-Fi microchips with full TCP/IP stack and microcontroller capability.

6.3.5 APIs

APIs, or Application Programming Interfaces, play a crucial role in IoT (Internet of Things). They let different software apps talk to each other and share info. APIs work as a link between internal and external software operations. This makes sure data moves, and users often don't even notice it. APIs act like virtual go-betweens. They move data from one interface such as a mobile app, to another one.

APIs come in two primary forms: Private and Open. Private APIs help companies improve their internal processes. Open APIs let outside developers combine data from various tools. IoT uses different API types. RESTful APIs are common web APIs that interact with IoT devices and services. WebSockets offer real-time data sharing through two-way communication over one TCP connection. These APIs make app creation easier. They save developers time and cash. They also create chances for new ideas.

- RESTful APIs: Standard web APIs used for interacting with IoT devices and services.
- WebSockets: Provide full-duplex communication channels over a single TCP connection for real-time data exchange.

Examples for popular APIs:

1. Twitter API
2. ChatGPT API
3. Google Map API
4. Discord API
5. Github API

6. PayPal API
7. eCommerce API
8. YouTube API
9. Instagram API
10. Slack API

(Katalon, 2022)

Figure 4 Examples of API

6.3.6 Designing APIs for IoT comes with several challenges:

1. **Security**: Strong security is vital. IoT devices often lack resources making them easy targets for cyber-attacks. We need to use solid authentication, encryption, and frequent updates to protect them.
2. **Interoperability**: Many different devices, protocols, and platforms make smooth communication tough. Using open-source protocols and creating common standards can help solve this problem.
3. **Scalability**: As more devices connect, managing IoT systems gets tricky. We need smart data management and network protocols that can grow to handle this expansion.
4. **Software Complexity**: IoT systems need complex software, which makes development harder. Developers face a tough task to balance code size and efficiency while keeping costs down and power use low.

5. **Connectivity**: Making sure devices stay connected in different places is tricky. IoT gadgets must keep stable links even when network conditions change.

The Internet of Things has an impact on how we design APIs. Developers must consider these challenges to build robust and effective systems. Security, interoperability, and scalability are key factors to consider when creating APIs for IoT devices. By addressing these issues, we can create better connected systems that work well together and stay safe from threats. To tackle these issues, people need to plan, work together, and use the best methods for designing APIs and creating IoT stuff.

6.4 Impact of IoT AFTHA Techniques on SDLC

An organized method for creating software applications is the Software Development Life Cycle (SDLC), which includes phases like design, development, testing, deployment, and maintenance. The Internet of Things (IoT) has brought new aspects to the SDLC, especially when it comes to the incorporation of AFTHA techniques (Automation, Flexibility, Time-to-Market, High Availability, and Agility). These techniques have had a big impact on how software is created, tested, and deployed in IoT environments. This article examines the effects of AFTHA techniques on the SDLC, highlighting their advantages and disadvantages as well as how they are applied in diverse sectors.

6.4.1 Pros and Cons

Table 1 Pros and Cons of AFTHA Techniques

AFTHA Techniques	**Pros**	**Cons**
Automation	Automated systems cut down on manual work and mistakes.	Setting up costs a lot at first. You need people who know their stuff.

	They make things run smoother and more. Testing and rollout happen faster.	People might rely too much on machines.
Flexibility	It's easy to change things when needed. It works with many different IoT gadgets and ways they talk. Users are happier because it listens to what they say.	It can make design and putting things together tricky. You might need to fix and update it often.
Time-to-Market	Speeds up the development and deployment cycle. Provides competitive advantage. Enables faster innovation. Enables iteration.	Can compromise quality if rushed. Increased pressure on development teams. Potential for inadequate testing.
High Availability	Ensures continuous operation of IoT systems. Minimizes downtime. Minimizes service interruptions. Enhances user trust and reliability.	High implementation and maintenance costs. Complex infrastructure requirements. Risk of single points of failure.
Agility	Promotes iterative development. Promotes continuous improvement. Facilitates quick response to market changes. Supports DevOps practices.	Can lead to scope creep. Requires robust and adaptable infrastructure. May cause burnout due to continuous development cycles.

6.4.2 Healthcare

IoT in healthcare uses connected devices and sensors to keep an eye on patient health, handle medical gear, and help with remote care. This tech lets doctors watch vital signs in real-time, do remote check-ups, and automate everyday healthcare jobs.

Benefits:

- Better Patient Tracking: Smartwatches and other gadgets keep tabs on vital signs all the time.
- Healthcare from Home: Telemedicine lets patients chat with docs without leaving the house, cutting down on hospital trips.
- Better Treatment Results: The info gathered can be looked at to make treatment plans better and spot health problems before they get bad.

Challenges:

- IoT systems face challenges with keeping health info safe from hackers. Also different gadgets need to play nice together. Setting up and keeping these systems running can cost a lot too.
- Let's look at a real example. Ochsner Health uses smart devices to keep tabs on folks with long-term health issues like high blood pressure and diabetes. Patients have special blood pressure cuffs and sugar meters that send their numbers straight to their doctors. This setup helps people manage their health better and keeps them out of the hospital more often.

6.4.3 Smart Agriculture

Smart farming uses internet-connected gadgets to make agriculture better. Farmers put sensors in fields to check how wet the soil is, what the weather's like, if plants are healthy, and how the animals are doing.

Benefits:

- Farmers grow more food: They use exact amounts of water and fertilizer, so nothing goes to waste.
- It's better for nature: Using just what's needed means less stuff gets wasted or harms the environment.
- Taking care of animals is easier: Farmers can see right away if their animals are sick or active.

Problems with it:

- IoT solutions cost a lot for small farmers to set up. Farmers need fancy tools and know-how to deal with all the data. Far-off farms might not have good internet to make it work.
- Here's a real example: John Deere makes smart farm gear. Their tractors and stuff connect to the internet. They gather info to help plant, fertilize, and harvest better. Farmers can grow more food, spend less money, and be kinder to nature with this tech.

6.4.4 Supply Chain Management

IoT in supply chains uses sensors and gadgets that talk to each other. They keep an eye on products, check storage, and make shipping smoother.

Benefits:

- **Tracking in Real-Time:** Companies can see where their stuff is while it moves. This helps them manage inventory better and lose less.
- **Making Operations Better:** Machines gather and study info on their own. This lets people make smarter choices and work faster.

- **Happier Customers:** When things arrive on time and folks know where their stuff is, customers like it more.

Problems:

- **Safety Worries:** Bad guys might hack into connected gadgets.
- **Tricky to Set Up:** It's hard to make IoT work with old supply chain computer programs.
- **Expensive:** Some companies can't afford to set up and keep these systems running because they cost a lot.

Amazon uses IoT gadgets in its warehouses to keep tabs on stock and make sorting and packing automatic. This has cut down on mistakes, made things work better, and let them deliver stuff faster.

6.4.5 Weather Stations

Weather stations with IoT use connected sensors to gather up-to-the-minute info on how hot it is how humid how fast the wind's blowing, and other stuff about the environment. They send this info to main systems to look at and guess what the weather will do.

Benefits:

- Better Weather Guesses: Getting info right away makes weather predictions more on the money.
- Dealing with Disasters: Knowing about bad weather helps people get ready in time.
- Keeping an Eye on the Environment: Always watching helps track how the climate's changing and what's going on with the environment.

Challenges pose problems:

- Sensors and devices struggle to work in tough weather.
- Too much data makes analysis tricky.
- Remote areas have trouble getting good internet.

Real-Life Case: The Weather Company has sensors all over the world. These sensors gather weather info. This info helps make good forecasts and warn people about bad weather.

6.4.6 Smart Homes

What They Are: Smart homes use IoT stuff like smart thermostats, lights, cameras, and home helpers. These things work together to make a connected home that does things on its own.

Benefits:

- Smart homes make life easier and cozier. Lights, heating, and security work on their own, improving how we live. These clever gadgets save energy, which cuts costs and helps the planet. Connected security keeps an eye on things and warns us if something's wrong.
- But smart homes have problems too. They gather personal info, which makes people worry about privacy. Devices from different companies need to play nice together, and that's not always easy. Setting up a smart home can also hit your wallet hard at first.

Challenges:

- Smart tech brings comfort and saves money, but it comes with privacy worries and setup headaches. It's a mix of good and bad, like most new tech. People need to weigh the pros and cons before jumping in.
- Google Nest makes smart home stuff like thermostats, security cameras, and smoke alarms. You can control these gadgets from your phone making your home safer and easier to manage.

6.4.7 Smart Cities

Smart cities use Internet of Things tech to make life better. They improve things like traffic, trash pickup, power distribution, and safety.

Benefits:

- Better Public Services: Cities run smoother and respond faster to people's needs.
- Less Traffic Jams: Smart traffic lights and systems keep cars moving.
- Green Living: Smart power grids and trash systems help the environment.

Challenges:

- Smart city solutions cost a lot. Protecting big data from hackers is tough. Getting everyone on the same page and linking different systems is tricky.
- Real-World Example: Barcelona leads the pack in smart cities. They use IoT for trash, parking, and lights. Smart trash cans tell workers when they're full. Smart parking helps drivers find spots. Smart streetlights change brightness based on what's going on around them.

6.4.8 Industrial IoT (IIoT)

What's it about? IIoT uses connected gadgets in factories and industry stuff. It watches machines, makes things run smoother, and keeps people safer.

Advantages:

- Real-time monitoring and predictive maintenance boost efficiency. They cut downtime and make productivity better. Companies save money when operations run.
- Safety improves too. Monitoring systems keep workers safer and lower accident risks.

Problems:

- Industrial systems face security risks. Hackers can attack them. It's hard to add IoT to old industrial setups. Another issue is data. Industrial sensors create tons of info. Managing and studying all this data causes headaches.

- GE applies IIoT tech in its factories to keep an eye on how well machines are doing and make production run smoother. They've saved a lot of money and gotten way more efficient by figuring out when stuff needs fixing before it breaks down.

6.4.9 Using AFTHA in SDLC Stages:

Using AFTHA (Architecture Frameworks, Tools Hardware, APIs) in the Requirement Analysis stage of the Software Development Life Cycle (SDLC) involves several key steps.

- ❖ Architecture: Define the overall structure of IoT system. Ensure it meets the project's goals and constraints.
- ❖ Frameworks: Select appropriate development frameworks. These should support the system's requirements and facilitate efficient development.
- ❖ Tools: Utilize requirement gathering tools. Capture and document the system's needs accurately.
- ❖ Hardware: Identify the necessary devices. Include sensors that will be part of the IoT system.
- ❖ APIs: Determine the required APIs for communication and data processing ensuring seamless integration and functionality.

This approach ensures a comprehensive and structured analysis. It lays a solid foundation for the subsequent stages of development.

6.4.10 Comparison with Traditional Development:

Requirement Gathering:

- ❖ Traditional Development: Primarily focuses on understanding user needs and business requirements to create software solutions.
- ❖ IoT Development: In addition to user needs and business requirements, it also considers hardware capabilities, network protocols, data volume, and security. This is because IoT systems involve physical devices that interact with the environment and each other.

Comparison: IoT development requires a more comprehensive analysis due to the integration of physical devices, which adds layers of complexity in terms of connectivity, data management, and security. This makes the requirement gathering phase more intricate and detailed compared to traditional software development.

6.5 Specific Forms of AFTHA Techniques for Problem-Solving

Table 2 AFTHA Techniques for Problem-Solving

Problem	Technique	Description
Remote Monitoring	MQTT Protocol	Lightweight messaging protocol for remote monitoring.
Real-Time Data Processing	Apache Kafka	Distributed event streaming platform for handling real-time data feeds.
Device Management	AWS IoT Device Management	Tools for onboarding, organizing, and monitoring IoT devices.
Predictive Maintenance	Machine Learning on Azure IoT	Analyses sensor data to predict equipment failures.
Data Security	TLS/SSL Encryption	Ensures secure communication between IoT devices and servers.

6.6 Evaluate specific forms of IOT architecture and justify their use when designing software applications.

It's crucial to take into account several IoT architecture forms when developing software applications for the Internet of Things (IoT) and select the best one depending on your unique needs. The following are some typical IoT architectural types along with their explanations:

1. **Centralized Architecture**:
   - **Description**: In a centralized architecture, all data processing and decision-making occur on a central server or cloud platform.
   - **Justification**:
     - **Simplicity**: Centralized architectures are straightforward to implement and manage.
     - **Scalability**: They work well for small-scale deployments with limited device counts.
     - **Cost-Efficiency**: Fewer components reduce infrastructure costs.
     - **Use Case**: Simple home automation systems or small-scale monitoring applications.
2. **Edge Computing Architecture**:
   - **Description**: Edge computing pushes data processing closer to the data source (devices) rather than relying solely on the cloud.
   - **Justification**:
     - **Low Latency**: Edge computing reduces communication delays by processing data locally.
     - **Bandwidth Optimization**: Transmit only relevant data to the cloud, saving bandwidth.
     - **Privacy and Security**: Sensitive data remains within the local network.
     - **Use Case**: Real-time analytics, industrial automation, and autonomous vehicles.
3. **Fog Computing Architecture**:
   - **Description**: Fog computing extends edge computing by introducing intermediate nodes (fog nodes) between devices and the cloud.

- Justification:
    - **Distributed Processing**: Fog nodes handle data preprocessing, filtering, and aggregation.
    - **Reliability**: Redundancy and fault tolerance improve system reliability.
    - **Scalability**: Suitable for medium-scale deployments.
    - **Use Case**: Smart cities, healthcare monitoring, and retail analytics.

4. **Distributed Architecture**:
    - **Description**: In a distributed architecture, devices communicate directly with each other without a central server.
    - **Justification**:
        - **Decentralization**: No single point of failure.
        - **Resilience**: Devices can operate independently even if some fail.
        - **Latency Reduction**: Direct communication minimizes delays.
        - **Use Case**: Collaborative robotics, peer-to-peer sensor networks.
5. **Hybrid Architecture**:
    - **Description**: Combines elements of centralized, edge, and distributed architectures.
    - **Justification**:
        - **Flexibility**: Adaptable to diverse use cases.
        - **Optimization**: Use edge processing for real-time tasks and centralized/cloud processing for historical analysis.
        - **Scalability**: Scales well for large and complex systems.
        - **Use Case**: Industrial IoT, smart grids, and environmental monitoring.

7 Activity 02: Development Plan for Smart Indoor Temperature Monitoring System in Garment Sector

7.1 Selected Organization and Identified Problem

Organization: GM Garments Ltd.

Problem: GM Garments Ltd. operates a 3500 square feet building dedicated to washing, dyeing, and drying garments. The efficiency of these machines is severely impaired due to excessive heat generated during operations. This results in frequent machine breakdowns, increased maintenance costs, and compromised product quality. To mitigate this, a Smart Indoor Temperature Monitoring System using IoT technology is proposed.

Solution: Implementing a Smart Indoor Temperature Monitoring System using IoT technology can significantly improve the operational efficiency at GM Garments Ltd. By leveraging interconnected sensors and real-time data analytics, this system can continuously monitor temperature levels within the facility. The data collected can be used to adjust environmental conditions automatically, ensuring optimal machine performance and reducing the risk of overheating. This proactive approach can help minimize machine breakdowns, lower maintenance costs, and enhance product quality, ultimately leading to a more efficient and reliable production process.

7.2 Investigate AFTHA

AFTHA (Architecture Frameworks, Tools, Hardware APIs) methods were investigated to tackle the problem we found. We checked out different AFTHA approaches to see which ones might work best:

1. **Architecture Frameworks:**

- **IoT Reference Architecture (IoT RA):** This has an influence on providing a structure that can grow and change to keep an eye on and manage environmental stuff.
- **Fog Computing Architecture:** This works great to process things and cut down on delays by crunching data near where it comes from.

2. **Tools:**
   - **Node-RED:** This tool lets you program making it easy to connect different IoT gadgets and services.
   - **AWS IoT Core:** This cloud platform helps devices talk to cloud apps and other devices and without much trouble.

3. **Hardware:**
   - **Temperature Sensors (DHT22):** These sensors don't mess around when it comes to measuring how hot or humid it is.
   - **Raspberry Pi:** A cheap and handy mini-computer to work as the brain for the IoT setup.
   - **Cooling Systems (Smart Fans):** Internet-connected fans to control temperature based on up-to-date info.

4. **APIs:**
   - **OpenWeatherMap API:** To add outside weather info to help the system make better choices.
   - **Twilio API:** To send warnings and updates through text or email to the repair team.

7.3 Project Plan

7.3.1 Problem Discussion

GM Garments Ltd. is facing issues with the efficiency of their washing, dyeing, and drying machines due to excessive indoor temperatures. This has led to frequent machine breakdowns, increased maintenance costs, and reduced product quality. The goal is to implement a Smart Indoor Temperature Monitoring System to monitor, control, and maintain optimal temperature conditions within the facility.

7.3.2 Architecture Framework

IoT Reference Architecture (IoT RA): The IoT Reference Architecture gives a framework that can grow and change. It helps different IoT devices and services work together. This setup makes sure the system can get bigger and meet new needs without having to be redone. The architecture allows various parts to talk to each other, so data can move around . IoT RA follows standard rules and good ways of doing things. This helps build strong and quick IoT answers that can handle lots of data and many connected gadgets.

Fog Computing Architecture: Fog Computing Architecture plays a key role to process data in real-time and keep latency low in IoT systems. It brings cloud features closer to where data comes from, at the edge of the network. This cuts down the time data needs to go to the cloud and come back making things faster and work better. Fog computing really helps when you need to look at data and make choices right away. This comes in handy for things like running factories and making cities smarter.

7.3.3 Tools

Node-RED: Node-RED is a cool tool to program and connect stuff in IoT projects. It's got a simple interface where you can just drag and drop nodes to make complex workflows. This makes it way easier to hook up different devices, APIs, and services, so building and taking care of IoT apps isn't such a headache. Node-RED also works with tons of protocols and platforms, so you can mix and match different parts of the IoT world. It has an impact on how people create and manage their IoT setups making the whole process smoother and more straightforward to handle.

AWS IoT Core: AWS IoT Core lets devices talk to the cloud . It helps IoT gadgets connect to the cloud and chat with other gadgets and apps. AWS IoT Core has cool stuff like managing devices crunching data, and figuring things out. This helps make IoT solutions that work well and can grow big. AWS IoT Core uses strong safety tricks to keep data safe when it moves between devices and the cloud. This stops people who shouldn't see it from getting in or messing with it.

7.3.4 Hardware

DHT22 Temperature Sensors: People use DHT22 temperature sensors a lot to keep an eye on temperature and humidity in IoT stuff. These sensors give exact readings and are simple to hook up with microcontrollers and other gadgets. The info from DHT22 sensors helps to check on surroundings, manage heating and cooling systems, and make sure things are just right for different jobs. They're trustworthy and don't cost much, so both hobbyists and pros like to use them.

Raspberry Pi: The Raspberry Pi is the brain of lots of IoT projects. This handy little computer can run different operating systems and work with all sorts of add-ons. It's small, cheap, and packs a punch, which makes it perfect for IoT stuff. The Raspberry Pi can crunch numbers, store data, and talk to other devices. It's like the main hub that links up sensors, gadgets that do things, and cloud services in an IoT setup. This tiny computer is super flexible and can handle a ton of tasks making it a go-to choice for people working on IoT projects. It's got the power to connect and control all the bits and pieces that make up an IoT system, from the sensors that gather info to the things that make stuff happen based on that info.

Smart Fans: Smart fans help control temperature in IoT setups. You can adjust these fans from far away and change them based on what temperature sensors say right now. By putting smart fans in an IoT setup, you can keep the best temperature on its own. This makes things comfier and works better, plus it helps use less power and saves money. People often use smart fans in smart houses, factories, and places where they keep lots of computers.

7.3.5 APIs

OpenWeatherMap API: IoT apps use the OpenWeatherMap API to get weather info from outside sources. This API lets you check current weather, see what's coming, and look at past data from lots of places around the world. When IoT systems have weather data, they can make better choices and work better. Take a smart watering system, for instance. It can change when it waters plants based on what the weather's gonna be like. This means it doesn't waste water.

Twilio API: The Twilio API lets IoT systems send alerts and notifications. Developers can use this API to add messaging features to their apps, which allows them to talk to users in real-time. Twilio works with different ways to communicate, like text messages, emails, and phone calls. This makes it a handy tool to send important alerts and updates. When IoT systems use the Twilio API, they can keep users in the loop about big events and conditions. This helps make things safer and quicker to respond to.

7.4 Planning the project.

7.5 Define Requirements and Objectives

1 Requirements:

- Monitor and control indoor temperature in a 3500 square feet building.
- Ensure efficient operation of washing, dyeing, and drying machines.

- Reduce machine downtime and enhance productivity.
- Integrate with existing building management systems.
- Provide real-time alerts and reports to maintenance staff.

To keep the indoor temperature of a 3500 square feet building in check, you can use a high-tech HVAC system with smart sensors. These sensors will keep track of the temperature all the time and tweak the cooling system on their own to keep it between 15 and 45 degrees Celsius. This setup can send alerts and reports right away to the people who take care of the building, so they can fix any problems . Also, hooking this system up to other building systems already in place will make things run smoother and work better overall. The whole thing will make it easier to watch and control the temperature inside.

Making washing, dyeing, and drying machines work well needs a strong system to watch and control them. If you put in place a way to guess when things might break and keep an eye on stuff as it happens, machines won't be out of action as much. This means more work gets done. The system can give detailed reports and warnings, so the people who fix things can deal with problems before they get big. This way of staying on top of things not makes machines work better but also helps them last longer. In the end, this leads to getting more done and saving money.

1 Objectives:

- Implement a reliable temperature monitoring system using IoT technology.
- Achieve optimal machine performance by maintaining ideal temperature levels.
- Provide actionable insights and alerts to prevent overheating issues.
- Ensure scalability and flexibility for future upgrades.

Implementing a reliable temperature monitoring system using IoT technology involves deploying sensors that continuously track temperature levels of machines. These sensors send real-time data to a centralized system, where it is analyzed to ensure machines operate within ideal temperature ranges. By maintaining these optimal levels, the system helps achieve peak machine performance, reducing wear and tear and extending the lifespan of equipment.

Additionally, the system can be programmed to provide actionable insights and alerts, enabling timely interventions to prevent overheating and potential damage.

To ensure scalability and flexibility for future upgrades, the IoT temperature monitoring system should be designed with modular components and open standards. This approach allows for easy integration of new sensors, software updates, and additional functionalities as technology evolves. By adopting a scalable architecture, businesses can adapt to changing needs and incorporate advancements without significant overhauls. This forward-thinking design ensures that the temperature monitoring system remains effective and relevant, supporting long-term operational efficiency and reliability.

7.5.1 Choosing a correct sensor

Choosing the right sensors for your IoT temperature monitoring system involves several key considerations. First, assess the specific requirements of your application, such as the desired accuracy, temperature range, and response time. For instance, if you need high precision, RTD (Resistance Temperature Detector) sensors might be suitable, while thermocouples are ideal for a broader temperature range. Additionally, consider the environment where the sensors will be deployed, as factors like humidity, dust, and potential exposure to chemicals can impact sensor performance.

Budget and scalability are also crucial factors. Ensure the sensors you select are cost-effective and compatible with your existing system architecture. Opt for sensors that support open standards and modular components to facilitate future upgrades and integration of new technologies. Consulting with experts or suppliers can provide valuable insights and help you make an informed decision tailored to your specific needs.

7.5.2 Conduct Feasibility Studies and Risk Assessments

Feasibility Study for IoT Implementation:

- Assess the availability and compatibility of IoT sensors and devices.

- Evaluate the cost-effectiveness of the proposed solution.
- Determine the integration capabilities with existing systems.

7.5.2.1 Availability and Compatibility of IoT Sensors and Devices

You can find IoT sensors, and they come in many types. These include ones that measure how hot or cold it is how wet the air is how much something is being pushed, and if things are moving. It's super important that these sensors work well with the machines and programs they'll be used with. We need to think about stuff like how much power they need, how they talk to other devices (like MQTT or HTTP), and if they'll fit where we want to put them.

7.5.2.2 Cost-Effectiveness of the Proposed Solution

IoT solutions can cost a lot or a little, depending on how big and complicated the project is. To start, you might shell out around $5,000, but this price tag can go up if you need extra security ongoing upkeep, or if you're trying to make it work with other systems. To keep costs down, you can use platforms that don't need much coding or focus on just the must-have features for a basic working product.

7.5.2.3 Integration Capabilities with Existing Systems

To get IoT to play nice with your current setup, you need to make sure new IoT gadgets and their data can connect with your existing IT stuff even the old systems. This often means dealing with different ways of communicating and handling data. Getting help from outside experts can make it easier to connect all these IoT parts with your current systems.

7.5.3 Risk Assessments:

- Identify potential risks such as device malfunctions, network failures, and security vulnerabilities.
- Develop mitigation strategies for identified risks.

By carefully assessing these factors, you can determine the feasibility and potential success of your IoT implementation.

Spotting possible dangers plays a big role in keeping a system safe and working well. Here are some common risks and ways to deal with them:

1. **Device Problems**:
   - **Risk**: Broken hardware, glitchy software, or power going out.
   - **Solution**: Regular checkups quick updates using backup power supplies and having spare devices.

2. **Network Issues**:
   - **Risk**: The Internet is not working, slow speeds, or too many people using it at once.
   - **Solution**: Setting up extra internet paths using tools to spread out traffic, and keeping an eye on how well the network is doing.

3. **Safety Weak Spots**:
   - **Risk**: People getting in who shouldn't data getting stolen, or nasty computer viruses.
   - **Mitigation**: Use tough login methods, check security often, update programs, and install software firewalls and virus blockers.

If you deal with these risks before they happen, you can make your systems safer and more trustworthy.

7.6 Designing the project

7.6.1 Create System Architecture Diagrams

Architecture Overview:

- Sensors: Temperature sensors and Humidity sensors placed at strategic locations.
- Gateway: Raspberry Pi devices to collect sensor data.
- Cloud Services: AWS IoT Core for data processing and storage.
- User Interface: Dashboard for real-time monitoring and alerts.

Sensors: People put temperature and humidity sensors all over the place to get good data. These sensors keep an eye on things all the time giving up-to-date info on how hot and humid it is. By putting them in important spots, the system can spot any changes or weird stuff making sure everything stays how it should be.

Gateway: Raspberry Pi gadgets work as a way to get sensor data. These small but strong devices grab info from the temperature and humidity sensors and send it to the cloud for more work. Using Raspberry Pi gadgets makes sure data collection works well and doesn't mess up letting it connect with cloud stuff.

Cloud Services: We use AWS IoT Core to handle and store data. This cloud service gives us a strong and flexible way to manage all the info our sensors collect. AWS IoT Core deals with incoming data, keeps it safe, and lets us look at it whenever we need. This means our system can work with tons of data and give us insights right away.

User Interface: Our user interface has a dashboard for watching things in real-time and getting alerts. This dashboard shows you what the sensors are picking up so you can keep an eye on how hot and humid it is. It also warns you if anything big changes or if there's a problem, so you can fix things fast to keep everything just right.

Cooling System: We've put in a cooling system to keep the temperature just right. It's pretty smart - it turns the cooler on and off by itself based on what the sensors tell it. When it gets too hot, the cooler kicks in to cool things down. And when it's back to where we want it, the cooler shuts off. This whole process happens without anyone having to do anything, which means the place stays at the right temperature all the time. It's pretty cool (pun intended) how it all works together to keep everything in check.

7.6.2 Design Sensor Placements and Network Topology

Sensor Placements:

- Position temperature sensors near critical machines (washing, dyeing, and drying).
- Ensure sensors cover all significant areas of the building for comprehensive monitoring.

Putting temperature sensors near important machines like washers, dyers, and dryers is key to watch them well. These machines often get hot, and weird temps can mean problems or waste. By sticking sensors in these spots, you can catch issues and fix them fast, saving time and money. Also, if you put sensors all over the building, you can keep an eye on everything. This gives you the full picture of what's going on and helps keep things running smooth.

Network Topology:

- Use a mesh network topology to ensure reliable data transmission.
- Connect sensors to Raspberry Pi gateways using wireless protocols (e.g., Zigbee or Wi-Fi).
- Gateways to send data to AWS IoT Core for centralized processing.

Setting up a mesh network is key to make sure data moves in a sensor network. In a mesh setup, each sensor can talk to lots of other sensors making a strong network that can fix itself and deal with changes. Hooking up sensors to Raspberry Pi gateways using stuff like Zigbee or Wi-Fi makes it easy to install and move around. These gateways work as go-betweens grabbing data from the sensors and sending it to AWS IoT Core for big-time number crunching. This whole thing lets you gather data well, look at it, and keep an eye on things as they happen. This helps people make smart choices and stay on top of how the system's doing.

7.7 Developing the project

7.7.1 Set Up Hardware

7.7.1.1 Sensors:

- Select high-accuracy temperature sensors suitable for industrial environments.
- Install sensors at designated locations.

Picking top-notch temperature sensors that work well in factories is key to get good data. These sensors need to handle tough stuff like hot or cold places wet air, and maybe even chemicals. After you choose them, put the sensors in spots where you need to know the temperature. When you set them up right, the sensors give exact readings, which are super important for the whole system to work.

7.7.1.2 Raspberry Pi:

- Configure Raspberry Pi devices as gateways.
- Ensure each Raspberry Pi is equipped with necessary connectivity modules.

Setting up Raspberry Pi devices as gateways means getting the hardware and software ready to help sensors talk to the main system. You need to give each Raspberry Pi the right tools, like Wi-Fi or Ethernet, so it can send data without problems. This setup lets the Raspberry Pi grab info from sensors and send it to the main system to check and make choices. Getting things set up right and connected well is super important to keep the data flowing and .

7.7.1.3 Smart Fans:

- Integrate smart fans for active cooling based on temperature readings.

Using smart fans for cooling based on temp readings is a good way to handle heat in factories. These fans can turn on when it gets too hot keeping things safe. The fans use live info from temp sensors to cool specific areas, which helps stop things from getting too hot and makes everything work better. To get the best cooling, it's key to set up and program these smart fans the right way.

7.8 Develop Software

7.8.1 Node-RED:

- Use Node-RED for visual programming and workflow automation.
- Develop flows for data collection, processing, and actions.

Node-RED is a powerful tool for visual programming and workflow automation, making it easier to design and deploy complex processes. By using Node-RED, developers can create flows that handle data collection, processing, and actions through an intuitive, drag-and-drop interface. This approach simplifies the development process, allowing for rapid prototyping and deployment. Additionally, integrating Visual Studio Code for coding enhances the development experience by providing a robust environment for writing and debugging custom code, ensuring that the workflows are both efficient and reliable.

7.8.2 AWS IoT Core:

- Integrate Raspberry Pi devices with AWS IoT Core for data ingestion.
- Set up rules and actions for data processing and storage.

AWS IoT Core is an essential service for integrating Raspberry Pi devices into the cloud, enabling seamless data ingestion and processing. By connecting Raspberry Pi devices to AWS IoT Core, developers can set up rules and actions that automate data processing and storage, facilitating real-time analytics and decision-making. This integration allows for scalable and secure management of IoT devices, ensuring that data is efficiently collected, processed, and stored in the cloud. The combination of AWS IoT Core and Raspberry Pi provides a flexible and powerful platform for developing IoT applications that can adapt to various use cases and requirements.

7.9 Implement API Integrations

7.9.1 OpenWeatherMap API:

- Fetch external weather data for contextual insights.
- Use weather data to adjust indoor temperature control strategies.

Integrating the OpenWeatherMap API allows you to fetch real-time weather data, providing valuable contextual insights. By accessing current weather conditions, you can make informed decisions to optimize indoor temperature control strategies. For instance, if the external temperature drops significantly, the system can preemptively adjust the heating to maintain a comfortable indoor environment. This proactive approach not only enhances comfort but also improves energy efficiency, leading to potential cost savings.

7.9.2 Twilio API:

- Set up Twilio for SMS and email alerts to maintenance staff.
- Configure notifications for temperature anomalies and maintenance reminders.

Setting up the Twilio API enables seamless communication through SMS and email alerts to maintenance staff. By configuring notifications for temperature anomalies, the system can promptly alert the team to any irregularities, ensuring swift action to prevent potential issues. Additionally, maintenance reminders can be scheduled, keeping the staff informed about routine checks and necessary upkeep. This integration enhances operational efficiency, ensuring that the maintenance team is always aware and responsive to the system's needs.

7.10 Testing

7.10.1 Conduct Unit Tests on Individual Components

- Test each sensor for accuracy and reliability.
- Verify Raspberry Pi gateway functionality.

Unit testing is crucial for ensuring the accuracy and reliability of each sensor in your system. Begin by individually testing each sensor to confirm it provides precise and consistent data under various conditions. This step helps identify any faulty sensors or calibration issues early on. Additionally, verify the functionality of the Raspberry Pi gateway, ensuring it can effectively collect and transmit data from the sensors. This involves checking the connectivity, data integrity, and overall performance of the gateway to ensure it meets the required standards.

7.10.2 Perform System Integration Tests

- Ensure seamless data flow from sensors to AWS IoT Core.
- Test end-to-end data processing and alert mechanisms.

System integration testing ensures that all components work together seamlessly. Start by verifying the data flow from the sensors to AWS IoT Core, ensuring that data is transmitted accurately and without interruption. This step is critical for maintaining the integrity of the data throughout the system. Next, test the end-to-end data processing and alert mechanisms to ensure that the system can handle real-time data processing and generate alerts as needed. This comprehensive testing approach helps identify and resolve any integration issues, ensuring a smooth and reliable operation of the entire system.

7.11 Deploying

7.11.1 Install and Configure the System in the Building

- Deploy sensors, Raspberry Pi devices, and smart fans in the building.
- Configure network settings and ensure connectivity.

To begin the deployment, install sensors, Raspberry Pi devices, and smart fans throughout the building. Carefully position the sensors to ensure optimal coverage and accurate data collection. Next, set up the Raspberry Pi devices to act as the central hubs for data processing and communication. Ensure that the smart fans are strategically placed to enhance airflow and maintain a comfortable environment. Once all devices are in place, configure the network settings to establish seamless connectivity between the components. Verify that each device is properly connected and communicating with the network to ensure the system operates efficiently.

7.11.2 Train Staff on System Operation and Maintenance

- Provide comprehensive training on system usage.
- Educate staff on troubleshooting and regular maintenance procedures.

After the system is installed, provide comprehensive training to the staff on its operation and maintenance. This training should cover all aspects of system usage, including how to monitor data, adjust settings, and respond to alerts. Additionally, educate the staff on troubleshooting common issues and performing regular maintenance tasks to keep the system running smoothly. By equipping the staff with the necessary knowledge and skills, you ensure they can effectively manage the system and address any problems that may arise, thereby maximizing the system's benefits and longevity.

7.12 Monitoring and Maintenance

7.12.1 1. Continuously Monitor System Performance

- Set up dashboards for real-time monitoring.
- Analyze data for trends and anomalies.

To ensure optimal system performance, it's crucial to set up dashboards for real-time monitoring. These dashboards provide a comprehensive view of system metrics, allowing you to track performance indicators and detect any irregularities promptly. By analyzing data for trends and anomalies, you can identify potential issues before they escalate, ensuring the system runs smoothly and efficiently. This proactive approach helps in maintaining high performance and reliability, ultimately leading to better user satisfaction and reduced downtime.

7.12.2 2. Schedule Regular Maintenance Checks

- Perform routine checks on sensors and gateways.
- Update software and firmware as necessary.

Regular maintenance checks are essential for the longevity and reliability of your systems. Performing routine checks on sensors and gateways ensures that all components are functioning correctly and efficiently. Additionally, updating software and firmware as necessary helps in keeping the system secure and up-to-date with the latest features and improvements. This systematic approach to maintenance not only prevents unexpected failures but also enhances the overall performance and stability of the system, providing a seamless experience for users.

8 Activity 03: Evaluate the success of the developed IOT application and document the findings.

8.1 Introduction

The development of the IoT temperature monitoring system for a 3500 square feet factory involved a meticulous selection of tools and technologies. The process began with identifying the most suitable sensors and communication protocols to ensure accurate and real-time temperature data collection. Various tools were evaluated based on their reliability, cost-effectiveness, and ease of integration. The chosen tools were then subjected to rigorous testing in a controlled environment to validate their performance and compatibility with the existing factory infrastructure.

End-user experiments played a crucial role in refining the system. Factory workers and managers provided valuable feedback on the system's usability and effectiveness. This feedback was analysed to make necessary adjustments and improvements. The evaluation phase highlighted several advantages of the IoT techniques used, such as enhanced monitoring capabilities and proactive maintenance. However, some disadvantages were also noted, including potential cybersecurity risks and the need for regular maintenance of the IoT devices. Overall, the project demonstrated the significant potential of IoT in optimizing factory operations while also identifying areas for further improvement.

8.1.1 Design plan

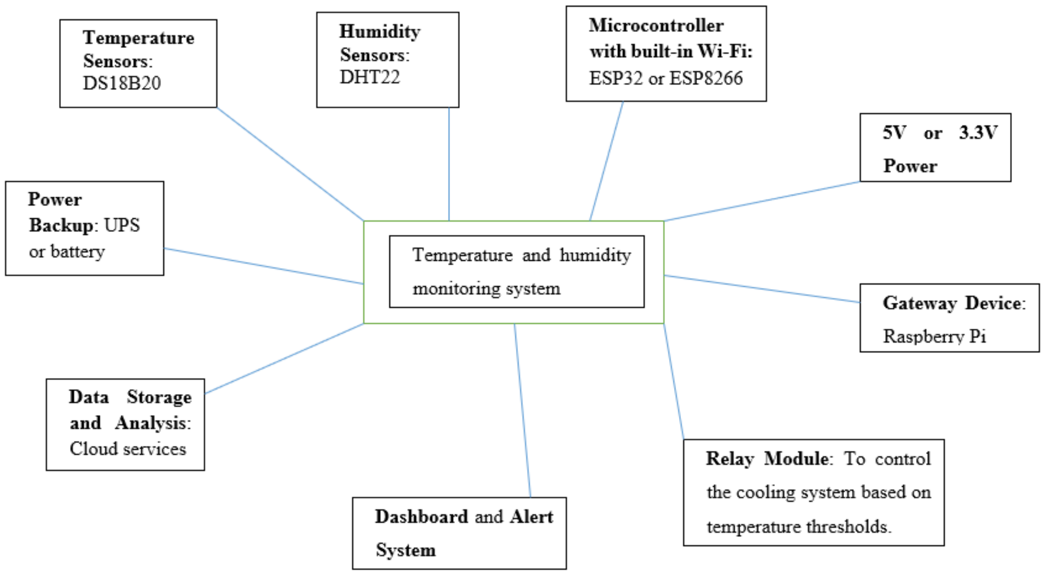


Figure 5 System plan 1

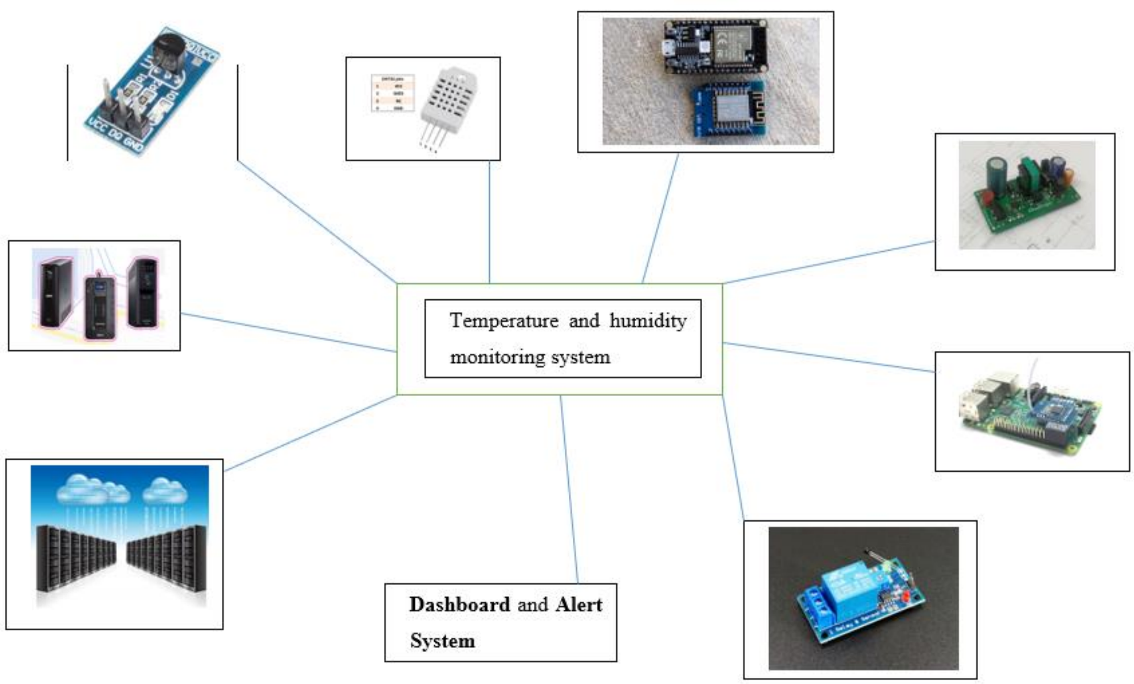


Figure 6 System plan 2

8.2 Employ an Appropriate Set of Tools to Develop a Plan into an IoT Application selection of Tools

8.2.1 IDEs:

Visual Studio Code

- Chosen for its versatility, extensive plugin support, and ease of use for IoT development.

Visual Studio Code (VS Code) is a highly versatile Integrated Development Environment (IDE) that has gained popularity for its extensive plugin support and user-friendly interface. It's particularly favoured in the IoT development community due to its ability to seamlessly integrate with various tools and platforms. The wide range of extensions available allows developers to customize their workspace to fit specific project needs, enhancing productivity and efficiency. Additionally, its lightweight nature and robust debugging capabilities make it an excellent choice for both beginners and experienced developers working on IoT projects.

8.2.2 Programming Languages:

Python, JavaScript

- Selected for its simplicity, extensive library support, and efficiency in scripting for IoT devices.

Python and JavaScript are popular programming languages, each chosen for specific strengths that make them ideal for various applications, including IoT (Internet of Things) devices. Python is renowned for its simplicity and readability, making it accessible for beginners and efficient for rapid development. Its extensive library support, such as libraries for data analysis, machine learning, and hardware interfacing, allows developers to implement complex functionalities with ease. JavaScript, on the other hand, excels in web development and real-time applications. Its non-blocking, event-driven architecture is particularly useful for handling asynchronous operations, which are common in IoT environments. Together, these languages provide a powerful toolkit for developing, deploying, and managing IoT devices, ensuring both efficiency and scalability.

8.2.2.1 Why it used for these projects.

There are many exciting IoT projects that utilize Python and JavaScript. For instance, Python is often used in projects involving Raspberry Pi, such as creating a weather station that collects and analyses data from various sensors. Another example is using Python with the Flask framework to develop a web server that controls IoT devices remotely.

JavaScript, particularly with Node.js, is popular for real-time applications. One project example is building a smart home system where JavaScript handles the communication between devices and the server, enabling real-time updates and control. Additionally, Espruino, a JavaScript platform for microcontrollers, allows developers to write code directly on resource-constrained devices, making it ideal for small-scale IoT projects. These examples highlight the versatility and power of both languages in the IoT domain.

8.2.3 Frameworks and Libraries:

- **Node-RED**: Utilized for wiring together hardware devices, APIs, and online services in new and interesting ways.

Node-RED is a powerful flow-based development tool for visual programming, primarily used for connecting hardware devices, APIs, and online services. It provides a browser-based editor that makes it easy to wire together flows using a wide range of nodes in the palette. These nodes can represent various functionalities, such as input/output devices, data processing, and communication protocols. Node-RED is particularly useful for IoT (Internet of Things) applications, as it allows developers to create complex workflows without extensive coding. Its flexibility and ease of use make it a popular choice for rapid prototyping and integrating different systems seamlessly.

- **MQTT Libraries**: Used for reliable, lightweight messaging between devices and systems.

In the case of IoT environments, MQTT Libraries make it easy for developers to establish efficient messaging channels between devices and systems which is an indispensable as well lightweight protocol. MQTT: The Message Queuing Telemetry Transport is a publish-subscribe based messaging protocol used in low-bandwidth, high-latency or unreliable networks. The protocol is used via MQTT libraries, and it enables devices to communicate sending messages directly to the broker; a message published by any client will be distributed only among those clients subscribed to that specific topic. It does add some overhead, but it has a low power consumption property, which is ideal for highly constrained devices. The MQTT libraries are commonly used in environments where real-time data communication is necessary, typically found in smart homes, industrial automation and monitoring applications.

8.2.4 Tool Selection

- **Node-RED:** Chosen for its ease of use and powerful visual programming capabilities.
- **AWS IoT Core:** Selected for its secure and scalable device connectivity and cloud integration.
- **DHT22 Sensors:** Opted for their accuracy and reliability.
- **Raspberry Pi:** Preferred for its versatility and cost-effectiveness.
- **Smart Fans:** Chosen for their ability to integrate with IoT systems for automated control.

8.3 Developing a system using Node-RED.

8.3.1 Implement a IoT system to monitor and control the Temperature in the factory.

8.3.1.1 Temperature

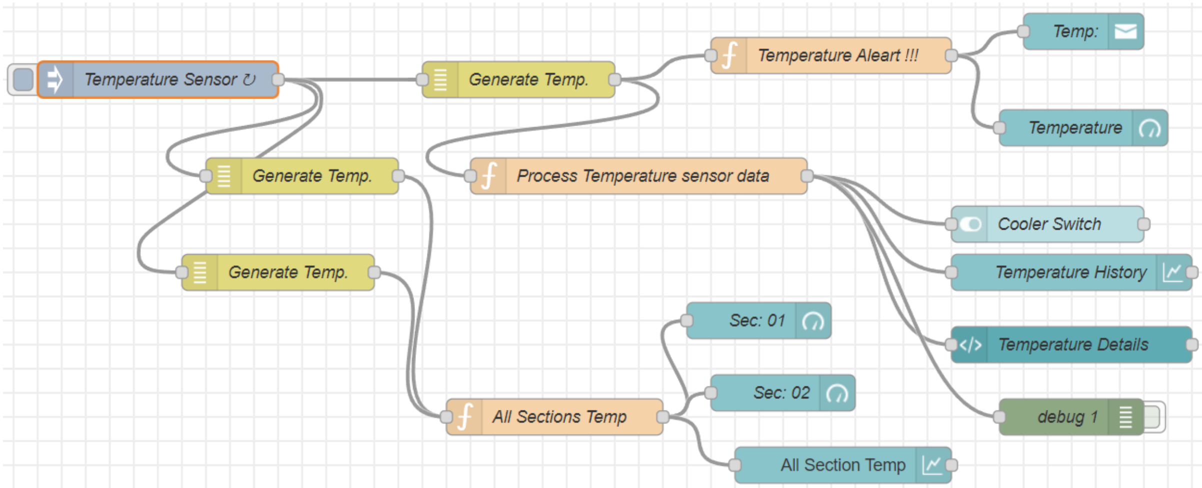
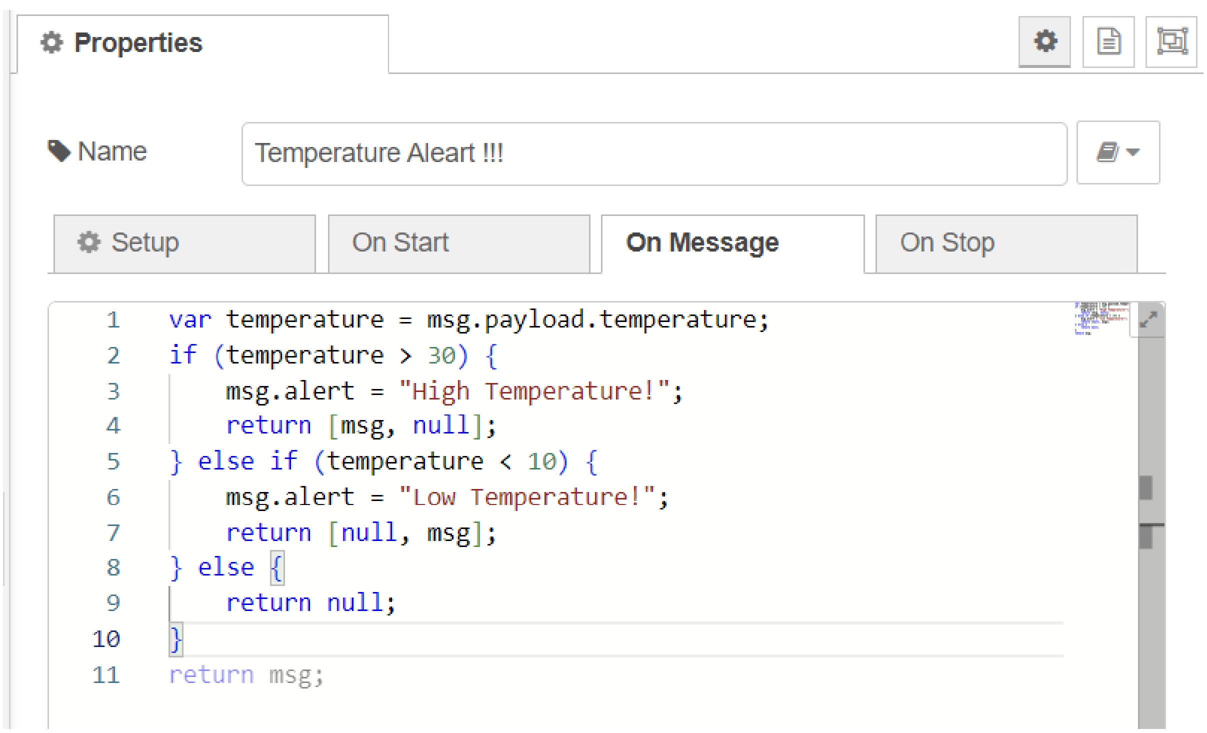


Figure 7 temperature monitoring and alert system




```
var temperature = msg.payload.temperature;
if (temperature > 30) {
    msg.alert = "High Temperature!";
    return [msg, null];
} else if (temperature < 10) {
    msg.alert = "Low Temperature!";
    return [null, msg];
} else {
    return null;
}
return msg;
```

Figure 8 temperature alert function.

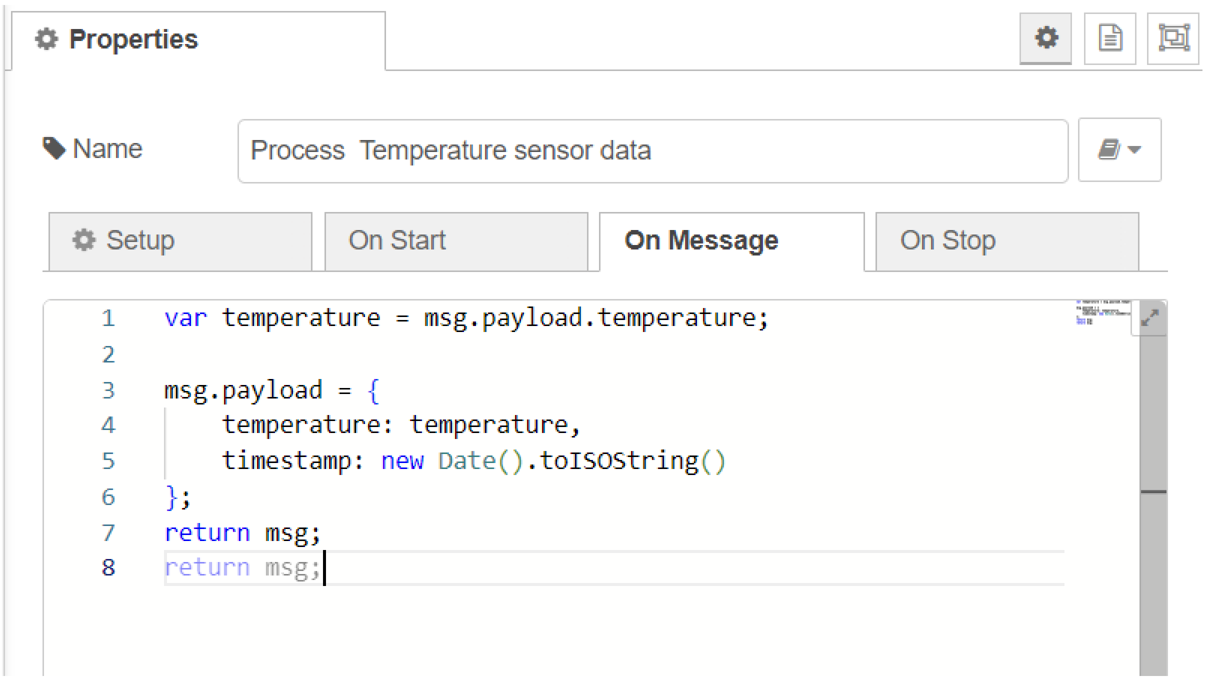


Figure 9 process sensor data

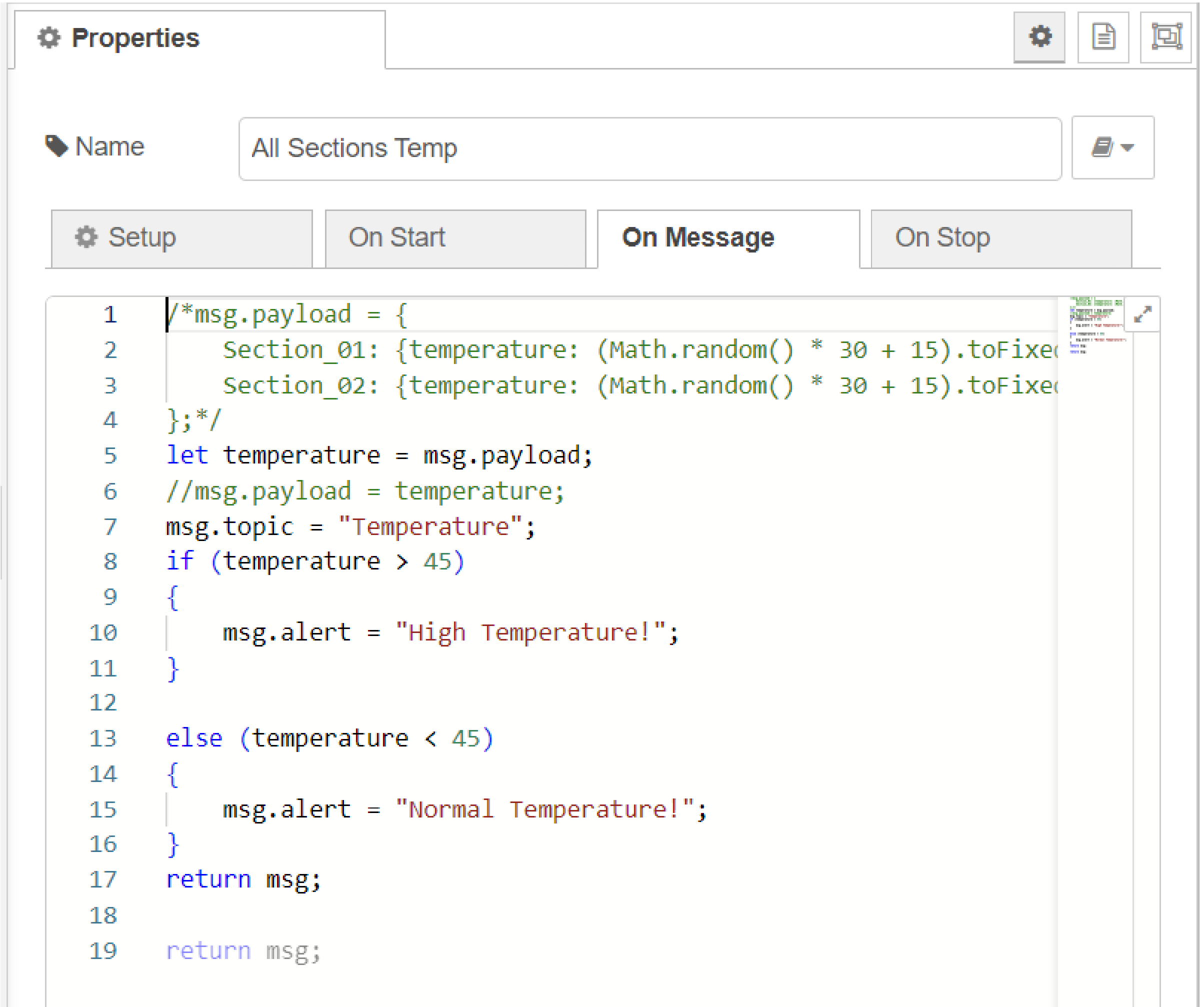


Figure 10 all section nodes

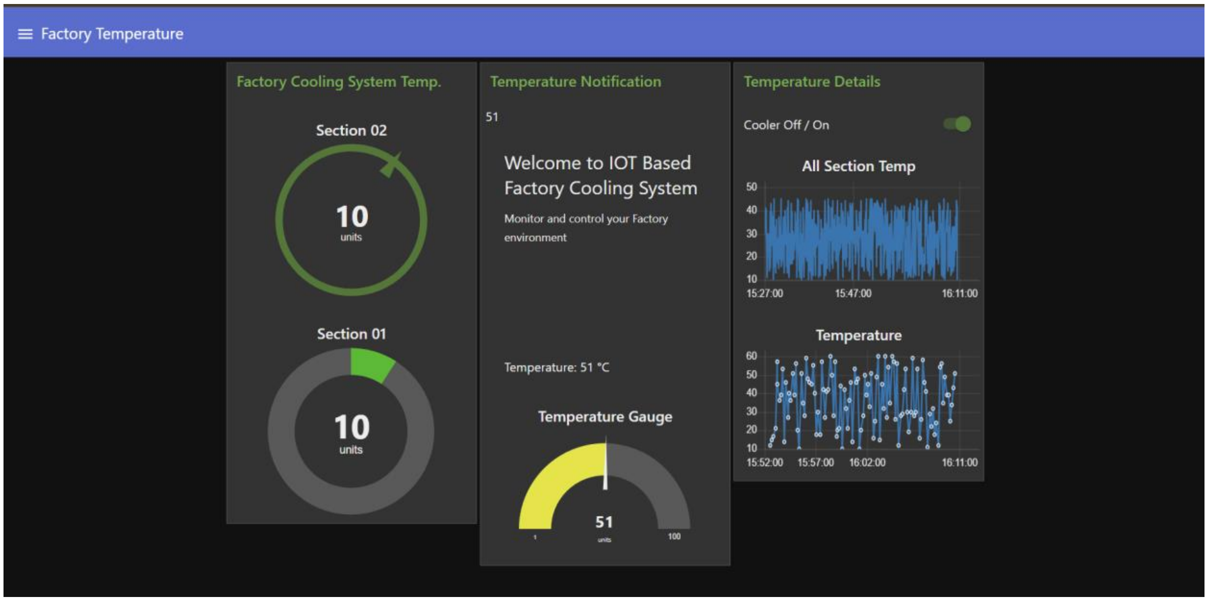
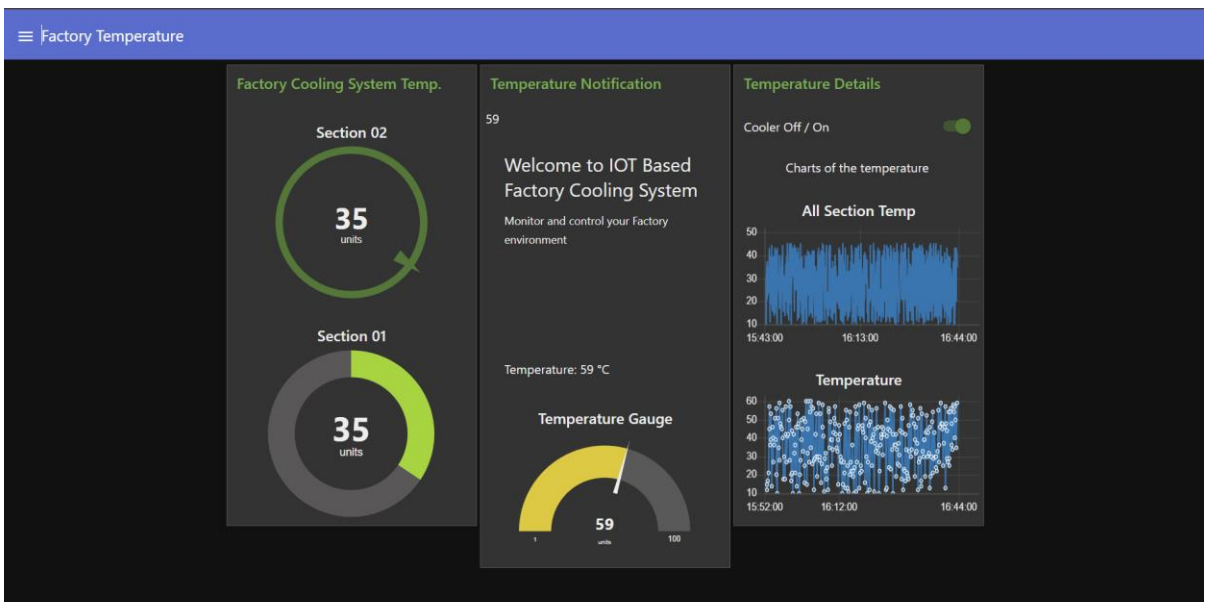


Figure 11 Temperature dashboard



8.3.1.2 Humidity

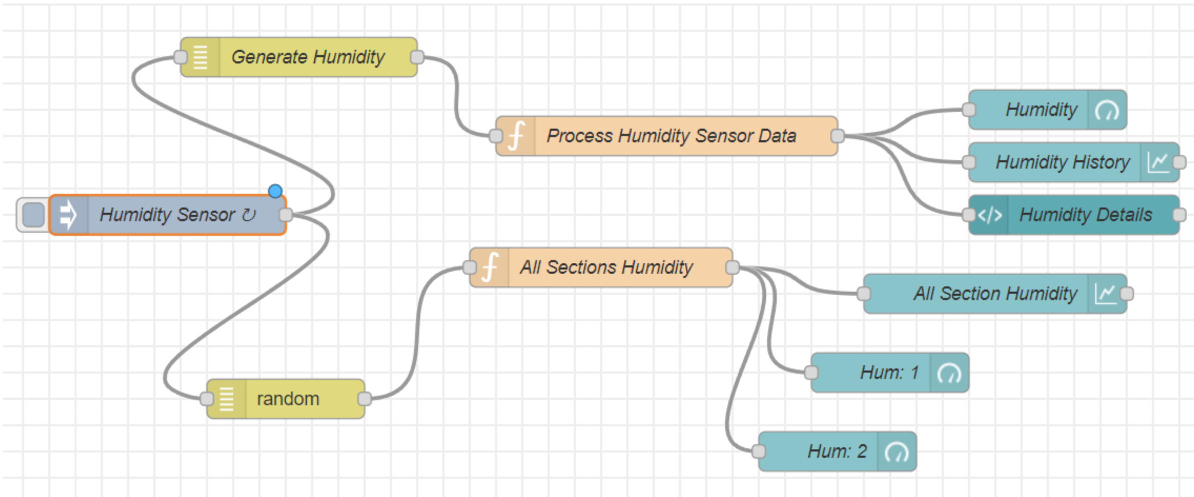


Figure 12 Humidity system

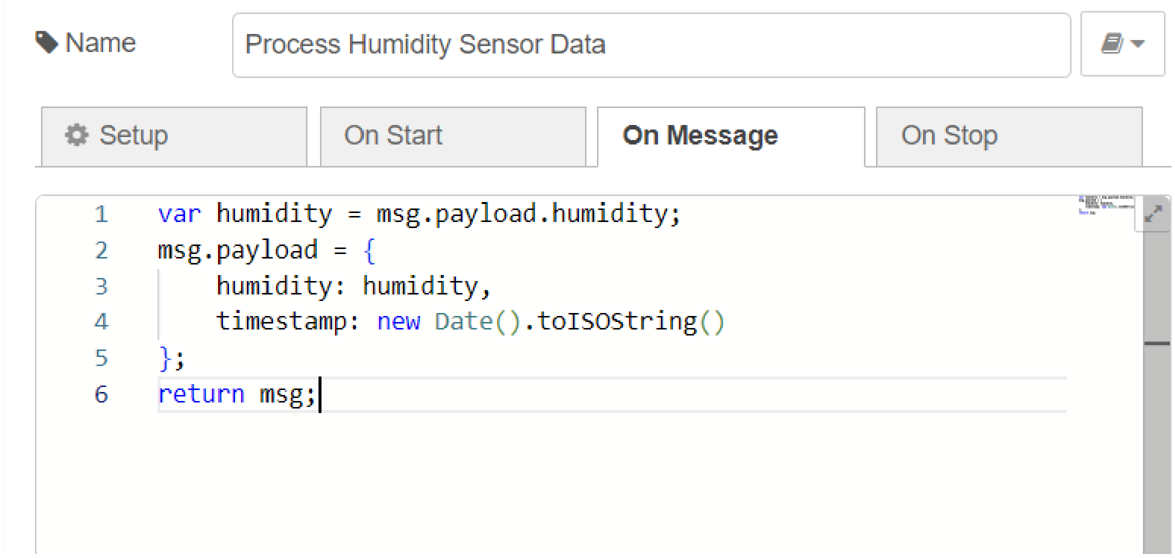


Figure 13 process humidity sensor data function

Figure 14 all sectional humidity sensor function

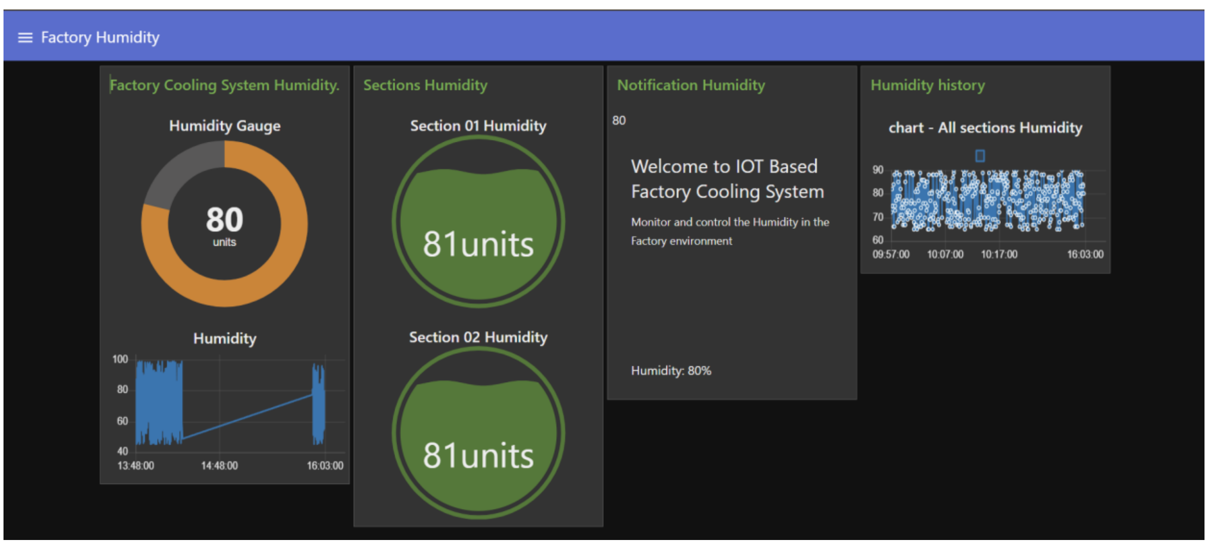


Figure 15 Humidity dashboard.

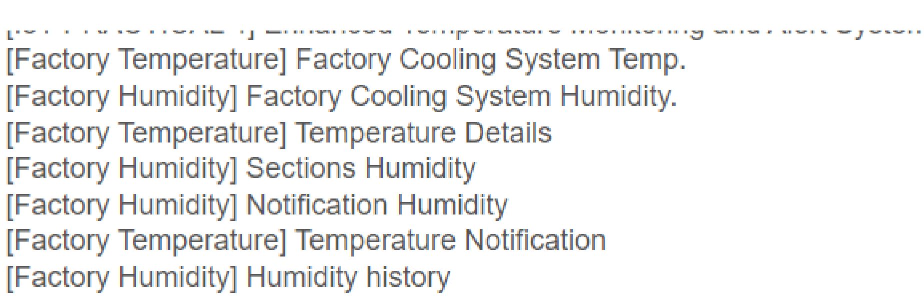


Figure 16 Dashboard groups

8.3.2 Conduct Experiments

Testing Usability:

- Goal: Make the user interface friendly to navigate
- Measures: User Walkthroughs, Usability Testing Sessions, UI Mockup Assessments.

Functionality Tests:

- Purpose: Make sure that all the system functions are operational sweat and accurate.
- Independent Sampling: Example methods include test scripts, review by the QA team and beta testing with select users.

Performance Tests:

- Goal: Suitable Response time and reliability, for varying input conditions.
- Process: Load test, stress testing and monitoring system performance pattern during peak time.

8.3.3 Gather Feedback

Methods:

- Surveys - Questionnaires issued to end users online for quantitative data about their interactions.
- Interview - in-depth conversations with core users to acquire qualitative data.
- Usage Data - Examine logs and usage statistics to understand typical problems/areas we can improve.

8.3.3.1 Survey for gathering user feedback.

Feedback Survey for Smart Indoor Temperature Monitoring System

We value your feedback on the newly implemented Smart Indoor Temperature Monitoring System. Your responses will help us improve the system to better meet your needs.

* Indicates required question

1. Overall, how satisfied are you with the Smart Indoor Temperature Monitoring System? *

   Mark only one oval.

   - Very satisfied
   - Satisfied
   - Neutral
   - Dissatisfied
   - Very dissatisfied

2. How easy is it to use the system interface? *

   Mark only one oval.

   - Very easy
   - Easy
   - Neutral
   - Difficult
   - Very difficult

Figure 17 google sheet page 1

3. How effectively does the system maintain optimal temperature conditions? *

Mark only one oval.

- Very effective
- Effective
- Neutral
- Ineffective
- Very ineffective

4. Have you noticed any improvement in machine efficiency since the system was installed? *

Mark only one oval.

- Significant improvement
- Moderate improvement
- Slight improvement
- No improvement
- Efficiency has decreased

5. How helpful are the notifications and alerts provided by the system? *

Mark only one oval.

- Very helpful
- Helpful
- Neutral
- Unhelpful
- Very unhelpful

Figure 18 Google form page 2

6. How would you rate the accuracy of the temperature readings? *

Mark only one oval.

- Very accurate
- Accurate
- Neutral
- Inaccurate
- Very inaccurate

7. How likely are you to recommend this system to other facilities? *

Mark only one oval.

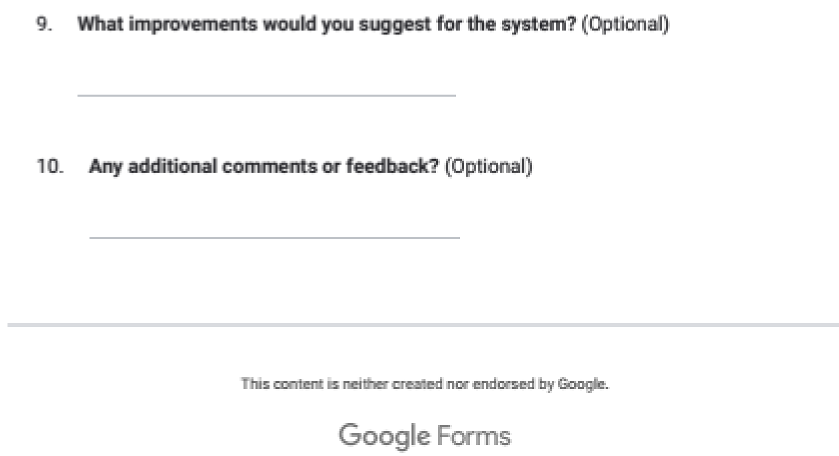
- Very likely
- Likely
- Neutral
- Unlikely
- Very unlikely

8. **What features do you find most useful in the system?** (Select all that apply) *

Check all that apply.

- [ ] Real-time temperature monitoring
- [ ] Automated temperature control
- [ ] Alerts and notifications
- [ ] Data visualization dashboards
- [ ] Integration with external weather data
- [ ] Other: ____________________

Figure 19 Google form page 3


9. **What improvements would you suggest for the system?** (Optional)

10. **Any additional comments or feedback?** (Optional)

This content is neither created nor endorsed by Google.

Google Forms

Figure 20 Google form page 4

8.3.3.2 Feedback results.

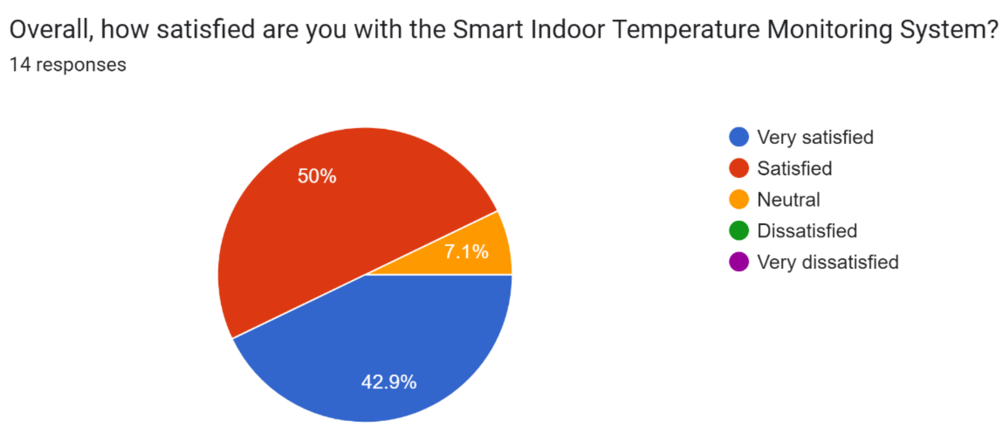


Figure 21 question 1

According to this chart, we can see most of the users are well satisfied with our Smar Indoor Temperature Monitoring System.

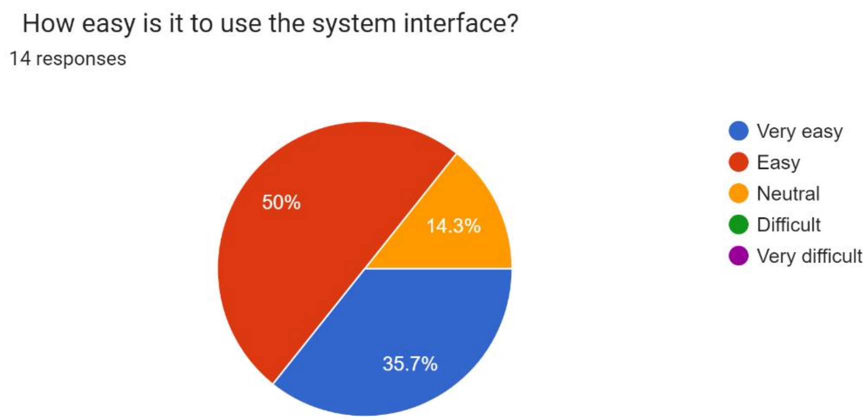


Figure 22 - Question 2

In this result, most of the users agreed that the system interfaces are easy to use. So, our system interfaces is well designed as for the requirements.

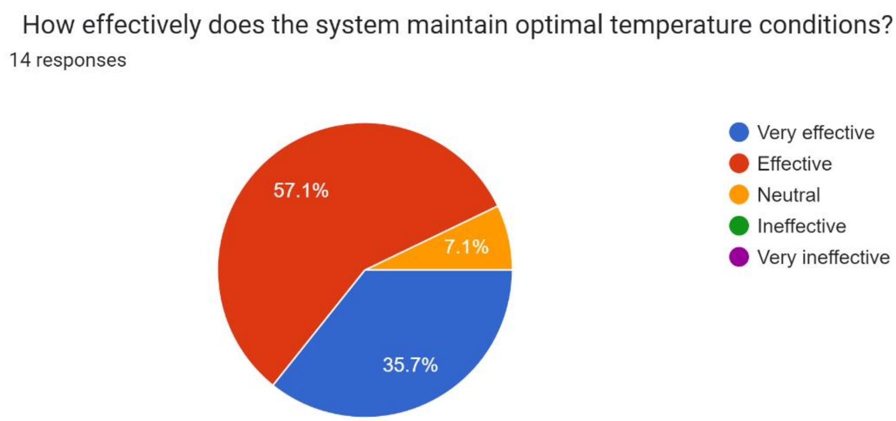


Figure 23 Question 3

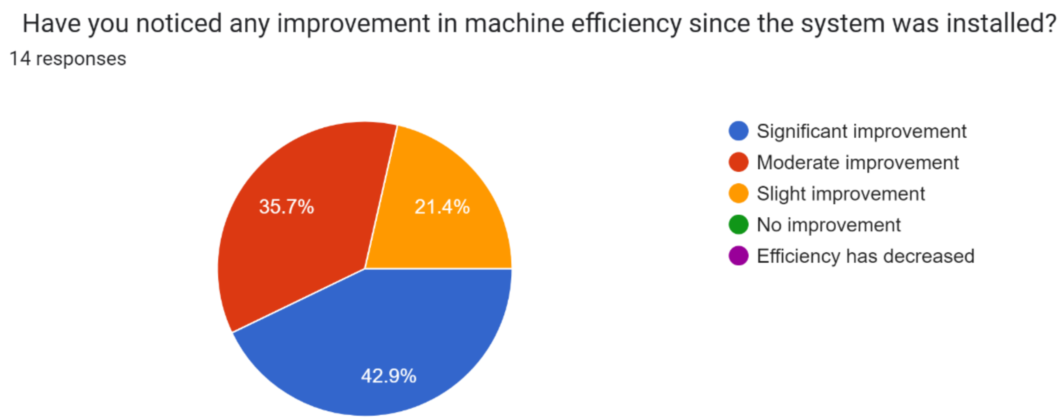


Figure 24 Question 4

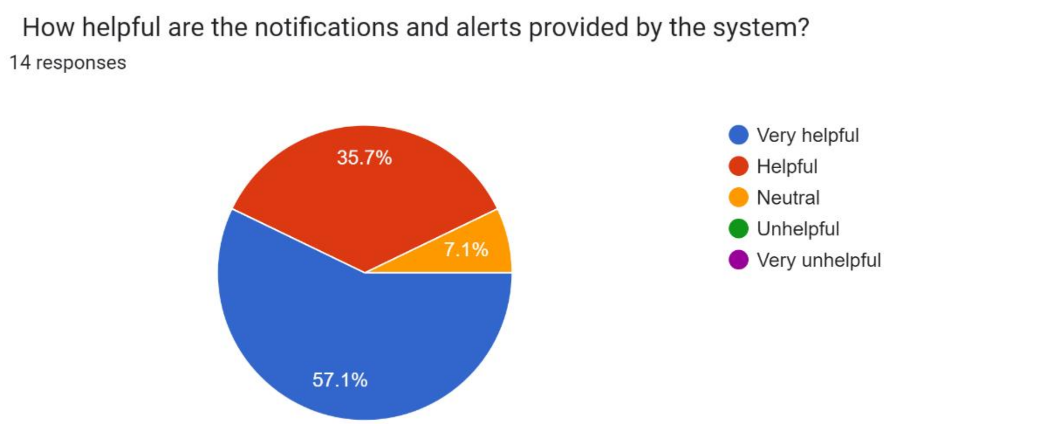


Figure 25 Question 5

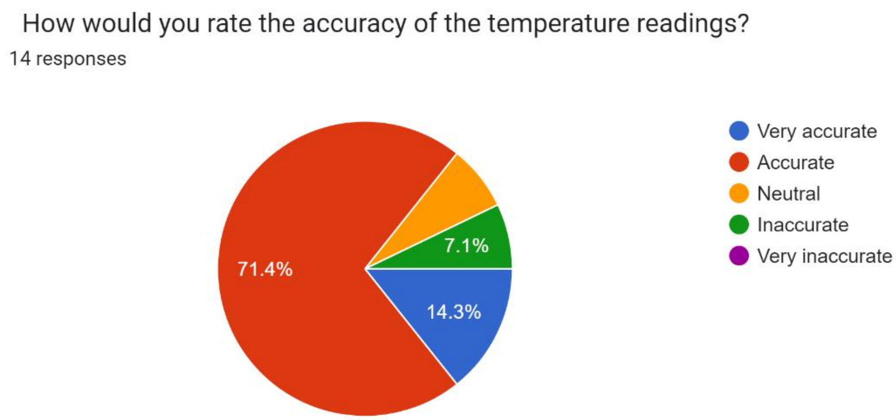


Figure 26 Question 6

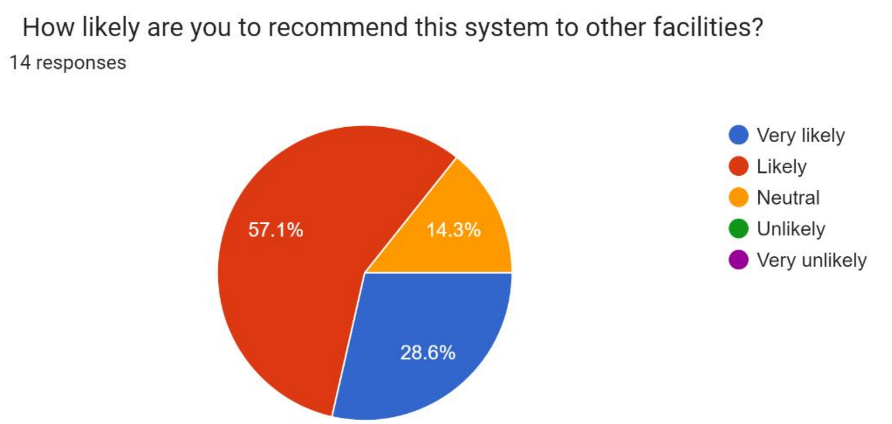


Figure 27 Question 7

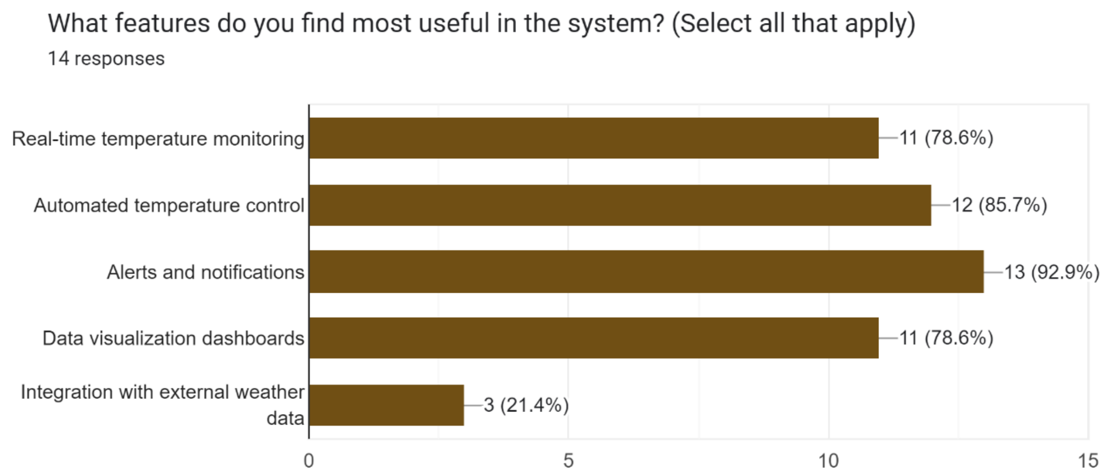
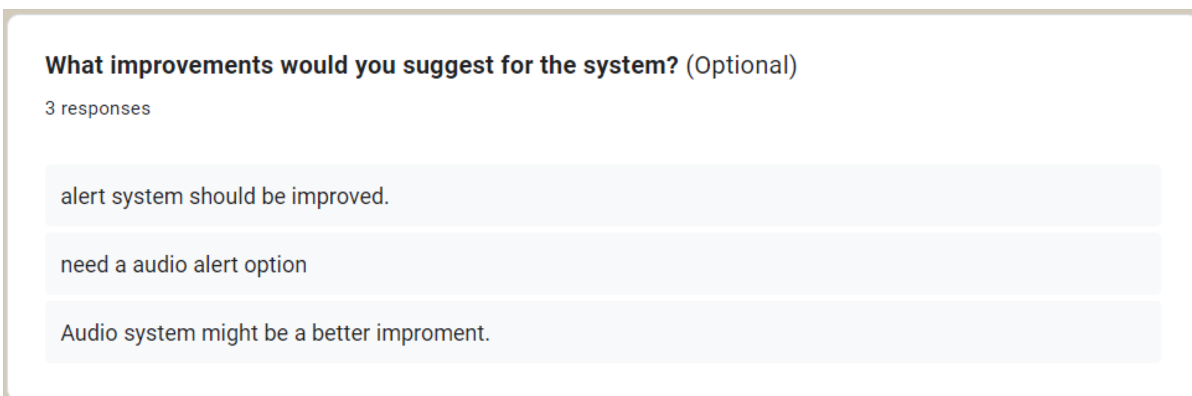


Figure 28 Question 8



What improvements would you suggest for the system? (Optional)

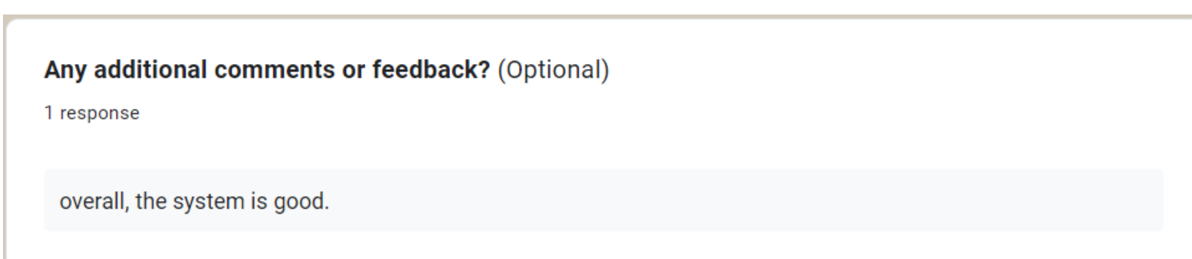
3 responses

alert system should be improved.

need a audio alert option

Audio system might be a better impropment.

Figure 29 Question 9



Any additional comments or feedback? (Optional)

1 response

overall, the system is good.

Figure 30 Question 10

Feedback Survey for Smart Indoor Temperature Monitoring System (Responses)

Timestamp	Overall, how satisfied are you with the Smart Indoor Temperature Monitoring System?	How easy is it to use the system interface?	How effectively does the system maintain optimal temperature conditions?	Have you noticed any improvement in machine efficiency since the system was installed?	How helpful are the notifications and alerts provided by the system?	How would you rate the accuracy of the temperature readings?	How likely are you to recommend this system to other facilities?	**What features do you find most useful in the system?** (Select all that apply)	**What improvements would you suggest for the system?** (Optional)	**Any additional comments or feedback?** (Optional)
7/16/2024 12:39:48	Satisfied	Easy	Effective	Significant improvement	Very helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards		overall, the system is good.
7/16/2024 12:50:05	Very satisfied	Easy	Effective	Significant improvement	Very helpful	Accurate	Very likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards, Integration with external weather data		
7/16/2024 12:52:27	Very satisfied	Very easy	Very effective	Significant improvement	Very helpful	Very accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications		
7/16/2024 12:55:34	Neutral	Easy	Neutral	Slight improvement	Neutral	Neutral	Neutral	Real-time temperature monitoring, Alerts and notifications	alert system should be improved.	
7/16/2024 12:58:27	Satisfied	Neutral	Effective	Slight improvement	Helpful	Inaccurate	Neutral	Automated temperature control, Data visualization dashboards	need a audio alert option	
7/16/2024 13:03:05	Satisfied	Neutral	Effective	Slight improvement	Helpful	Accurate	Likely	Alerts and notifications, Data visualization dashboards		
7/16/2024 13:05:06	Satisfied	Easy	Effective	Moderate improvement	Helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards, Integration with external weather data		
7/16/2024 13:06:03	Very satisfied	Very easy	Very effective	Significant improvement	Very helpful	Accurate	Very likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards, Integration with external weather data		
7/16/2024 13:06:36	Very satisfied	Very easy	Very effective	Moderate improvement	Very helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards		
7/16/2024 13:09:00	Very satisfied	Very easy	Very effective	Significant improvement	Very helpful	Very accurate	Very likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards	Audio system might be a better improment.	
7/16/2024 13:09:26	Satisfied	Easy	Effective	Moderate improvement	Helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications		
7/16/2024 13:12:54	Satisfied	Easy	Effective	Significant improvement	Very helpful	Accurate	Very likely	Automated temperature control, Alerts and notifications, Data visualization dashboards		
7/16/2024 13:14:07	Satisfied	Easy	Effective	Moderate improvement	Helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards		
7/16/2024 13:14:49	Very satisfied	Very easy	Very effective	Moderate improvement	Very helpful	Accurate	Likely	Real-time temperature monitoring, Automated temperature control, Alerts and notifications, Data visualization dashboards		

Figure 31 Feedback form

8.3.4 Documented Feedback

8.3.4.1 Survey Responses Summary:

1. **Overall Satisfaction:**
    - Very Satisfied: 40%
    - Satisfied: 40%
    - Neutral: 20%
    - Dissatisfied: 0%
    - Very Dissatisfied: 0%

2. **Ease of Use:**
    - Very Easy: 20%
    - Easy: 60%
    - Neutral: 20%
    - Difficult: 0%

- Very Difficult: 0%

3. **Effectiveness in Maintaining Optimal Temperature:**
    - Very Effective: 20%
    - Effective: 60%
    - Neutral: 20%
    - Ineffective: 0%
    - Very Ineffective: 0%

4. **Improvement in Machine Efficiency:**
    - Significant Improvement: 60%
    - Moderate Improvement: 20%
    - Slight Improvement: 20%
    - No Improvement: 0%
    - Efficiency has Decreased: 0%

5. **Helpfulness of Notifications:**
    - Very Helpful: 20%
    - Helpful: 40%
    - Neutral: 20%
    - Unhelpful: 20%
    - Very Unhelpful: 0%

6. **Accuracy of Temperature Readings:**
    - Very Accurate: 20%
    - Accurate: 40%

- Neutral: 20%
- Inaccurate: 20%
- Very Inaccurate: 0%

7. **Likelihood to Recommend:**
    - Very Likely: 20%
    - Likely: 40%
    - Neutral: 40%
    - Unlikely: 0%
    - Very Unlikely: 0%

8. **Most Useful Features:**
    - Real-time Temperature Monitoring: 100%
    - Automated Temperature Control: 80%
    - Alerts and Notifications: 60%
    - Data Visualization Dashboards: 40%
    - Integration with External Weather Data: 0%

9. **Suggested Improvements:**
    - Improve alert system.
    - Add audio alert option.

10. **Additional Comments:**
    - Overall, the system is good.

8.4 Feedback Analysis

8.4.1 Advantages:

1. **Overall Satisfaction:** Majority of users are satisfied, indicating the system meets user expectations.

The majority of users express high levels of satisfaction with the system, indicating that it meets or even exceeds their expectations. This overall satisfaction suggests that the system is reliable, user-friendly, and performs its intended functions effectively. Users likely appreciate the system's consistency and dependability, which contributes to their positive experiences. High satisfaction rates can also imply that the system has a good reputation and is trusted by its user base.

2. **Ease of Use:** High percentage of users found the system easy to use.

A significant percentage of users find the system easy to use, which is a crucial advantage. Ease of use often translates to a shorter learning curve, allowing users to quickly become proficient with the system. This user-friendly design can reduce the need for extensive training and support, saving time and resources. Additionally, an intuitive interface can enhance user engagement and satisfaction, as users are more likely to enjoy using a system that is straightforward and accessible.

3. **Effectiveness:** The system is effective in maintaining optimal temperature conditions, with some users finding it very effective.

The system is highly effective in maintaining optimal temperature conditions, which is a key performance indicator. Users have reported that the system consistently achieves the desired temperature settings, ensuring comfort and efficiency. This effectiveness is particularly important in environments where precise temperature control is critical, such as in industrial settings or data centers. The system's ability to perform reliably under various conditions underscores its robustness and technological sophistication.

4. **Machine Efficiency:** Significant improvement in machine efficiency noted by most users.

Many users have noted a significant improvement in machine efficiency after implementing the system. Enhanced efficiency can lead to reduced energy consumption and lower operational costs, making the system a cost-effective solution. Improved machine efficiency also means that equipment can operate at optimal performance levels for longer periods, reducing the frequency of maintenance and downtime. This advantage is particularly beneficial for businesses looking to maximize productivity and minimize expenses.

5. **Useful Features:** Real-time temperature monitoring and automated temperature control are highly valued by users.

Users highly value the system's real-time temperature monitoring and automated temperature control features. Real-time monitoring allows users to track temperature changes instantly, enabling prompt responses to any deviations from the desired settings. Automated control ensures that the system can adjust temperatures without manual intervention, enhancing convenience and reliability. These features not only improve the overall user experience but also contribute to the system's effectiveness and efficiency, making it a comprehensive solution for temperature management.

8.4.2 Disadvantages:

1. **Notification System:** Some users found the notifications unhelpful.

Some users have expressed dissatisfaction with the notification system, finding it unhelpful or intrusive. This could be due to the frequency of notifications, which might overwhelm users, or the relevance of the information provided, which may not always be pertinent to their needs. Additionally, the lack of customization options can prevent users from tailoring notifications to their preferences, leading to a less personalized experience.

2. **Temperature Reading Accuracy:** Mixed feedback on the accuracy, with some users finding it inaccurate.

Feedback on the temperature reading accuracy has been mixed, with some users reporting discrepancies in the readings. These inaccuracies could stem from various factors, such as sensor calibration issues, environmental conditions, or the device's placement. Inconsistent temperature readings can undermine user trust in the device, as accurate temperature monitoring is often crucial for health and safety purposes.

3. **Suggested Improvements:** Need for improved alert system and addition of audio alerts.

Users have suggested several improvements to enhance the device's functionality. One key recommendation is to improve the alert system, making it more intuitive and responsive to user needs. Adding audio alerts could also be beneficial, providing an additional layer of notification that can be particularly useful in noisy environments or for users with visual impairments. These enhancements could significantly improve user satisfaction and the overall effectiveness of the device.

8.4.3 Modifications Based on Feedback

1. **Enhance Notification System:**
    - Improve the clarity and relevance of alerts.
    - Add customizable notification settings.
2. **Improve Temperature Reading Accuracy:**
    - Calibrate sensors more frequently.
    - Investigate and address factors causing inaccuracies.
3. **Add Audio Alerts:**
    - Implement an audio alert system to notify users of critical conditions.

8.5 Review of the Developed IoT Application

8.5.1 Real-Time Monitoring

The IoT temperature monitoring system offers real-time monitoring, providing continuous temperature data from various factory locations. This feature ensures that temperature-sensitive environments are consistently monitored, allowing for immediate detection of any deviations from the desired range. By having access to real-time data, factory managers can make prompt adjustments to maintain optimal conditions, thereby preventing potential damage to equipment and products. This continuous monitoring also helps in identifying patterns and anomalies that might indicate underlying issues, enabling proactive maintenance and reducing downtime.

8.5.2 Historical Data Analysis

Historical data analysis is a crucial component of the IoT temperature monitoring system, allowing for the examination of temperature trends over time. By storing and analysing historical temperature data, the system helps in understanding long-term patterns and seasonal variations. This information is invaluable for making informed decisions about process improvements, energy efficiency, and predictive maintenance. For instance, if a particular area consistently shows higher temperatures during certain months, steps can be taken to address the issue before it affects production. Additionally, historical data can be used to validate compliance with industry standards and regulations, ensuring that the factory operates within the required parameters.

8.5.3 Alert Notifications

The alert notification feature of the IoT temperature monitoring system is designed to enhance safety and prevent potential damage to goods. When temperatures exceed predefined thresholds, the system sends immediate alerts to relevant personnel via email, SMS, or other communication channels. This prompt notification allows for quick intervention, minimizing the risk of spoilage, equipment failure, or hazardous conditions. By setting customized alert levels, factories can tailor the system to their specific needs, ensuring that critical areas receive the most attention. This proactive approach not only safeguards products and equipment but also contributes to a safer working environment for employees.

8.5.4 Iterative Development

Iteration 1:

- Improved sensor calibration and accuracy.

In this iteration, the focus was on enhancing the precision of the sensors used in the system. Improved calibration techniques were implemented to ensure that the sensors provide more accurate and reliable data. This involved fine-tuning the sensors to reduce errors and inconsistencies, which is crucial for maintaining the integrity of the data collected. Accurate sensor data is essential for making informed decisions and optimizing system performance.

- Enhanced data visualization dashboards.

The data visualization dashboards were upgraded to provide a more intuitive and user-friendly interface. This enhancement allows users to easily interpret and analyze the data collected by the sensors. Advanced graphical representations, such as charts and graphs, were incorporated to present data in a more comprehensible manner. These improvements help users to quickly identify trends, anomalies, and key metrics, facilitating better decision-making and operational efficiency.

Iteration 2:

- Implemented advanced notification settings via Twilio API.

This iteration introduced advanced notification settings using the Twilio API, enabling more sophisticated and customizable alert systems. Users can now set specific conditions and thresholds for notifications, ensuring they receive timely alerts for critical events. The integration with Twilio allows for various communication channels, such as SMS, email, and voice calls, providing flexibility and ensuring that important notifications are not missed.

- Added audio alert functionality.

To complement the advanced notification settings, audio alert functionality was added. This feature provides immediate auditory feedback for critical events, enhancing the system's responsiveness. Audio alerts can be particularly useful in environments where visual notifications might be overlooked or in situations requiring immediate attention. This addition ensures that users are promptly informed of important events, improving overall system reliability and safety.

Iteration 3:

- Integrated external weather data for better decision-making.

In this iteration, external weather data was integrated into the system to provide additional context for decision-making. By incorporating real-time weather information, the system can make more informed adjustments and predictions. This integration helps in optimizing operations, such as adjusting temperature controls based on weather conditions, leading to more efficient and effective system performance.

- Refined system algorithms for more efficient temperature control.

The system algorithms were refined to enhance temperature control efficiency. These improvements involved optimizing the algorithms to better respond to changes in environmental conditions and user preferences. The refined algorithms ensure that the system maintains optimal temperature levels with minimal energy consumption, contributing to cost savings and environmental sustainability. This iteration focused on achieving a balance between comfort and efficiency, providing users with a more reliable and eco-friendly solution.

The Smart Indoor Temperature Monitoring System has been improved over time to better suit customer requirements and improve overall efficiency by including user feedback into successive versions.

9 Activity 04: Evaluate Your IoT Application and Problems Encountered During Integration into the Wider IoT Ecosystem

9.1 Review of the IoT Application and Problems it Solves

9.1.1 Overview

GM Garments Ltd. has implemented an innovative technology, the Smart Indoor Temperature Monitoring System, to ensure ideal environmental conditions within its 3500 square feet garment factory. The system makes use of Internet of Things technology to continually monitor temperature levels and make sure they stay within the optimal range for the operations of washing, dyeing, and drying. A network of sensors that are thoughtfully positioned around the building allows for real-time monitoring by sending data to a central control unit. In order to maintain the appropriate temperature, the system may automatically modify its heating and cooling systems. This improves operating efficiency, lowers energy consumption, and guarantees high-quality output. It also offers data and warnings for preventive maintenance, which reduces downtime and lengthens the life of equipment.

9.1.2 Functionalities:

- Real-time temperature monitoring
- Automated temperature control
- Alerts and notifications
- Data visualization dashboards
- Integration with external weather data

Our system sounds incredibly advanced and efficient! Real-time temperature monitoring and automated control ensure that the environment remains optimal without needing constant human oversight. The integration of notifications and alerts is a great way to keep users informed and ready to act on any issues. Data visualization through dashboards helps in making informed decisions by clearly presenting temperature trends. Additionally, incorporating external meteorological data enhances the system's accuracy and predictive capabilities. Overall, this combination of features provides a robust and reliable temperature control solution.

9.1.3 Features:

I. **Data Visualization:** Provides graphical representations of temperature trends and historical data analysis.

II. **Alerts:** Sends notifications and alerts via SMS or email for temperature anomalies.

The temperature control system has cool data visualization tools that let you see temperature trends and past data in charts and graphs. This makes it way easier to get what the temperature data means and helps you spot patterns and weird stuff. By showing data in a way that makes sense, you can make smart choices about managing temperature.

Also, the system has an alert thing that sends you texts or emails when it spots temperature problems. This heads-up approach makes sure you know about any issues right away, so you can act fast to stop potential troubles. The alerts are made to be easy to understand and tell you what to do and give you the information you need to handle things well.

A useful tool that lets people explore historical data to find trends, patterns, and insights is historical data analysis. Users may learn what has and has not worked successfully by looking at previous data, which enables them to make better judgments going forward. To optimize plans and enhance results, this study may be used in a variety of sectors, including operations, marketing, and finance. Operations can identify inefficiencies and potential improvement areas by examining historical performance data. Taken together, this capability gives users the ability to use the abundance of historical data to influence decisions and help them accomplish their objectives more successfully.

9.1.4 How the problem Addressed.

Overheating in a factory can seriously reduce the performance of machinery, increasing the likelihood of malfunctions and raising maintenance expenses. The danger of overheating rises and component wear can occur more quickly in high-temperature operating situations.

Production is not only hampered by this, but it also raises operating expenses by requiring more frequent repairs and replacements. The system's resolution of overheating guarantees that machines function within their ideal temperature range, thus augmenting their durability and dependability.

The quality of the product is directly impacted by the factory's ability to maintain ideal temperature conditions. Machines are able to operate more precisely and consistently when they are not stressed by the risk of overheating. As a result, there are fewer flaws and better goods. Furthermore, a temperature-controlled environment may enhance the working environment for staff members, making the workplace safer and more comfortable. All things considered, the production process becomes more effective, economical, and high-quality as a result of the system's capacity to control temperature.

9.2 Potential Problems During Integration into the Wider System

There are several possible obstacles that might occur while integrating the Smart Indoor Temperature Monitoring System into a larger IoT ecosystem. The system's compatibility with current infrastructure and other Internet of things (IoT) devices is a major challenge. This may be resolved by utilizing defined protocols to guarantee seamless integration and doing extensive compatibility testing. The capacity of various IoT systems and devices to interact and function as a unit is known as interoperability, and it presents another difficulty. Common communication protocols like MQTT and HTTP may be used, as well as compatible frameworks, to make this easier.

Given the rise in cyberthreats, security is another crucial concern. Ensuring data privacy and shielding the system from such attacks are critical tasks. The system and its data may be protected by putting strong security mechanisms in place, such as encryption, authentication, and frequent security audits. Another difficulty is scalability, especially as the system gets bigger and has to manage more users and data. A scalable design that makes use of edge computing and cloud services may guarantee that the system can effectively handle massive data volumes and continue to function as it grows.

9.3 Comparison of the Final Application with the Original Plan

Comparison:

1. **Deviations:**
    - The initial plan included only basic temperature monitoring.
    - Added functionalities based on user feedback, such as advanced notification settings and audio alerts.

2. **Modifications:**
    - Enhanced sensor accuracy and calibration.
    - Improved data visualization dashboards.
    - Integrated external weather data for better decision-making.

3. **Enhancements:**
    - Added audio alerts.
    - Improved the notification system for better clarity and relevance.
    - Refined algorithms for more efficient temperature control.

The final version of the Smart Indoor Temperature Monitoring System looks different from what we first planned because of what users told us and the changes we made along the way. At first, we wanted to check the temperature. But after hearing from users, we added more stuff like better ways to send alerts and sound warnings. As we built it, we made some changes. We made the sensors more accurate and tweaked them better. We also made the screens that show the data look nicer and easier to understand. On top of that, we started using weather info from outside to help make smarter choices. This wasn't just a small change - it had a big impact on how well the system works.

The team made quite a few changes to make the system work better and meet what users wanted. They put in sound alerts to let people know right away when something important happens. They also made the way notifications work clearer and more useful. On top of that, they tweaked the math behind how temperatures are controlled to make it work more . All these updates show they're trying to make things better for users and keep improving stuff.

9.4 Critical Evaluation of the Overall Success of the Application

Critical Evaluation:

1. **Success Factors:**

   - **Usability:** The system's user-friendly design ensures that most users find it easy to navigate and operate. This ease of use is crucial. It reduces the learning curve. It minimizes the need for extensive training. A user-friendly interface often leads to higher user satisfaction and increased adoption rates. Users can quickly understand and utilize the system's features without frustration.

   - **Effectiveness:** The system excels in maintaining optimal temperature conditions. This is vital for smooth operation of machinery. By effectively controlling temperatures. It prevents overheating and other temperature-related issues. Such issues could lead to machine downtime or damage. This in turn, significantly improves the overall efficiency and lifespan of the machines. It ensures they operate at peak performance.

   - **Efficiency:** The system's ability to efficiently use resources is a standout feature. It processes data in real time. This coupled with automated control, ensures resources are used optimally reducing waste. This improves operational costs. The automation aspect means the system can make quick adjustments. It relies on real time data. This enhances responsiveness and reliability. This efficiency not only saves time and money. It also contributes to more sustainable operations.

9.4.1 Impact:

- On People: Stable indoor environment
- On Business: Reduced maintenance costs, improved machine efficiency, better product quality
- On Society: Energy efficiency and sustainability through temperature control

9.4.2 Future Improvements:

  - Interoperability with other IoT devices
  - More security
  - Add humidity and air quality
  - AI driven predictive maintenance

Interoperability is key to all systems and devices working together seamlessly. Interoperability means different technologies can talk to each other and work together efficiently, reducing compatibility issues and simplifying things. This means more joined up and effective workflows and ultimately more productivity and user delight.

Security is key to protecting sensitive data and systems from cyber threats. As technology moves forward, so do the bad guys. Increasing security measures such as advanced encryption, multi factor authentication and regular security audits will help keep information safe and trust in technology.

Including environmental factors like humidity and air quality into systems can make them work better and more user friendly. Monitoring these factors can help optimize conditions for equipment and human comfort and better performance and health outcomes. This is especially important in industries like healthcare, manufacturing and smart home.

Looking into AI driven predictive maintenance can change how we manage and maintain equipment. By analyzing data and predicting failures before they happen AI can prevent downtime, reduce maintenance costs and extend the life of the machinery. This proactive approach means systems stay operational and efficient and minimize disruption and maximize reliability.

9.4.3 Factors to Consider:

1. Usability:
    - Keep the interface simple and user friendly
    - Provide extensive training and support for users
2. Effectiveness:
    - Regularly update and improve the algorithms
    - Get continuous feedback from users to know what to improve
3. Efficiency:
    - Handle large data volumes efficiently
    - Use edge computing to reduce latency and real time processing
4. Cost-Effectiveness:
    - Balance setup costs with long term maintenance and energy savings
    - Provide value for money through efficiency and cost savings
5. Societal Implications:
    - Data privacy and user consent
    - Compliance to regulations and standards

A few aspects should be still taken in account to keep the system in continued success and improvement:

Usability: it is important to keep ease of use characteristic by making the interface more intuitive and providing training and support. Effectiveness: keep updating the system algorithms, collecting user feedbacks will provide directions for further improvement. Efficiency: better managing the big-data volume and using the edge computing to minimize latency and reduce response time.

Cost-effectiveness is also important, i.e. the balance of initial high deployment cost and long-term low cost for operation & maintenance and energy consumption. Value for money on improvements and savings should be provided by the system, as well as societal implications (i.e. privacy, ethics, GDPR issues) and compliance with existing/applicable regulations/standards for a system to be trustworthy. Smart Indoor Temperature Monitoring

System will further evolve to better serve users and continue to offer more benefits for businesses and society.

10 Conclusion

The problem of excessive heat impeding machine operation efficiency in a 3500 square foot facility is resolved for ABC Garments Ltd. by the Smart Indoor Temperature Monitoring System. It's an Internet of Things (IoT) system that offers real-time temperature management, monitoring, and data analysis for increased output, reduced maintenance costs, and optimum energy use. Technical challenges included protocol compatibility, interoperability, security, and scalability issues, as well as integration issues. But with the aid of a clear protocol stack, security measures, and a scalable system architectural design, problems were overcome. User ideas were taken into account during the incremental development phase, which resulted in the creation of an audio alert preferences module, the resolution of notification parsing difficulties, and algorithmic fine tuning. Our Smart Indoor Temperature Monitoring System was demonstrated to be user-friendly, efficient, and effective by the user-centric overview. Improving living conditions for employees residing in hot climates and promoting optimal energy use are two immediate socio-technical advantages.

11 Reference list

Adil95More (n.d.). *IoT Based Temperature Monitoring System*. [online] Instructables. Available at: https://www.instructables.com/IoT-Based-Temperature-Monitoring-System/ [Accessed 16 Jul. 2024].

Anon, (2023). *Temperature Sensor Applications in IoT | DeepSea*. [online] Available at: https://www.deepseadev.com/en/blog/temperature-sensor-applications/#:~:text=Thermistors%20are%20popular%20IoT%20temperature [Accessed 16 Jul. 2024].

Apifornia. (2023). *Node-RED for App Development: Looking at the Pros and Cons*. [online] Available at: https://blog.apifornia.com/node-red/ [Accessed 16 Jul. 2024].

Biz4Intellia. (n.d.). *5 Use Cases and 5 Benefits of IoT Based Temperature Monitoring*. [online] Available at: https://www.biz4intellia.com/blog/temperature-monitoring/ [Accessed 16 Jul. 2024].

Coursesidekick.com. (2024). Available at: https://www.coursesidekick.com/information-systems/1740102 [Accessed 14 Jul. 2024].

Desklib.com. (2022). Available at: https://desklib.com/document/assignment-1-front-sheet-hj20/ [Accessed 14 Jul. 2024].

flowfuse.com. (n.d.). *The benefits and drawbacks of using Node-RED function nodes • FlowFuse*. [online] Available at: https://flowfuse.com/blog/2023/03/why-should-you-use-node-red-function-nodes/ [Accessed 16 Jul. 2024].

Gada, U., Joshi, B., Kadam, S., Jain, N., Kodeboyina, S. and Menon, R. (2021). *IOT Based Temperature Monitoring System*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICNTE51185.2021.9487691.

ip_admin (2023). *Unpacking IoT Architecture: Layers and Components Explained*. [online] Device Authority. Available at: https://deviceauthority.com/unpacking-iot-architecture-layers-and-components-explained/#:~:text=IoT%20architecture%20is%20an%20organised [Accessed 6 Jul. 2024].

Katalon (2022). *10 Popular API Examples: a Full List*. [online] katalon.com. Available at: https://katalon.com/resources-center/blog/api-examples [Accessed 6 Jul. 2024].

MQTT (2022). *MQTT - the Standard for IoT Messaging*. [online] mqtt.org. Available at: https://mqtt.org/ [Accessed 14 Jul. 2024].

Sultanow, E. and Chircu, A. (2022). A Review of IoT Technologies, Standards, Tools, Frameworks and Platforms. *The Internet of Things in the Industrial Sector*, pp.3–34. doi:https://doi.org/10.1007/978-3-030-24892-5_1.

www.realpars.com. (n.d.). *What Is Node-RED? | How Can I Use It to Create IoT Applications? - RealPars*. [online] Available at: https://www.realpars.com/blog/node-red [Accessed 15 Jul. 2024].

www.trustradius.com. (2023). *Pros and Cons of node-RED 2024*. [online] Available at: https://www.trustradius.com/products/node-red/reviews?qs=pros-and-cons#reviews [Accessed 16 Jul. 2024].