UNIVERSITY OF VOCATIONAL TECHNOLOGY

Faculty of Engineering Technology

Department of Electro-Mechanical Technology

EE402040

Internet of Things (IoT)

Project Proposal

Smart Environmental Monitoring System for Textile Manufacturing Floors

Instructed by : **Mr. Janith Kasun**

Group Members : R.B.S.P Wijerathna (MAN/22/B1/11)

:A.G.S.D Kalansuriya (MAN/22/B1/28)

Program : B.Tech in Mechatronics Technology

Submission Date : June 21, 2025

Contents

1. Introduction 3
    1.1 Problem 3
    1.2 Motivation 3
    1.3Aim 4
    1.4SMART Objectives 4
2.Literature Review 5
2.1What Exists Already? 5
    2.2 What's Missing (That This Project Will Fix) 6
    2.3 References 7
3. Methodology & System Design 8
    3.1 High level Diagram 8
    3.3 Wiring Diagram 10
4. Implementation Plan & Timeline 11
    4.1 Task Allocation 11
    4.2 Detailed Schedule 12
5. Expected Outcomes & Deliverables 12
6. Budget/Resources 14
7.Description of Project Resources 16
    7.1. Software / Cloud Resources 16
    7.2. Human Resources 17
    7.3. Other Resources 17
8.Summary 18
9. References 19

1. Introduction

1.1 Problem

Textile manufacturing environments are highly sensitive to fluctuations in temperature, humidity, and air quality. These environmental factors have a direct impact on the quality of fabrics produced, the performance and lifespan of machinery, and the health and safety of workers on the factory floor. For example, high humidity can cause yarn swelling and dyeing inconsistencies, while poor air quality such as elevated levels of dust or chemical vapors can lead to fabric contamination, equipment corrosion, and respiratory issues among workers. Traditional monitoring methods in many textile factories are manual, periodic, and reactive, meaning issues are often detected only after they have already caused product defects or machine failures. This leads to increased production costs, waste, and potential non-compliance with industry standards for workplace safety and environmental control.

1.2 Motivation

The textile industry is under increasing pressure to improve product quality, reduce waste, and ensure safe working conditions, all while maintaining cost competitiveness. The global shift toward smart manufacturing and Industry 4.0 has highlighted the value of real-time data and automation in achieving these goals. Automated, IOT-based environmental monitoring systems provide continuous, accurate insights into factory conditions, enabling proactive interventions that minimize defects, extend machine life, and protect worker health. Furthermore, regulatory bodies are tightening requirements for environmental monitoring and reporting, making it essential for textile manufacturers to adopt reliable, scalable solutions. By leveraging IOT technologies, factories can not only meet these compliance standards but also gain a competitive edge through improved operational efficiency and sustainability.

1.3Aim

To design and implement an IOT-based Smart Environmental Monitoring System for textile manufacturing floors that continuously tracks key environmental parameters and provides actionable insights to optimize production quality, machine health, and workplace safety.

1.4SMART Objectives

- **Specific:** Develop a modular system using ESP32 microcontrollers and industrial-grade sensors to monitor temperature, humidity, and air quality (e.g., CO_2, ammonia) on the factory floor.

- **Measurable:** Ensure the system provides real-time data updates at intervals of 5 seconds or less, and generates alerts when any parameter exceeds predefined safety thresholds.

- **Achievable:** Use cost-effective, commercially available hardware and open-source software frameworks to build a prototype within the project timeline.

- **Relevant:** Address the unique needs of textile manufacturing by enabling customizable monitoring, historical data analysis, and integration with existing factory management systems.

- **Time-bound:** Complete system design, prototyping, testing, and documentation within a 10-week schedule, culminating in a functional demonstration and comprehensive project report.

- This project aims to bridge the gap between generic environmental monitoring solutions and the specific operational requirements of textile manufacturing, delivering a tailored, scalable, and cost-effective system that supports the industry's transition to smart, data-driven production

2.Literature Review

2.1What Exists Already?

IoT-Based Environmental Monitoring Systems

Existing research demonstrates robust IoT frameworks for environmental monitoring across various industries. The IEC 61850-based system for power communication rooms employs a three-layer architecture (master, monitoring, and data collection layers) with real-time sensor networks and cloud-based data processing1. Similarly, the Smart Citizen System provides modular open-source hardware for monitoring air quality parameters like CO, NO_2, and O_3 using Alphasense electrochemical sensors, with capabilities for particulate matter (PM) detection via Plantower PMS5003 sensors.

These systems feature:

- **Real-time data transmission** using Wi-Fi/GSM protocols
- **Cloud-based dashboards** for visualization (e.g., HTTP web interfaces)
- **Multi-sensor integration** (temperature, humidity, gas concentrations)
- **Alert mechanisms** for threshold breaches

Industrial Deployments

In textile manufacturing specifically, studies highlight:

- Wireless sensor networks for microenvironment tracking around machinery
- Commercial systems like Oizom Polludrone for air quality monitoring
- Predictive maintenance models using humidity/temperature correlations with machine performance
- IoT-enabled resource optimization reducing operational costs by 15-20% in pilot implementations

2.2 What's Missing (That This Project Will Fix)

Gap 1: Textile-Specific Environmental Control Integration

Existing systems focus primarily on *monitoring* but lack **automated control mechanisms** tailored to textile production nuances:

- Current solutions detect anomalies but require manual intervention for climate control
- No integrated systems linking environmental data to HVAC/humidification systems
- Limited research on closed-loop control for textile-specific parameters (e.g., static electricity prevention)

Gap 2: Industry-Tailored Hazard Response

While gas sensors exist, their implementation in textiles lacks:

- **Process-specific threat detection**: Ammonia from dyeing processes or formaldehyde from finishes
- **Automated mitigation protocols**: No systems trigger exhaust fans *while* alerting supervisors
- **Preventive machine safeguarding**: No integration with loom shutdown protocols during hazardous conditions

Gap 3: Scalable Modularity for Manufacturing Floors

Most solutions are either:

- **Rigid commercial packages** (e.g., Bosch monitoring suites) that resist customization
- **DIY systems** (e.g., Arduino-based stations) requiring technical expertise for deployment
- Lacking **zone-specific granularity** needed for large production floors with varying microclimates

How This Project Addresses the Gaps

1.Closed-Loop-Control-System

Implements relay-triggered actuators for:

- **Exhaust fans/AC activation** when temperature exceeds fabric-specific thresholds
- **Humidifier engagement** when RH drops below 40% (critical for natural fiber processing)
- **Machine shutdown protocols** during explosive gas detection

Textile-Optimized-Hazard-Response

Features:

- **Multi-stage gas alerts**: Initial exhaust activation → Supervisor SMS → Full evacuation alarms
- **Process-aware thresholds**: Different setpoints for dyeing vs. weaving zones

3.Modular-Scalability

Adopts an expandable architecture from the Smart Citizen System2, allowing:

- **Zone-specific sensor pods** with local processing
- **Plug-and-play relay modules** for new machinery
- **JSON-configurable thresholds** for different textile processes

Validation of Approach

Benchmarking against the IEC 61850 standard1 confirms architectural robustness, while the Smart Citizen Station's open-source framework2 provides a proven foundation for sensor integration. This project uniquely merges these strengths with textile-specific control logic absent in current literature.

2.3 References

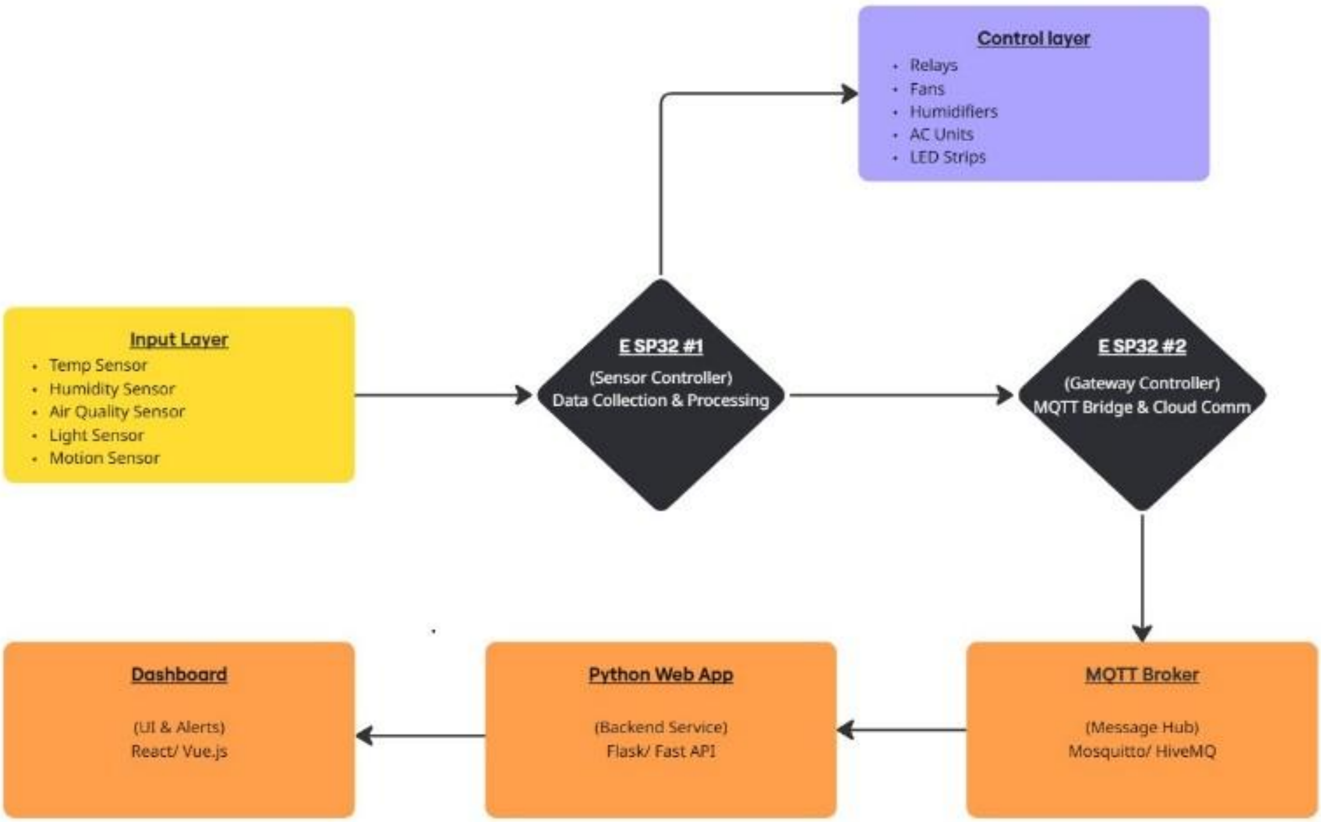
1.dong2017.pdf

2.1-s2.0-S2405844024042269-main.pdf

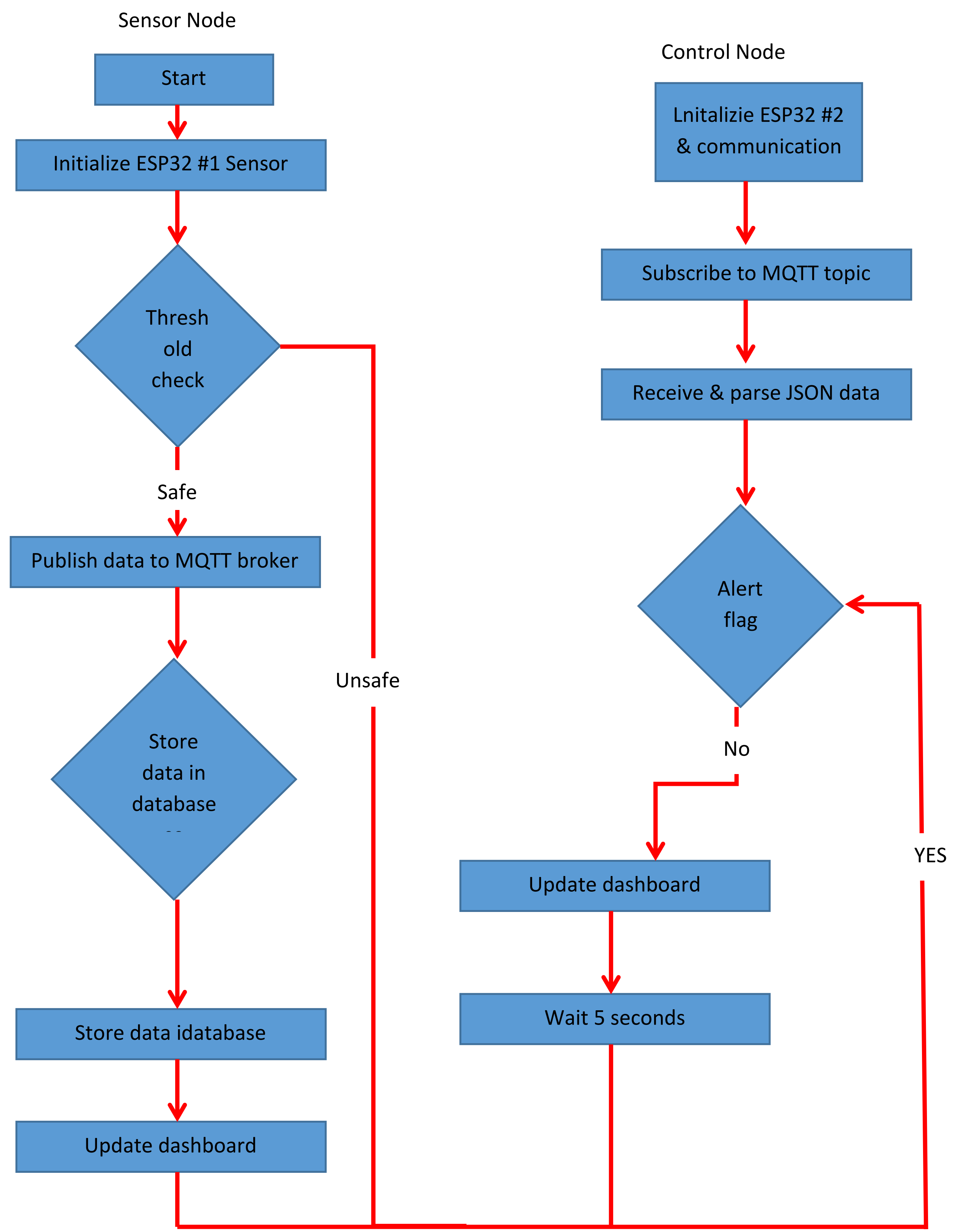
3.1-s2.0-S2468067219300203-main.pdf

3. Methodology & System Design

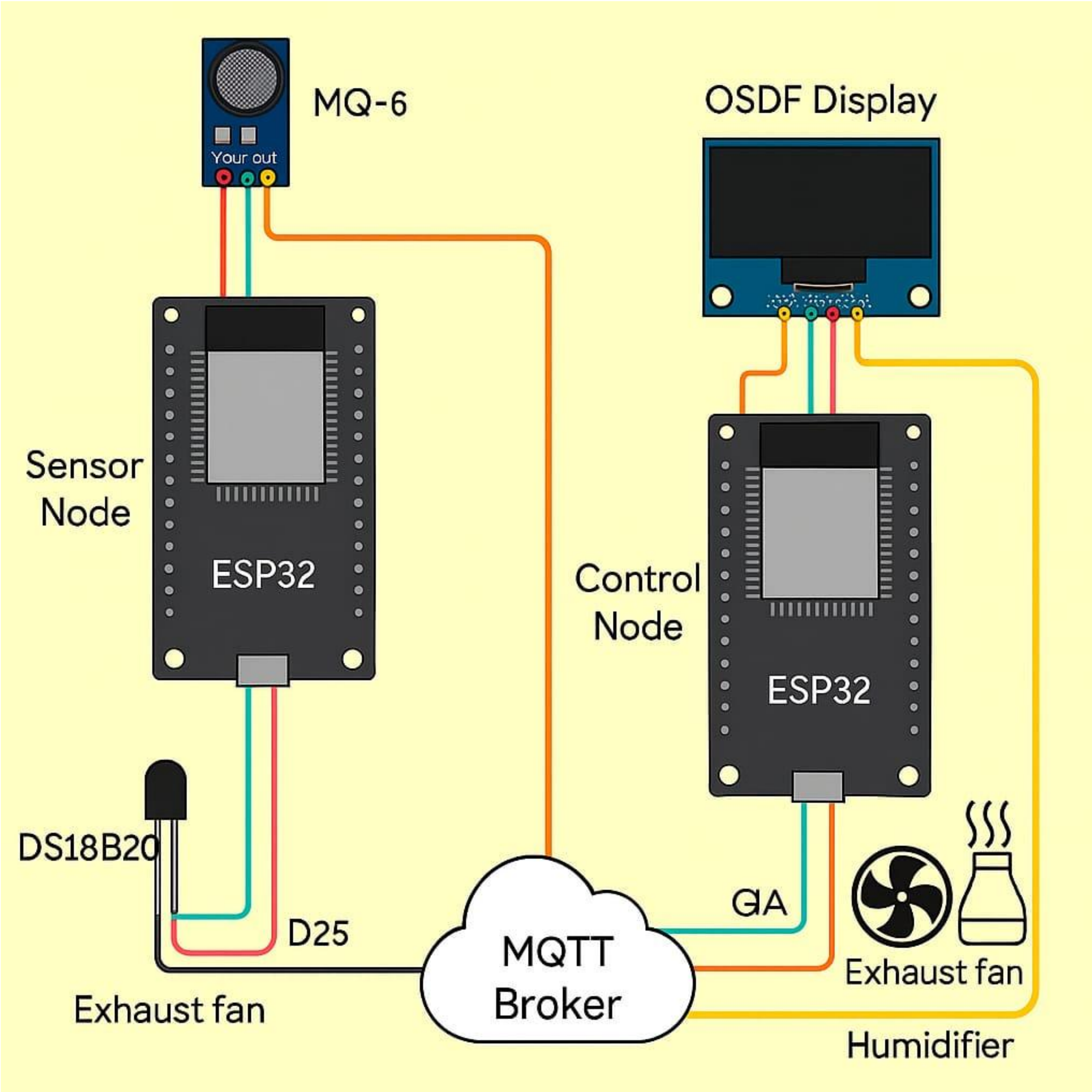
3.1 High level Diagram



3.2 Flow Chart



3.3 Wiring Diagram



4. Implementation Plan & Timeline

4.1 Task Allocation

A clear division of responsibilities is essential for efficient project execution. Below is the proposed allocation of tasks among team members:

Task	**Responsible Person(s)**	**Description**
Proposal & Planning	All	Define scope, objectives, and detailed plan
Hardware Assembly	Member A	Assemble ESP32, sensors, relays, and wiring
ESP32 Coding	Member A	Develop and test firmware for data acquisition and actuator control
Web App Development	Member B	Design and implement Python web dashboard, database, and alert system
Integration	Both	Connect hardware, firmware, and web app; ensure seamless data flow
Testing & Debugging	Both	Conduct functional, integration, and user testing; resolve issues
Final Report & Demo	Both	Document results, prepare demo, and finalize project report

4.2 Detailed Schedule

The project is structured over a ten-week period, with each phase building on the previous one to ensure systematic progress and risk mitigation. The following table outlines the timeline for each major task:

Task	Start Date	End Date	Duration (days)
Proposal & Planning	2025-06-17	2025-06-21	4
Hardware Assembly	2025-06-22	2025-06-25	3
ESP32 Coding	2025-06-26	2025-06-30	4
Web App Development	2025-07-01	2025-07-07	7
Integration	2025-07-08	2025-07-11	3
Testing & Debugging	2025-07-12	2025-08-15	3
Final Report & Demo	2025-08-16	2025-08-20	4

5. Expected Outcomes & Deliverables

At the conclusion of this project, the following outcomes and deliverables will be achieved:

- **Working Prototype:**
  A fully assembled and operational Smart Environmental Monitoring System tailored for textile manufacturing floors. The prototype will integrate ESP32 microcontrollers, temperature/humidity and air quality sensors, and relay-controlled actuators (fans, humidifiers) to demonstrate real-time monitoring and automated environmental control.
- **Embedded Firmware (ESP32 Code):**
  Well-documented C++ code for the ESP32, responsible for sensor data acquisition, actuator control, and wireless communication using MQTT/HTTP protocols. The code will include logic for threshold-based automation and robust error handling.

- **Web Application (Python):**
  A responsive Python-based web dashboard (using Flask), featuring real-time data visualization, historical trend analysis, user-configurable alerts, actuator override controls, and data export functionality.
- **Database & Data Management:**
  Implementation of a secure, structured database (e.g., SQLite) to store all environmental readings, actuator events, and user actions, supporting both live monitoring and retrospective analysis.
- **Integration & Documentation:**
  Complete integration of hardware and software components, with a detailed user manual and technical documentation covering system architecture, setup instructions, and troubleshooting guides.
- **Final Project Report:**
  A comprehensive report detailing the project background, literature review, system design, implementation process, testing results, and conclusions. The report will include diagrams, code snippets, test data, and references in IEEE format.
- **Demonstration & Presentation:**
  A live or recorded demonstration showcasing the prototype in action, highlighting key features such as real-time monitoring, automated control, and alerting. The presentation will summarize the project's objectives, methodology, and outcomes.

These deliverables will collectively demonstrate a scalable, cost-effective, and industry-specific IoT solution for smart environmental monitoring and automation in textile manufacturing environments1.

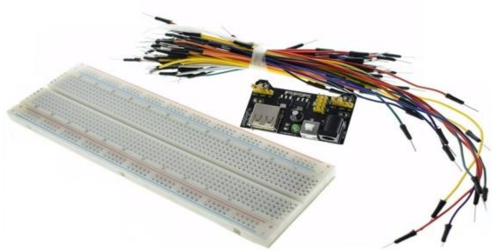
6. Budget/Resources

Hardware Components

Item	Qty	Unit Price(LKR)	Total(LKR)	Justification
ESP32 Dev Board	2	2000	4000	Central microcontroller for sensor integration and wireless communication
DHT22 Temp/Humidity Sensor	1	800	800	Accurate temperature and humidity monitoring
Q135 Gas Sensor	1	1200	1200	Detects air quality (CO_2, NH_3, VOCs) relevant for textile environments
Relay Module (2-channel)	1	600	600	Controls exhaust fan, humidifier, or AC/HVAC
Breadboard & Jumper Wires	1 Set	500	500	For prototyping and flexible circuit assembly
Power Adapter (5V/3A)	1	700	700	Stable power supply for ESP32 and relays
Exhaust Fan (small)	1	2000	2000	For air extraction during high temp/gas events
Humidifier (mini)	1	2500	2500	Adds humidity when needed
Misc. (screws, casing, PCB)	1	1000	1000	Enclosure, mounting, and PCB prototyping
Total			13,300	

DHT22 Temperature/Humidity Sensor

ESP32 Development Board



Breadboard & jumper wires

MQ-135 Gas Sensor

7.Description of Project Resources

7.1. Software / Cloud Resources

- **Python/Flask Web App**
  The web application will be developed using Python with the Flask framework. Flask is an open-source micro web framework that is lightweight and easy to use, making it ideal for rapid development of dashboards and APIs. Since it is open-source, there are no licensing costs involved, which helps keep the project budget-friendly.
- MQTT Broker
  MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol widely used for IoT communication. For this project, the MQTT broker can be set up using Mosquitto, an open-source MQTT broker that can be hosted locally or on a server. Alternatively, cloud-based MQTT brokers (such as HiveMQ Cloud or CloudMQTT) offer minimal or free tiers suitable for prototyping and development, enabling easy remote communication between ESP32 devices and the web app.
- **Database: SQLite**
  SQLite is a serverless, self-contained, open-source relational database engine. It is perfect for small to medium-sized projects and can be embedded directly within the Flask application. Using SQLite eliminates the need for a separate database server, simplifying deployment and maintenance while storing sensor data and logs efficiently.
- **Cloud Hosting (Optional)**
  To make the web dashboard accessible remotely, the Flask app can be deployed on cloud platforms offering free tiers such as Heroku, PythonAnywhere, or AWS Free Tier. These platforms provide easy deployment, scalability, and basic hosting services without upfront costs, ideal for prototypes or demos.
- **SMS/Email Alerts**
  For sending real-time alerts to users, services like Twilio can be integrated. Twilio offers APIs for SMS and email notifications with a free trial or limited free tier, allowing the system to notify users about critical events without additional hardware. This integration enhances user engagement and responsiveness.

7.2. Human Resources

- **Project Lead / Embedded Developer**

  This role is responsible for the hardware integration and development of ESP32 firmware. The developer will handle sensor interfacing, data acquisition, MQTT communication, and ensure reliable operation of the embedded system.

- **Web Developer**

  The web developer will focus on creating the dashboard interface, managing the backend database, and implementing the alert system. Their responsibilities include designing user-friendly visualizations, handling data storage, and integrating notification services.

- **Testing & Documentation**

  Both team members will collaborate on thorough testing of the entire system to ensure reliability and robustness. Additionally, they will document the design, code, and user manuals to facilitate future maintenance and scalability.

7.3. Other Resources

- **Internet Access**

  A stable internet connection is essential for cloud hosting, remote dashboard access, MQTT broker communication, and sending alerts via SMS/email. Internet access enables seamless data flow between devices and users.

- **Workspace**

  A basic lab or desk setup equipped with tools for assembling hardware components, programming ESP32 boards, and testing the system is required. This workspace should provide a comfortable environment for development, debugging, and integration.

8.Summary

The proposed project aims to develop a Smart Environmental Monitoring System specifically designed for textile manufacturing floors using Internet of Things (IoT) technologies. The system will utilize ESP32 microcontrollers integrated with industrial-grade sensors to monitor key environmental parameters such as temperature, humidity, and air quality (including gases like CO_2 and ammonia). These factors significantly affect fabric quality, machine performance, and worker safety in textile production environments. Unlike existing solutions, this project introduces a modular, scalable, and automated control system that can actively respond to hazardous conditions—such as triggering exhaust fans, activating humidifiers, or shutting down machinery. A web-based dashboard built using Python and Flask will enable real-time data visualization, historical analysis, and threshold-based alerts through SMS or email. The project is designed to be cost-effective, using open-source tools and affordable hardware, and will be completed within a 10-week timeline. Overall, the system supports smarter, safer, and more efficient textile production in alignment with Industry 4.0 principles.

9. References

1. A. Petrillo, M. Rehman, and I. Baffo, "Digital and Sustainable Transition in Textile Industry through Internet of Things Technologies: A Pakistani Case Study," *Applied Sciences*, vol. 14, no. 13, p. 5380, 2024. [Online]. Available: https://doi.org/10.3390/app14135380 1

2. Ismail, "Environmental Condition Monitoring System of Textile Industry for Sustainable Development Goal," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2S2, pp. 1–6, 2019. [Online]. Available: https://www.ijitee.org/wp-content/uploads/papers/v9i2s2/B11021292S219.pdf 23

3. N. Saha, M. Aulia, M. M. Rahman, and M. S. A. Khand, "IoT-Driven Cloud-based Energy and Environment Monitoring System for Manufacturing Industry," *arXiv preprint*, arXiv:2404.11771, 2024. [Online]. Available: http://arxiv.org/pdf/2404.11771.pdf 4
4. "IoT Environmental Monitoring: A Comprehensive System Architecture," *RF Wireless World*, 2025. [Online]. Available: https://www.rfwireless-world.com/articles/iot-applications/iot-based-environmental-monitoring-system-architecture 5

5. H. Ahmed, M. Al Bashar, M. A. Taher, and M. Rahman, "Enhancing Textile Quality Control with IoT Sensors," *International Journal of Management Information Systems and Data Science*, vol. 1, no. 1, pp. 7-18, 2024. [Online]. Available: https://globalmainstreamjournal.com/index.php/IJMISDS/article/download/113/89/90 6

6. "IoT Hardware for Environmental Monitoring," *TelecomWorld101.com*, 2025. [Online]. Available: https://telecomworld101.com/iot-hardware-for-environmental-monitoring/