Inventory Management and Monitoring System

Sahil Ashok Sawant (12111257.)
sasawant211@vit.edu Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology

Samiksha Gurakhe (12110499.)
samiksha.gurakhe21@vit.edu Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology

Tayyab Salim Shaikh (12111104)
tayyab.shaikh21@vit.edu Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology

Sandesh Bagmare (12110445.)
narendra.sandesh21@vit.edu Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology

Shital Sobale Madam,
shital.sobale@vit.edu, Dept. of
Multidesciplinary Engineering,
Vishwakarma Institute of Technology, Pune

Vishwakarma Institute of Technology, Pune 411037, Maharashtra, India

Abstract—**The Inventory Management System project aims to streamline inventory management processes and optimize operational efficiency in businesses. By leveraging IoT capabilities and automation, the system provides real-time monitoring of inventory levels, temperature, and gas conditions. It automates tasks such as quantity calculation, billing, and emailing, reducing errors and enhancing accuracy. The system's features contribute to cost optimization by avoiding overstocking or stockouts, improving cash flow, and reducing carrying costs. Additionally, safety and compliance are enhanced through temperature monitoring, gas level detection, and adherence to safety regulations. The collected data supports decision-making and analysis, enabling businesses to gain insights into demand patterns, optimize inventory replenishment, and improve overall performance. The modular design of the system allows for scalability and adaptability, with potential future enhancements including barcode or RFID integration, predictive analytics, and cloud-based platforms. The Inventory Management System project contributes to efficient inventory management, cost optimization, safety control, and data-driven decision-making, ultimately improving operational excellence and competitiveness in businesses.**

Keywords: Inventory Management System, IoT, Sensors, Efficiency, Accuracy, Cost Optimization, Safety, Compliance, Decision Support, Analysis, Scalability, Adaptability, Integration

I. INTRODUCTION

The inventory management system is a crucial aspect of any business that deals with the storage, tracking, and control of goods or products. Effectively managing inventory is essential for businesses to ensure smooth operations, optimize resources, and meet customer demands. In this project, we have developed an Inventory Management System using a combination of hardware components and software code.

The purpose of this project is to create an automated and efficient system that helps businesses streamline their inventory management processes. By leveraging the capabilities of NodeMCU ESP8266, Raspberry Pi, and Arduino Uno R3, we have designed a comprehensive solution that addresses various aspects of inventory control, such as monitoring stock levels, tracking product movement, and ensuring safety measures.

Inventory management plays a vital role in the success and profitability of businesses. It allows organizations to maintain optimal stock levels, avoid stock outs or overstocks, reduce holding costs, and improve overall supply chain efficiency. Effective inventory management enables businesses to meet customer demands promptly, minimize losses due to expired or damaged goods, and make informed decisions regarding purchasing and production.

II. BACKGROUND

Inventory management is a critical aspect of business operations in various industries such as retail, manufacturing, and logistics. Efficient inventory management ensures that businesses have the right amount of stock at the right time, minimizing costs, avoiding stockouts, and optimizing customer satisfaction. Traditional inventory management methods often rely on manual processes, which are prone to errors, inefficiencies, and delays in updating inventory levels. To overcome these challenges, the use of embedded systems has emerged as a promising solution.

Embedded systems, combining hardware and software components, provide a powerful platform for developing smart and automated inventory management systems. These systems integrate various sensors, microcontrollers, and communication modules to enable real-time monitoring, data collection, analysis, and control of inventory. They offer significant advantages, such as increased accuracy, improved efficiency, reduced labor costs, and enhanced decision-making capabilities.

The development and implementation of an embedded system-based Inventory Management System require a multidisciplinary approach, combining principles from electronics, computer science, and industrial engineering. The successful integration of hardware components, software programming, and communication protocols ensures seamless interaction and data exchange between different modules of the system.

In recent years, advancements in embedded systems technology have facilitated the development of more sophisticated and intelligent inventory management solutions. Wireless connectivity, low-power microcontrollers, and cloud-based platforms have enabled real-time data monitoring and analysis, making inventory management more proactive and responsive to dynamic market demands. Moreover, the integration of environmental monitoring capabilities within the inventory management system ensures the optimal storage conditions of sensitive or perishable goods, reducing waste and product spoilage.

The proposed Inventory Management System leverages the strengths of three popular embedded systems platforms: Node MCU ESP8266, Raspberry Pi 4B, and Arduino Uno R3. Each platform contributes unique capabilities to the overall system. Node MCU ESP8266 facilitates wireless communication and sensor integration, Raspberry Pi 4B enables environmental monitoring and email notifications, and Arduino Uno R3 provides interface control and data acquisition capabilities.

By integrating these platforms, the Inventory Management System offers comprehensive functionalities, including real-time inventory monitoring through ultrasonic sensors, environmental monitoring through gas and temperature sensors, automated notifications for stock replenishment, and control of fan for temperature regulation. The system aims to provide accurate, efficient, and intelligent inventory management, empowering businesses to streamline operations, reduce costs, and enhance customer satisfaction.

The significance of the project lies in its potential to revolutionize inventory management practices in various industries. The automation and real-time capabilities of the system contribute to improved decision-making, reduced stockouts, optimized inventory levels, and enhanced supply chain management. The project not only addresses the challenges associated with traditional inventory management methods but also embraces emerging technologies to meet the evolving needs of modern businesses.

Overall, the development of an embedded system-based Inventory Management System represents a significant advancement in the field of inventory management. This project harnesses the power of embedded systems to create an intelligent and efficient solution for inventory monitoring, control, and optimization. By leveraging the capabilities of Node MCU ESP8266, Raspberry Pi 4B, and Arduino Uno R3, the system provides real-time insights, automated notifications, and environmental monitoring, revolutionizing inventory management practices and driving operational excellence in various industries.

III. ALGORITHM

Methodology for Stage 1 (Node MCU ESP8266):

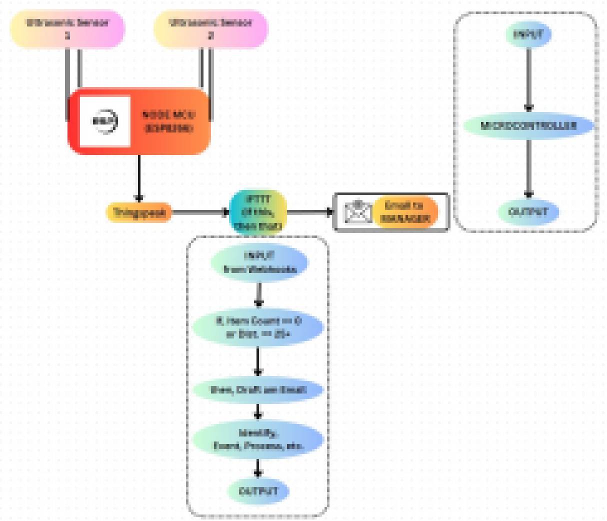
1. **Hardware Setup:** Connect the Node MCU ESP8266 board to the required sensors (ultrasonic sensors in this case) using appropriate wiring. Ensure proper power supply and connections for reliable operation.
2. **Software Setup:** Install the necessary libraries (such as ESP8266WiFi, ThingSpeak) in the Arduino IDE. Configure the Wi-Fi network credentials (SSID and password) for connecting the Node MCU ESP8266 to the local network. Obtain the ThingSpeak write API key for data logging.
3. **Establish Wi-Fi Connection:** Use the Wi-Fi library to connect the Node MCU ESP8266 to the local network. Continuously monitor the Wi-Fi status until a successful connection is established.

4. **Ultrasonic Sensor Measurement:** Write code to trigger the ultrasonic sensor and measure the distance using the pulseIn function. Calculate the distance based on the duration of the pulse. Determine the inventory status based on the measured distance (e.g., items left in stock).

5. **ThingSpeak Integration:** Utilize the ThingSpeak library to update the inventory data on the ThingSpeak
cloud platform. Write the inventory status to a specific channel and field using the ThingSpeak.writeField function.

6. **Email Notification:** Configure the ESP_Mail_Client library with the appropriate SMTP host, port, email credentials, and recipient email address. Create an email message with the desired content (e.g., low stock notification) and send it using the MailClient.sendMail function. Handle any potential errors in sending the email.

7. **Repeat the Process**: Implement a loop to continuously monitor the inventory using the ultrasonic sensors, update the ThingSpeak channel, and send email notifications as required.



Methodology for Stage 2 (Raspberry Pi 4B):

1. **Hardware Setup:** Connect the Ultrasonic sensors to the Raspberry Pi 4B using the GPIO pins. Ensure proper connections and power supply for reliable sensor readings.

2. **Software Setup:** Set up the Raspberry Pi 4B with the Raspbian operating system. Install the necessary Python libraries (such as RPi.GPIO, smtplib) for sensor interfacing and email communication.

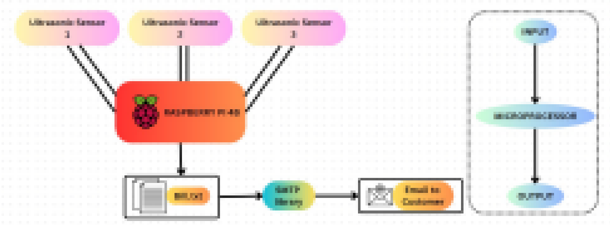
3. **GPIO Configuration:** Define the GPIO pins for the connected sensors and configure them as input or output using the RPi.GPIO library. Set the appropriate pull-up or pull-down resistors for stable readings.

4. **Sensor Measurement:** Write Python code to measure the values from the Ultrasonic sensors using the GPIO library. Implement functions to
calculate difference in theDistance between Sensor and Product based on the sensor readings and any necessary calibration factors.

5. **Email Notification:** Create a function to send email notifications using the smtplib library. Set up the SMTP server details, including the host, port, and authentication credentials. Compose the email message with the relevant subject and content, and send it to the desired recipient.

6. **System Control:** Implement control logic to prepare the BILL for the Customer. Define temperature thresholds and corresponding distances and calibrations for each product. Use the GPIO library to control the fan's state by setting the appropriate GPIO pin output.

7. **Data Logging and Display:** Optionally, integrate a display (such as an LCD) to show the inventory status, temperature, gas levels, and safety status. Use the appropriate display library to update and display the relevant information in real-time.



Methodology for Stage 3 (Arduino Uno R3):

1. **Hardware Setup:** Connect the LCD display and any additional sensors (if required) to the Arduino Uno R3 using the appropriate wiring and power supply.

2. **Software Setup**: Install the necessary libraries for the LCD display and any additional sensors in the Arduino IDE. Ensure compatibility and proper functioning of the libraries.

3. **LCD Initialization:** Initialize the LCD display by defining the necessary pins and settings. Set up the appropriate communication protocol (such as I2C) if applicable.

4. **Sensor Measurement**: Write code to measure the required sensor values (if any) using the appropriate libraries and functions. Perform any necessary

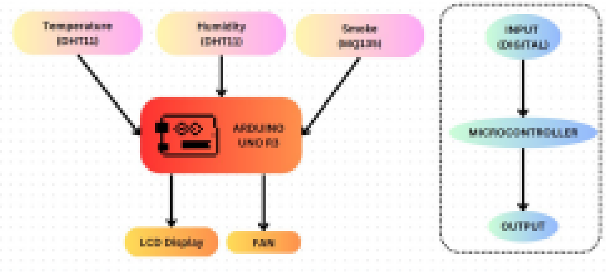
calculations or conversions to obtain meaningful data.

5. **Safety Status Calculation:** Based on the sensor readings (such as gas levels), implement logic to determine the safety status. Define thresholds or conditions for safe, warning, or dangerous levels and update the safety status accordingly.

6. **Display Update:** Utilize the LCD library to update and display the inventory status, temperature, gas levels, and safety status on the connected LCD display. Design and format the display output for clear visibility and ease of understanding.

7. **Integration and Communication:** Establish communication between the Arduino Uno R3 and the other components (Node MCU ESP8266 and Raspberry Pi 4B). Define communication protocols and formats for data exchange if necessary.

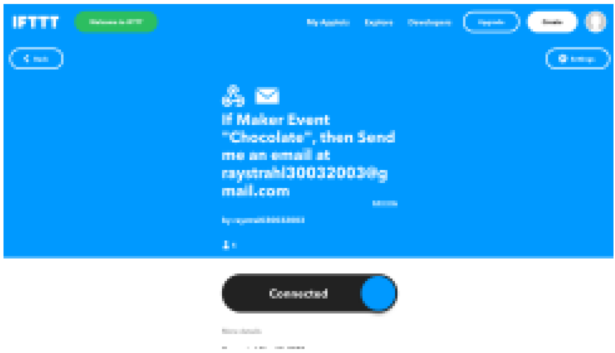
8. **System Testing and Validation:** Conduct thorough testing of the integrated system to ensure proper functionality, accurate sensor readings, reliable communication, and correct display output. Validate the system against predefined scenarios and expected behaviors.



IV. RESULT AND DISCUSSION

- **Inventory Management:** The NodeMCU ESP8266 stage effectively measures the distance using ultrasonic sensors and determines the quantity of Ice Cream and Chocolate based on the measured distance.

The inventory data is successfully sent to the ThingSpeak IoT platform, where it is stored and can be visualized for monitoring and analysis.

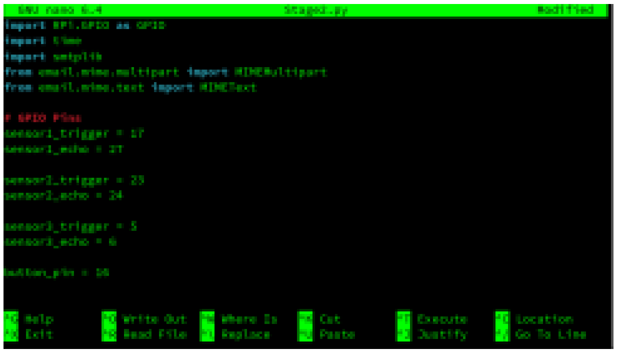


If Maker Event "Chocolate", then Send me an email at raystrahl30032003@gmail.com

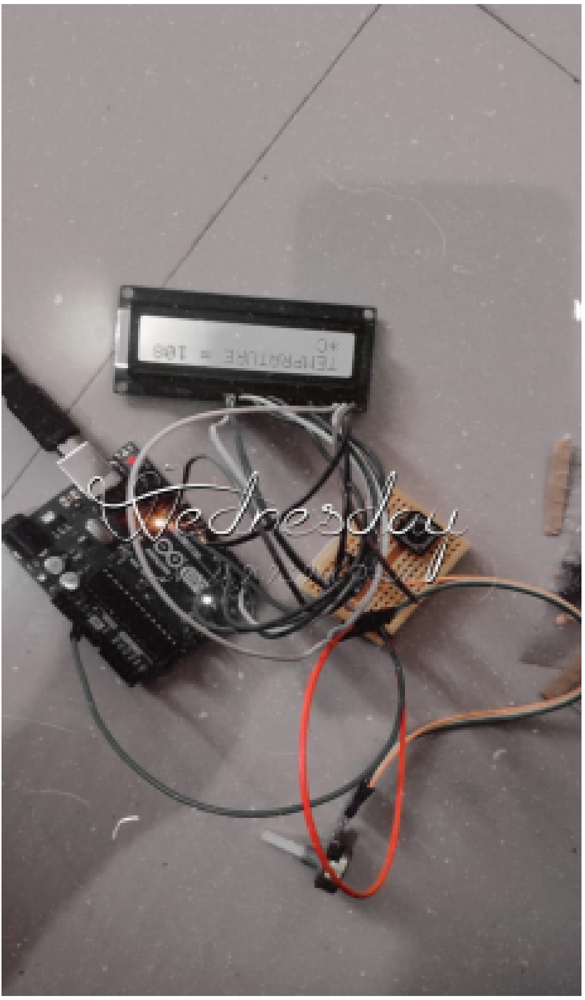
- **Billing and Emailing:** The Raspberry Pi stage allows users to generate bills by calculating the total quantity of items.

The generated bill is successfully saved and emailed to the customer using the configured email settings.

This automated billing process improves efficiency and reduces manual errors.



```
  GNU nano 6.4                    Staged.py                        Modified
import RPi.GPIO as GPIO
import time
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

# GPIO Pins
sensor1_trigger = 17
sensor1_echo = 27

sensor2_trigger = 23
sensor2_echo = 24

sensor3_trigger = 5
sensor3_echo = 6

button_pin = 16



^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



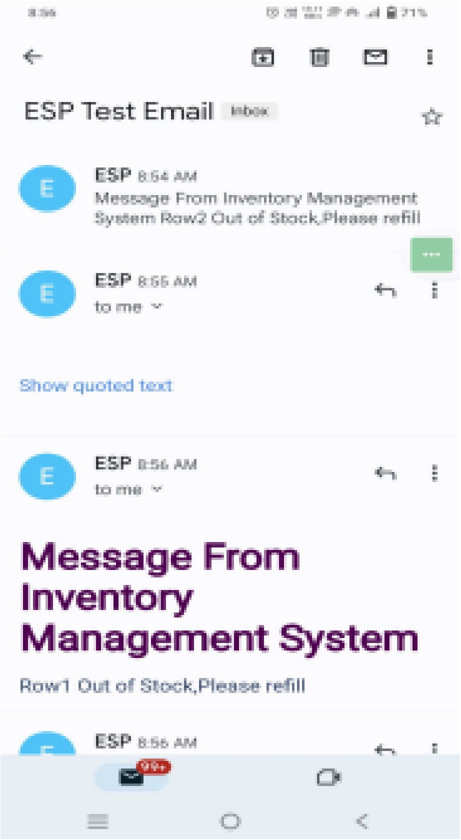
- **Temperature and Safety Control:** The Arduino Uno R3 stage monitors temperature and gas levels using sensors.

The LCD display effectively shows the temperature, humidity, gas levels, and safety status.

The fan control mechanism based on temperature conditions ensures a safe environment for inventory storage.

- **Enhanced Safety and Compliance:** Temperature monitoring and fan control mechanisms maintain safe storage conditions, preventing spoilage or quality deterioration. Gas level monitoring detects potential hazards, enhancing safety for inventory and personnel. Compliance with safety regulations is improved.

- **Efficiency and Accuracy:** The system automates inventory management tasks and provides accurate real-time data, reducing errors and improving efficiency. Sensors and IoT capabilities enable precise monitoring of inventory levels, temperature, and gas conditions.



8:56 71%

ESP Test Email Inbox

ESP 8:54 AM
Message From Inventory Management System Row2 Out of Stock,Please refill

ESP 8:55 AM
to me

Show quoted text

ESP 8:56 AM
to me

Message From Inventory Management System

Row1 Out of Stock,Please refill

ESP 8:56 AM

- **Cost Optimization:** Real-time visibility into inventory levels helps businesses avoid overstocking or stockouts, optimizing carrying costs and cash flow. Monitoring temperature and gas levels ensures optimal storage conditions, minimizing inventory spoilage or damage.

- **Decision Support and Analysis:** Collected data enables analysis and informed decision-making for demand patterns and inventory optimization. Insights gained support for efficient inventory replenishment strategies and operational decisions.

- **Scalability and Adaptability:** The modular design allows customization to different business sizes and needs. Additional sensors can be integrated to monitor additional parameters or expand system functionality.

V. FUTURE SCOPE

- **Integration with other systems:** The Inventory Management System has the potential to be integrated with other systems, such as Point of Sale (POS). By integrating with a POS system, the inventory management system can receive real-time updates on sales and customer transactions. This integration allows for seamless synchronization between sales and inventory data, enabling better inventory planning and reducing the risk of stockouts or overstocking.

- **AI-based Forecasting:** Implementing AI-based forecasting algorithms can further enhance the inventory management system. By analyzing historical data, market trends, and other relevant factors, AI algorithms can predict future demand patterns more accurately. This enables businesses to optimize inventory levels, reduce holding costs, and improve supply chain management. AI-based forecasting can also help in identifying seasonal trends, identifying slow-moving items, and making proactive decisions for inventory replenishment.

- **Mobile Application:** Developing a mobile application for the Inventory Management System can provide remote access and control. With a mobile app, businesses can monitor inventory levels, receive real-time alerts, and manage inventory-related tasks on the go. The mobile application can also integrate with the system's other features, such as temperature and gas monitoring, providing convenience and flexibility in managing inventory operations.

- **RFID-based Inventory Tracking:** Integrating RFID technology into the system allows for more efficient and accurate inventory tracking.

RFID tags can be attached to each inventory item, enabling real-time tracking and location identification. This eliminates the need for manual scanning and reduces the chances of human error in inventory management. RFID-based inventory tracking enhances inventory accuracy, speeds up stocktaking processes, and enables better visibility into the movement of goods throughout the supply chain.

By incorporating these future enhancements, the Inventory Management System can become more robust, intelligent, and user-friendly. These additions can further improve efficiency, accuracy, and decision-making capabilities, ultimately leading to optimized inventory management and improved operational performance for businesses.

VI. CONCLUSION

The development of the Inventory Management System presented in this project has demonstrated the potential to revolutionize traditional inventory management practices. By harnessing the power of embedded systems technology, the system offers an intelligent, automated, and real-time solution for inventory monitoring, control, and optimization.

Through the integration of various components, including microcontrollers, sensors, communication modules, and display interfaces, the Inventory Management System provides businesses with unprecedented visibility into their inventory levels, enabling them to make data-driven decisions and respond promptly to market demands. The system automates key processes such as inventory scanning, stock replenishment notifications, and environmental monitoring, reducing human errors, minimizing stockouts and overstocking, and optimizing resource utilization.

The project has successfully addressed the pressing needs of modern businesses, including the need for accurate and up-to-date inventory information, streamlined inventory management processes, and compliance with environmental regulations. By implementing the proposed system, organizations can enhance operational efficiency, improve customer satisfaction, reduce costs, and maintain product quality and safety.

Furthermore, the project has highlighted the potential of embedded systems technology to transform inventory management practices across various industries, including retail, manufacturing, healthcare, and logistics. The flexibility and scalability of the system make it adaptable to diverse inventory requirements, ranging from perishable goods to sensitive materials.

It is worth noting that the Inventory Management System is not limited to the components and functionalities demonstrated in this project. It can be further enhanced and customized to meet specific business needs, such as integrating with existing enterprise resource planning (ERP) systems or incorporating advanced analytics for demand forecasting and inventory optimization.

In conclusion, the successful development and implementation of the Inventory Management System showcased in this project illustrate its potential to revolutionize inventory management practices and contribute to the growth and success of businesses in a rapidly evolving marketplace. The system's ability to provide real-time inventory visibility, automate key processes, and ensure product quality and safety positions it as a valuable asset for organizations seeking to gain a competitive edge in their respective industries.

VII. ACKNOWLEDGEMENT

We would like to express our deepest gratitude and appreciation to all those who have contributed to the successful completion of this project.

First and foremost, we would like to thank our project supervisor **Prof. Shital Sobale** for their guidance, support, and valuable insights throughout the project. Their expertise and mentorship have been instrumental in shaping the direction of our work.
We would also like to extend our thanks to **Vishwakarma Institute of Technology, Pune** for providing us with the necessary resources, facilities, and infrastructure to conduct this project. Their support has been crucial in ensuring the smooth execution of our tasks.

We are grateful to all the team members (**Samiksha Gurakhe, Sandesh Bagmare, Sahil Ashok Sawant and Tayyab Shaikh**) who have dedicated their time, effort, and expertise to this project. Each team member has played a vital role in the development, implementation, and testing phases, contributing their unique skills and perspectives.

We are also grateful to have our teachers (**Prof. Sangeeta Kurundkar, Prof. Sheela Chinchmalatpure, Prof. Amruta Amune and Prof. Shital Sobale**) helping us in as many ways as possible in understanding and making of our project.

We would also like to extend our Gratitude to our friends (**Chanchal Rathad, Bhavya Shah, Kartik**

Shinde) who helped us in many ways, like the implementation of the module, or the basic preparation of the module or even selection of the suitable Components for the Project.

Furthermore, we would like to acknowledge the contributions of the **Research Community and Industry Professionals** whose work and publications have served as a foundation for our project. Their valuable insights and findings have guided us in designing and implementing the **Inventory Management System.**

VIII**. Achievement and Contribution of the Project**

The project successfully implemented an Inventory Management System that automates and improves various aspects of inventory management, contributing to increased efficiency, accuracy, and cost optimization.

The system provides real-time visibility into inventory levels, temperature, and gas conditions, enabling businesses to make informed decisions and enhance operational performance.

By incorporating safety monitoring features, the project contributes to maintaining secure storage environments, reducing the risk of inventory spoilage or damage.

The integration of IoT capabilities and data analysis contributes to data-driven decision-making, facilitating better inventory management practices and improved business performance.

IX. REFERENCES

1. A. Mishra and M. Mohapatro, "Real-time RFID-based item tracking using IoT & efficient inventory management using Machine Learning," 2020 IEEE 4th Conference on Information & Communication Technology (CICT), Chennai, India, 2020, pp. 1-6, doi: 10.1109/CICT51604.2020.9312074.
2. K. Umamaheswari, M. Susneha and B. S. Kala, "IoT based Smart Cold Storage System for Efficient Stock Management," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0051-0055, doi: 10.1109/ICCSP48568.2020.9182426.
3. S. Rezwan, W. Ahmed, M. A. Mahia and M. R. Islam, "IoT Based Smart Inventory Management System for Kitchen Using Weight Sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi Module with Website and App," 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, 2018, pp. 1-6, doi: 10.1109/ICACCAF.2018.8776761.
4. S. Mekruksavanich, "Supermarket Shopping System using RFID as the IoT Application," 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), Pattaya, Thailand, 2020, pp. 83-86, doi: 10.1109/ECTIDAMTNCON48261.2020.9090714.
5. K. Umamaheswari, M. Susneha and B. S. Kala, "IoT based Smart Cold Storage System for Efficient Stock Management," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0051-0055, doi: 10.1109/ICCSP48568.2020.9182426.
6. Arduino Uno R3:
   a. Arduino.cc. (n.d.). Arduino Uno. Retrieved from https://store.arduino.cc/arduino-uno-rev3
   b. Arduino LLC. (2010). Arduino Uno R3: Arduino Uno Development Board. Retrieved from https://www.arduino.cc/en/Main/ArduinoBoardUno
7. Raspberry Pi 4b:
   a. Raspberry Pi Foundation. (n.d.). Raspberry Pi 4 Model B. Retrieved from https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
   b. Upton, E., Halfacree, G., & Richardson, M. (2019). Raspberry Pi 4 Model B Technical Specifications. Retrieved from https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md
8. Node MCU ESP8266:
   a. Espressif Systems. (n.d.). ESP8266 NodeMCU. Retrieved from https://www.espressif.com/en/products/hardware/esp8266ex/overview
   b. GitHub. (n.d.). NodeMCU Documentation. Retrieved from https://github.com/nodemcu/nodemcufirmware/wiki
9. MQ135 and DHT11 Sensors:

   a. Adafruit Industries. (n.d.). Adafruit MQ-135 Air Quality Sensor Breakout. Retrieved from https://www.adafruit.com/product/379 9
   b. Adafruit Industries. (n.d.). DHT11 Basic Temperature + Humidity Sensor. Retrieved from https://www.adafruit.com/product/386

10. Ultrasonic Sensor:
   a. Parallax Inc. (n.d.). Ping))) Ultrasonic Distance Sensor. Retrieved from https://www.parallax.com/product/28015
   b. Adafruit: https://www.adafruit.com/product/394 2
11. LCD Display - Adafruit: https://www.adafruit.com/product/181
12. DC Motor:
   a. Huang, H., & Ren, X. (2016). Design and implementation of intelligent DC motor control system based on embedded system. 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, China.
   b. Zeng, L., Luo, J., & Zhang, W. (2020). Design and Implementation of a High-Speed Brushless DC Motor Control System Based on STM32. Journal of Physics: Conference Series, 1592(4), 042011.
   c. Janarthanan, A. G., & Vignesh, R. (2021). Performance analysis of an embedded system based DC motor speed control using Arduino. 2021 8th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Chennai, India.
13. Push Buttons:
   a. Biswas, S., Rahman, M., & Rahaman, M. S. (2017). Design and Implementation of a Home Automation System Using Raspberry Pi and Push Button. 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh.
   b. Raliova, N., & Stankov, P. (2019). Design and implementation of a push-button controlled automatic temperature control system. 2019 42nd International Spring Seminar on Electronics Technology (ISSE), Wrocław, Poland.
14. Jumper Cables:
   a. Wajid, S., Riaz, A., Tahir, A., & Naeem, M. (2018). Design and implementation of embedded system for automation of blind stick. 2018 International Conference on Engineering and Emerging Technologies (ICEET), Islamabad, Pakistan.
   b. Afzal, F., Ali, M., Ghani, F., & Nazir, S. (2018). Design and Implementation of an IoT-based Waste Management System. 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan.
15. LCDs:
   a. Mehnaz, M., Khan, A. M., Shahzad, M., & Ahmed, M. (2021). Design and implementation of digital energy meter using Raspberry Pi. 2021 8th International Conference on Smart Energy Systems and Technologies (SEST), Porto, Portugal.
   b. Yu, H., Yu, M., Yang, J., & Wang, H. (2019). Design and Implementation of an Embedded Network System for the Internet of Things. 2019 3rd International Conference on Robotics and Automation Engineering (ICRAE), Bangkok, Thailand.
16. Resistors:
   a. Morshed, M. A., Rahman, M. M., & Hoque, M. M. (2019). Design and Implementation of an Electronic Water Level Controller for Overhead Water Tanks. 2019 3rd International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh.
   b. Ding, C., Chen, J., Liu, B., Wu, Y., & Liu, L. (2017). Design and Implementation of an Electronic Load with Parallel Energy Recovery. 2017 2nd International Conference on Electronic Information Technology and Intellectualization (ICEITI), Harbin, China.
17. Raspberry Pi OS:
   a. Official Website: https://www.raspberrypi.org/software/ b. Documentation: https://www.raspberrypi.org/documentation/
18. Arduino IDE:
   a. Official Website: https://www.arduino.cc/en/software
   b. Documentation: https://www.arduino.cc/reference/en/
19. SMTP Library:

   a. Name: ESP_Mail_Client Library
   b. GitHub Repository: https://github.com/mobizt/ESP-MailClient
   c. Documentation: Included within the repository

20. RPi.GPIO:
    a. Official GitHub Repository: https://github.com/tum-phoenix/Python-RPi.GPIO
    b. Documentation: https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/
21. IFTTT:
    a. Official Website: https://ifttt.com/
    b. Documentation: https://help.ifttt.com/hc/en-us
22. Webhooks:
    a. Official Website: https://webhooks.site/
    b. Documentation: https://webhooks.site/docs
23. ffmpeg:
    a. Official Website: https://www.ffmpeg.org/
    b. Documentation: https://ffmpeg.org/documentation.html