Faculty of Electrical Technology and Engineering

DEVELOPMENT OF AN IOT-BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER

MUHAMMAD FAIZZAT BIN AZMAWI

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2023

DEVELOPMENT OF AN IOT-BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER

MUHAMMAD FAIZZAT BIN AZMAWI

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics)
with Honours

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek: Development Of An IoT-Based Smart Inventory Management System Using A Microcontroller

Sesi Pengajian : 2023/2024

Saya Muhammad Faizzat Bin Azmawi mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	**TIDAK TERHAD**	

(TANDATANGAN PENULIS)

Alamat Tetap: Lot 11683 A,
Kampung Delek Baru
41250 Klang,
Selangor

Tarikh: 12 / 1 / 2024

Disahkan oleh:

Ts. MASLAN BIN ZAINON
Senior Lecturer
Department of Electrical Engineering Technology
Faculty of Electrical Technology and Engineering
Universiti Teknikal Malaysia Melaka

Tarikh: 12 / 1 / 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “DEVELOPMENT OF AN IOT-BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

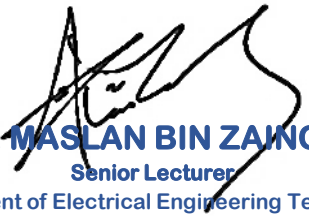
Signature : 

Student Name : MUHAMMAD FAIZZAT BIN AZMAWI

Date : 12 / 1 / 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature : 

Ts. MASLAN BIN ZAINON
Senior Lecturer
Department of Electrical Engineering Technology
Faculty of Electrical Technology and Engineering
Universiti Teknikal Malaysia Melaka

Supervisor Name : TS. MASLAN BIN ZAINON

Date : 12 / 1 / 2024

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

In completing my senior year project, I want to express my deepest gratitude to my father, Azmawi Bin Abdullah, and mother, Maini Binti Arifin. Your unwavering support, love, and understanding have been my pillars of strength. Your sacrifices and encouragement have fueled my determination to reach this milestone, and I am profoundly grateful for the opportunities you've provided. To all my fellow friends, your constant support, encouragement, and shared moments have been invaluable throughout this journey. Your presence has made the challenges more manageable and the successes more meaningful. I am truly thankful for your friendship. Although it's been a challenging journey, I want to express my heartfelt thanks to everyone who has supported me during these moments. Your encouragement, guidance, and belief in my abilities have been instrumental in overcoming obstacles and achieving success. This achievement is as much yours as it is mine. I dedicate this accomplishment to my family, friends, and all who have been part of this challenging yet rewarding journey.

ABSTRACT

Establishing an effective and efficient inventory system is paramount for industries like supermarkets and mini-markets, aiming to proficiently manage their stock and mitigate challenges such as unnoticed expired products. Using NodeMCU ESP8266 as a microcontroller and integrating the Internet of Things (IoT) represents a significant stride towards creating a more sophisticated inventory management system. In this project, the ESP8266 serves as a Wi-Fi module, facilitating the device's connection to a database that continuously monitors the system. An ultrasonic sensor is employed to gauge the quantity of items remaining on the shelves, while an IR sensor is utilized to track the precise location of products. The IoT platform, implemented through a MySQL database with PHP programming, requires users to input expiration date data for effective system monitoring. The primary objective of this project is to attain high accuracy in stock monitoring and enable real-time tracking. The system is designed to send notifications to users when the quantity of items approaches the minimum threshold or when expiration dates are nearing. This proactive approach empowers users to promptly remove nearly expired items or replenish stocks with new products. In conclusion, this project exhibits promising potential and could significantly optimize stock management, curbing waste in the retail industry.

ABSTRAK

Menetapkan sistem inventori yang berkesan dan cekap adalah sangat penting untuk industri seperti pasar raya dan pasar mini, yang berusaha untuk mengurus stok mereka dengan mahir dan mengatasi cabaran seperti produk luput yang tidak diperhatikan. Penggunaan NodeMCU ESP8266 sebagai mikropengawal, bersama-sama dengan penyatupaduan Internet of Things (IoT), menandakan langkah penting ke arah pembangunan sistem pengurusan inventori yang lebih canggih. Dalam projek ini, ESP8266 berperanan sebagai modul Wi-Fi, memudahkan sambungan peranti ke pangkalan data yang secara berterusan memantau sistem. Menggunakan sensor ultrasonik membolehkan pengukuran yang tepat terhadap kuantiti barang yang masih tinggal di rak, sementara sensor IR digunakan untuk mengesan lokasi tepat produk. Platform IoT, dilaksanakan melalui pangkalan data MySQL dengan pengaturcaraan PHP, memerlukan pengguna untuk memasukkan data tarikh luput bagi pemantauan sistem yang berkesan. Objektif utama projek ini adalah untuk mencapai ketepatan yang tinggi dalam pemantauan stok dan membolehkan penjejakan waktu nyata. Sistem ini direka untuk menghantar pemberitahuan kepada pengguna apabila kuantiti barang mendekati ambang minimum atau apabila tarikh luput semakin hampir. Pendekatan proaktif ini memberdayakan pengguna untuk segera mengeluarkan barang yang hampir luput atau mengisi semula stok dengan produk baru. Sebagai kesimpulannya, projek ini menunjukkan potensi yang menjanjikan dan memiliki keupayaan untuk mengoptimalkan pengurusan stok secara signifikan, secara berkesan mengurangkan pembaziran dalam industri runcit.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Maslan Bin Zainon for his precious guidance, words of wisdom and patience throughout this project.

I am also indebted to University Technical Malaysia Melaka (UTeM) and my parents for the financial support that enabled me to accomplish the project. Not forgetting my fellow friend's willingness to share their thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayers during my study. An honourable mention also goes to my close friend for all the motivation and understanding. And to close friend, thanks for your continuous support.

Finally, I would like to thank all the University Technical Malaysia Melaka (UTeM) staff, colleagues and classmates, faculty members, and other individuals who are not listed here for being cooperative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

V	-	Voltage
IoT	-	Internet Of Things
IFTT	-	If This Then That
WiFi	-	Wireless Fidelity
RFID	-	Radio Frequency Identification
OLED	-	Organic Light-Emitting Diodes
SQL	-	Structured Query Language
GPIOs	-	General Purpose Input/Output
EOQ	-	Economic Order Quantity
ROL	-	Reorder Level
IDE	-	Integrated Development Environment
PHP	-	Hypertext Preprocessor
PWM	-	Pulse Width Modulation
USB	-	Universal Serial Bus
TCP/IP	-	Transmission Control Protocol/Internet Protocol
RAM	-	Random Access Memory
D	-	Time
C	-	Speed
US	-	Ultrasonic Sensor
IR	-	Infrared Sensor

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Effective retail management relies heavily on inventory control. Proper pre-planning and decision-making result from keeping track of your goods. Nowadays, our local supermarkets and major companies store their products on high shelves or in enormous storage spaces. The proprietors of these establishments must physically count the packets to determine the current stock. Inaccuracy can stem from either human or system errors. Incorrect physical inventory count is often the leading cause of data inaccuracies[1]. This approach is time-consuming and creates mistakes that were not intended. It becomes challenging to physically verify the inventory regularly in extensive storerooms since the situation gets rather complex.

1.2 Addressing Societal and Global Issues Through Smart Inventory Management System Project

Inventory management is critical to business operations, impacting societal and global levels. Inefficient practices have significant societal implications, leading to stockouts, delays in customer orders, and decreased satisfaction. Excess inventory resulting from poor management contributes to wastage and environmental degradation and hinders economic growth. On a global scale, ineffective inventory management disrupts supply chains, creates imbalances, and affects economic stability. It increases transportation and storage costs, creates bottlenecks, and hinders resource utilization efficiency. To address

these challenges, efficient inventory management practices are crucial. Real-time tracking, automation, and technology integration improve accuracy, responsiveness, and decision-making. They minimize stockouts, reduce wastage, and optimize operational efficiency, benefiting businesses and communities. Furthermore, effective inventory management practices contribute to a sustainable global supply chain by reducing excess inventory, minimizing environmental impact, and promoting economic stability.

To address these issues effectively, an IoT-based smart inventory management system emerges as a valuable solution. By leveraging IoT technology, businesses can automate and streamline inventory management. Real-time tracking and monitoring of inventory quantities, expiration dates, and locations become possible, allowing for proactive decision-making and optimization of stock movement.

This IoT-based system facilitates efficient inventory management by reducing accuracy, minimizing errors, and improving overall operational efficiency. By providing real-time data and insights, businesses can make informed decisions regarding inventory replenishment, stock movement, and product lifecycle management. Ultimately, implementing an IoT-based smart inventory management system enhances inventory accuracy, reduces wastage, and improves customer satisfaction in the global marketplace.

Although we have the law to follow, further action will be taken if the compounds are not addressed within the time limit provided. Violators might face a fine of no more than RM10,000, up to two years in prison, or both if found guilty. In addition, smart inventory system management can check the shelves with minimum supervision. The shelves will be at maximum stock with the ultrasonic sensor, and consumers can prevent overstocking. Besides that, the Internet of Things can monitor the shelves with prior time. When the shelves are half empty or nearly expired, it can notify the consumer to take action immediately. That way, retail staff can monitor the expiration date and reduce some work.

1.3 Problem Statement

For the past year, manually recorded stockings have been estimated in every marketplace. It is because checking the shelves for expired daily or twice a week is no longer effective. Therefore, it will become inefficient, and the product will be obsolete due to human error. The Selangor Health Department [2] found 24 found 406 damaged and expired foods still on the shelves in 36 premises, including supermarkets, mini-markets, and retail stores all over the state.

Although the law provides clear guidelines, further action will be taken if the compounds are not addressed within the time limit provided. Violators might face a fine of no more than RM10,000, up to two years in prison, or both if found guilty. In addition, smart inventory system management can check the shelves with minimum supervision. The shelves will be at maximum stock with the ultrasonic sensor, and consumers can prevent overstocking. Besides that, the Internet of Things can monitor the shelves with prior time. When the shelves are half empty or nearly expired, it can notify the consumer to take action immediately. That way, retail staff can monitor the expiration date and reduce some work. It can benefit both parties.

1.4 Project Objective

The main aim of this project is to propose a systematic and practical methodology accuracy. Specifically, the objectives are as follows:

a) To develop a smart inventory management system for the retail industry using a microcontroller.

b) To develop an IoT-based system for real-time monitoring and notification of goods inventory status using the database.

c) To analyze the system's performance regarding inventory status and goods expiry date accuracies.

1.5 Scope of Project

The scope of this project is as follows:

a) The project consists of microcontroller programming and IoT platform configuration.

b) The system will function in the retail market with WiFi coverage only.

c) The system must manually key the product expiration date before monitoring the inventory in real-time.

d) An Infrared sensor to detect the item's location on the shelf.

e) Applicable for box items only.

f) An Ultrasonic sensor with a formula distance detects the quantity of the product.

g) The project will be focused on checking the inventory quantity and expiration date control.

1.6 Report Structure and Organization

This study is organized into four chapters, as follows:

The background establishes the foundation by underscoring the pivotal significance of inventory control in inefficient retail management. The study's objectives are to delve into the complexities of proprietors in maintaining accurate inventory data, paving the way for potential technological solutions to enhance efficiency in retail management.

In Chapter 2, a comprehensive literature review is conducted to assimilate knowledge and gain insights from previous research, aiming to address unresolved issues and avoid redundancy in the project. Comparing past research endeavours aims to scrutinize the advantages and drawbacks of diverse methodologies.

In Chapter 3, the methodology is structured around four key milestones aligned with the project's objectives. The system architecture development draws on insights from previous research studies involving selecting appropriate methods and components for integration into the project. Testing and troubleshooting are evaluated in alignment with the algorithmic development within this system.

In Chapter 4, the outcomes of the system component testing and data analysis are showcased. The smart inventory management system's integration is underscored, focusing on the database, user interfaces, and ESP8266 connectivity. The chapter delves into the exploration of real-time monitoring, email notifications, and the presentation of the final project design. Summarizing respondent insights obtained from surveys and performance tests evaluating sensor accuracy, the chapter culminates by presenting key findings and implications for the project.

1.7 Summary

This project introduces a smart inventory management system for the retail industry, utilizing a microcontroller and IoT technology. The current manual inventory tracking poses challenges, leading to inaccuracies and inefficiencies. The proposed system addresses these issues by providing real-time monitoring, automation, and technology integration. It seeks to improve efficiency, minimize errors, ensure compliance with regulations, and contribute to environmental sustainability. The project's objectives include developing a systematic methodology, creating an IoT-based system, and analyzing system performance. With a focus on accuracy, the scope covers microcontroller programming, IoT platform configuration, and application in retail markets with WiFi coverage. The project holds significance for its potential to revolutionize inventory practices and enhance overall operational efficiency.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explores existing research on the selected topic and aims to comprehensively understand the areas that have undergone examination and those that require further investigation. The historical trajectory of the topic is explored, outlining its evolution over time. The examination extends to encompass various theories and concepts posited by researchers while concurrently assessing the strengths and weaknesses of preceding studies. Furthermore, ongoing debates or disagreements within the literature are addressed. Culminating in synthesising the main findings, this chapter plays a pivotal role in summarizing the collective knowledge and pinpointing the gaps in understanding the research endeavours to fill. The significance of this chapter lies in establishing the foundational framework for the study and underscoring its academic relevance in the field.

2.2 Understanding Societal and Global Issues Through Smart Inventory Management System Project in the Literature

Through an extensive review of the literature on inventory management, this section examines the understanding of societal and global issues associated with inefficient inventory practices. Research indicates that inadequate inventory management leads to societal challenges such as stockouts, delays in customer orders, and decreased customer satisfaction. These issues significantly affect businesses and local communities, hindering economic growth and limiting access to essential products and services. Moreover, excess

inventory resulting from poor inventory management contributes to wastage, environmental degradation, and resource depletion, impacting the environment and society.

By leveraging IoT sensors and data analytics, businesses can gain valuable insights into inventory levels, demand patterns, and product expiration dates. This enables accurate demand forecasting, optimized production planning, and streamlined distribution processes, ultimately reducing excess inventory and waste generation. The system's real-time monitoring capabilities also facilitate better coordination and collaboration across supply chain partners, reducing imbalances and disruptions in the global flow of goods. An IoT-based smart inventory management system holds great potential to play a critical role in addressing the global challenge of future inventory management.

2.3 Development of an IoT-based Smart Inventory Management System

Inventory is now more precious than ever, yet it must be in the correct location at the appropriate time to avoid harming the company's daily operations. As an effective method of controlling the movement of items into and out of an existing inventory, inventory management systems will have beneficial impacts, particularly on issues related to the industrial sector. Tracking materials, sometimes called work in progress or products, determines and modifies the ordering amount before inventory becomes low and becomes risky to the company. Many organizations rely on manual inventory systems, but the company have drawbacks such as time consumption, communication, difficulty counting, difficulty keeping track of daily inventory movements, and inconsistency in ordering materials. These methods are frequently restricted to essential tools like the computation of economic order quantities and rough approximations of reorder points or base stocks for achieving target service levels.

Nevertheless, almost all large companies and many small and medium-sized enterprises increasingly try applying scientific methods to manage their inventory better. IoT's built-in connection and real-time analytics might solve merchants' inventory problems. Stocking the appropriate quantity of goods is essential. If companies place fewer orders, customers will look for better choices. There is a possibility of overordering by businesses. To prevent sustaining a significant loss, the company will eventually be forced to sell at clearance prices due to the issue of overstocking and near expiration. Our study article aims to find the ideal inventory level to uphold high service standards.

2.3.1 RFID Tag and Reader

Developing a Smart Inventory Management System benefits companies such as local supermarkets and large factories. Radiofrequency identification is used in one of these systems (RFID). RFID is widely used to identify and track several inventory items. Attaching an RFID tag to every item in the inventory makes the tagged object part of the network. The inventory administrator uses handheld RFID readers to scan the objects, and handheld readers nowadays can be wireless and smart with processing capabilities. There are different types of tags, but passive tags that reflect the received signal from the reader are the most used due to their low cost and power efficiency[3].

The network of inventory objects in Figure 2.1 shows that they are connected to the middleware using various computer cables or wireless using Wi-Fi communication protocol IEEE 802.11. The reading distance between the reader and the tags cannot exceed a few centimetres, and both ends must be held appropriately to achieve the required polarization angle. The cloud serves as the middleware provider for the management system.

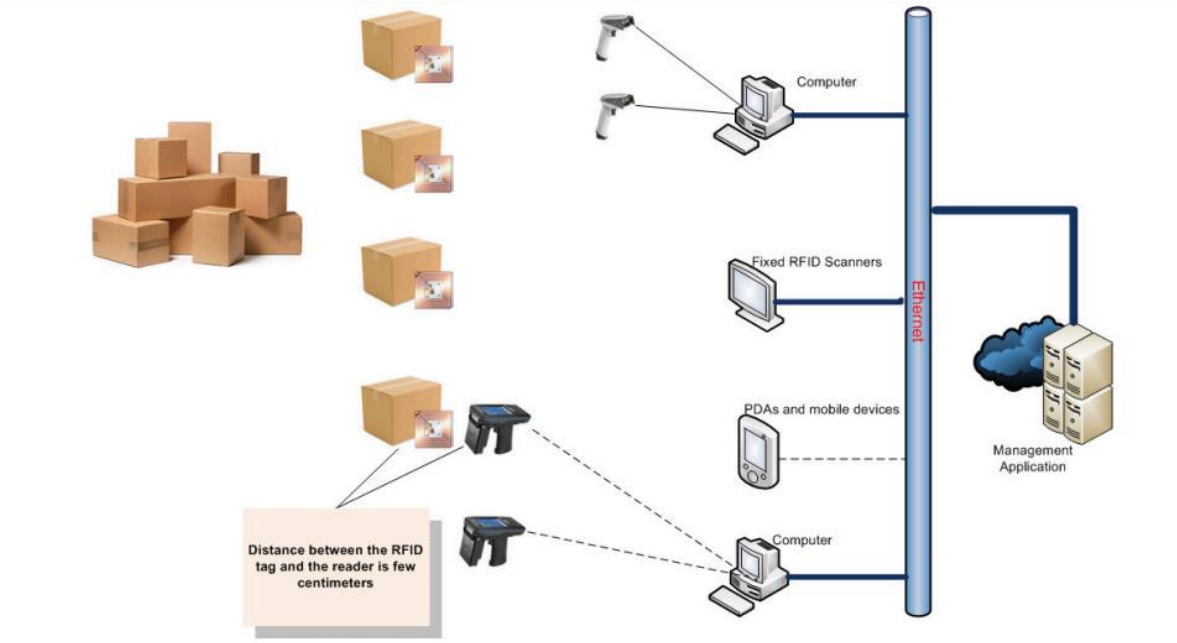


Figure 2.1 Inventory System Architecture

2.3.2 Load Cell

This study explained how the inventory can be measured using weight in simple calculations. The key component of this project is the load cell. The load cell HX711 load cell amplifier module is operated to calculate the weight of items in the inventory. To know the number of packets available in the inventory, we must know the total and individual weights. We can measure the total weight and do all the necessary calculations with the help of a microcontroller. A strain gauge is a component that is used to measure strain. With the help of the Wheatstone Bridge, the bridge is balanced. A NodeMCU microcontroller is the processing unit, and the IoT technology is connected via ESP8266 Wi-Fi SoC and 128×64 pixels OLED display. Figure 2.2 below shows the front view of a design implementation of the Automatic Inventory Management System.[4]

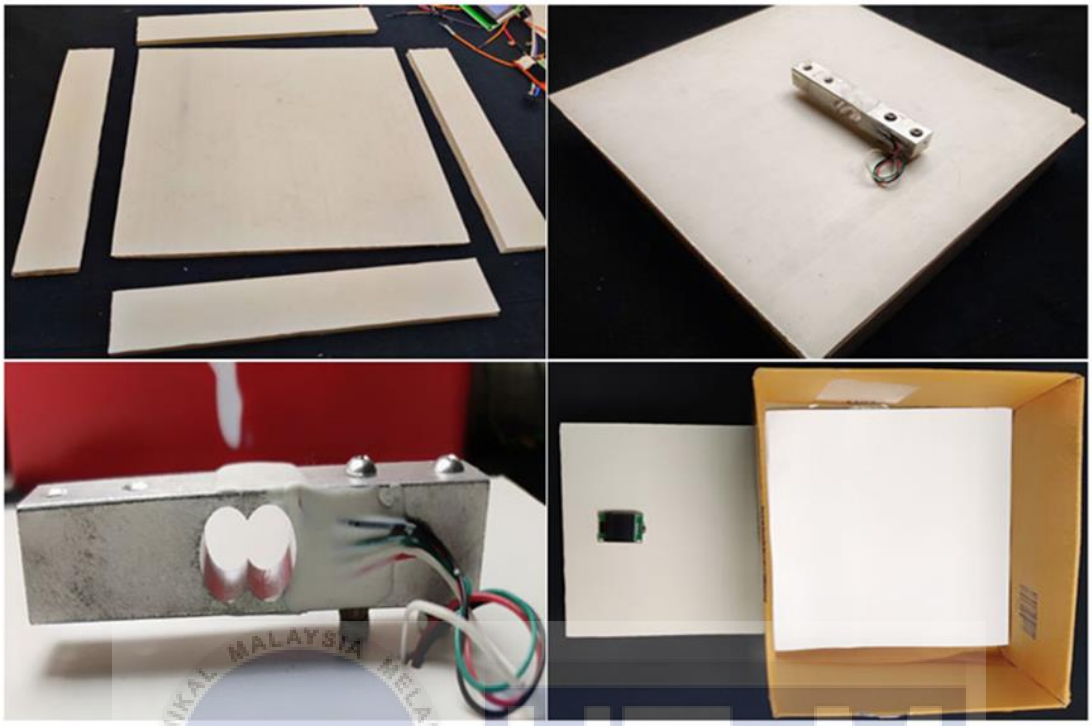


Figure 2.2 Front view design implementation Automatic Inventory Management System

2.3.3 Ultrasonic Sensor

In this study, the researcher used an ultrasonic sensor to detect the distance of items or goods in the form of a box. In this system, the Ultrasonic sensor counts the number of products stored in a rack, ThingSpeak is used for live data monitoring, and IFTTT sends mail whenever the product count exceeds the user-defined criteria. Ultrasonic sensors measure distance by using ultrasonic waves. The sensor position is fixed so that the distance between the sensor and the box number will be permanently fixed.

The inventory management systems using ultrasonic sensors highlight the benefits of using this technology to automate inventory management processes. Ultrasonic sensors can monitor inventory levels in real-time, allowing companies to optimize their inventory levels and reduce waste. Several studies have proposed inventory management systems that

use ultrasonic sensors to monitor inventory levels and alert companies when inventory levels run low. Figure 2.3 shows the design of the implementation inventory system using an Ultrasonic sensor.[5]

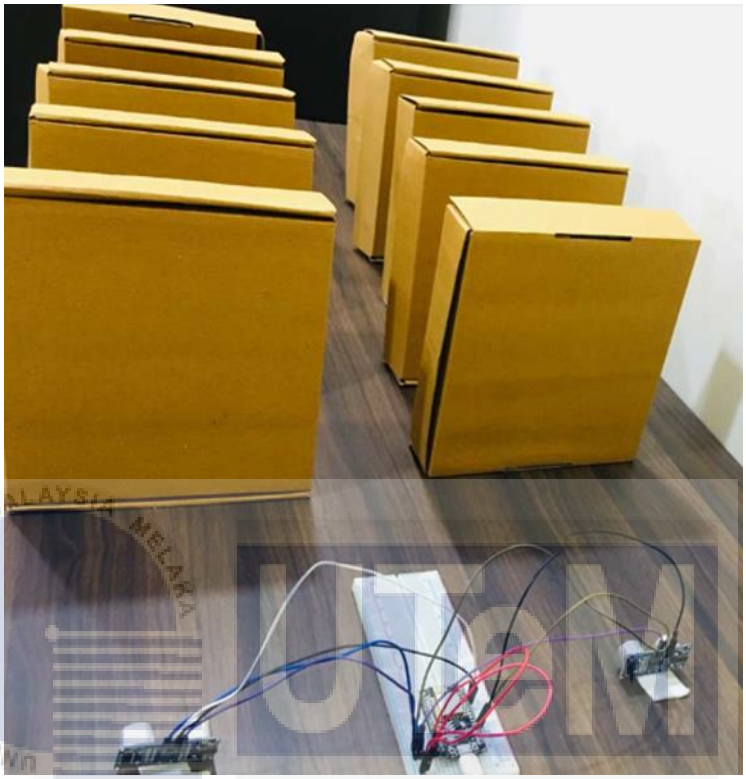


Figure 2.3 The design implementation Inventory System using an Ultrasonic sensor

2.3.4 Barcode Scanner

This research aims to design an Android-based goods inventory system application with a barcode scanner. The application consists of a desktop application developed in the Basic language and an Android application developed in Java. The system uses a MySQL database stored online, allowing mobile access to the inventory system application. Data exchange between the Android application and the database is facilitated through a web service for efficient data distribution. The application's main functionality is to check stock information availability by scanning the goods' barcode using an Android smartphone. This feature improves the service provided to customers in the store by quickly providing information on stock availability. The application utilizes barcode scanning technology to

match the scanned barcode with the corresponding item's stock information stored in the database.

The research discusses the importance of inventory systems in modern businesses, emphasizing the need for efficient and mobile solutions to manage stock. Several related studies highlight previous research on inventory applications using barcode scanners. When the barcode is scanned with a barcode scanner device, the code automatically connects to the product data stored in the database. The scanning result contains information about various products, such as vendor name, product name, price, and other relevant data entered into the database[6]. The methodology involves defining the problem, collecting relevant data, understanding the system processes and required software, designing and developing the application using Android Studio and MySQL, testing an Android smartphone, and evaluating the results.

The system overview is depicted in a diagram illustrating the process of checking stock availability using a barcode scanner on an Android smartphone. The application connects to the database server, retrieves stock information, and displays it on the smartphone screen, allowing users to identify nearby stores with the desired item in stock. This also includes modelling the system using Data Flow Diagrams (DFD) to represent the relationships between entities and processes. The DFD Level 0 diagram illustrates six processes: login validation, store data management, operator data management, employee data management, item data management, and sales data management.

The database design is presented through an entity-relationship diagram, demonstrating the relationships between tables such as User, Store, Item, Sales, Sale Details, and Stock Details. Figure 2.4 shows an example of a barcode that enhances inventory management and improves customer service by providing real-time stock information through barcode scanning.[7]

Figure 2.4 Example of barcode

2.3.5 Temperature and Humidity Sensor (DHT11)

The proposed method is an Artificial Intelligence-based foodstuff monitoring system used for inventory management in various settings such as industrial kitchens, restaurants, canteens, and vegetable stores. The system addresses the risks and challenges when the storekeeper cannot monitor the grocery and orders. The method utilizes various sensors to monitor different aspects of the food items. An ultrasonic sensor is used for level estimation detection. It measures the level of groceries in containers, and if a container is empty, the information is sent to the storekeeper.

This ensures that the availability of specific food items can be determined, and unavailable things can be ordered for purchase. The system also incorporates a DHT11 sensor to monitor the humidity and temperature inside the containers. If the humidity or temperature exceeds certain thresholds, notifications are sent to alert the storekeeper. This helps in preventing spoilage or contamination of food items.

Additionally, the system uses an MQ3 sensor to identify decomposed organic items. The MQ3 sensor detects alcoholic gas produced by organic items, allowing the system to identify spoiled items. The sensor data, including grocery level, humidity, temperature, and decomposition range of organic items, is collected from smart containers. The data transmission between the sensors and the system is facilitated by a microcontroller known as NodeMCU. The microcontroller handles the communication between the sensors and the system, enabling the collection and processing of sensor data. To trigger notifications and

automate the ordering of kitchen items, the system integrates with a mobile application called IFTTT. The IFTTT app allows the system to send mail notifications to the storekeeper based on predefined conditions.

The method leverages sensors, microcontrollers, and AI-based monitoring to provide real-time information about grocery levels, quality, and potential issues such as decomposition or unfavourable environmental conditions. This helps in efficient inventory management and timely ordering of items. Figure 2.5 shows the flowchart of the grocery monitoring system; this flows from measuring the sensor values from the corresponding containers, which will be stored in the cloud. When sensor values are stored, they will be compared with the threshold value to send the mail notification.[8]

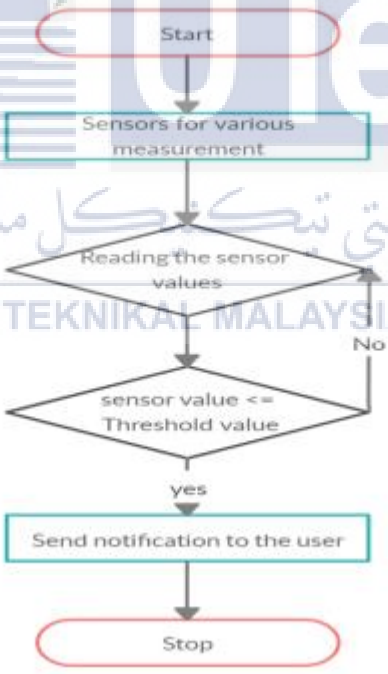


Figure 2.5 Flowchart of the proposed system using the DHT11 sensor

2.4 Studies related to Smart Inventory Management Systems

This section clearly explains the components that researchers often utilize. Such component requirements are referred to in the datasheet to prevent inaccurate information.

2.4.1 ARDUINO ATMEGA-328 MICROCONTROLLER

Arduino is an open-source microcontroller, and there is no feedback present in the microcontroller. This Arduino board consists of an I2C bus that transfers data from the Arduino board to the output devices. Arduino ATMEGA-328 microcontroller consists of 14 input and output analogue and digital pins (from which six pins are PWM pins), six analogue inputs, and the remaining digital inputs. A power jack cable connects the Arduino board to the computer.

Arduino can be programmed easily because the programming language has been simplified and uses open-source systems[9]. Some examples of Arduino microcontrollers are Arduino Duemilanove, Arduino UNO, Arduino Leonardo, Arduino Mega, Arduino MEGA 2560 R3, Arduino MEGA 2560 R3, Arduino Nano, Arduino Due, LilyPad Arduino, micro Arduino. Arduino has been programmed by using C and C++ programming languages. C and C++ are high-level languages. Figure 2.6 shows the example of Arduino Atmega-328.

Figure 2.6 Example of Arduino Atmega-328

Compared to other microcontroller development platforms, Arduino boards are relatively inexpensive, making them an attractive choice for prototyping, experimentation, and educational purposes. The accessibility and affordability of the Atmega328-based Arduino boards enable individuals and educational institutions with limited budgets to explore the world of electronics and programming without financial constraints. This democratization of technology empowers a broader range of users to engage in innovative projects and fosters creativity and innovation.[9]

2.4.2 NodeMCU

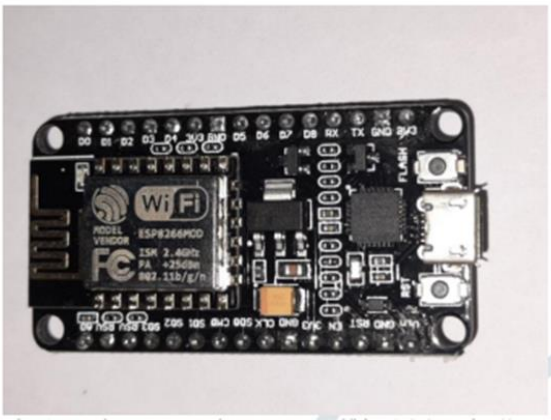


Figure 2.7 NodeMCU ESP8266 board with Wi-Fi solution

NodeMCU is an open-source and LUA programming language-based firmware developed for the ESP8266 Wi-Fi chip. Espruino, Mongoose OS, software development kit provided by Espressif, and ESP8266 add-on for Arduino are some development platforms that may program the ESP8266. ESP8266 may host the application or offload all Wi-Fi networking functions from another application processor through its self-contained Wi-Fi networking solution.

Figure 2.7 shows that the board of NodeMCU ESP8266 with Wi-Fi solution has a powerful onboard processing capability and sufficient storage, allowing it to be integrated with minimal development up-front and loading during runtime through its GPIOs with the sensor's specific devices. ESP8266 has a low cost and features, making it an ideal module for the Internet of Things (IoT). It can be used in any application that requires connecting a device to a local network or internet. Table 2.1 shows the NodeMCU specification table.[10]

Table 2.1 NodeMCU specification table

Model	NodeMCU ESP8266
Bluetooth	No
SRAM	No
MCU	L106 Xtensa Single-core 32-bit
Frequency of use	80Mhz
Flash	No
GPIO Pins	17
Sensor of temperature	No
Sensor of touch	No
Wi-Fi 802.11 b/g/n	HT20

2.4.3 Internet of Thing(IoT)

The Internet of Things is a platform where everyday devices become more innovative, everyday processing becomes intelligent, and everyday communication becomes informative. While the Internet of Things is still seeking shape, its effects have already made incredible strides as a universal solution media for the connected scenario. The architecture-specific study does always pave the conformation of related fields. The lack of overall architectural knowledge resists the researchers' understanding of the scope of Internet of Things-centric approaches.

This literature surveys Internet of Things-oriented architectures capable enough to improve the ability of related tools, technology, and methodology to facilitate developer's requirements. Directly or indirectly, the presented architectures propose solving real-life problems by building and deploying powerful notions of the Internet of Things. A main contribution of this survey paper is that it systematically summarizes the current state-of-the-art of Internet of Things architectures in various domains. Figure 2.8 presents the domain tree showing all its leaves as subdomains.[11]

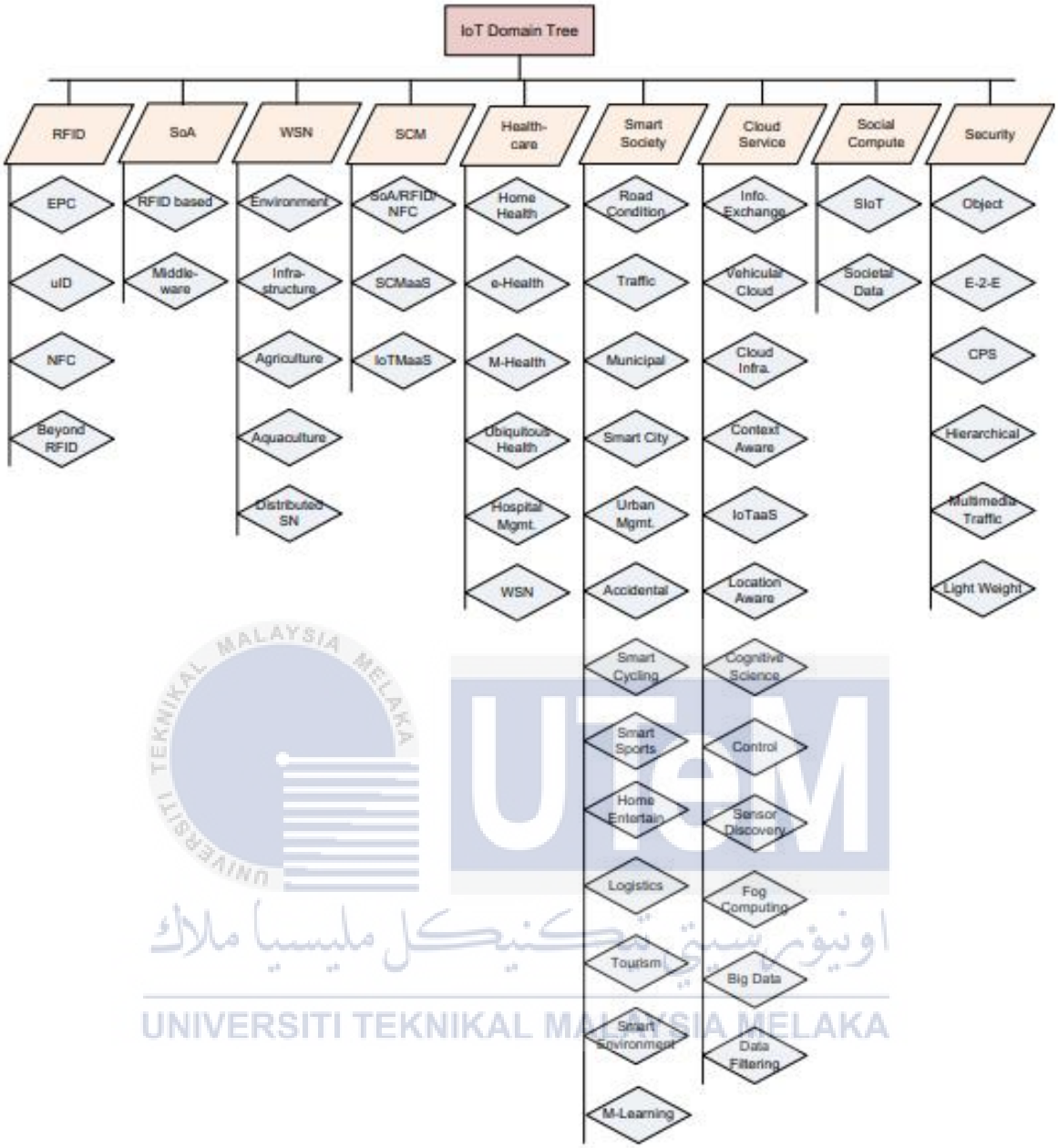


Figure 2.8 Application Domains of IoT Cloud Platforms

2.5 Previous Related Research of Work

In preparation for developing a Smart Inventory Management System, thoroughly reviewing previous research and papers is crucial. The primary aim of this review is to examine the ideas and concepts employed in addressing the challenges the project will encounter. These sources were selected based on their relevance to the project's objectives. By delving into prior research, the project team can gain valuable insights and leverage existing knowledge to enhance the development of the smart inventory management system.

2.5.1 STUDY OF SMART INVENTORY MANAGEMENT SYSTEM BASED ON THE INTERNET OF THINGS (IOT) [12]

In this study, the researcher has developed Inventory management based on IoT and RFID. It addresses the limitations of traditional inventory management models in handling the demands of product diversity and the need for efficiency. The system utilizes RFID technology, automatically tracking inventory items and recording the data in a central warehouse. This eliminates the need for manual tracking and reporting, saving time and reducing the probability of human error.

The integration of IoT and RFID technology provides significant advantages in terms of efficiency and accuracy, making it a promising solution for inventory management. RFID systems feature three main components: RFID tags, antennas, and readers. Radio-frequency identification (RFID) uses radio waves to read and capture information stored on a tag attached to an object. The Internet of Things (IoT) is crucial in the proposed inventory management system. By connecting smart devices to the internet and enabling communication, the method leverages Social IoT (SIoT) to facilitate information sharing and collaboration.

However, interoperability, security, and privacy must be addressed to realize IoT's potential fully. Security, in particular, is a priority due to the vulnerability of devices with limited resources. Nonetheless, the benefits of IoT in automating asset tracking and reporting in inventory management are evident, resulting in significant time savings and reduced errors. Supply Chain Management (SCM) is integral to the proposed inventory management system. SCM systems handle the flow of products, data, and finances throughout the supply chain, from product development to logistics and shipping. By incorporating IoT and RFID technology, the system enables efficient tracking and management of inventory across multiple stages, including raw materials, production facilities, and finished products.

The use of RFID tags with unique IDs attached to each object facilitates accurate tracking and reduces the need for manual inventory management. With the integration of IoT and SCM, organizations can optimize inventory levels, improve customer service, and reduce costs. By leveraging IoT and RFID technology, the system enables efficient and accurate tracking of inventory items, automates asset tracking and reporting, and improves overall inventory management in terms of cost optimization and customer service. While challenges are to be overcome, the potential benefits and prospects for developing this system are significant.[12]

2.5.2 SMART MANAGEMENT SCHEME FOR THE EFFICIENT CONTROL OF INDUSTRIAL INVENTORY [13]

In this study, the researcher developed a cost-effective and efficient database management system for industrial inventory. The manual management system many companies use is expensive and generates unfriendly reports, leading to stockouts and increased inventory costs. To address these issues, the researchers developed a database management system using Visual Basic 2017 for the front end and an SQL server for the back end. The system ensures zero error tolerance and allows easy identification and resolution of issues. It also enables companies to determine the Economic Order Quantity (EOQ) and Reorder Level (ROL) and reduce stockouts. The system offers versatility and can be used for vendor-managed inventory, benefiting companies with multiple branches.

Overall, this research presents a cost-effective solution that improves inventory management and allows for a comparison between automated and manual inventory systems. The methodology employed in this research involved database design, login form design, purchasing process design, sales process design, search options, and client data management. The database design included arranging data relationships and mathematical equations to calculate EOQ, safety stock, and ROL. The researchers utilized Visual Basic 2017 for the front-end development, allowing for an easy-to-use graphical interface, and SQL server for the back-end, ensuring data protection and integrity. The purchasing process involved entering product information and calculating safety stock and EOQ, while the sales process recorded product sales and updated inventory quantities.

The search option enabled users to find specific products in the inventory, and client data management stored information about employees, suppliers, vendors, and stakeholders. The system's functionalities were designed to streamline inventory management and improve efficiency. The results and discussions emphasized the practical

implementation of the software. The developed database management system offered comprehensive functionality, from customer orders to product delivery to dealers. It provided real-time inventory updates for top management and authorized personnel, including those in multiple branches.

The system's features, such as automatic calculation of EOQ and ROL, ensured accurate inventory control and reduced stockouts. The research highlighted the developed system's cost-effectiveness, efficiency, and reliability, making it a valuable tool for inventory management in various industrial settings. By incorporating this system, companies can enhance their inventory management processes, reduce costs, and improve overall productivity. Figure 2.9 shows a simplified diagram to illustrate the main phases of the purchasing process.[13]

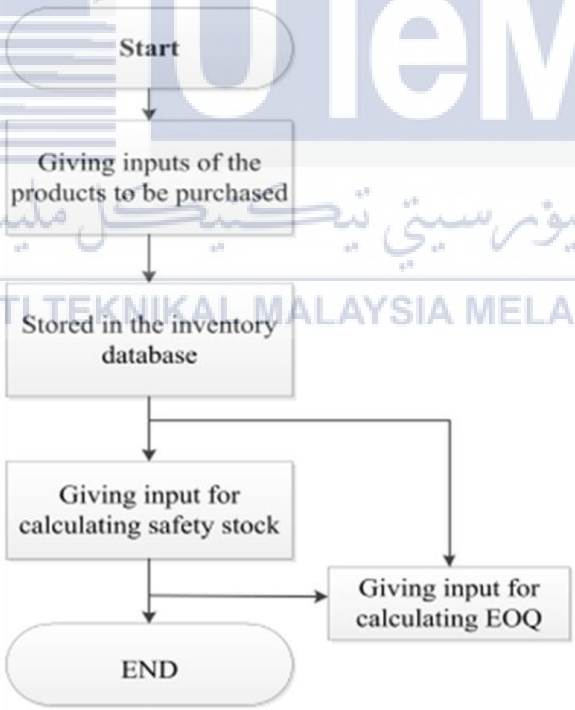


Figure 2.9 A simplified diagram to illustrate the main phases of the purchasing process

2.5.3 DEVELOPMENT OF INVENTORY MANAGEMENT SYSTEM BASED ON RADIO FREQUENCY IDENTIFICATION TECHNOLOGY[14]

The researcher proposed an inventory management system to check the tools and equipment in industrial plants and warehouses automatically. The system utilizes radio frequency identification (RFID), a cloud database (MySQL), an Android-based application program, and a PC-based computer program to construct a real-time raw material inventory system. This innovative approach enables real-time, automated inventory tracking, offering significant accuracy, time savings, and cost reduction benefits. In implementing this inventory management system is to attach RFID tags to the items that need to be tracked. RFID tags are small, adhesive chips that are affixed to each item. These tags contain a unique identification code, allowing them to be easily identified and tracked. Once the tags are attached, the items become "smart" and can interact with RFID readers. RFID readers are strategically installed throughout the facility. These readers communicate wirelessly with the RFID tags, capturing real-time data.

By strategically placing the readers at critical locations such as entry and exit points, storage areas, or workstations, the system can accurately track the movement of items as they enter, leave, or move within the facility. The information collected by the readers is then processed by a central database or inventory management software, providing up-to-date information on inventory levels, locations, and any status changes. This real-time visibility enables organizations to monitor inventory efficiently, reduce manual effort, minimize errors, and make informed decisions regarding reordering, restocking, and utilization.

Figure 2.10 shows the software architecture of the proposed system in implementing an RFID-based inventory management system, which offers a transformative solution for businesses seeking to enhance their inventory control processes.

Organizations can benefit from accurate and real-time tracking, improved efficiency, and reduced costs by attaching RFID tags to items and strategically placing RFID readers. Embracing this technology enables organizations to streamline operations, optimize inventory levels, and achieve greater productivity.[14]

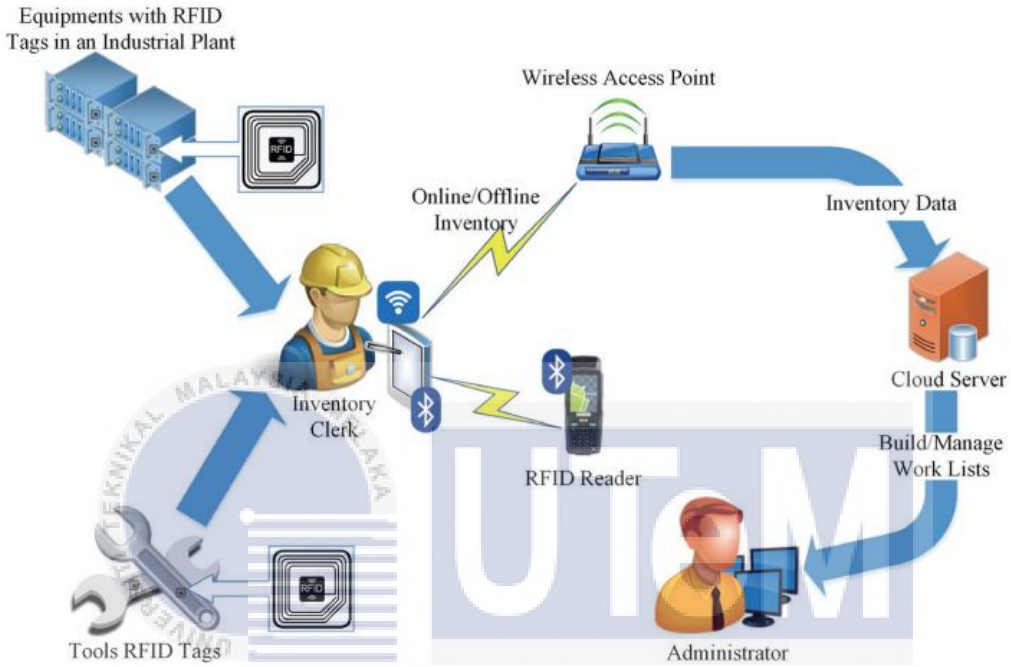


Figure 2.10 Proposed system architecture

2.5.4 SMART INVENTORY SYSTEM SCENARIO BASED ON THE INTERNET OF THINGS.[15]

The research method used for this system is inventory management by implementing an IoT-based intelligent inventory system (SIS). The system leverages the power of RFID tags, Wi-Fi technology, and a centralized server to create a dynamic and efficient inventory tracking system. The SIS provides real-time information about item availability, location, and quantity by combining RFID tags with the existing inventory management system. This enables customers to locate items within the inventory store

quickly and eliminates the need for manual stock checks or reorganization efforts. The SIS architecture incorporates Wi-Fi-enabled LPS, allowing customers to track a specific item's location using smartphones. By integrating LPS with RFID technology, the system provides accurate and up-to-date information about the current position of items.

This feature dramatically enhances the customer experience, as they can quickly and effortlessly find the things they need without needing manual searching or assistance from store staff. Furthermore, the SIS architecture includes a centralized server that maintains the inventory database and receives real-time updates from shelf monitors equipped with RFID readers. These updates ensure the server has the most accurate and current information about item availability and quantity.

The server also handles customer authentication, providing secure access to the system. Additionally, the proposed approach has the potential for further expansion, such as automating the billing process by integrating item prices. By combining RFID technology, Wi-Fi-enabled LPS, and a centralized server, the SIS provides real-time information about item availability, location, and quantity. This enhances inventory management efficiency, improves the customer experience, and eliminates the need for manual stock checks.

The proposed system has the potential to transform inventory management processes in various industries and offers opportunities for further advancements in automating billing and integrating additional functionalities.[15]

2.6 Research Comparison

Table 2.2 Research Work Comparison

Aspect/Feature	**Research Work 2.5.1**	**Research Work 2.5.2**	**Research Work 2.5.3**	**Research Work 2.5.4**
Objective	- Improve efficiency and accuracy in inventory management through IoT and RFID technology.	- Improve accuracy in inventory tracking through RFID technology.	- Real-time tracking and integration with a cloud database for efficient inventory management.	- Real-time tracking, precise item location information, and potential for expansion.
Main Components	RFID tags, RFID antennas, RFID readers	RFID tags, RFID antennas, RFID readers	RFID tags, RFID readers, Cloud Database	RFID tags, Wi-Fi-enabled LPS, Centralized server

Real-Time Tracking	Yes	No	Yes	Yes
Accuracy	High	High	High	High
Time Savings	Yes	No	Yes	Yes
Cost Reduction	Yes	No	Yes	Yes
Integration with Existing Systems	No	No	Yes	No
Item Availability Information	Yes	No	Yes	Yes

Item Location Information	No	No	Yes	Yes
Item Quantity Information	No	No	Yes	Yes
Customer Experience	N/A	N/A	Enhanced	Enhanced
Expansion Potential	N/A	N/A	Billing Automation, Additional Functionality	Billing Automation, Additional Functionality
Disadvantage	- No integration with existing systems.	- Lack of real-time tracking capabilities.	- Potential security and privacy concerns with cloud-based storage.	- Limited integration potential and potential reliance on Wi-Fi connectivity.

Table 2.2 displays the comparison between the four Research Works. Research Work 2.5.1 introduces an inventory management system that combines IoT and RFID technology. This system provides real-time tracking, high accuracy, time savings, and cost reduction to improve efficiency and accuracy. It offers the potential for integration with existing systems, enabling organizations to achieve comprehensive inventory management solutions. Research Work 2.5.1 presents a promising solution for optimizing inventory management processes by leveraging IoT and RFID. On the other hand, Research Work 2.5.2 focuses solely on RFID technology for inventory tracking.

While it offers improved accuracy, it lacks real-time tracking capabilities and integration features. Although it enhances the accuracy of inventory tracking, it may not provide the same efficiency and time savings as systems that incorporate real-time monitoring. Research Work 2.5.3 combines RFID technology with a cloud database to achieve real-time monitoring, integration capabilities, and improved decision-making. This system offers the advantages of real-time monitoring, providing up-to-date information about item availability, location, and quantity. Additionally, it enables integration with existing systems, offering a comprehensive solution for inventory management. However, potential security and privacy concerns may arise due to cloud-based storage, which must be addressed and mitigated to ensure data protection.

Research Work 2.5.4 introduces a comprehensive inventory management system incorporating IoT, RFID, Wi-Fi, and a centralized server. This system provides real-time tracking and enhances the customer experience by offering precise item location information. It also has the potential for expansion, such as automating billing processes and integrating additional functionalities. However, it is essential to note that this system may rely on Wi-Fi connectivity, which can be a potential limitation and may have limited integration potential.

Overall, the comparison highlights the importance of selecting an inventory management system that aligns with an organization's specific needs and objectives. Research Work 2.5.1 and 2.5.3 are comprehensive solutions offering real-time tracking, accuracy, time savings, cost reduction, and potential integration capabilities. Research Work 2.5.4 enhances customer experience and expansion possibilities by integrating Wi-Fi and location positioning systems. On the other hand, Research Work 2.5.2 focuses solely on RFID technology, offering improved accuracy but lacking real-time tracking and integration features. Ultimately, organizations should carefully consider the features and objectives of each system to make an informed decision and optimize their inventory management processes accordingly.

2.7 Summary

In conclusion, Chapter 2 highlighted the significance of inventory management and the challenges organizations face using manual systems. It discussed various components and technologies used in smart inventory management, such as RFID tags, load cells, and ultrasonic sensors. The review examined previous studies on integrating IoT and RFID technology, database management systems, and real-time inventory tracking systems. These studies demonstrated improved efficiency, accuracy, and cost reduction in inventory management. Overall, the review provides a comprehensive overview of IoT-based smart inventory management systems, emphasizing the need for automation and the potential benefits for organizations. It also identifies areas for further research in this field.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter presents a thorough overview of the project's execution, emphasizing the step-by-step process of designing and developing the smart inventory management system. The incorporation of hardware and software components is detailed, showcasing the successful methods employed. The execution is acknowledged as well-executed, resulting in a functional mechanism for the smart inventory management system. The chapter provides a comprehensive account of the project's implementation, underscoring the effective utilization of hardware and software elements to achieve the system's objectives.

3.2 Selecting and Evaluating Tools for a Sustainable Development

Developing an IoT-based smart inventory management system using Arduino involves carefully selecting and evaluating sensors, ensuring compatibility, considering environmental impacts, addressing social and economic implications, and utilizing methodological approaches. The system can efficiently monitor inventory and streamline operations by selecting accurate and reliable sensors, establishing IoT connectivity, and developing appropriate software. Considering energy efficiency, material selection, and conducting cost-benefit analysis helps mitigate environmental impacts and optimize resources: accessibility, usability, and iterative development address social and economic implications. Testing, validation, and iterative improvements ensure the system's accuracy and reliability.

3.3 Project Milestone

Identifying milestones in a project serves the important function of indicating the project's advancement and ensuring continuous progress. These milestones are instrumental in estimating the project's duration by setting specific deadlines for each milestone. Furthermore, they contribute to developing a comprehensive project plan, which can be visually represented in a flowchart Figure 3.1, facilitating easy comprehension of the project's flow and key achievements.

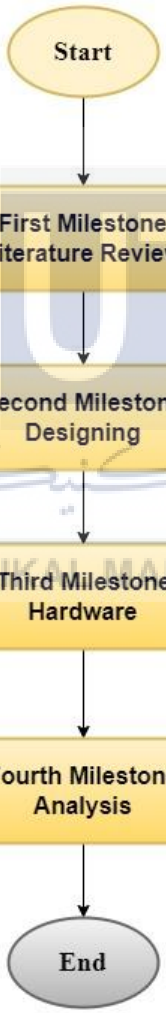


Figure 3.1 Flowchart of Methodology

3.4 First Milestone

Initiating the project, the supervisor and the researcher engage in discussions to define the objectives. Once the project objectives are finalized, the subsequent task is to complete a literature review. The literature review provides a comprehensive overview of previous research undertaken by scholars or institutions. By examining prior studies, the literature review offers valuable insights into the employed components, problem-solving methodologies, and analytical techniques. Upon reviewing the literature, the supervisor approves the specific scopes to accomplish the project goals. Figure 3.2 visually presents a flow chart representing the sequential steps in the literature review process.

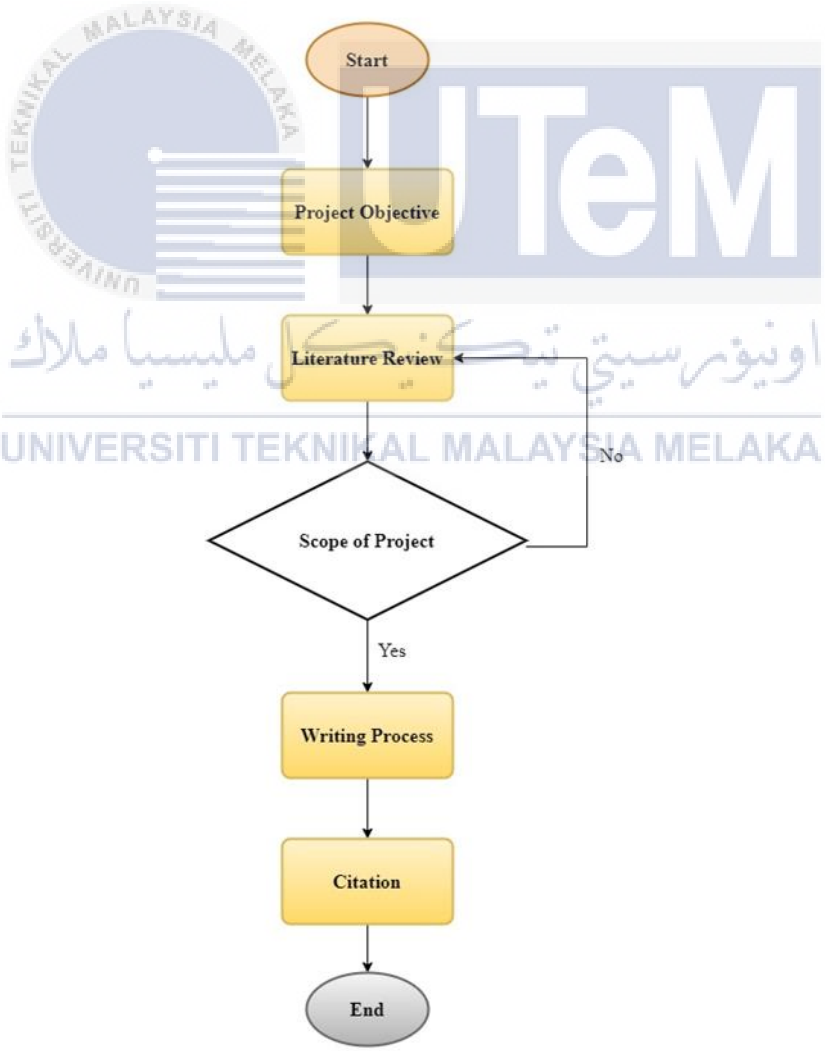


Figure 3.2 Flowchart of Literature Review

3.5 Second Milestone

This section comprehensively explains the methods and steps for developing the inventory system and IoT applications. The project aimed to create a real-time inventory system that integrates IoT technology, resulting in a smart inventory management system. Various essential processes must be completed to ensure the anticipated outcome, including data collection, implementation, testing, and troubleshooting.

These processes are utilized to analyze the collected data and generated output. The stages involved in accomplishing the project are illustrated in Figure 3.3. By employing these methods, it becomes feasible to construct and assess the project effectively.

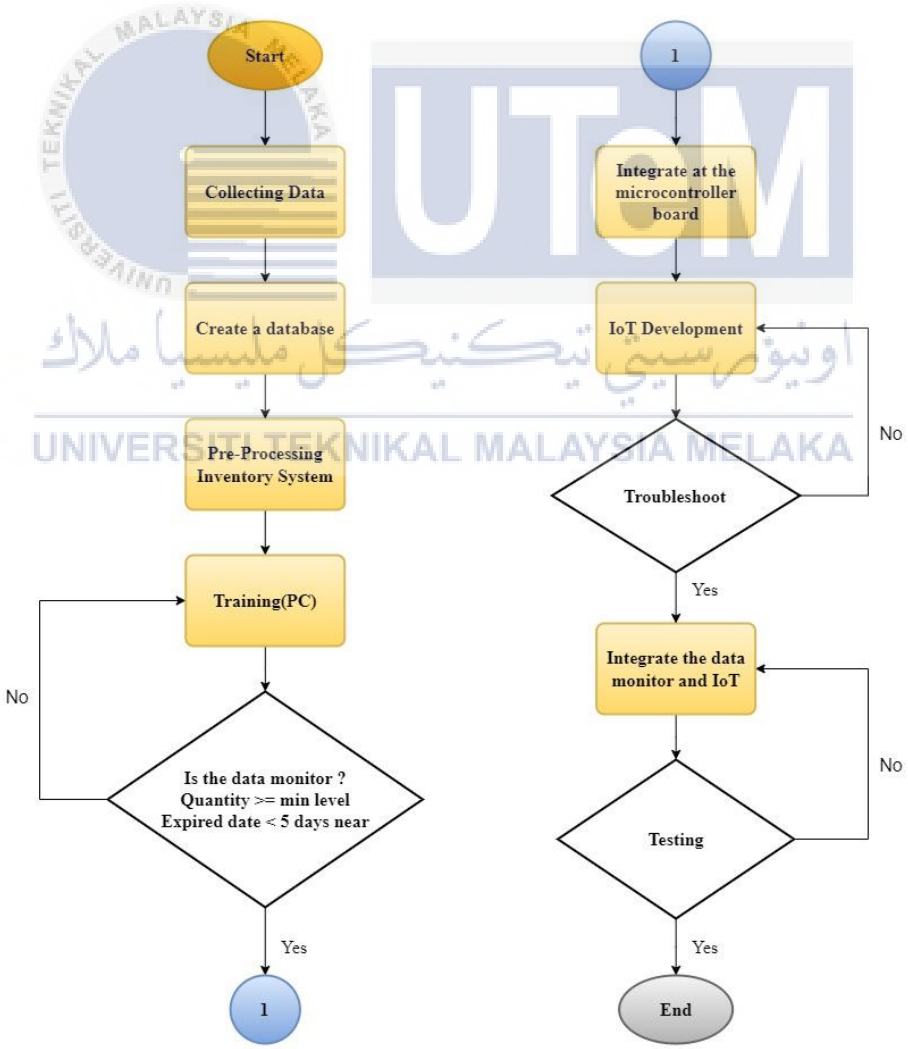


Figure 3.3 Flowchart of Designing the Project

3.5.1 Software Development

This section provides an overview of the software employed with the hardware component. The coding and simulation aspect of the project is particularly highlighted due to its crucial role in overseeing and managing the entire monitoring system. Emphasizing its significance, the coding and simulation process is considered the critical component that ensures the effective operation and functionality of the monitoring system as a whole.

3.5.1.1 Proteus

Proteus software is for virtual simulation in microcontroller experimental teaching. The traditional system design model is compared with the proposed model based on Proteus. Proteus software has been chosen as the preferred tool for designing the electrical circuit in this project due to its user-friendly nature and widespread use among design engineers and technicians. In addition to schematic layout, PCB layout, circuit simulation, and other features, the most significant component of Proteus can simulate the microcontroller and its peripheral devices[16].

It forms a new system design model: schematic design, simulation circuit design, and modification of physical production and testing[17]. The findings emphasize the significance of Proteus in microcontroller experimental teaching and suggest its potential for transforming the traditional teaching model and improving teaching quality.[18] Figure 3.4 shows the circuit diagram using the Proteus diagram.

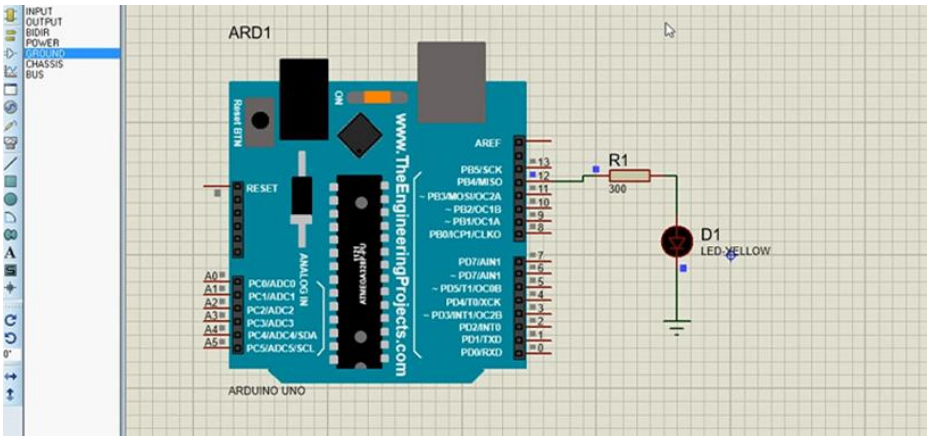


Figure 3.4 Circuit Diagram using Proteus Software

3.5.1.2 Arduino IDE

Arduino IDE is an open-source software developed by Arduino.cc for editing, compiling, and uploading code to Arduino devices. It is an Integrated Development Environment and compatible with various Arduino modules. Arduino IDE is user-friendly, making it accessible even to individuals without technical knowledge. It is available for multiple operating systems, including MAC, Windows, and Linux, and operates on the Java platform.

The software facilitates code debugging, editing, and compilation and supports various Arduino modules such as Arduino Uno, Arduino Mega, Arduino Leonardo, and Arduino Micro. Each module contains a microcontroller that executes the uploaded code. The main code, a sketch, is created in the IDE environment and generates a Hex File that is transferred and uploaded to the microcontroller on the Arduino board. Figure 3.5 shows the interface of IDE for Arduino. The IDE environment consists of an Editor for writing code and a Compiler for compiling and uploading the code to the Arduino module. Arduino IDE supports both the C and C++ programming languages.[18]

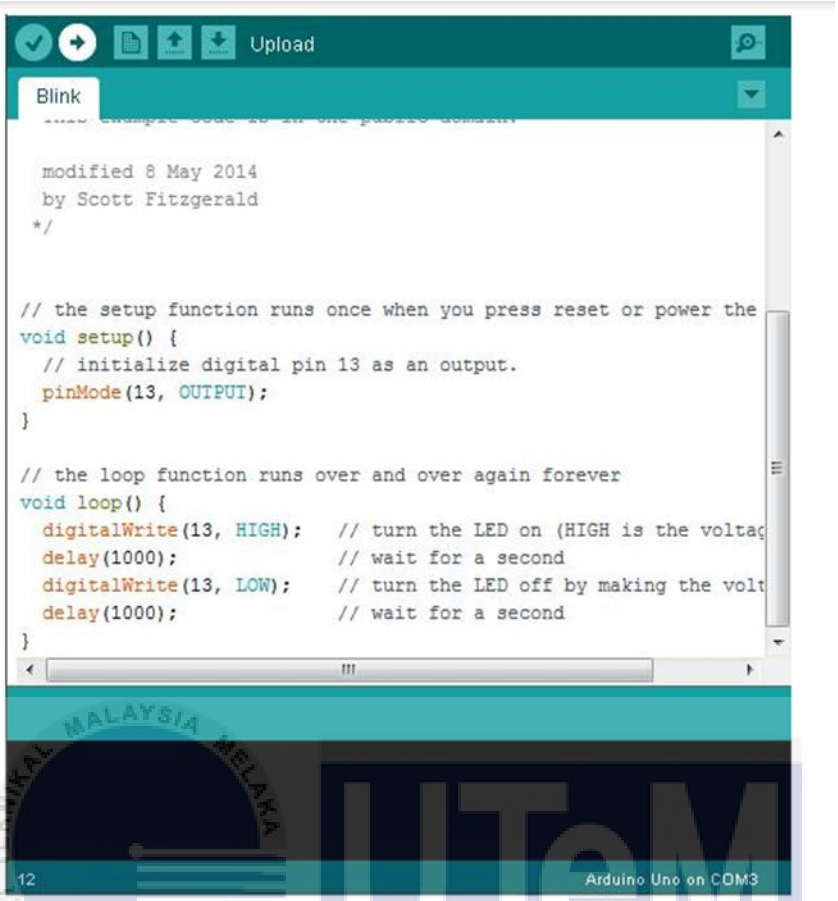


Figure 3.5 Interface IDE for Arduino

3.5.1.3 MySQL Database with PHP

MySQL is a widely used database system for web applications. It runs on a server, is fast and reliable, and can be used for small and large applications. MySQL is free to download and use, and it supports standard SQL for querying and manipulating data. The data in MySQL databases are organized in tables consisting of columns and rows. It is commonly used with PHP to create dynamic and data-driven websites.

MySQL is popular among websites with large amounts of data and high traffic. It can also be scaled down for embedded database applications. One of the languages for developing Web projects is PHP, a scripting programming language [19]. PHP supports MySQL, Oracle, and MS Access. An open-source software environment, PHP (personal

homepage), scripting language, and MySQL database offer alternative, efficient, cost-effective solutions for creating data-driven, dynamic, and personalized Web applications.[20] Figure 3.6 revealed the essential MySQL functions.

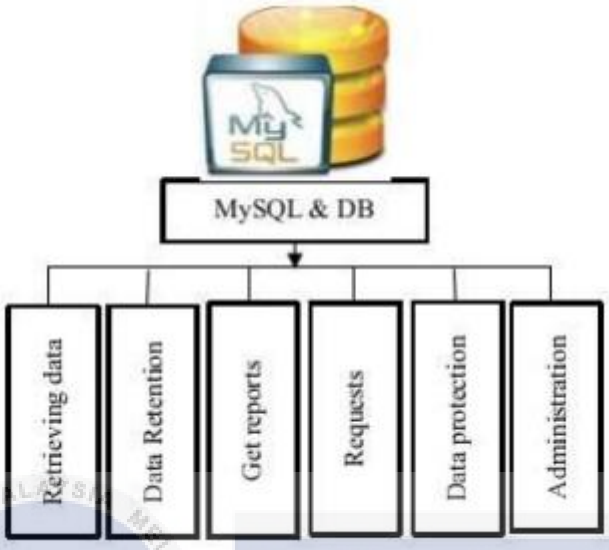


Figure 3.6 Basic MySQL Functions

3.6 Third Milestone

This section covers the circuit hardware and the controller implemented in this project. It is an essential component of a project.

3.6.1 NodeMCU V2 ESP8266

NodeMCU is a versatile and open-source development board designed to facilitate the prototyping and implementation Internet of Things (IoT) projects. At its core, NodeMCU v2 integrates the ESP8266, a powerful and cost-effective Wi-Fi module developed by Espressif Systems. The ESP8266 boasts a Tensilica L106 32-bit microcontroller operating at a clock speed of 80 MHz and comes equipped with 4 MB of flash memory. NodeMCU's strength lies in its firmware, explicitly built for the ESP8266,

leveraging the Lua scripting language. This Lua-based firmware simplifies the development process for IoT applications, making it accessible to many developers.

Figure 3.7 shows the ESP8266 board. Its development board features a CH340G USB-to-serial chip, enabling seamless communication with the ESP8266 and simplifying the programming process. With a regulated 5V power output and a dedicated 3.3V power source, the board accommodates various external components and sensors. The GPIO pins are conveniently broken out, providing easy interfacing with other electronic devices. Moreover, the board includes a built-in LED connected to GPIO 2, serving as a handy indicator during development.

In terms of connectivity, NodeMCU supports the 802.11 b/g/n Wi-Fi standards, offering multiple modes such as Station, SoftAP, and Station+SoftAP. This flexibility allows the board to function simultaneously as a client, access point, or both. Additionally, the board provides interfaces for standard communication protocols like UART, SPI, and I2C, enhancing its compatibility with various sensors and peripherals. A flash button and reset button further contribute to the ease of development. Table 3.1 shows the specification of ESP8266.

Despite its compact dimensions of approximately 49mm x 24mm, NodeMCU packs a punch in terms of functionality. It has become famous for IoT enthusiasts and developers, offering a user-friendly platform for experimenting with connected devices and creating innovative solutions. The active NodeMCU community continually contributes to its development, ensuring a wealth of resources and support for users embarking on IoT projects.[21]

Table 3.1 ESP8266 V2 Specification

Specification	Details
Microcontroller	ESP8266 (Tensilica L106 32-bit MCU)
Clock Speed	80 MHz
Flash Memory	4 MB
Firmware	NodeMCU (Lua-based)
USB-to-Serial	CH340G
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1 (Max voltage: 3.3V)
UART	1
SPI	1
I2C	1
Flash Button	Yes
Reset Button	Yes
Built-in LED	Yes (Connected to GPIO 2)
Wi-Fi Standard	802.11 b/g/n
Wi-Fi Modes	Station, SoftAP, Station+SoftAP
Dimensions	Approximately 49mm x 24mm

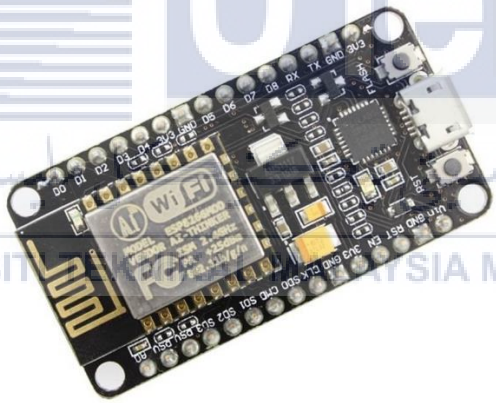


Figure 3.7 NodeMCU V2 ESP8266 Board

3.6.2 HC-SR04 Ultrasonic Range Detector Distance Sensor

Figure 3.8 Ultrasonic Sensor

Figure 3.8 shows the ultrasonic sensor known as the HC-SR04 ultrasonic ranging module. It is a device that allows non-contact measurement of distances ranging from 2cm to 400cm with an accuracy of 3mm. It consists of ultrasonic transmitters, a receiver, and a control circuit. The user provides a 10uS pulse to the trigger input to use the module, which initiates the ranging process. The module then emits eight bursts of ultrasonic waves at 40 kHz and detects the echo signal when it bounces back. By measuring the duration of the echo signal, you can calculate the distance to the target using the formula provided.

While ultrasonic waves generally have better directivity than low-frequency waves, they are attenuated more rapidly[22]. The module operates on a 5V DC power supply and has a working current of 15mA. It has a maximum range of 4m and a minimum coverage of 2cm. The module provides an echo output signal proportional to the distance, and the measuring angle is 15 degrees. Table 3.2 shows the ultrasonic sensor parameters.

Table 3.2 Ultrasonic sensor parameters

Specification	Details
Microcontroller	ESP8266 (Tensilica L106 32-bit MCU)
Clock Speed	80 MHz
Flash Memory	4 MB
Firmware	NodeMCU (Lua-based)
USB-to-Serial	CH340G
Operating Voltage	3.3V
Digital I/O Pins	11

3.6.3 Infrared Module Sensor

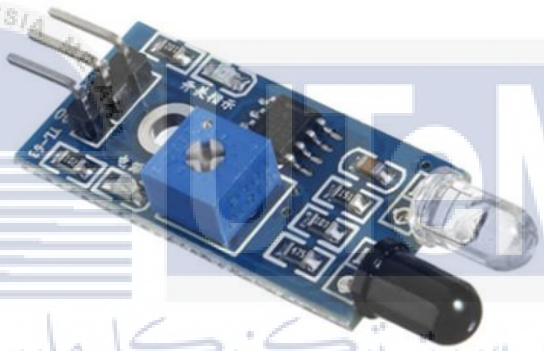


Figure 3.9 Infrared Sensor

The sensor module is an infrared proximity sensor that can adapt to different environments. Figure 3.9 shows the infrared sensor, which operates within a voltage range of 3.3V to 5V DC and utilizes a pair of infrared transmitting and receiving tubes. When an obstacle or reflecting surface is detected within its adjustable detection distance (2cm to 30cm), the module processes the signal and activates a green indicator light. It also outputs a digital signal (low-level) through the signal output interface. [23]

This module is commonly used in applications such as robot obstacle avoidance, obstacle avoidance cars, and line counting. It is also suitable for detecting an object. Table 3.3 shows the infrared sensor parameters.

Table 3.3 Infrared sensor parameters

Specification	Description
Sensor Type	Infrared
Voltage Range	3.3V - 5V DC
Current Consumption	45mA (at 5V)
Detection Distance	Adjustable: 2cm - 30cm (via potentiometer)
Size	32mm x 14mm
Weight	3g

3.6.4 Circuit Diagram

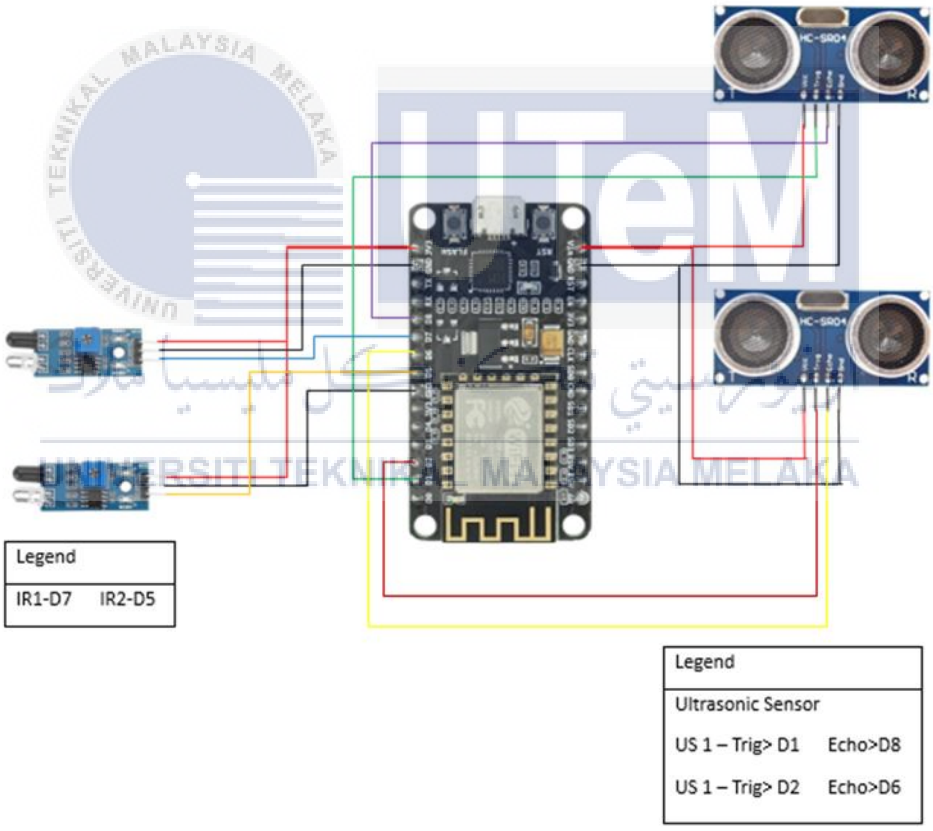


Figure 3.10 Wiring Connection Between Infrared Sensor and Ultrasonic Sensor with ESP8266

Figure 3.10 represents the wiring connection constructed between the Infrared Sensor (IR) and Ultrasonic Sensor with ESP8266. In the hardware configuration of the project, IR 1 is connected to the input D7, and IR 2 is connected to D5. For the first ultrasonic

sensor, the trigger is connected to D1, and the echo is connected to D8. Similarly, the second ultrasonic sensor's trigger is connected to D2, and the echo is connected to D6. These specific pin assignments establish the connections between the sensors and the microcontroller, allowing for effective data input and interaction within the system.

3.7 System Architecture

3.7.1 Block Diagram of the project

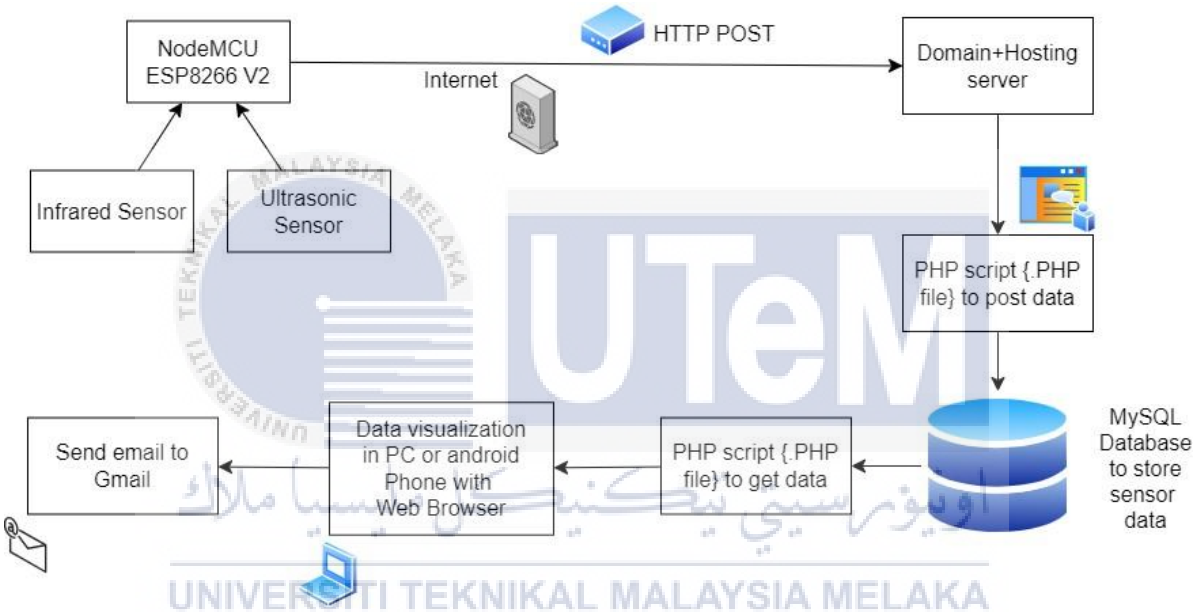


Figure 3.11 Block diagram of Smart Inventory Management System using Internet of Things.

Figure 3.11 shows that our innovative system has the engineering architecture as above. The sensors are connected to the Arduino microcontroller. Two types of sensors have been used for different purposes. These sensor data are collected and stored in MySQL Database cloud framework via a microcontroller and the Wi-Fi module ESP8266. Depending on the sensor data, the intimation is sent to the user. If the sensor value is less than the threshold value, the system generates an email and sent to the corresponding user. Expired dates are also crucial in this system. If the expiration date is near the current date,

the system will generate an email and send it to the corresponding user. Otherwise, it will continue monitoring the data.

3.7.2 Flowchart of the project

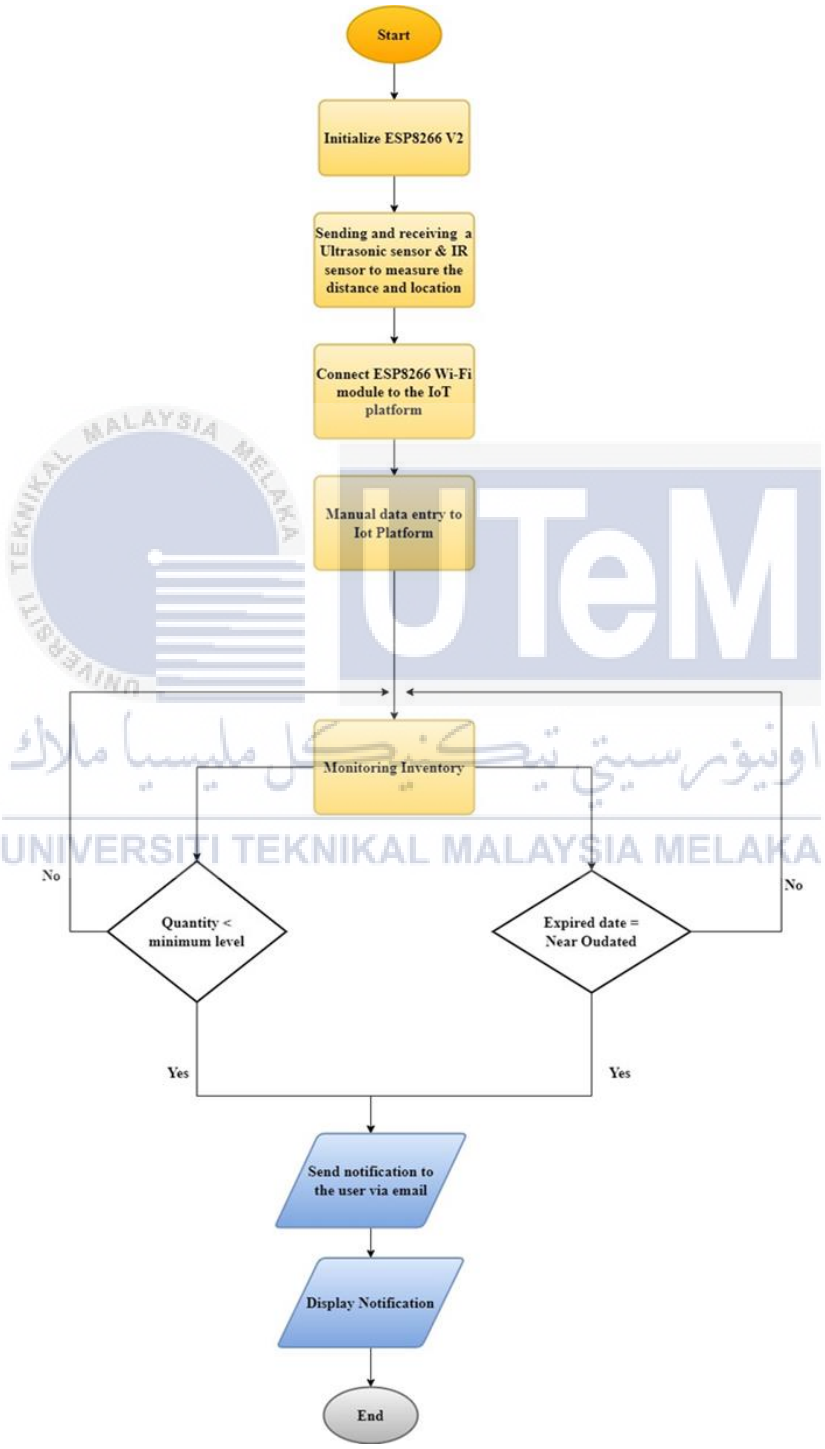


Figure 3.12 Flowchart of the project

The flowchart represents a system that utilizes an ESP8266 to interface with sensors such as the Ultrasonic Sensor and IR Sensor. It connects to the database through an ESP8266 WiFi, allowing manual data entry and inventory monitoring. The system includes a real-time clock at HTML to check for products nearing expiration (for example, nearly five days from the date). A notification is sent to the user if such products are nearly outdated.

Additionally, the system checks if the quantity of a product is below the minimum level, and if so, it sends a notification to the user. If neither condition is met, the system continues monitoring the inventory and displays notifications to the user when necessary. This flowchart outlines the steps and decision points involved in the system's operation, visually representing its workflow. Figure 3.12 displays the flowchart of the project.

3.8 Fourth Milestone

The fourth milestone in the project involves analyzing the system's performance by collecting data through various methods. The objective is to evaluate whether the system can accurately recognize the user's inventory in real-time. Data from the system's operation will be gathered and examined to achieve this. The collected data will then be evaluated to determine if the system is meeting the desired level of accuracy and real-time recognition of the user's inventory. Modifications will be made if the system's performance is poor or not meeting the requirements.

The modifications may involve revisiting the previous milestones and reviewing the system's design, algorithms, or any other components that could affect the system's performance. Making the necessary adjustments aims to improve the system's performance and ensure it accurately recognizes the user's inventory in real-time. Figure 3.13 shows the flowchart of the Milestones 4 process.

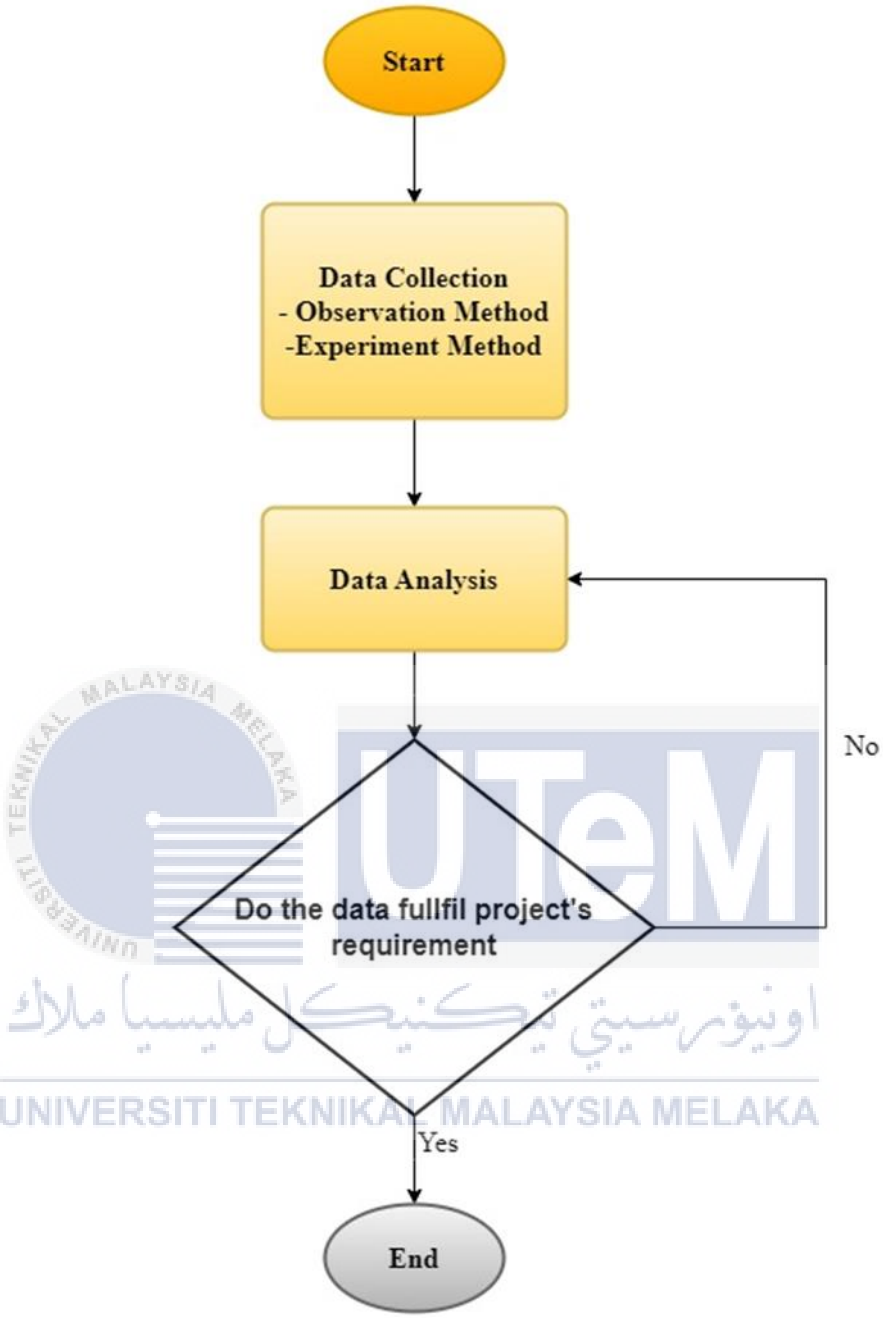


Figure 3.13 Flowchart of Milestone 4 process

3.9 Summary

The text outlines the execution of a project to develop a smart inventory management system using IoT technologies. It discusses milestones, including tool selection, project flowcharts, and key steps such as literature review and software development. The software development section introduces tools like Proteus, Arduino IDE, and MySQL Database with PHP. Hardware components include NodeMCU V2 ESP8266, HC-SR04 Ultrasonic Range Detector, and Infrared Module Sensor. The system architecture is represented through block diagrams and flowcharts. The project aims for sustainability, accuracy, and real-time inventory recognition, with regular evaluations and modifications to enhance performance.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter revealed the results, analysis, and discussion of the data obtained from the smart inventory management system, which are used to identify areas for improvement and maximize its performance. Data analysis identifies patterns and discrepancies, and the system's performance is evaluated against benchmarks. Root cause analysis helps understand the underlying causes of issues, and modification strategies are proposed to enhance the design. The system can achieve better inventory accuracy, order fulfilment, and operational efficiency by implementing these modifications.

4.2 System Design

In this project, a foundational database system utilizing HTML, PHP, and SQL has been expanded to include two essential features: a login form and a registration form. The project's core pages are Home, Register Product, Data Monitoring, and User Data. These pages help organize information. Home is like a central hub. Register Product deals with adding new products, Data Monitoring keeps track of data, and User Data manages user info for expiration. This simple setup keeps everything running smoothly and user-friendly.

4.2.1 User Login Interface

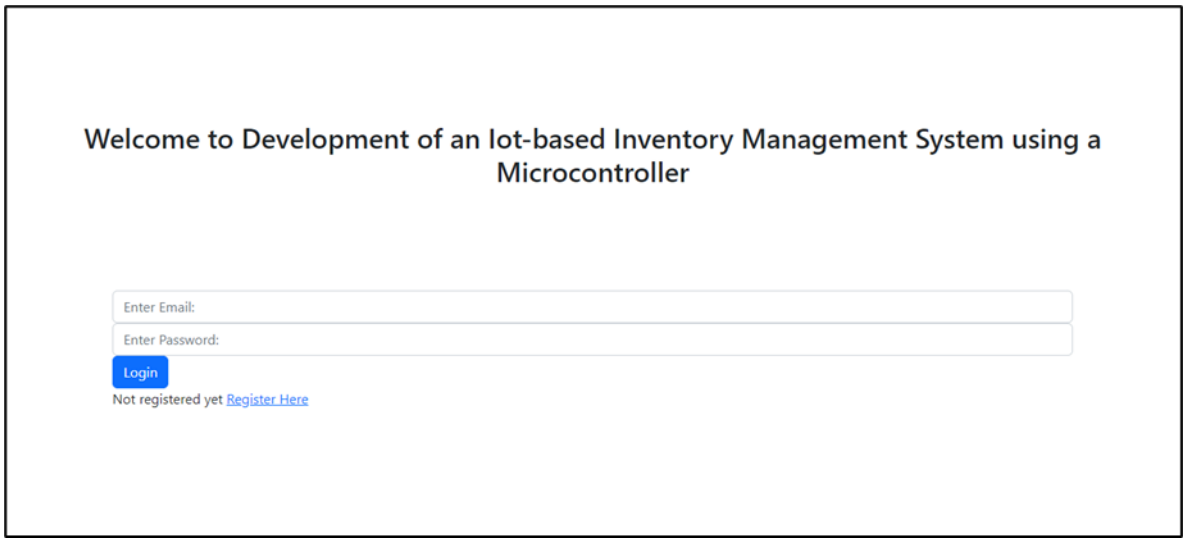


Figure 4.1 Login Form Interface

Figure 4.1 shows the login form interface. To enhance the functionality of the web application, a user-friendly login system has been made as a main page. This new feature allows registered users to access personalized data and manage their information securely. The login form is thoughtfully designed and seamlessly integrated into the existing layout. Users can easily input their credentials, consisting of a username and password, to gain access to their accounts. This login functionality ensures a more personalized experience, enabling users to interact with specific features such as product registration, data monitoring, and user data management.

4.2.2 User Registration Interface

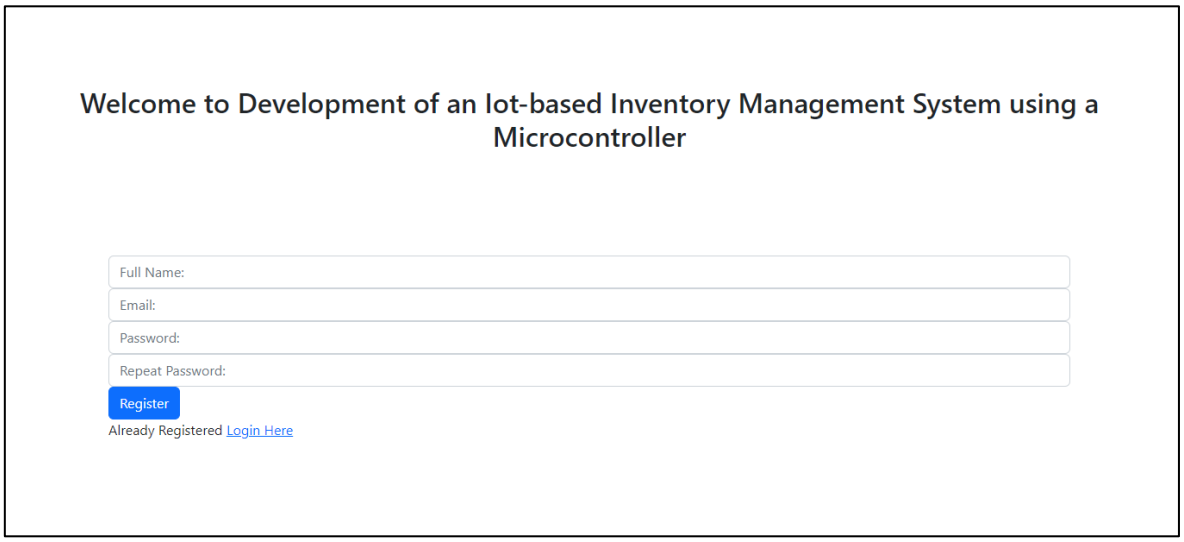


Figure 4.2 Registration Form Interface

Registration forms contribute significantly to data security by implementing secure authentication mechanisms, such as password protection and encryption, safeguarding user information from unauthorized access. Additionally, collecting user contact information through registration forms supports effective communication, enabling the delivery of updates, notifications, or personalized messages to users. Successful registrations result in a smooth redirection to a login page or home page, while any issues trigger a redirection back to the home page with an appended error message. This integrated registration system adds a layer of personalization to the application, allowing users to access tailored features and securely store their information for future interactions. Figure 4.2 shows the registration form interface.

4.2.3 Homepage Interface

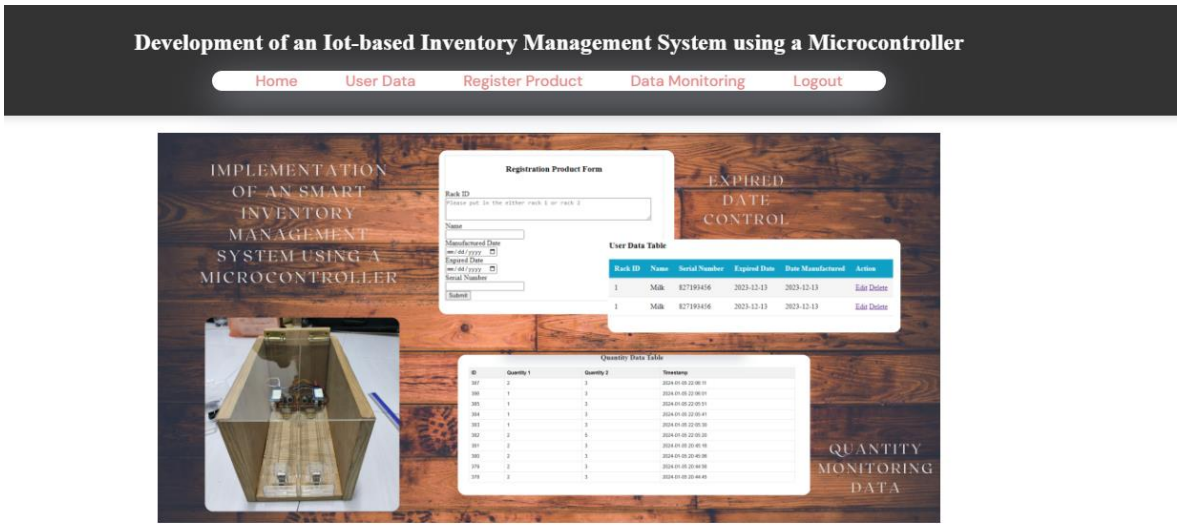


Figure 4.3 Homepage Interface

Figure 4.3 shows the homepage interface. The homepage is a visual introduction to this project's functionality. It features straightforward navigation and a cohesive design with interactive elements showcasing processes like product registration and data monitoring. Responsive and user-friendly, it encourages engagement through call-to-action buttons. Project branding is consistent, and the footer provides valuable information. Overall, the homepage is a compelling entry point, concisely communicating the project's purpose and functionality.

4.2.4 User Data Interface

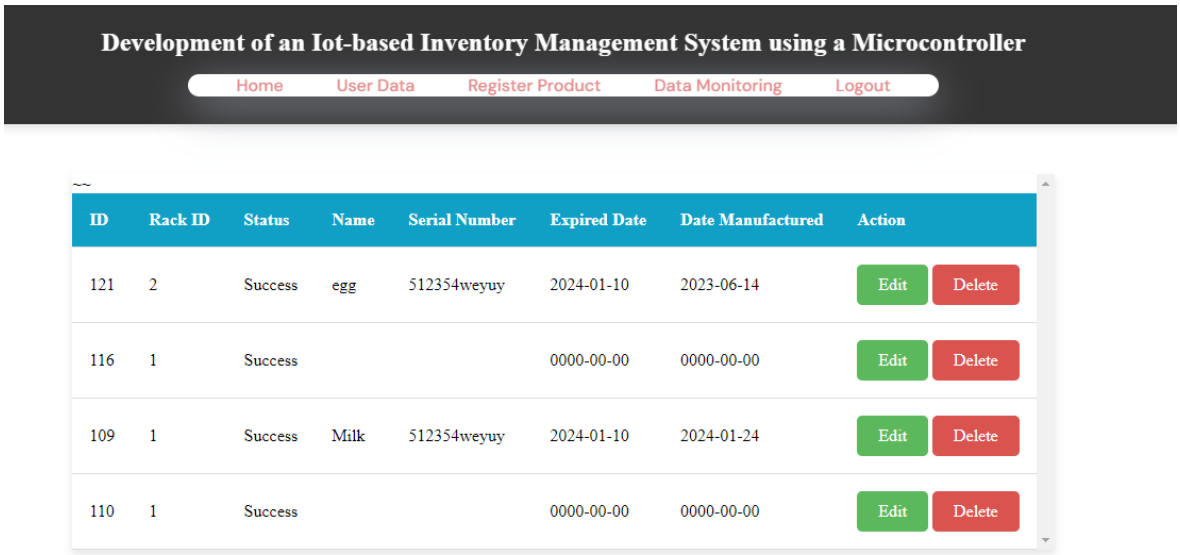


Figure 4.4 User Data Interface

The user data section presents a table containing essential information about registered products. Each entry in the table includes an ID, rack ID, status, name, serial number, expiration date, and date of manufacture. Each row has buttons for editing and deleting entries to enhance data management efficiency. The edit button lets users update the table in case of errors or modifications, ensuring accurate and up-to-date information. Conversely, the delete button configures and removes unwanted data entries, streamlining the database and maintaining the integrity of the product registry. Figure 4.4 shows the user data.

4.2.5 Registration Product Interface

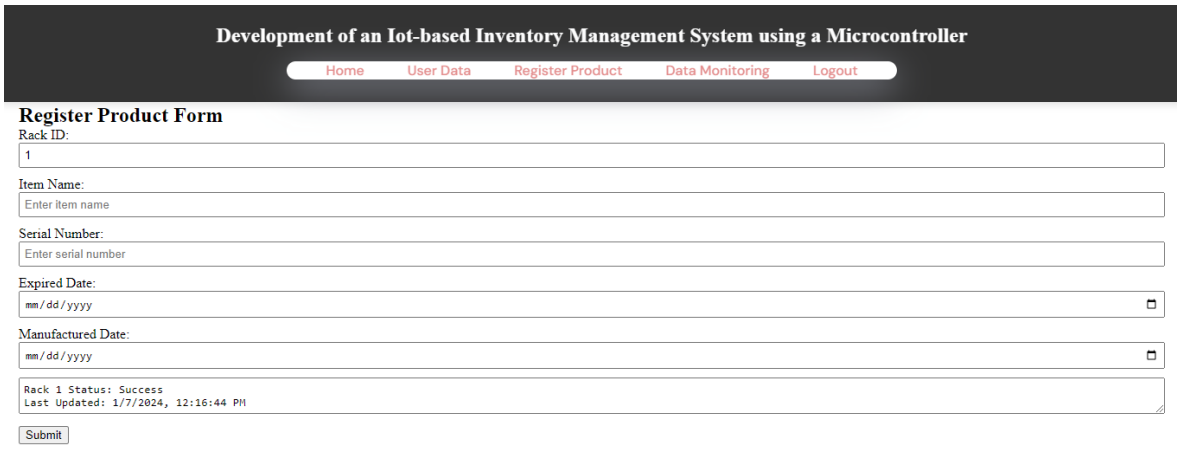


Figure 4.5 Registration Product Interface

The registration product form facilitates the input of item details for storage within the system. Essential information such as item name, serial number, expiration date, and manufactured date must be provided. Additionally, visibility into rack status and availability is offered. After completing the required fields, the submit button can store data in the database. This streamlined process ensures accurate and organized storage of item information, enhancing the system's data management efficiency. Figure 4.5 visually represents the registration product form.

4.2.6 Data Monitoring Interface

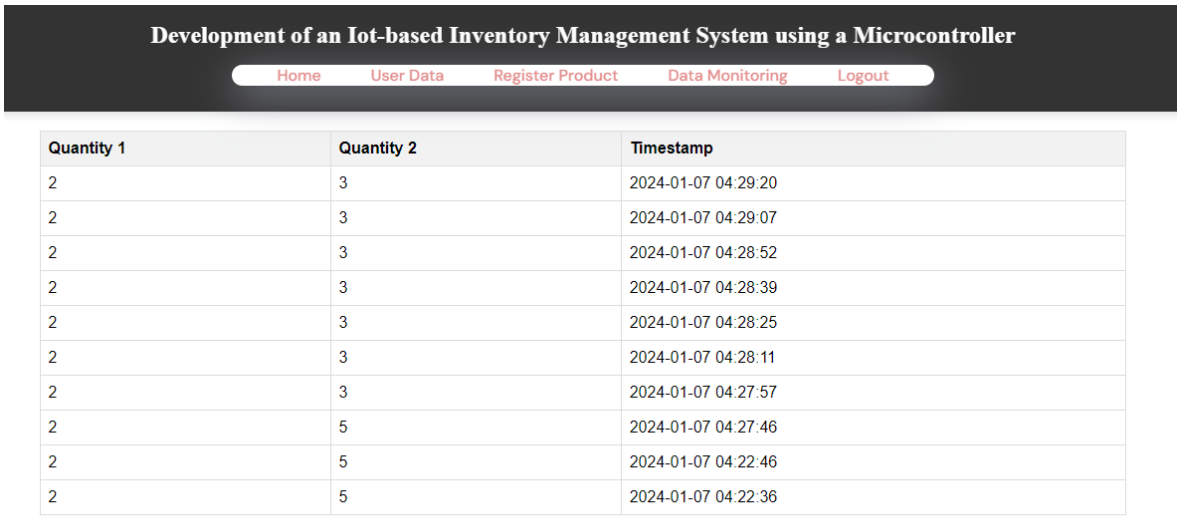


Figure 4.6 Data Monitoring Interface

Figure 4.6 displays real-time data on the quantity of objects using information sourced from an ESP8266 with an ultrasonic sensor. The system provides a timestamp, allowing users to track the quantity of objects from the system's inception. The data is gathered by measuring the object's distance using the ultrasonic sensor, offering a dynamic and up-to-the-minute overview of the object's quantity. This feature provides valuable insights into the system's historical trends and current status, enhancing user monitoring capabilities.

4.3 Project Integration

This chapter represents the pivotal phase of project integration, a crucial component in the development lifecycle where disparate elements are brought together to form a cohesive and fully functional system. The project's various components, modules, or subsystems are meticulously unified within this stage. The integration process extends beyond mere combination; it involves merging diverse technologies, databases, and

software components to enhance interoperability and elevate the system's overall performance. This chapter delves into the intricacies of project integration, shedding light on the planning, testing, and coordination essential for the seamless fusion of different project elements.

4.3.1 Software Results

4.3.1.1 Connection with ESP8266V2 with Database MySQL

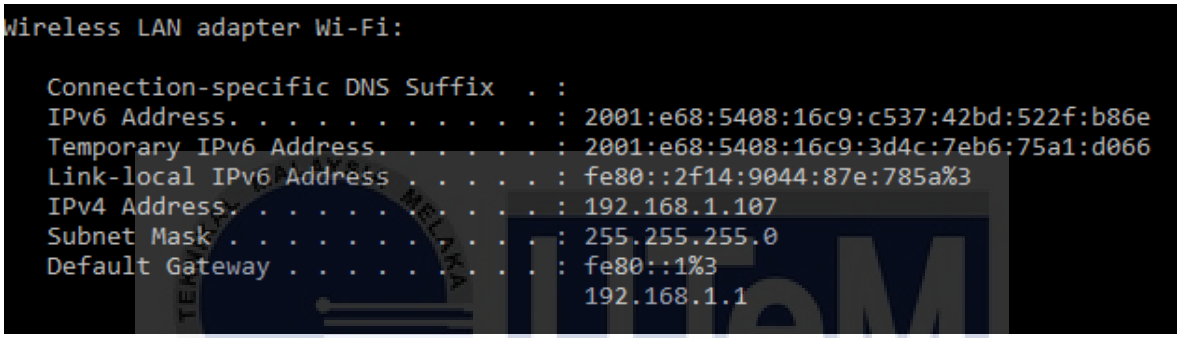


Figure 4.7 IP Address IPv4 of the WiFi

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char *ssid = "POCO F5";
const char *password = "Fairy753159";
const char *phpServer = "127.0.1.1"; // Change to your server IP
const char *phpPage = "/real/datamonitoring.php";
```

Figure 4.8 ESP WiFi Code

Figure 4.8 shows that the communication framework hinges on utilising IP addresses within the Wi-Fi network to establish a seamless connection between hardware and software. The ESP8266 microcontroller, equipped with its integrated Wi-Fi module, streamlines the process, eliminating the need for manual configuration. However, a meticulous setup is required when the ESP-01 Wi-Fi module is employed.

For effective communication, Figure 4.7 shows the system's PHP page must be intricately directed, encompassing the correct IP address of the server. Typically, the default server IP address is designated as 127.0.0.1, and the alias 'localhost' serves as an alternative reference. This meticulous configuration ensures a harmonious flow of data from the ESP8266 hardware to the server, underlining the critical role of precise IP addressing in fostering successful integration between the physical components and the software architecture.

4.3.1.2 The Database Setup

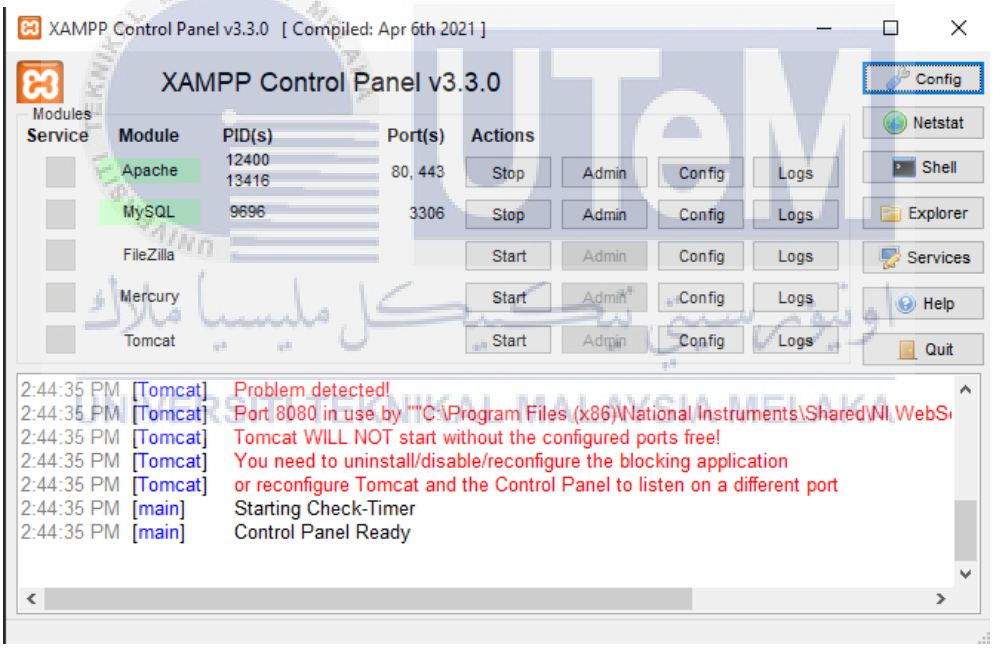


Figure 4.9 XAMMP Control Panel

Figure 4.9 shows the XAMMP control panel. Setting up XAMPP involves a straightforward process. Begin by downloading the XAMPP installer from the official website and choosing the version compatible with your operating system. Once downloaded, run the installer and follow the on-screen instructions. During installation, select the

components you want, such as Apache, MySQL, PHP, and phpMyAdmin. After installation, start the XAMPP Control Panel and initiate the Apache and MySQL modules.

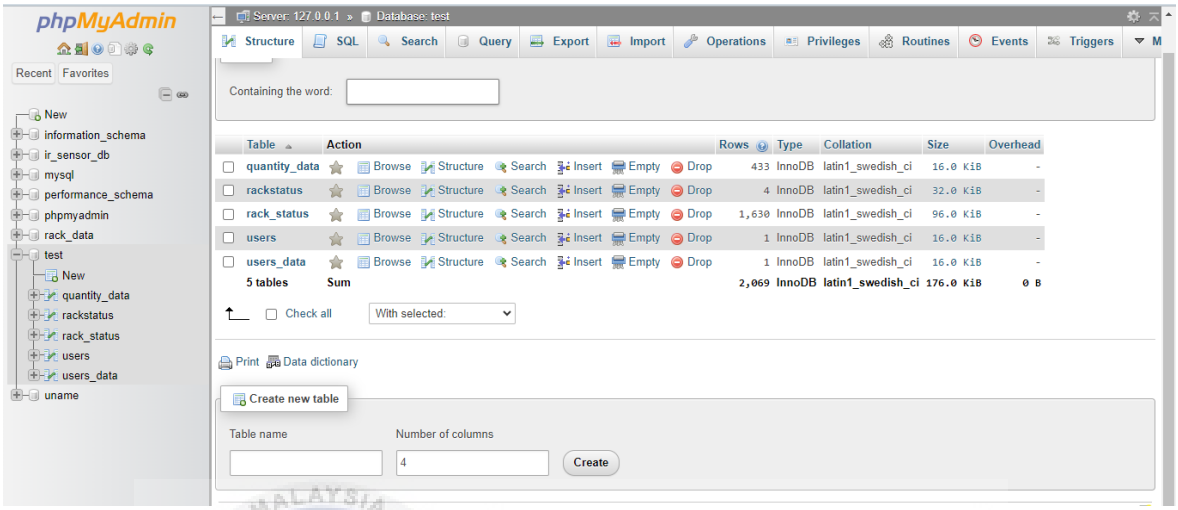


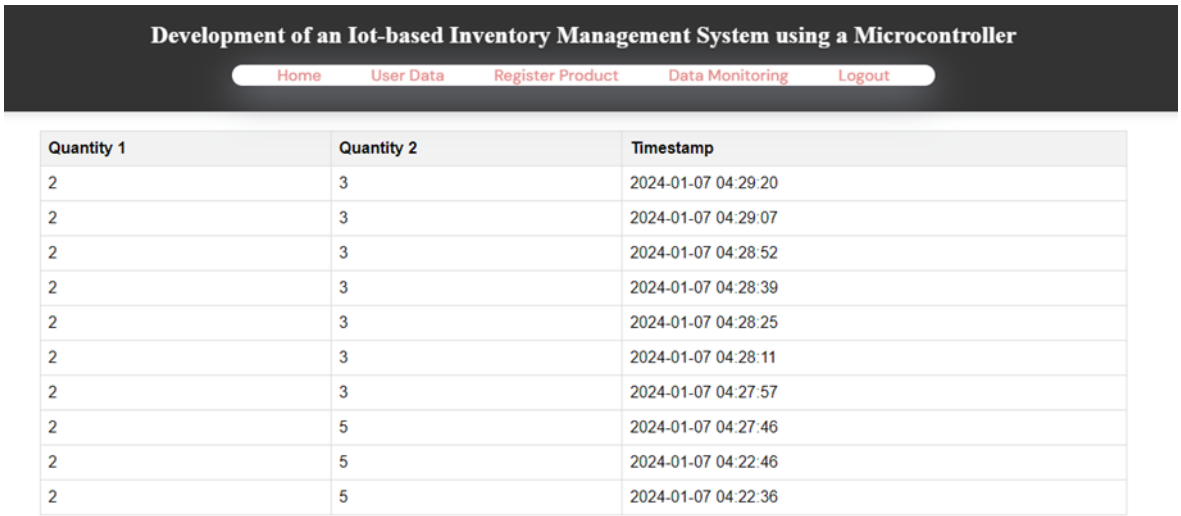
Figure 4.10 Creating a Data Table in the Database

To create a table in MySQL, access phpMyAdmin through a web browser by navigating to http://localhost/phpmyadmin/. Log in with the default credentials ("root" for username and no password). Once in phpMyAdmin, create a new database by clicking "Databases," entering a name, and clicking "Create". Figure 4.10 shows the creation of a data table in the database.

4.3.1.3 Data Monitoring

Figure 4.11 Design Project for Ultrasonic Sensor

The project design in Figure 4.11 utilizes ultrasonic sensors to gauge the distance of objects. It employs a formula to estimate the number of boxes in a row based on this distance. Two ultrasonic sensors are assigned to a different line of boxes and positioned at fixed locations to maintain a consistent distance from each box. This design has a deliberate restriction, allowing each rack to hold a maximum of five items. This intentional limitation is implemented to enhance organizational efficiency by combining precise distance measurements, fixed sensor placements, and a cap on the number of items each rack can hold.



Quantity 1	Quantity 2	Timestamp
2	3	2024-01-07 04:29:20
2	3	2024-01-07 04:29:07
2	3	2024-01-07 04:28:52
2	3	2024-01-07 04:28:39
2	3	2024-01-07 04:28:25
2	3	2024-01-07 04:28:11
2	3	2024-01-07 04:27:57
2	5	2024-01-07 04:27:46
2	5	2024-01-07 04:22:46
2	5	2024-01-07 04:22:36

Figure 4.12 Data Monitoring Table

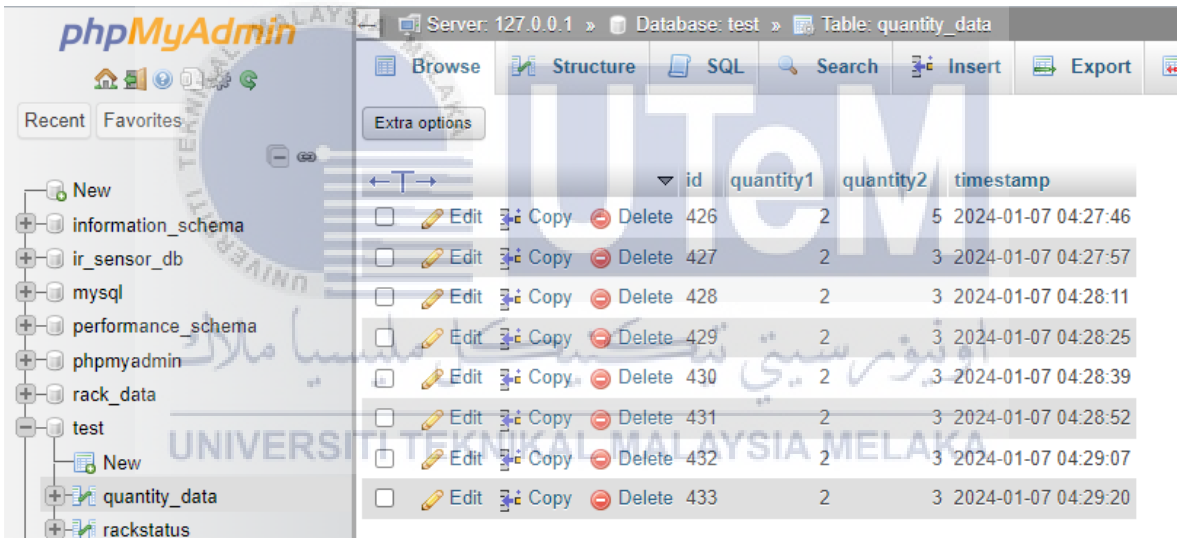


Figure 4.13 Data Monitoring Database

After establishing the connection between ESP8266 and the database, the user can monitor the data on the webpage in real time. Figure 4.12 shows the data monitoring table of quantity one and quantity two, along with the timestamp on the webpage. The condition in each item is different because the distance depends on the ultrasonic sensor. The webpage will be refreshed in 5 seconds. Figure 4.13 shows the data monitoring database.

```
16:14:47.639 -> HTTP Code: 200
16:14:57.647 -> Distance1: 46 cm
16:14:57.647 -> Distance2: 41 cm
16:14:58.063 -> POST data: quantity1=2&quantity2=2
```

Figure 4.14 The Serial Monitor of ESP8266

Figure 4.14 shows the serial monitor of ESP8266, which has an HTTP code of 200. This indicates the connection with the server is successful. This project calculates the data in cm. The distance between 36cm and 46cm will be calculated as the quantity item has two. The POST data showed the data was transferred from ESP8266 to the database.

4.3.1.4 Expired Date Control

Figure 4.15 The position of the IR sensor

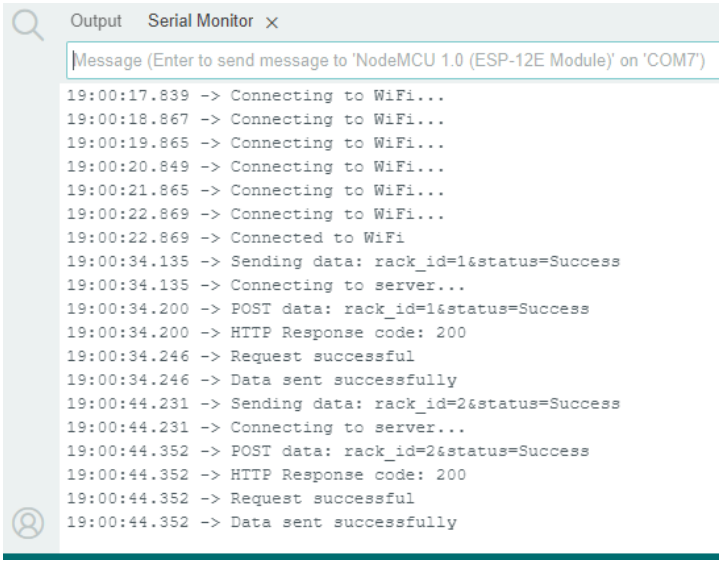


Figure 4.16 ESP8266 Serial Monitor

The IR sensor is positioned at a fixed location on the top cover, maintaining a height of 15cm. Figure 4.15 shows the position of the IR sensor. Its intentional fixed placement aims to distinguish between rack one and rack two. When the sensor registers at a HIGH state, it transmits the corresponding rack ID from ESP8266 to the database. Figure 4.16 shows the ESP8266 Serial Monitor.

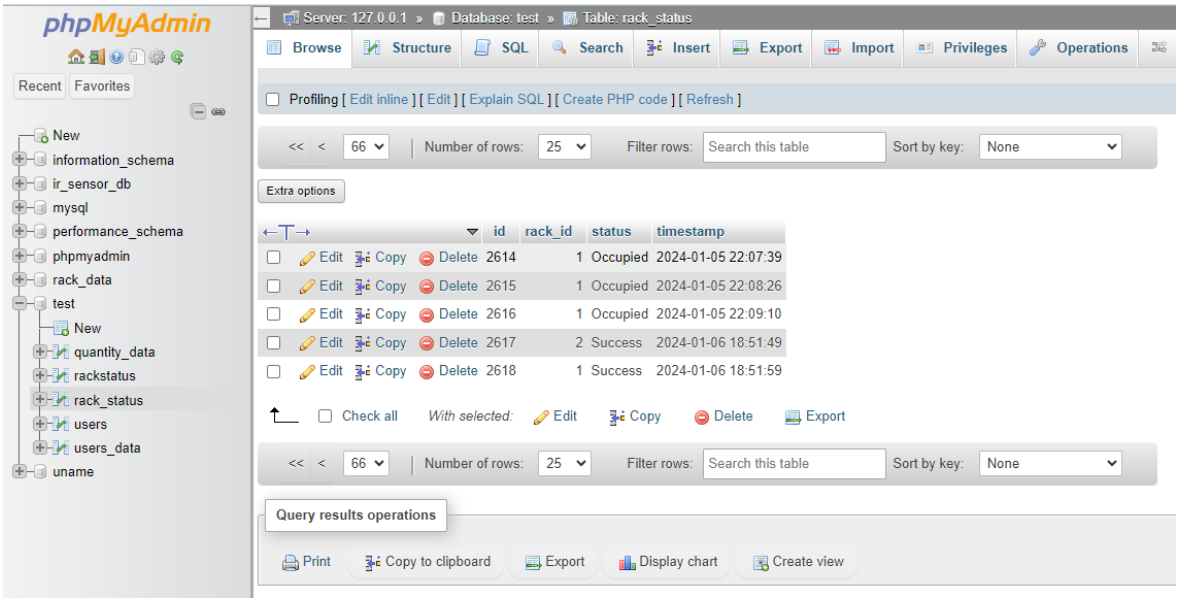


Figure 4.17 Database of IR sensor

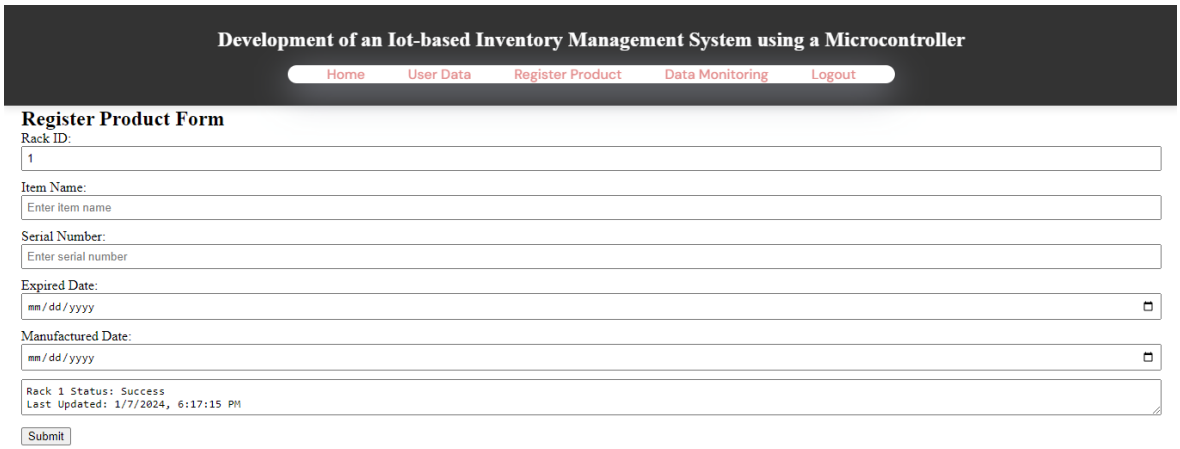


Figure 4.18 Register Product Form Interface

Before initiating data transmission, it is crucial to set up a prerequisite by creating a data table named "rack_status" in phpMyAdmin. Once this table is established, the registered product form on the webpage becomes operational, enabling users to input data such as the product name, serial number, expiration date, and manufacturing date. Figure 4.18 shows the interface of the register product form. Subsequently, when the submit button is pressed on the register product form, the entered data is transmitted and securely stored in the "rack_status" table within the database. Figure 4.17 shows the database of IR sensors.

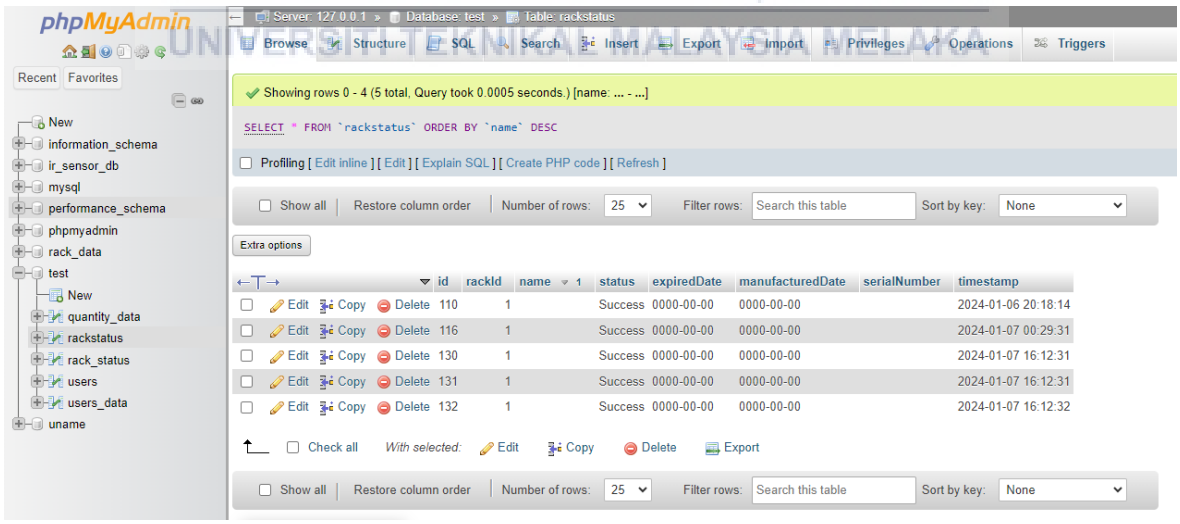


Figure 4.19 Database of User Data Table

To maintain a smooth data flow from the ESP8266 without any disruption, a separate data table named "rackstatus" is specifically created. The "rackstatus" table is dedicated to user data, providing real-time insights, including the display of either rack one or rack two during specific time intervals within the register product form. Figure 4.19 shows the database of the user data table.

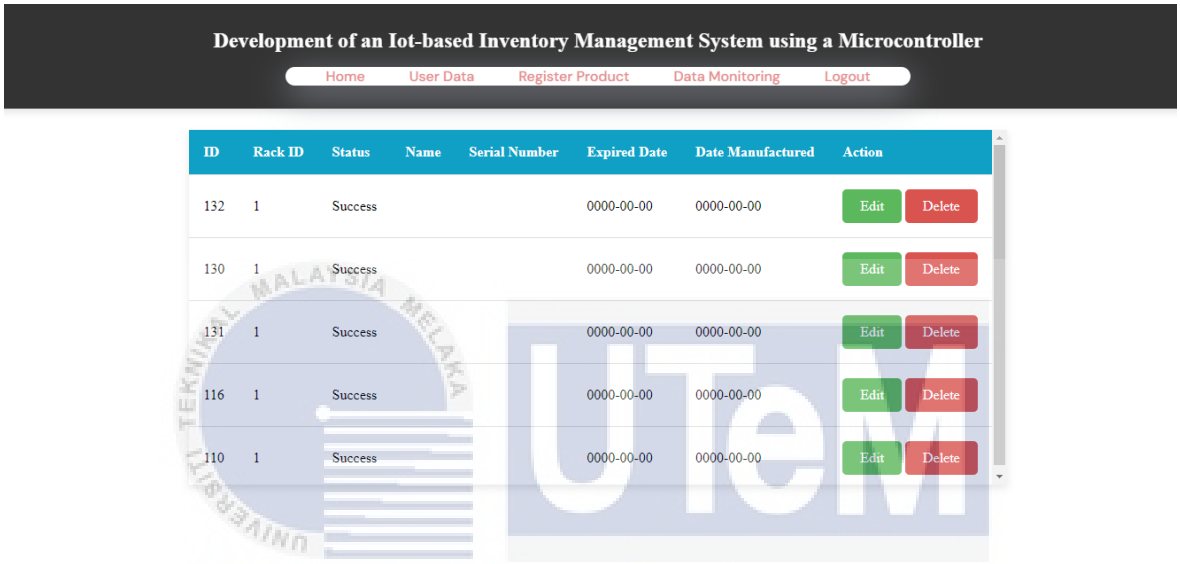


Figure 4.20 User Data Table

Figure 4.20 shows the user data table plays a crucial role as an efficient monitoring tool for expiration dates, facilitating the timely removal of items as they approach or exceed the present day. Two additional features, the update and delete buttons, have been incorporated into the system to enhance user control and data accuracy. The update button allows users to modify data in case of slight mistakes or changes. This feature ensures the accuracy of the information stored in the database and provides a user-friendly way to keep data up-to-date. On the other hand, the delete button empowers users to eliminate any unwanted data from the system. This functionality streamlines data management and maintains a clean and organized database. This adaptability to real-time changes in the data contributes significantly to the overall effectiveness of the project.

4.3.1.5 Email

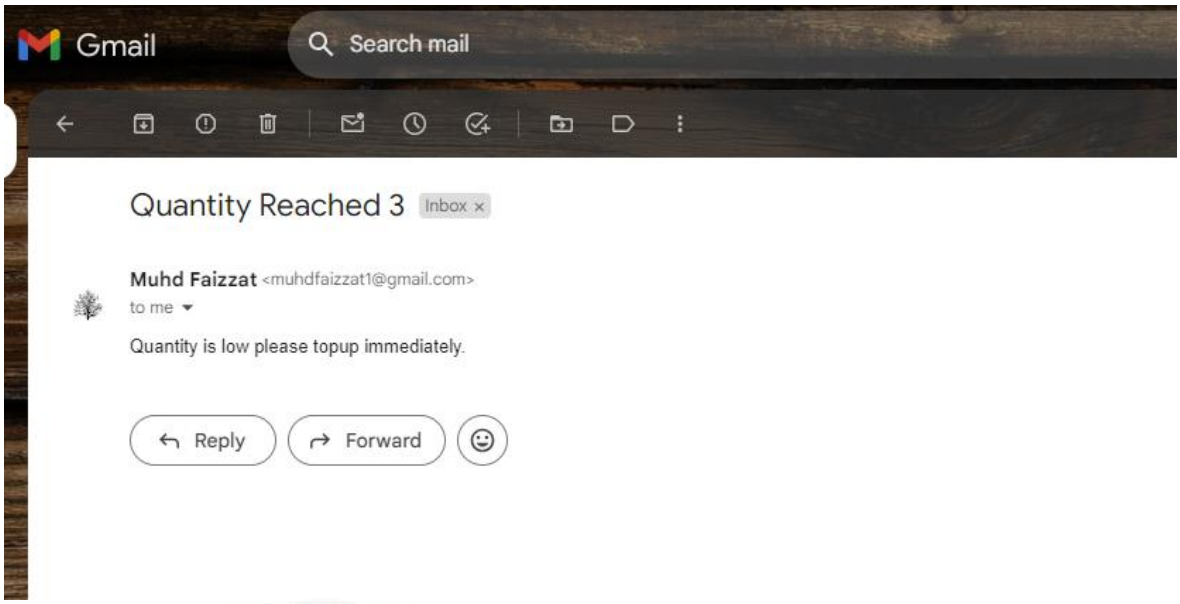


Figure 4.21 The Quantity Threshold Email

When the quantity data meets the condition of equal to three or below, an email alert is sent to the user via Gmail. Utilizing an ESP8266 ultrasonic sensor, the PHP code triggers the email transmission, displaying a concise message: "Quantity is low! Please Top-up immediately! TQ." This setup ensures quick notifications to users, prompting timely action for replenishing items as needed. Figure 4.21 shows the quantity threshold email.

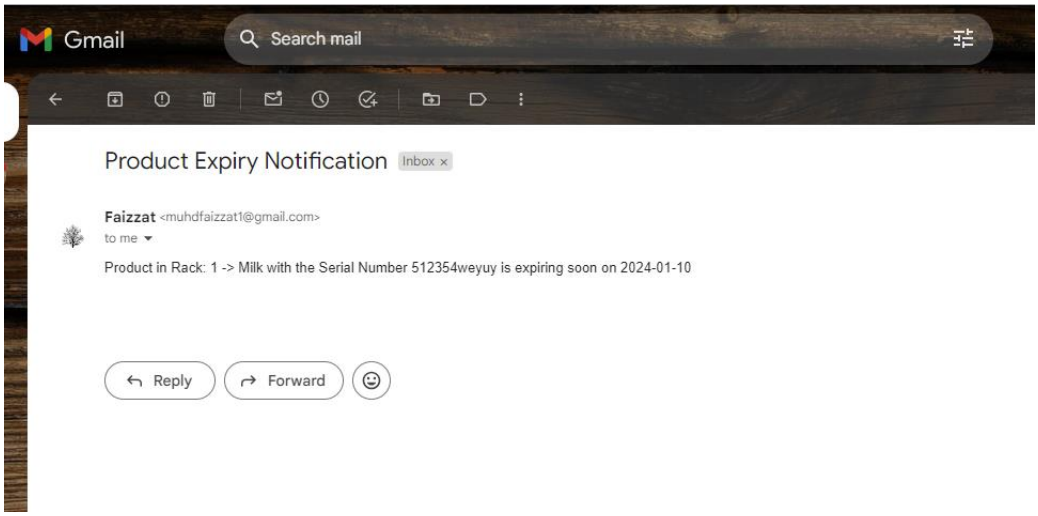


Figure 4.22 The Product Expired Date Notification

When the data meets the condition of reaching the expiration date, the system initiates an automatic email notification to the user. Gmail serves as the email recipient in this scenario. The condition for triggering the alert is seven days before the expiration date. Utilizing a datestamp in the PHP code, the system activates the email notification process a week before the expiration date. The email content is designed to inform the user about the impending expiration, displaying a message such as "Product in Rack: 1 -> Milk with Serial Number “496xxx0yy10zzzz7” is expiring soon on 2024-01-10." This automated email alert ensures that users receive timely notifications, allowing them to take necessary actions in advance of the product's expiration. Figure 4.22 shows the product's expired date notification in Gmail.

4.4 Hardware Result

4.4.1 Final Project Design

Figure 4.23 Final Project Design

The project's design incorporates wood, acrylic paper, and a clear container dispenser. Wood provides structural support and a natural aesthetic, while acrylic paper contributes to a visually appealing and customizable surface. The dimension is 15cm x 15cm x 36cm for this final prototype design. The clear container dispenser enhances the project's functionality by allowing easy visibility and monitoring of stored items. Figure 4.23 shows the final project design.

4.5 Data Analysis

This chapter initiates the exploration of data analysis, encompassing survey methodology and examining subsequent results. It delves into the processes of data collection and the methods employed to extract meaningful insights. Adopting a comprehensive outlook aims to unveil patterns and correlations within survey results, providing valuable input for research, decision-making, and strategic planning. This chapter establishes the foundation for a comprehensive exploration, addressing survey design intricacies and nuanced data interpretation to enhance the overall understanding.

4.5.1 Survey Analysis

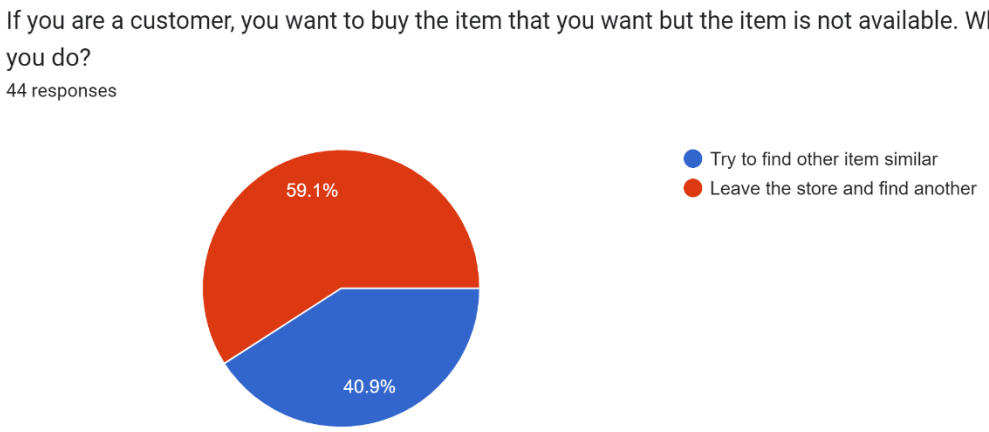


Figure 4.24 Item is not available.

The survey revealed that 59.1% of respondents prefer seeking alternatives when faced with unavailability, emphasizing the importance of offering a diverse range of similar items. Additionally, 40.9% indicated a tendency to leave the store and search elsewhere if their desired item is not in stock, highlighting the potential impact on sales and customer loyalty. These findings underscore the critical role of effective inventory management in influencing customer satisfaction and retention, as product availability significantly shapes purchasing decisions. Figure 4.24 illustrates a pie chart depicting respondents' feedback regarding the unavailability of certain items.

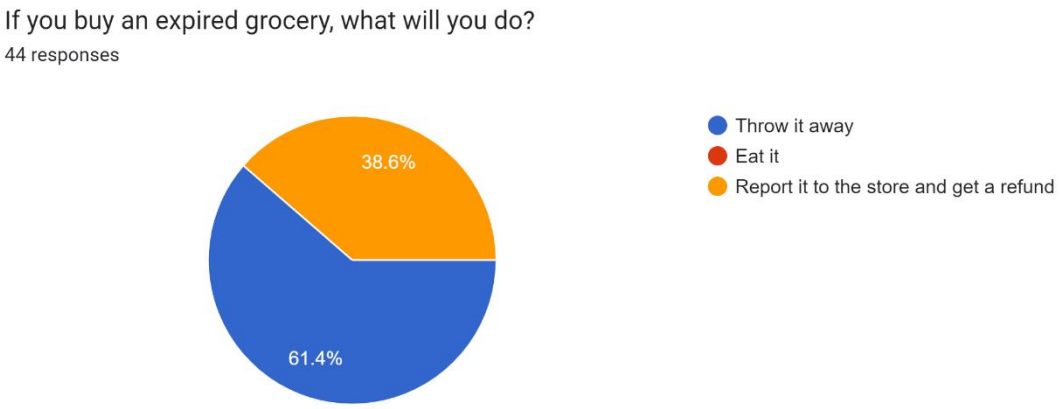


Figure 4.25 Expired Grocery

A significant majority, 61.4%, indicated a commitment to food safety by discarding expired grocery items. Notably, none were willing to consume expired products, underscoring a strong aversion to such items. Additionally, 38.6% expressed an intention to report expired items to the store, seeking a refund or replacement. This emphasizes the importance of accurate expiration date labelling and responsive customer support in grocery retail, ensuring a positive and safe consumer experience. Figure 4.25 presents a pie chart reflecting respondents' perspectives on expired grocery items.

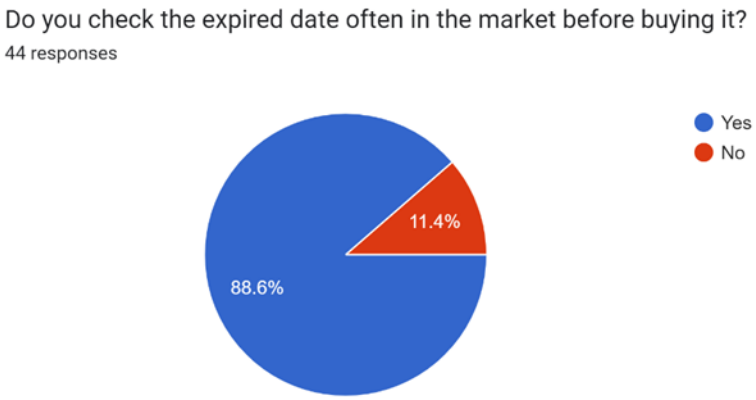


Figure 4.26 Check the expiration date at the market

The survey reveals that 88.6% of respondents actively check the expiration date on food and grocery items before purchase, indicating a strong commitment to food safety practices. A smaller but notable 11.4% reported not checking, prompting further exploration into factors such as trust in the retailer or a lack of awareness. This insight offers valuable considerations for grocery retailers to enhance overall consumer awareness and ensure continued vigilance over expiration dates, a crucial aspect of food management. Figure 4.26 displays respondent feedback concerning the verification of expiration dates.

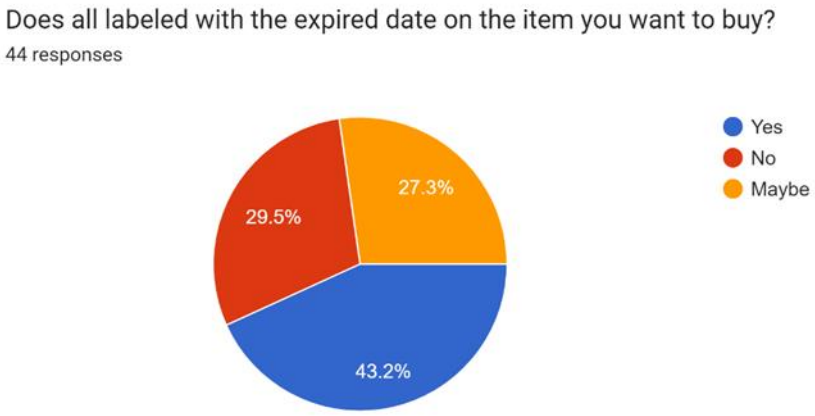


Figure 4.27 Labeled expired date

The survey indicates that 43.2% of respondents consistently find expiration date labels on items they wish to buy, showcasing positive labelling practices. However, 29.5% rarely or never find such labels, revealing potential inconsistencies in retail labelling. A significant minority, 27.3%, experiences uncertainty in label visibility, indicating room for improvement in ensuring a more predictable shopping experience. These insights highlight areas for grocery retailers to enhance labelling practices and promote a safer and more informed consumer experience. Figure 4.27 illustrates respondent alerts regarding labelled expiration dates.

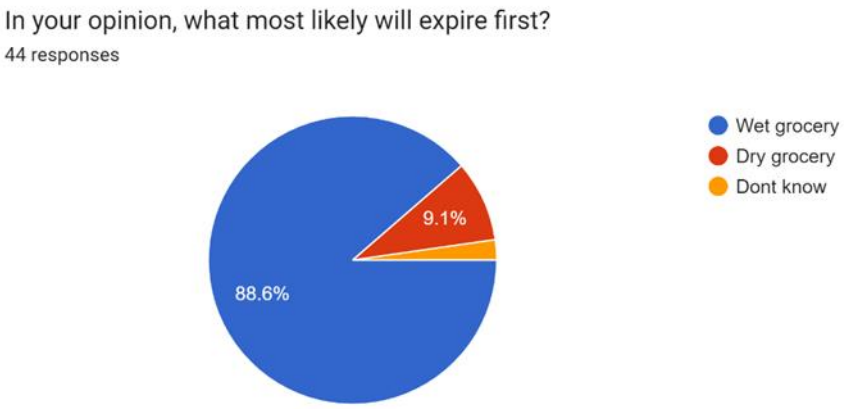


Figure 4.28 Expired item

The survey indicates that 88.6% of respondents believe wet grocery items like fresh produce and dairy are most likely to expire first, aligning with the common understanding that perishables have a shorter shelf life. In contrast, 9.1% think dry grocery items, such as packaged goods, might expire sooner, possibly influenced by packaging or interpretation of expiration. A small 2.3% expresses uncertainty, highlighting the need for consumer education on product shelf life. This nuanced understanding underscores the importance of tailored communication and inventory management for wet and dry grocery items. Figure 4.28 depicts respondents' opinions on which item will most likely expire first.

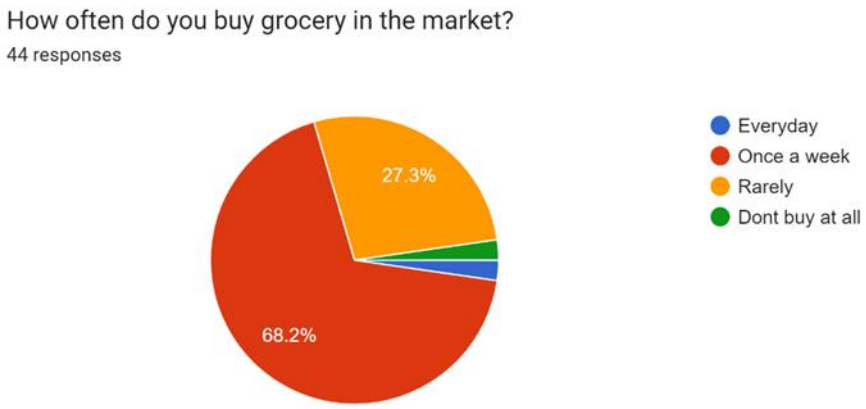


Figure 4.29 Buying in the market

The survey shows that 68.2% of respondents shop for groceries once a week, offering retailers insights for effective inventory planning and promotions. A notable 27.3% shop rarely, prompting the exploration of targeted initiatives to encourage more frequent visits. A small 2.3% don't buy groceries at all, and understanding their reasons provides valuable insights for retailers to tailor strategies to the diverse shopping habits of their customer base. Figure 4.29 depicts the buying routines of individual respondents.

4.5.2 Performance and Functionality Test

4.5.2.1 The accuracy of ultrasonic and infrared sensors is determined by comparing the value data.

Ultrasonic Sensor Testing

Table 4.1 Ultrasonic Sensor Testing Table

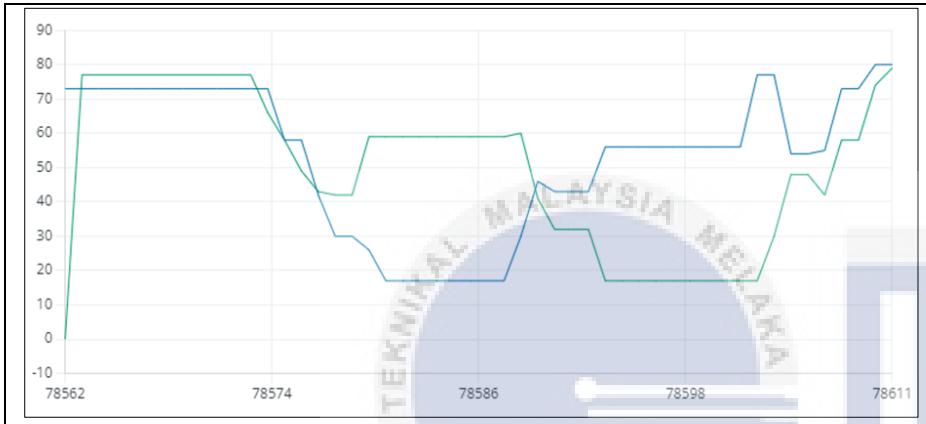
Condition	1st Test	2nd Test	3rd Test	4th Test	5th Test	Quantity
distance >=77	82 cm	80 cm	79cm	90cm	120cm	0
distance <= 76 && distance >= 60	70 cm	72 cm	69cm	65cm	67cm	1
distance <= 59 && distance >= 48	56 cm	58cm	59cm	45cm	48cm	2
distance <= 47 && distance >= 33	47 cm	40 cm	33cm	45cm	39cm	3
distance <= 32 && distance >= 21	32 cm	30 cm	25cm	27cm	31cm	4
distance <=20	17 cm	15 cm	16cm	10cm	5cm	5

Analyzing the provided table unveils a meticulous assessment of distances conducted through five tests. All are carried out under the same conditions. These tests are strategically designed to evaluate distances within specific criteria, systematically categorising six distinct ranges. These categories span measurements equal to or greater than 77 cm and those equal to or less than 20 cm. Upon a detailed examination of the results, a consistent trend emerges for each recorded distance within a particular category that impeccably aligns with the anticipated conditions, resulting in an impressive 90% accuracy across all five tests. This consistent alignment underscores the dependability and precision

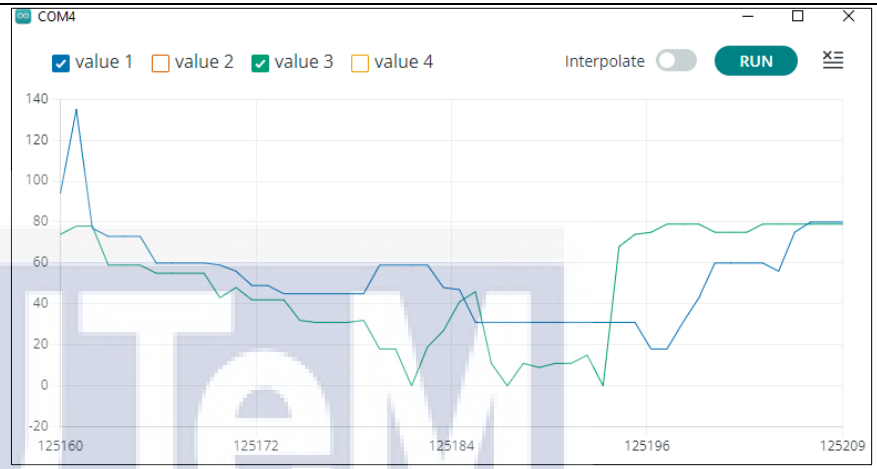
of the testing methodology, showcasing its effectiveness in consistently classifying distances within the specified criteria under the same conditions.

The results derived from these five tests under consistent conditions highlight the accuracy and reliability of the testing methodology in consistently evaluating distances based on predefined criteria. Including additional metrics like precision and recall further enriches the evaluation, providing a more comprehensive perspective on the overall performance and effectiveness of the testing approach. Table 4.1 presents the results of five tests of the ultrasonic sensor.

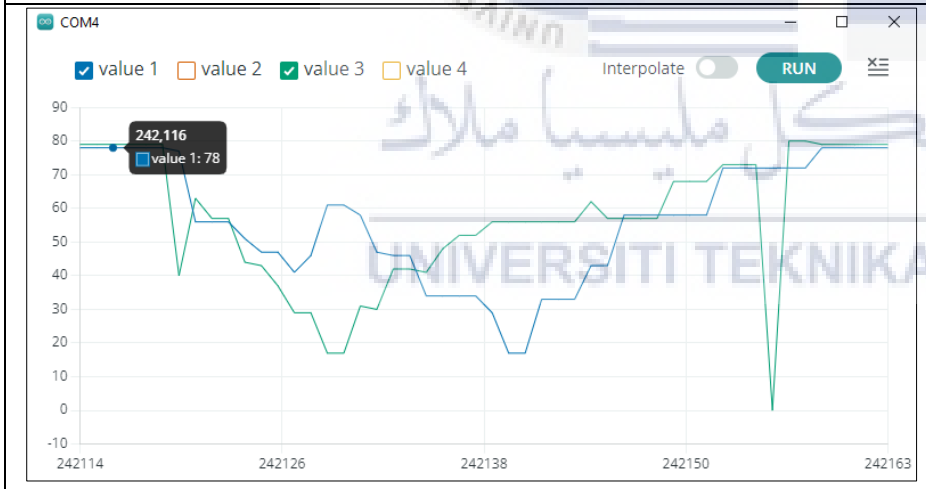
Table 4.2 Ultrasonic Sensor Distance Graph Testing



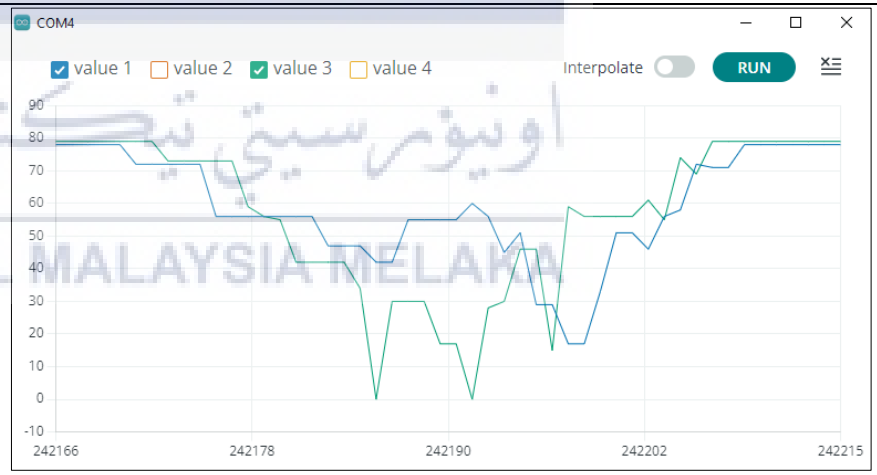
1st Testing



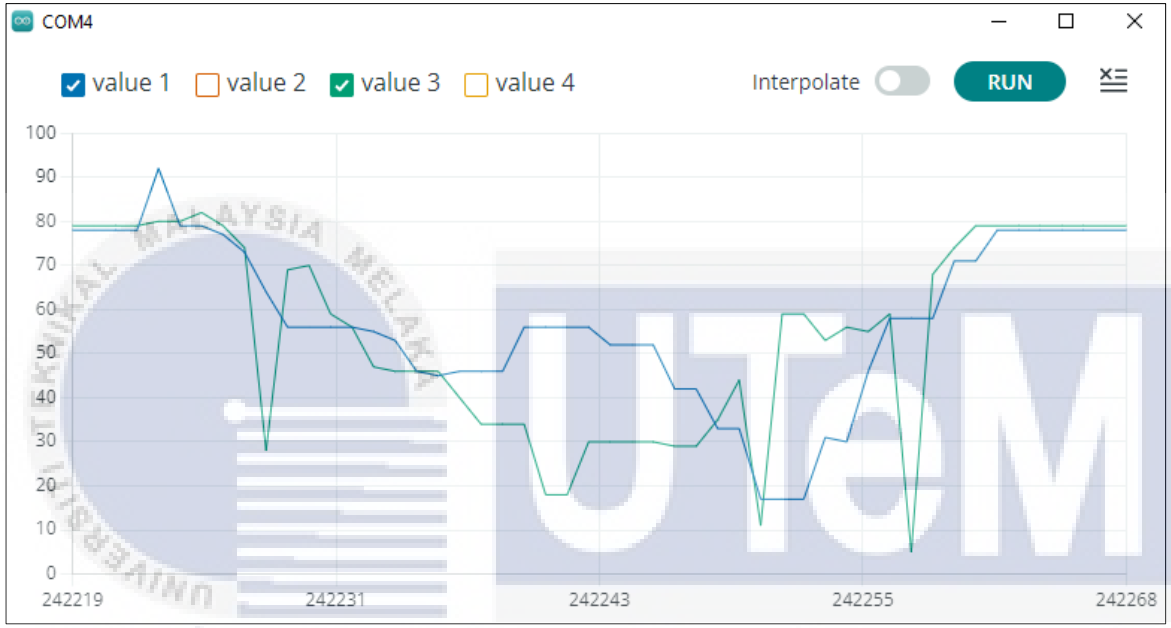
2nd Testing



3rd Testing

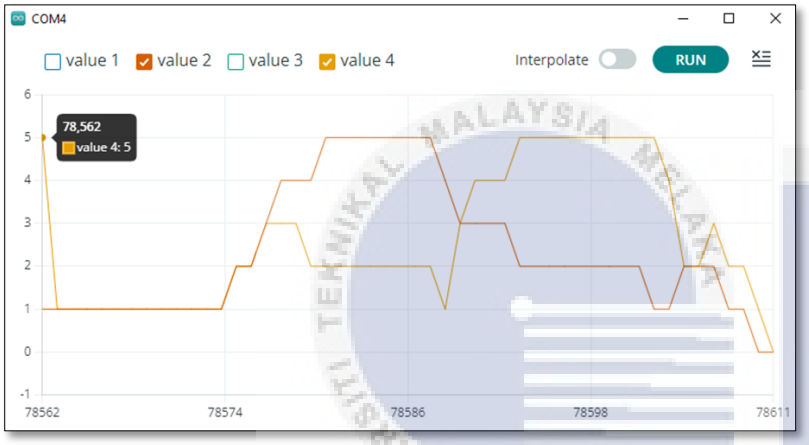


4th Testing

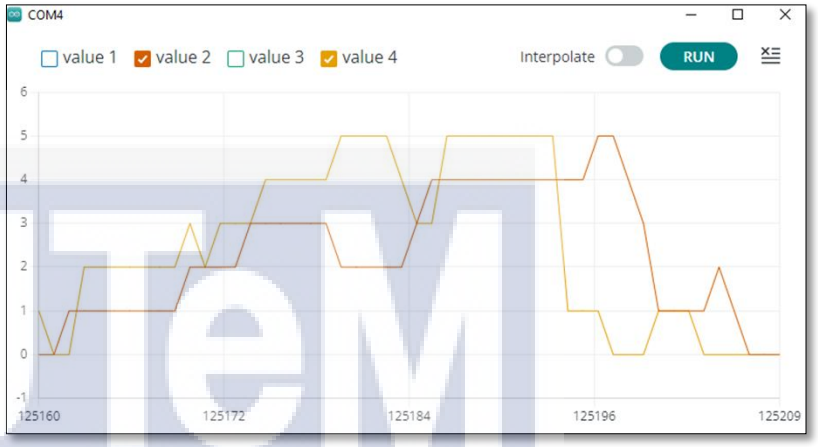


5th Testing

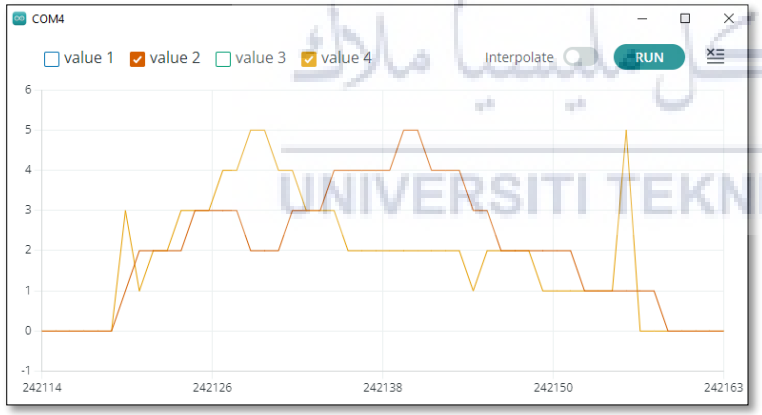
Table 4.3 Ultrasonic Sensor Quantity Graf



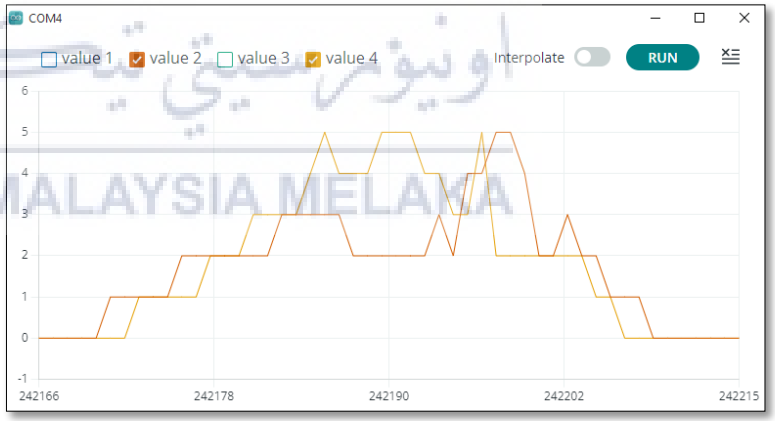
1st Testing



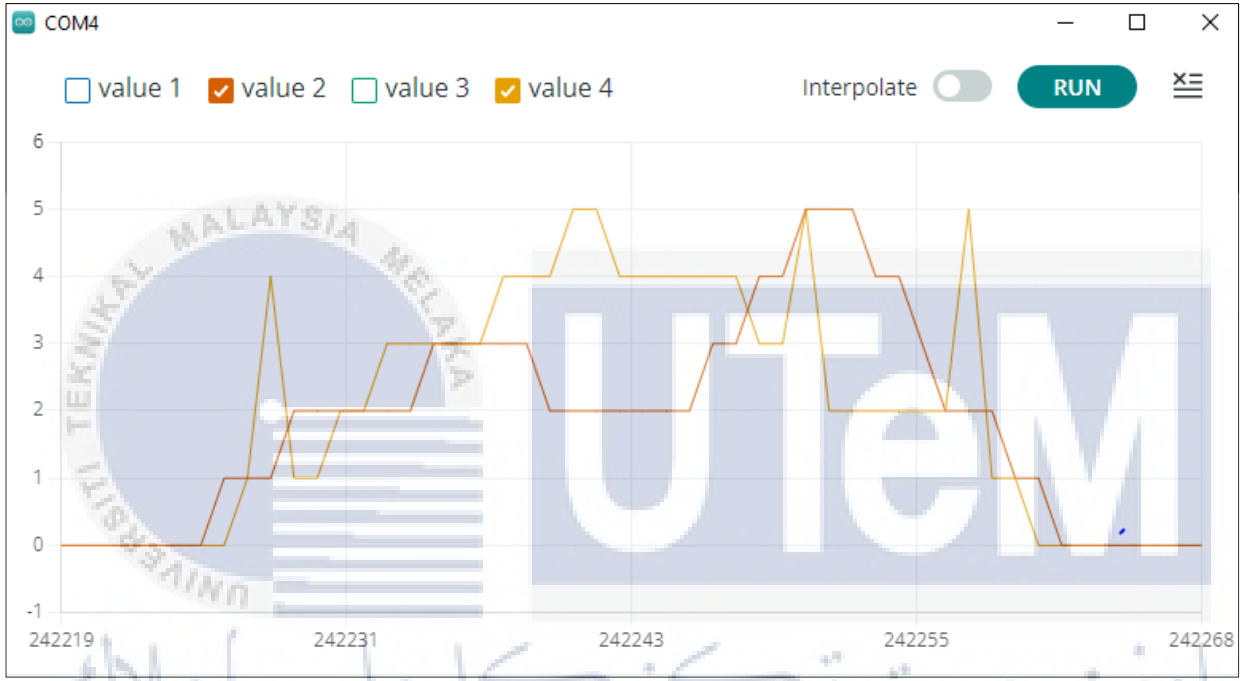
2nd Testing



3rd Testing



4th Testing



5th Testing

The visually generated graph captures the outcomes of five distinct tests performed on the ultrasonic sensor. Each data point on the graph represents a measured distance intricately linked with its corresponding quantity, bearing in mind that these quantities are restricted to a maximum value of 5. Noteworthy is the observation of intermittent errors in the sensor's performance, where instances arise in which readings are not successfully registered or recorded values exceed the anticipated quantity limit. These irregularities inject a degree of variability into the accuracy of the measurements, signalling potential challenges faced during the testing phases. Table 4.2 shows the ultrasonic sensor distance graph testing.

An in-depth scrutiny of the graph in isolation from the table enables a focused exploration of the sensor's conduct across the five tests, with specific attention directed towards the prescribed quantity limit of 5. Table 4.3 ultrasonic sensor quantity graf testing. The data paints a nuanced portrait of the sensor's behaviour, underscoring the imperative to address occasional discrepancies and refine the accuracy, particularly within the defined constraints of quantity. This recognition prompts a strategic approach to enhance the sensor's reliability and precision, which is crucial for optimal performance in diverse testing scenarios.

Infrared Sensor Accuracy

Table 4.4 Infrared Sensor 1 (Rack ID 1)

Test Number	Status	Sensor State
1	Success	HIGH
2	Success	HIGH
3	Failed	LOW
4	Success	HIGH
5	Success	HIGH
Result	Success	80% SUCCESS

Table 4.5 Infrared Sensor 2 (Rack ID 2)

Test Number	Status	Sensor State
1	Success	HIGH
2	Success	HIGH
3	Failed	LOW
4	Success	HIGH
5	Failed	LOW
Result	Success	60% SUCCESS

Based on Tables 4.4 and 4.5, when comparing the Infrared Sensor (IR 1) and Infrared Sensor (IR 2), it's clear that IR 1 performed better in the tests. IR 1 had 80% success, meaning it got the right results in 4 out of 5 tests. On the other hand, IR 2 had a success rate of 60%, getting the correct results in only 3 out of 5 tests.

Database connection between NodeMCU ESP8266

Table 4.6 Database Connection between ESP8266

Test Number	Connection(Payload)	Connection ESP8266 WiFi
1	SUCCESS	SUCCESS
2	SUCCESS	SUCCESS
3	SUCCESS	SUCCESS
4	SUCCESS	SUCCESS
5	SUCCESS	SUCCESS
Result	SUCCESS	SUCCESS

Table 4.6 confirms the successful completion of the connection between the database and the ESP8266 WiFi module.

4.6 Summary

In conclusion, the project documentation delves into Chapter 4, offering an exhaustive exploration of the Smart Inventory Management System. The chapter comprehensively details the system design, project integration, hardware development, and data analysis components. Within the system design, the expansion of the foundational database with user-friendly interfaces and login functionalities is highlighted, fostering enhanced security and personalized user experiences. The project integration phase is meticulously examined, emphasizing the intricate connection between the ESP8266 hardware and the database, the setup of the MySQL database, and the implementation of various interfaces to manage user data, product registration, and real-time data monitoring. The hardware development section presents the final project design, incorporating structural support and visual appeal materials.

Moreover, the data analysis segment provides valuable insights into customer behaviours and preferences through survey results, shedding light on crucial aspects of grocery shopping. Performance and functionality tests scrutinize the accuracy of ultrasonic and infrared sensors, demonstrating their reliability and showcasing potential areas for improvement. The successful connection between NodeMCU ESP8266 and the database ensures consistent and reliable data transmission. Chapter 4 encapsulates a thorough exploration of the system's various facets, paving the way for informed decision-making regarding system modifications to optimize inventory accuracy, order fulfilment, and operational efficiency.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In conclusion, the development and implementation of an innovative IoT-based Smart Inventory Management System, seamlessly integrated with a microcontroller, represent a significant stride in addressing challenges inherent in traditional inventory control methodologies. With a dedicated emphasis on precisely tracking both quantity and expiration dates, this system sets itself apart for its potential to transform retail operations. By highlighting existing inefficiencies and errors in manual inventory systems, the study underscores the pivotal role of advanced technologies in enhancing precision and efficiency.

The proposed system's incorporating of ultrasonic sensors and a sophisticated alert mechanism offers a proactive solution to issues like stock depletion and impending expiration dates. Beyond mitigating the risk of stockouts and preventing product obsolescence, it substantially improves overall inventory accuracy. The implementation of an IoT infrastructure not only enables remote monitoring but also streamlines inventory control processes, reducing dependence on physical presence.

Highlighting the paramount importance of integrating technological advancements in inventory management, this conclusion urges for a shift towards practices that are not only more efficient but also more precise. The concrete benefits, ranging from heightened product availability to reduced waste and improved operational efficiency, position this cutting-edge system as a promising solution for the ever-evolving landscape of retail operations. The progress achieved in this research sets the stage for future innovations in inventory management, pointing towards a positive trajectory for the industry.

5.2 Potential for Commercialization

Inventory management projects are poised for significant commercialization, driven by the escalating demand for precise inventory data monitoring. This technology finds versatile applications across diverse sectors, including retail, manufacturing, healthcare, and logistics. In retail, inventory management systems are pivotal for optimizing stock levels, preventing stockouts, and enhancing supply chain efficiency. Manufacturing processes benefit from these systems by ensuring timely access to raw materials, reducing excess inventory, and streamlining production. Healthcare relies on efficient inventory management to maintain adequate medical supplies, medications, and equipment, ultimately improving patient care. Logistics and distribution companies leverage accurate inventory data for optimizing warehouse operations, tracking shipments, and improving order fulfilment. However, it is crucial to balance the potential for commercialization and the imperative to ensure that inventory data is accessible and usable for businesses of all sizes. This inclusivity is vital for addressing the diverse needs of various enterprises, fostering widespread benefits across industries. Additionally, prioritizing user-friendly interfaces and customizable features enhances usability, making these inventory management solutions adaptable to specific business requirements. Striking this balance ensures that the advantages of inventory management technology are accessible to a broad spectrum of businesses, promoting operational efficiency and success across different sectors.

5.3 Future Works

For future improvements, the accuracy of the Smart Inventory Management System estimation results could be enhanced as follows:

i) Implement machine learning algorithms to enhance data quality and accuracy over time.

ii) Implement automated data validation checks within the system. These checks can include range validations, logic checks, and consistency checks to ensure that incoming sensor data aligns with predefined criteria, reducing the likelihood of inaccuracies.

iii) Develop error-handling algorithms to identify and address anomalies in sensor data.

REFERENCES

[1] Kaleigh Moore, "Inventory Shortage: Causes, Tips + the Impact on Retail (2023)," *Industry News & Trends*, 2023. https://www.shopify.com/my/retail/inventory-shortages

[2] Bernama, "Selangor spot checks find 406 damaged, expired food items still on shelves," 2022. https://www.nst.com.my/news/nation/2022/12/864059/selangor-spot-checks-find-406-damaged-expired-food-items-still-shelves

[3] A. Alwadi, A. Gawanmeh, S. Parvin, and J. N. Al-Karaki, "Smart Solutions for RFID based Inventory Management Systems: A Survey," *Scalable Comput. Pract. Exp.*, vol. 18, no. 4, pp. 347–360, Nov. 2017, doi: 10.12694/scpe.v18i4.1333.

[4] Alaybeyoglu et al., "IoT Based Inventory Management System using NodeMCU and Ultrasonic Sensor", Accessed: May 30, 2023. [Online]. Available: https://iotdesignpro.com/projects/iot-based-inventory-management-system-using-load-cell-and-nodemcu

[5] Sharath. S.Rao & Dan, "IoT Based Inventory Management System using NodeMCU and Ultrasonic Sensor", Accessed: May 30, 2023. [Online]. Available: https://iotdesignpro.com/projects/smart-inventory-management-system-using-iot

[6] Anonim, "Apa Itu Barcode Scanner," 2012, [Online]. Available: http://www.axopos.com/article/apa-itubarcode-scanner--69.html

[7] I. P. A. Putra Yudha, M. Sudarma, and P. Arya Mertasana, "PERANCANGAN APLIKASI SISTEM INVENTORY BARANG MENGGUNAKAN BARCODE SCANNER BERBASIS ANDROID," *J. SPEKTRUM*, vol. 4, no. 2, 2018, doi: 10.24843/spektrum.2017.v04.i02.p10.

[8] S. Palanisamy and S. N, "Sensor Based Industrial Kitchen Foodstuffs Monitoring System," *Int. J. Comput. Commun. Informatics*, vol. 3, no. 1, 2021, doi: 10.34256/ijcci2113.

[9] R. H. Sudhan, M. G. Kumar, A. U. Prakash, S. A. R. Devi, and S. P., "ARDUINO ATMEGA-328 MICROCONTROLLER," *IJIREEICE*, vol. 3, no. 4, 2015, doi: 10.17148/ijireeice.2015.3406.

[10] Y. Parihar, Sing, "Internet of Things and Nodemcu: A review of use of Nodemcu ESP8266 in IoT products," *J. Emerg. Technol. Innov. Res.*, vol. 6, no. 6, 2019.

[11] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud*

University - Computer and Information Sciences, vol. 30, no. 3. 2018. doi: 10.1016/j.jksuci.2016.10.003.

[12] S. Paul, A. Chatterjee, and D. Guha, "Study of Smart Inventory Management System Based on the Internet of Things (Iot)," *IJRTBT Int. J. Recent Trends Bus. Tour.* |, vol. 3, no. 3, 2019.

[13] P. Saha and M. A. Alam, "Smart Management Scheme for the Efficient Control of Industrial Inventory," *Am. J. Ind. Bus. Manag.*, vol. 12, no. 04, 2022, doi: 10.4236/ajibm.2022.124028.

[14] M. C. Chen, Y. T. Cheng, and C. Y. Siang, "Development of Inventory Management System Based on Radio Frequency Identification Technology," *Sensors Mater.*, vol. 34, no. 3, pp. 1163–1177, 2022, doi: 10.18494/SAM3497.

[15] Isha, A. K. Luhach, and R. Sobti, "Smart inventory system scenario based on internet of things," in *Far East Journal of Electronics and Communications*, 2016. doi: 10.17654/ECSV3PII16565.

[16] L. Li, "Simulation Design of Data Acquisition System Based on Proteus," *J. Electr. Power*, vol. 6, pp. 226–227, 2009.

[17] J.-J. Wang, H.-C., Zhang, "Proteus-based microcontroller multi-channel temperature acquisition system simulation design," *J. Shaanxi Univ. Sci. Technol.*, 2007.

[18] M. Fezari and A. Al Dahoud, "Integrated Development Environment ' IDE ' For Arduino," *ResearchGate*, no. October, 2018.

[19] T. Butler, "PHP & MySQL: Novice to Ninja," *Sitepoint*, 2022.

[20] & et al. . Veza, O., "Sat Nusapersada Batam: Business management information system at the cantine of PT," *Sat Nusaersada Batam. Eng. Technol. Int. J.*, vol. 2, no. 1, pp. 55–69, 2020.

[21] Y. S. Parihar, "Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products," *JETIR1907U33 J. Emerg. Technol. Innov. Res.* , vol. 6, no. 6, 2019.

[22] J. M. M. M. G. Wiczyn´ ski, "Distance estimation with a long range ultrasonic sensor system," *IEEE Sens.*, vol. 9, no. 7, pp. 767–773, 2009.

[23] C. Cheng, Q. Sha, B. He, and G. Li, "Path planning and obstacle avoidance for AUV: A review," *Ocean Engineering*, vol. 235. 2021. doi: 10.1016/j.oceaneng.2021.109355.

APPENDICES

Appendix 1 Turnitin Report Result

DEVELOPMENT OF AN IOT-BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER

ORIGINALITY REPORT

12%	10%	6%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	sietjournals.com Internet Source	1%
2	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	1%
3	scpe.org Internet Source	1%
4	www.researchgate.net Internet Source	1%
5	Submitted to University of New Haven Student Paper	1%
6	Palash Saha, Md. Ariful Alam. "Smart Management Scheme for the Efficient Control of Industrial Inventory", American Journal of Industrial and Business Management, 2022 Publication	1%
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%

Appendix 2 Ultrasonic Sensor Arduino IDE Code

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char *ssid = "POCO F5";
const char *password = "Fairy753159";
const char *phpServer = "127.0.1.1"; // Change to your server IP
const char *phpPage = "/real/datamonitoring.php";

const int trigPin1 = D1; // Replace with your actual pin
const int echoPin1 = D8; // Replace with your actual pin
const int trigPin2 = D2; // Replace with your actual pin
const int echoPin2 = D6; // Replace with your actual pin

WiFiClient client; // Create a WiFiClient object

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    int quantity1 = getUltrasonicDistance1();
    int quantity2 = getUltrasonicDistance2();
    sendQuantityToServer(quantity1,quantity2);
    delay(10000); // Adjust the delay based on your requirements
  }
}

int getUltrasonicDistance1() {
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
```

```
  delayMicroseconds(10);
  long duration = pulseIn(echoPin1, HIGH);
  int distance = duration / 17;
  Serial.print("Distance1: ");
  Serial.print(distance);
  Serial.println(" cm");

 // Define quantity ranges based on a 24 cm reference distance
if (distance >=77){
  return 0; //Quantity 0
}
  else if (distance <= 76 && distance >= 60) {
  return 1; // Quantity 1
} else if (distance <= 59 && distance >= 48) {
  return 2; // Quantity 2
} else if (distance <= 47 && distance >= 33) {
  return 3; // Quantity 3
} else if (distance <= 32 && distance >= 21) {
  return 4; // Quantity 4
} else {
  return 5; // Quantity 5
}
}

int getUltrasonicDistance2() {
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
 long duration = pulseIn(echoPin2, HIGH);
  int distance = duration /17;
  Serial.print("Distance2: ");
  Serial.print(distance);
  Serial.println(" cm");
// Define quantity ranges based on a 24 cm reference distance
if (distance >=77){
  return 0; //Quantity 0
}
  else if (distance <= 76 && distance >= 60) {
  return 1; // Quantity 1
} else if (distance <= 59 && distance >= 48) {
  return 2; // Quantity 2
} else if (distance <= 47 && distance >= 33) {
  return 3; // Quantity 3
} else if (distance <= 32 && distance >= 21) {
  return 4; // Quantity 4
} else {
  return 5; // Quantity 5
```

```
}
}

void sendQuantityToServer(int quantity1,int quantity2) {
 HTTPClient http;

 http.begin(client, "http://192.168.186.177/real/datamonitoring.php"); //
Use the WiFiClient instance for HTTPClient
 http.addHeader("Content-Type", "application/x-www-form-urlencoded");
//Specify content-type header
 String postData = "quantity1=" + String(quantity1)+ "&quantity2=" +
String(quantity2);

int httpCode = http.POST(postData);
Serial.println("POST data: " + postData);

String payload = http.getString();

Serial.println("Data : ");
Serial.println("httpCode : ");Serial.println(httpCode);
Serial.println("payload : ");Serial.println(payload);

if (httpCode > 0) {
    Serial.printf("HTTP Code: %d\n", httpCode);
  } else {
    Serial.printf("HTTP Error: %s\n",
http.errorToString(httpCode).c_str());
  }

  http.end();
}
```

Appendix 3 Infrared Sensor Arduino IDE Code

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char* ssid = "POCO F5";
const char* password = "Fairy753159";
const char *phpServer = "192.168.245.177"; // Change to your server IP
const char *phpPage = "/gg/index.php";

const int irPin1 = D7; // IR sensor 1 pin
const int irPin2 = D5; // IR sensor 2 pin

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  pinMode(irPin1, INPUT);
  pinMode(irPin2, INPUT);
}

void loop() {
  int irValue1 = digitalRead(irPin1);
  int irValue2 = digitalRead(irPin2);

  if (irValue1 == LOW || irValue2 == LOW) {
    String postData = "rack_id=";
    if (irValue1 == LOW) {
      postData += "1";
    } else {
      postData += "2";
    }

    postData += "&status=";
    postData += (irValue1 == LOW) ? "Success" : "Success";

    if (sendData(postData)) {
      Serial.println("Data sent successfully");
    } else {
      Serial.println("Error sending data");
```

```
    }

    delay(10000); // Adjust the delay according to your needs
  }
}

bool sendData(String data) {
  Serial.println("Sending data: " + data);
  Serial.println("Connecting to server...");
  WiFiClient client;
  HTTPClient http;

  // Construct the full URL, including the PHP page
  String url = "http://" + String(phpServer) + String(phpPage);
  http.begin(client, url);

  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  int httpResponseCode = http.POST(data);
  Serial.println("POST data: " + data);

  Serial.print("HTTP Response code: ");
  Serial.println(httpResponseCode);

  if (httpResponseCode == 200) {
    Serial.println("Request successful");
  } else {
    Serial.println("Error in HTTP request");
  }

  http.end();
  return httpResponseCode == 200;
}
```

Appendix 4 Connection Database PHP Code

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

Appendix 5 Login Form PHP Code

```
<?php
session_start();
if (isset($_SESSION["user"])) {
   header("Location: index.php");
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.
css" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2 style="text-align: center;">Welcome to Development of an Iot-based
Inventory Management System using a Microcontroller</h2>
<br>
<br>
<br>
<br>
<br>
<div class="container">
        <?php
        if (isset($_POST["login"])) {
           $email = $_POST["email"];
           $password = $_POST["password"];
            require_once "connection.php";
            $sql = "SELECT * FROM users WHERE email = '$email'";
            $result = mysqli_query($conn, $sql);
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
            if ($user) {
                if (password_verify($password, $user["password"])) {
                    session_start();
                    $_SESSION["user"] = "yes";
                    header("Location: home.php");
                    die();
                }else{
                    echo "<div class='alert alert-danger'>Password does
not match</div>";
```

```
                }
            }else{
                echo "<div class='alert alert-danger'>Email does not
match</div>";
            }
        }
        ?>
      <form action="login.php" method="post">
        <div class="form-group">
            <input type="email" placeholder="Enter Email:" name="email"
class="form-control">
        </div>
        <div class="form-group">
            <input type="password" placeholder="Enter Password:"
name="password" class="form-control">
        </div>
        <div class="form-btn">
            <input type="submit" value="Login" name="login" class="btn
btn-primary">
        </div>
      </form>
     <div><p>Not registered yet <a href="registration.php">Register
Here</a></p></div>
    </div>
</body>
</html>
```

Appendix 6 Registration User PHP Code

```
<?php
session_start();
if (isset($_SESSION["user"])) {
   header("Location: index.php");
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.
css" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2 style="text-align: center;">Welcome to Development of an Iot-based
Inventory Management System using a Microcontroller</h2>
<br>
<br>
<br>
<br>
<br>
    <div class="container">
        <?php
        if (isset($_POST["submit"])) {
           $fullName = $_POST["fullname"];
           $email = $_POST["email"];
           $password = $_POST["password"];
           $passwordRepeat = $_POST["repeat_password"];

           $passwordHash = password_hash($password, PASSWORD_DEFAULT);

           $errors = array();

           if (empty($fullName) OR empty($email) OR empty($password) OR
empty($passwordRepeat)) {
            array_push($errors,"All fields are required");
           }
           if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            array_push($errors, "Email is not valid");
```

```
            }
            if (strlen($password)<8) {
             array_push($errors,"Password must be at least 8 charactes
long");
            }
            if ($password!==$passwordRepeat) {
             array_push($errors,"Password does not match");
            }
            require_once "connection.php";
            $sql = "SELECT * FROM users WHERE email = '$email'";
            $result = mysqli_query($conn, $sql);
            $rowCount = mysqli_num_rows($result);
            if ($rowCount>0) {
             array_push($errors,"Email already exists!");
            }
            if (count($errors)>0) {
             foreach ($errors as  $error) {
                 echo "<div class='alert alert-danger'>$error</div>";
             }
            }else{

             $sql = "INSERT INTO users (full_name, email, password) VALUES
( ?, ?, ? )";
             $stmt = mysqli_stmt_init($conn);
             $prepareStmt = mysqli_stmt_prepare($stmt,$sql);
             if ($prepareStmt) {
                 mysqli_stmt_bind_param($stmt,"sss",$fullName, $email,
$passwordHash);
                 mysqli_stmt_execute($stmt);
                 echo "<div class='alert alert-success'>You are registered
successfully.</div>";
             }else{
                 die("Something went wrong");
             }
            }


        }
        ?>
        <form action="registration.php" method="post">
            <div class="form-group">
                <input type="text" class="form-control" name="fullname"
placeholder="Full Name:">
            </div>
            <div class="form-group">
                <input type="emamil" class="form-control" name="email"
placeholder="Email:">
            </div>
```

```
            <div class="form-group">
                <input type="password" class="form-control"
name="password" placeholder="Password:">
            </div>
            <div class="form-group">
                <input type="password" class="form-control"
name="repeat_password" placeholder="Repeat Password:">
            </div>
            <div class="form-btn">
                <input type="submit" class="btn btn-primary"
value="Register" name="submit">
            </div>
        </form>
        <div>
        <div><p>Already Registered <a href="login.php">Login
Here</a></p></div>
      </div>
    </div>
</body>
</html>
```

Appendix 7 Homepage PHP Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=2.0">
    <title>Home</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="navbar.css"> <!-- Add the link to the new
CSS file -->

    <title>Home : Development of an Iot-based Inventory Management System
using a Microcontroller</title>
</head>
<body>
    <div class="navbar">
        <h2>Development of an Iot-based Inventory Management System using
a Microcontroller</h2>
        <ul class="nav">
            <li><a href="home.php" class="nav-item is-active" active-
color="orange">Home</a></li>
            <li><a href="userdata.php" class="nav-item" active-
color="green">User Data</a></li>
            <li><a href="registerproduct.php" class="nav-item" active-
color="blue">Register Product</a></li>
            <li><a href="datamonitoring.php" class="nav-item" active-
color="red">Data Monitoring</a></li>
            <li><a href="logout.php" class="nav-item" active-
color="rebeccapurple">Logout</a></li>
            <span class="nav-indicator"></span>
        </ul>
    </div>

    <div class="content" style="text-align: center;">
    <br>
    <img src="home.jpg" alt="" style="width:60%; display: block; margin: 0
auto;">
</div>
</body>
</html>
```

Appendix 8 User Data Table PHP Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Data</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="navbar.css">
    <style>
    .navbar {
        margin-bottom: 20px; /* Adjust the margin */
    }

    .container {
        max-width: 1200px; /* Increase the max-width to make the container
wider */
        margin: 20px auto; /* Adjust the margin */
        padding: 20px;
    }

    .table-container {
            max-height: 400px;
            overflow-y: scroll;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            margin-top: 100px; /* Adjust margin-top for more space between
heading and table */
        }

    table {
        width: 100%; /* Make the table take up 100% of the container width
*/
        border-collapse: collapse;
    }

    th, td {
        padding: 16px;
        text-align: left;
        border-bottom: 1px solid #ddd;
    }

    th {
        background-color: #10a0c5;
        color: #fff;
    }

    tr:hover {
```

```
        background-color: #f5f5f5;
    }

    .container {
        max-width: 1200px; /* Increase the max-width to make the container
wider */
        margin: 20px auto; /* Adjust the margin */
        padding: 20px;
    }

    .row {
        margin-bottom: 30px; /* Adjust the margin-bottom to move the table
down */
    }

    .btn {
        display: inline-block;
        padding: 10px 20px;
        text-align: center;
        text-decoration: none;
        color: #fff;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    .btn-success {
        background-color: #5cb85c;
    }

    .btn-danger {
        background-color: #d9534f;
    }

    .row h3 {
        height: 0vh;
        display: flex;
        align-items: center;
        justify-content: center;
        margin: 0;
    }
    </style>
</head>
<body>
    <div class="navbar">
        <h2>Development of an Iot-based Inventory Management System using
a Microcontroller</h2>
        <ul class="nav">
```

```
            <li><a href="home.php" class="nav-item is-active" active-
color="orange">Home</a></li>
            <li><a href="userdata.php" class="nav-item" active-
color="green">User Data</a></li>
            <li><a href="registerproduct.php" class="nav-item" active-
color="blue">Register Product</a></li>
            <li><a href="datamonitoring.php" class="nav-item" active-
color="red">Data Monitoring</a></li>
            <li><a href="logout.php" class="nav-item" active-
color="rebeccapurple">Logout</a></li>
            <span class="nav-indicator"></span>
        </ul>
    </div>

    <div class="container">
        <div class="row table-container">
            <table>
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Rack ID</th>
                        <th>Status</th>
                        <th>Name</th>
                        <th>Serial Number</th>
                        <th>Expired Date</th>
                        <th>Date Manufactured</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
<?php
include 'database.php';
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

// Load Composer's autoloader
require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

$pdo = Database::connect();
$sql = 'SELECT * FROM rackstatus ORDER BY timestamp DESC LIMIT 10';
foreach ($pdo->query($sql) as $row) {
    echo '<tr>';
    echo '<td>'. $row['id'] . '</td>';
    echo '<td>'. $row['rackId'] . '</td>';
    echo '<td>'. $row['status'] . '</td>';
```

```
    echo '<td>'. $row['name'] . '</td>';
    echo '<td>'. $row['serialNumber'] . '</td>';
    echo '<td>'. $row['expiredDate'] . '</td>';
    echo '<td>'. $row['manufacturedDate'] . '</td>';

    // Check if the expired date is within 3 days
    $expiredDate = new DateTime($row['expiredDate']);
    $currentDate = new DateTime();
    $interval = $currentDate->diff($expiredDate);

    if ($interval->days <= 3) {
        // Send email notification using PHPMailer
        $mail = new PHPMailer;
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com'; // Set your SMTP host
        $mail->SMTPAuth = true;
        $mail->Username = 'muhdfaizzat1@gmail.com'; // Set your SMTP
username
        $mail->Password = 'bmgt xnsq tppp ttxn'; // Set your SMTP password
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;

        $mail->setFrom('muhdfaizzat1@gmail.com', 'Faizzat');
        $mail->addAddress('muhdfaizzat1@gmail.com', 'Users');

        $mail->Subject = 'Product Expiry Notification';
        $mail->Body = 'Product in Rack: '. $row['rackId'] .' -> '.
$row['name'] .' with the Serial Number '. $row['serialNumber'] .' is
expiring soon on ' . $row['expiredDate'];

        if ($mail->send()) {
            echo '~';
         } else {
             echo 'Error: ' . $mail->ErrorInfo;
         }
    }

    echo '<td>
            <a href="user data edit page.php?id='.$row['id'].'" class="btn
btn-success">Edit</a>
            <a href="user data delete page.php?id='.$row['id'].'"
class="btn btn-danger" onclick="return confirm(\'Are you sure you want to
delete this record?\')">Delete</a>
          </td>';
    echo '</tr>';
}
Database::disconnect();
```

```
?>
                </tbody>
            </table>
        </div>
    </div> <!-- /container -->
</body>
</html>
```

Appendix 9 Product Registration Form PHP Code

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register Product</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="navbar.css"> <!-- Link to the navbar
styles -->

    <style>
        .status-container {
            margin: 20px;
        }

        input,
        textarea {
            width: 100%;
            margin-bottom: 10px;
            padding: 5px;
        }
    </style>
</head>

<body>
    <div class="navbar">
        <h2>Development of an Iot-based Inventory Management System using
a Microcontroller</h2>
        <ul class="nav">
            <li><a href="home.php" class="nav-item is-active" active-
color="orange">Home</a></li>
            <li><a href="userdata.php" class="nav-item" active-
color="green">User Data</a></li>
            <li><a href="registerproduct.php" class="nav-item" active-
color="blue">Register Product</a></li>
            <li><a href="datamonitoring.php" class="nav-item" active-
color="red">Data Monitoring</a></li>
            <li><a href="logout.php" class="nav-item" active-
color="rebeccapurple">Logout</a></li>
            <span class="nav-indicator"></span>
        </ul>
    </div>

    <div class="status-container">
        <h2>Register Product Form</h2>
```

```
        <label for="rackId">Rack ID:</label>
        <input type="text" id="rackId" placeholder="Enter rack ID"
readonly>

        <label for="itemName">Item Name:</label>
        <input type="text" id="itemName" placeholder="Enter item name">

        <label for="serialNumber">Serial Number:</label>
        <input type="text" id="serialNumber" placeholder="Enter serial
number">

        <label for="expiredDate">Expired Date:</label>
        <input type="date" id="expiredDate" placeholder="Enter expired
date">

        <label for="manufacturedDate">Manufactured Date:</label>
        <input type="date" id="manufacturedDate" placeholder="Enter
manufactured date">


        <textarea id="rackStatusTextarea" readonly>Loading...</textarea>
        <br>
        <button onclick="submitRackStatus()">Submit</button>
    </div>
```

Appendix 10 Retrieve IR Sensor Data PHP Code

```php
<?php
// Connect to your MySQL database (replace with your actual database
credentials)
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch the latest rack status
$sql = "SELECT * FROM rack_status ORDER BY timestamp DESC LIMIT 1";
$result = $conn->query($sql);

// Close the database connection
$conn->close();
?>
```

Appendix 11 User Data Edit Table PHP Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit User Data</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="navbar.css">

    <style>
        .navbar {
            margin-bottom: 20px;
        }

        .container {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
        }

        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }

        th, td {
            padding: 10px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }

        .btn-container {
            margin-top: 10px;
        }

        .btn {
            display: inline-block;
            padding: 10px 20px;
            text-align: center;
            text-decoration: none;
            color: #fff;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
```

```
        .btn-success {
            background-color: #5cb85c;
        }

        .btn-danger {
            background-color: #d9534f;
        }
    </style>
</head>
<body>
    <div class="navbar">
        <h2>Development of an Iot-based Inventory Management System using a Microcontroller</h2>
        <ul class="nav">
            <!-- Your existing navigation links -->
        </ul>
    </div>

    <div class="container">
        <h3>Edit User Data</h3>
        <?php
        include 'database.php';

        if (isset($_GET['id']) && !empty($_GET['id'])) {
            $id = $_GET['id'];

            // Retrieve data for the specified ID
            $pdo = Database::connect();
            $sql = "SELECT * FROM rackstatus WHERE id = ?";
            $stmt = $pdo->prepare($sql);
            $stmt->execute([$id]);
            $row = $stmt->fetch();
            Database::disconnect();

            if ($row) {
                // Display a form with existing data
                echo '<form method="POST" action="updateuser.php">';
                echo '<table>';
                echo '<tr><td>ID:</td><td>' . $row['id'] . '</td></tr>';
                echo '<tr><td>Rack ID:</td><td><input type="text" name="rackId" value="' . $row['rackId'] . '"></td></tr>';
                echo '<tr><td>Status:</td><td><input type="text" name="status" value="' . $row['status'] . '" readonly></td></tr>';
                echo '<tr><td>Name:</td><td><input type="text" name="name" value="' . $row['name'] . '"></td></tr>';
                echo '<tr><td>Serial Number:</td><td><input type="text" name="serialNumber" value="' . $row['serialNumber'] . '"></td></tr>';
```

```
                echo '<tr><td>Expired Date:</td><td><input type="date"
name="expiredDate" value="' . $row['expiredDate'] . '"></td></tr>';
                echo '<tr><td>Manufactured Date:</td><td><input
type="date" name="manufacturedDate" value="' . $row['manufacturedDate'] .
'"></td></tr>';
                echo '</table>';
                echo '<input type="hidden" name="id"
value="'.$row['id'].'">';
                echo '<div class="btn-container">';
                echo '<input class="btn btn-success" type="submit"
value="Update">';
                echo '<a href="userdata.php" class="btn btn-
danger">Back</a>';
                echo '</div>';
                echo '</form>';
            } else {
                echo '<p>User not found</p>';
            }
        } else {
            echo '<p>Invalid request</p>';
        }
        ?>
    </div> <!-- /container -->
</body>
</html>
```

Appendix 12 User Data Delete Table PHP Code

```php
<?php
include 'database.php';

if (isset($_GET['id']) && !empty($_GET['id'])) {
    $id = $_GET['id'];

    // Delete record from the database
    $pdo = Database::connect();
    $sql = "DELETE FROM rackstatus WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$id]);
    Database::disconnect();

    // Redirect back to the user data page after deletion
    header("Location: userdata.php");
    exit();
}
?>
```

Appendix 13 Product Register Submit PHP Code

```php
<?php

// Replace these values with your actual database connection details
$host = "localhost";
$db_user = "root";
$db_password = "";
$db_name = "test";

// Enable error reporting for debugging
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Create connection
$conn = new mysqli($host, $db_user, $db_password, $db_name);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if it's a POST request
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Fetch data from get_status.php
    $get_status_url = "http://localhost/gg/get_status.php";
    $get_status_data = file_get_contents($get_status_url);
    $get_status_json = json_decode($get_status_data, true);

    // Check if data retrieval was successful
    if ($get_status_json !== null) {
        $rackId = $get_status_json['rack_id'];
        $status = $get_status_json['status'];

        // Retrieve additional data from the POST request
        $request_raw = file_get_contents('php://input');
        $request = json_decode($request_raw);

        $manufacturedDate = isset($request->manufacturedDate) ? $request->manufacturedDate : 'Unknown';
        $expiredDate = isset($request->expiredDate) ? $request->expiredDate : 'Unknown';
        $itemName = isset($request->itemName) ? $request->itemName : 'Unknown';
        $serialNumber = isset($request->serialNumber) ? $request->serialNumber : 'Unknown';
```

```
        // Insert data into the database using prepared statement
        $stmt = $conn->prepare("INSERT INTO rackstatus (rackId, status,
manufacturedDate, expireDdate, name, serialNumber, timestamp)
                                VALUES (?, ?, ?, ?, ?, ?, NOW())");

        $stmt->bind_param("ssssss", $rackId, $status, $manufacturedDate,
$expiredDate, $itemName, $serialNumber);

        if ($stmt->execute()) {
            $response = array('success' => true, 'message' => 'Status
submitted successfully');
        } else {
            $response = array('success' => false, 'error' => 'Error
submitting status: ' . $stmt->error);
        }

        $stmt->close();
    } else {
        $response = array('success' => false, 'error' => 'Error fetching
data from get_status.php');
    }
} else {
    $response = array('success' => false, 'error' => 'Invalid request
method');
}

// Set the content type to JSON
header('Content-Type: application/json');

// Send the JSON-formatted response
echo json_encode($response);

// Close the database connection
$conn->close();
?>
```

Appendix 14 User Data Update PHP Code

```php
<?php
include 'database.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id = $_POST['id'];
    $rackId = $_POST['rackId'];
    $status = $_POST['status'];
    $name = $_POST['name'];
    $serialNumber = $_POST['serialNumber'];
    $expiredDate = $_POST['expiredDate'];
    $manufacturedDate = $_POST['manufacturedDate'];

    // Update record in the database
    $pdo = Database::connect();
    $sql = "UPDATE rackstatus SET rackId=?, status=?, name=?,
serialNumber=?, expiredDate=?, manufacturedDate=? WHERE id=?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$rackId, $status, $name, $serialNumber, $expiredDate,
$manufacturedDate, $id]);
    Database::disconnect();

    // Redirect back to the user data page after update
    header("Location: userdata.php");
    exit();
}
?>
```

Appendix 15 Data Monitoring Table PHP Code

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="refresh" content="5">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quantity Data Monitoring</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="navbar.css"> <!-- Link to the navbar
styles -->
</head>

<body>
    <div class="navbar">
        <h2>Development of an Iot-based Inventory Management System using
a Microcontroller</h2>
        <ul class="nav">
            <li><a href="home.php" class="nav-item is-active" active-
color="orange">Home</a></li>
            <li><a href="userdata.php" class="nav-item" active-
color="green">User Data</a></li>
            <li><a href="registerproduct.php" class="nav-item" active-
color="blue">Register Product</a></li>
            <li><a href="datamonitoring.php" class="nav-item" active-
color="red">Data Monitoring</a></li>
            <li><a href="logout.php" class="nav-item" active-
color="rebeccapurple">Logout</a></li>
            <span class="nav-indicator"></span>
        </ul>
    </div>
</body>
</html>

<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

// Load Composer's autoloader
require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

// Database connection details
$servername = "localhost";
```

```
$username = "root";
$password = "";
$dbname = "test";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if (isset($_POST["quantity1"]) && isset($_POST["quantity2"])) {
    $q = $_POST["quantity1"];
    $r = $_POST["quantity2"];

    // Sanitize the input (to prevent SQL injection)
    $q = $conn->real_escape_string($q);
    $r = $conn->real_escape_string($r);

    // Insert data into the database
    $sql = "INSERT INTO quantity_data (quantity1, quantity2) VALUES ('$q',
'$r')";

    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";

        // Check if either quantity1 or quantity2 is 3
        if ($q == 3 || $r == 3) {
            // Send email logic
            $mail = new PHPMailer(true);

            try {
                // Server settings
                $mail->isSMTP();
                $mail->Host       = 'smtp.gmail.com';
                $mail->SMTPAuth   = true;
                $mail->Username   = 'muhdfaizzat1@gmail.com'; // Your
Gmail email address
                $mail->Password   = 'bmgt xnsq tppp ttxn';    // Your
Gmail app password (if 2-step verification is enabled)
                $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
                $mail->Port       = 587;

                // Sender and recipient settings
                $mail->setFrom('muhdfaizzat1@gmail.com', 'Muhd Faizzat');
// Your name and email address
                $mail->addAddress('muhdfaizzat1@gmail.com'); //
Recipient's email address
```

```
                // Email content
                $mail->isHTML(true);
                $mail->Subject = 'Quantity Reached 3';
                $mail->Body    = 'Quantity is low please
topup  immediately.';

                // Send the email
                //$mail->send();
                $mail->SMTPDebug = SMTP::DEBUG_SERVER;  // Enable verbose
debug output
                $mail->Debugoutput = function ($str, $level) {
                    file_put_contents('php://stdout', "$str\n");
                };
                echo 'Email sent successfully';
            } catch (Exception $e) {
                echo "Failed to send email. Error: {$mail->ErrorInfo}";
            }
        }
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
// Fetch data from the quantity_data table
$sqlSelect = "SELECT * FROM quantity_data ORDER BY timestamp DESC LIMIT
10";
$result = $conn->query($sqlSelect);

// Display the table of quantity data
echo "<style>";
echo "table {";
echo "  font-family: Arial, sans-serif;";
echo "  border-collapse: collapse;";
echo "  width: 80%;";
echo "  margin: 20px auto;";
echo "}";
echo "th, td {";
echo "  border: 1px solid #dddddd;";
echo "  text-align: left;";
echo "  padding: 8px;";
echo "}";
echo "th {";
echo "  background-color: #f2f2f2;";
echo "}";
echo "</style>";

echo "<h2>Quantity Data Table</h2>";
echo "<table>";
```

```
echo "<tr>
      <th>Quantity 1</th>
      <th>Quantity 2</th>
      <th>Timestamp</th>
      </tr>";

// Check if there is data in the result set
if ($result !== false && $result->num_rows > 0) {
    // Output data of each row
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["quantity1"] . "</td>";
        echo "<td>" . $row["quantity2"] . "</td>";
        echo "<td>" . $row["timestamp"] . "</td>";
        echo "</tr>";
    }
} else {
    // If no data is available
    echo "<tr><td colspan='4'>No data available</td></tr>";
}

echo "</table>";

// Close the database connection
$conn->close();
?>
```

Appendix 16 Stylesheet For All Page CSS Code

```
@import
url('https:fonts.googleapis.com/css?family=DM+Sans:500,700&display=swap');

*{
    box-sizing: border-box;
}

body {
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    height: 100vh;
    width: 100%;
    text-align: left;
}
.nav{
    display: inline-flex;
    position: relative;
    overflow: hidden;
    max-width: 100%;
    background-color: #fff;
    padding: 0 20px;
    border-radius: 40px;
    box-shadow: 0 10px 40px rgba(159, 162, 177, .8);
}
h2 {
    margin-bottom: 0px; /* Adjust the margin as needed */
}
.nav li {
    margin: 0 10px; /* Adjust the margin as needed */
}
.nav-item{
    color: #83818c;
    padding: 20px;
    text-decoration: none;
    transition: .3s;
    margin: 0 6px;
    z-index: 1;
    font-family: 'DM Sans', sans-serif;
    font-weight: 500;
    position: relative;

    &:before{
```

```
        content: "";
        position: absolute;
        bottom: -6px;
        left: 0;
        width: 100%;
        height: 5px;
        background-color: #dfe2ea;
        border-radius: 8px 8px 0 0;
        opacity: 0;
        transition: .3s;
    }
}
.nav-item:not(.is-active):hover:before{
    opacity: 1;
    bottom: 0;
}
.nav-item:not(.is-active):hover{
    color: #333;
}
.nav-indicator{
    position: top;
    left: 0;
    bottom: 0;
    height: 4px;
    transition: .4s;
    height: 5px;
    z-index: 1;
    border-radius: 8px 8px 0 0;
}
@media (max-width: 580px){
    .nav{ overflow: auto;}
}
```

Appendix 17 Style Navbar For All Page CSS Code

```
body {
    margin: 0;
    padding: 0;
}

.navbar {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background-color: #333; /* Choose your desired background color */
    color: white;
    padding: 10px;
    text-align: center;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Optional: Add a subtle
shadow */
}

.navbar a {
    text-decoration: none;
    color: rgb(228, 142, 142);
    margin: 0 10px;
    padding: 5px;
    border-bottom: 2px solid transparent; /* Add a transparent border for
a neat hover effect */
    transition: border-bottom 0.3s ease;
}

.navbar a:hover {
    border-bottom: 2px solid orange; /* Choose your desired hover color */
}

.content {
    margin-top: 50px; /* Adjust the margin to accommodate the fixed navbar
*/
}
.btn {
    display: inline-block;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #10a0c5;
    border: none;
    border-radius: 5px;
    cursor: pointer;
```

```
}

.btn-success {
    background-color: #5cb85c;
}

.btn-danger {
    background-color: #d9534f;
}
```

Appendix 18 Gantt Chart PSM1

DEVELOPMENT OF AN IOT BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER

Universiti Teknikal Malaysia Melaka(UTeM)
Ts.Maslan Bin Zainon

Project Start: Mon, 3/20/2023
Display Week: 11

TASK	ASSIGNED TO	PROGRESS	START	END
Chapter 1: Introduction				
Title & Synopsis		100%	3/20/23	3/26/23
Research background and objectives		100%	3/27/23	4/1/23
Identify research gap		100%	4/2/23	4/7/23
Review related literature		100%	4/8/23	4/23/23
Finish draft report and send to supervisor		100%	6/20/23	6/26/23
Chapter 2: Literature Review				
Conduct literature search		100%	4/24/23	5/1/23
Select relevant sources		100%	4/26/23	5/1/23
Review and summarize literature		100%	5/1/23	5/9/23
Analyze and synthesize findings		100%	5/9/23	5/20/23
Finish draft report and Gantt chart and send to supervisor		100%	5/20/23	5/21/23
Meeting with supervisor		100%	5/29/23	6/26/23
Weekly progress updates		100%	5/29/23	6/26/23
Feedback and guidance		100%	5/29/23	6/26/23
Chapter 3: Methodology				
Define research methodology		100%	5/21/23	5/26/23
Determine data collection methods		100%	5/27/23	6/1/23
Design data collection instruments		100%	6/2/23	6/8/23
Collect and analyze data		100%	6/9/23	6/13/23
Chapter 4: Preliminary Result				
Present initial findings		100%	6/14/23	6/17/23
Analyze results and draw conclusions		100%	6/17/23	6/19/23
Refine findings and discussion		100%	6/19/23	6/19/23
Chapter 5: Conclusion				
Summarize the research findings, implications, and limitations		100%	6/19/23	6/20/23
Final Presentation		100%	6/21/23	6/26/23

May 29, 2023 | Jun 5, 2023 | Jun 12, 2023 | Jun 19, 2023 | Jun 26, 2023 | Jul 3, 2023 | Jul 10, 2023 | Jul 17, 2023

29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S

Appendix 19 Gantt Chart PSM2

DEVELOPMENT OF AN IOT - BASED SMART INVENTORY MANAGEMENT SYSTEM USING A MICROCONTROLLER

Universiti Teknikal Malaysia Melaka(UTeM)
Ts.Maslan Bin Zainon

Project Start: Wed, 10/11/2023
Display Week: 7

TASK	ASSIGNED TO	PROGRESS	START	END
Chapter 5: Analysis				
Present research findings		100%	10/11/23	10/17/23
Analyze and interpret data		100%	10/17/23	10/21/23
Compare findings with literature		100%	10/21/23	10/25/23
Identify patterns and trends		100%	10/25/23	10/29/23
Chapter 6: Result				
Structure and programming		100%	10/29/23	1/2/24
Perform testing		100%	10/29/23	1/2/24
Analyze testing results		100%	10/29/23	1/2/24
Debug and fix issues		100%	10/29/23	1/2/24
Finish writing a draft report and send it to the supervisor		80%	1/2/24	1/10/24
Meeting with supervisor				
Weekly progress updates		80%	10/11/23	1/14/24
Feedback and guidance		80%	10/11/23	1/14/24
Chapter 7: Discussion				
Discuss research implications of the results		100%	1/2/24	1/4/24
Explore limitations and compare with existing research		100%	1/4/24	1/6/24
Propose future directions		100%	1/7/24	1/9/24
Chapter 8: Conclusion				
Summarize the main findings and contributions		100%	1/2/24	1/3/24
Provide recommendations for future research		100%	1/3/24	1/4/24
Finalize the conclusion and wrap up the report		80%	1/4/24	1/5/24
Final Presentation			1/14/24	1/15/24
Submit Final Report			1/14/24	1/15/24