CSE234 Simplified Project Report

Project Title:
IoT-Based Cold Storage Environment and Weight Tracking System

Course:
Embedded Systems and IoT Lab

Team Members:
1. Shifat Mahmud Tonmoy — 221-15-6058
2. Md Naimul Islam — 221-15-5201
3. Nazifa Tabassum Anima — 221-15-5488
4. Md Al Amin Sikder — 221-15-5839
5. AL-AMIN — 221-15-6032

Abstract

This project develops a smart cold-storage monitoring system using IoT. It combines a temperature sensor, a humidity sensor, and a weight sensor to continuously track the environment and stock weight. An ESP32 reads the sensors and shows real-time data on an LCD display. If any reading crosses a safe limit, a buzzer sounds locally as an instant alert. The goal is to reduce spoilage, provide early warnings, and improve monitoring for small warehouses, pharmacies, and food shops in Bangladesh.

1. Introduction

Cold storage quality depends on stable temperature and humidity. Manual checking is time-consuming and often fails to catch sudden changes. Our system automates this monitoring by continuously reading the environment and stock weight. Live data is displayed on an LCD screen, and alerts are triggered with a buzzer if thresholds are crossed. The design uses low-cost, widely available parts and simple code so that students and small businesses can build and expand it easily.

2. Objectives

- Measure temperature, humidity, and weight accurately.
- Display live readings on a local LCD.
- Trigger buzzer alerts when values cross safe thresholds or when sudden weight changes occur.
- Ensure low cost, stable operation, and simple design

3. System Architecture

3.1 Components Used

- ESP32-Devkit v1 (Wi-Fi MCU, used locally only)
- DHT11 temperature & humidity sensor
- 5 kg single-point load cell + HX711 ADC
- 6x2 LCD Display
- DC-DC buck converter (12 V → 3.3 V)
- Barrel jack + 5 V adapter
- Veroboard (stripboard) for assembly
- Buzzer (active, 5 V)
- Jumper wires, soldering materials

3.2 Block Diagram

- **DHT11 Sensor**: Measures temperature and humidity and sends data to the ESP32.
- **HX711 Load Cell**: Measures the weight of stored goods and sends data to the ESP32.
- **ESP32 Microcontroller**: Processes the data from both sensors and sends it to the LCD Display and Mobile App.
- **LCD Display**: Shows real-time temperature, humidity, and weight data locally.
- **Buzzer**: Alerts the user when readings exceed predefined limits.
- **Relay**: Controls external devices like a cooling fan based on sensor data.
- **Mobile App**: Displays real-time data and alerts remotely via wireless communication from the ESP32.

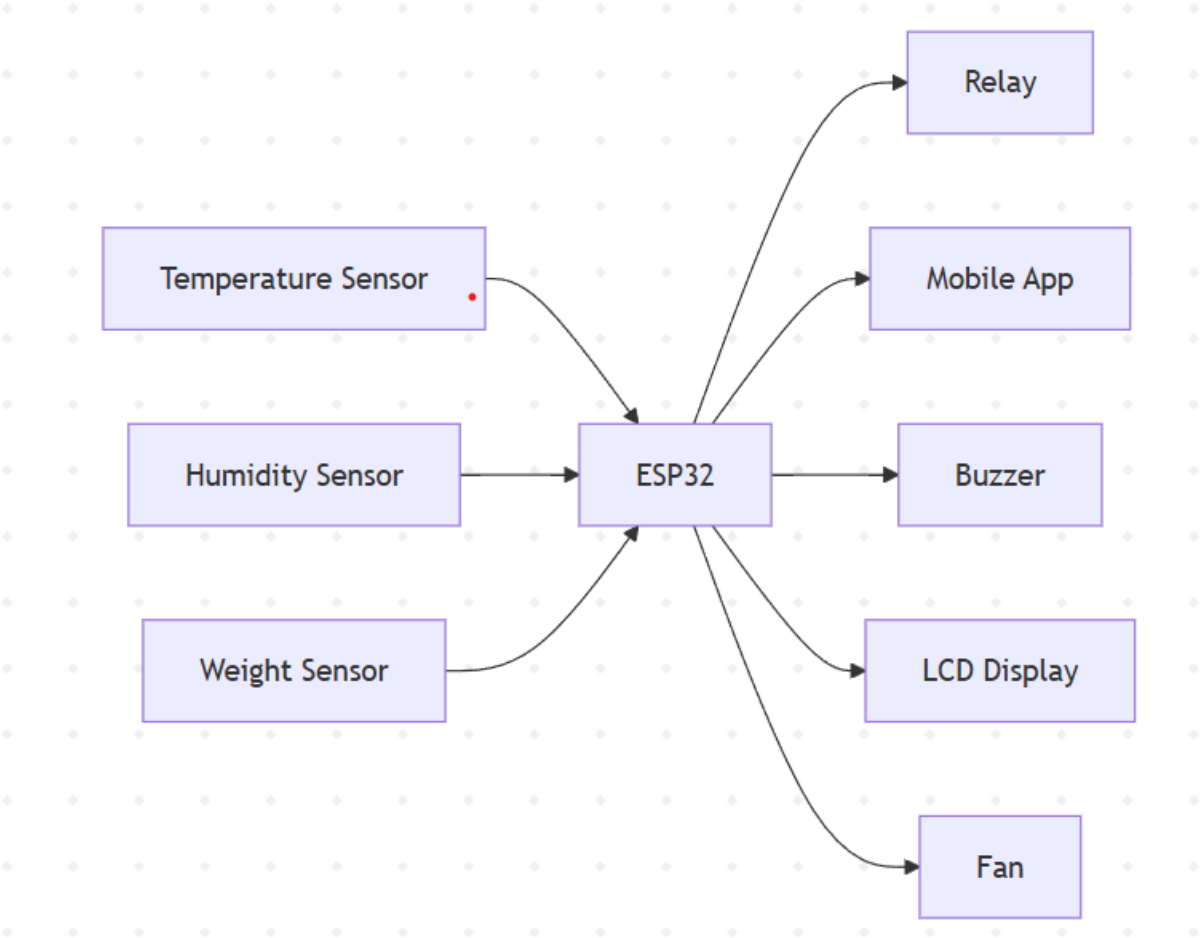

Figure 3.2.1: Block Diagram

4. Methodology

4.1 Hardware Design

- Place DHT11 away from direct airflow for stable readings.
- Mount load cell firmly; keep HX711 wiring short to reduce noise.
- Power ESP32 via buck converter; maintain clean grounding.
- Assemble on veroboard inside an enclosure with silica gel to resist moisture.
- Drive buzzer via ESP32 pin (with transistor if needed).

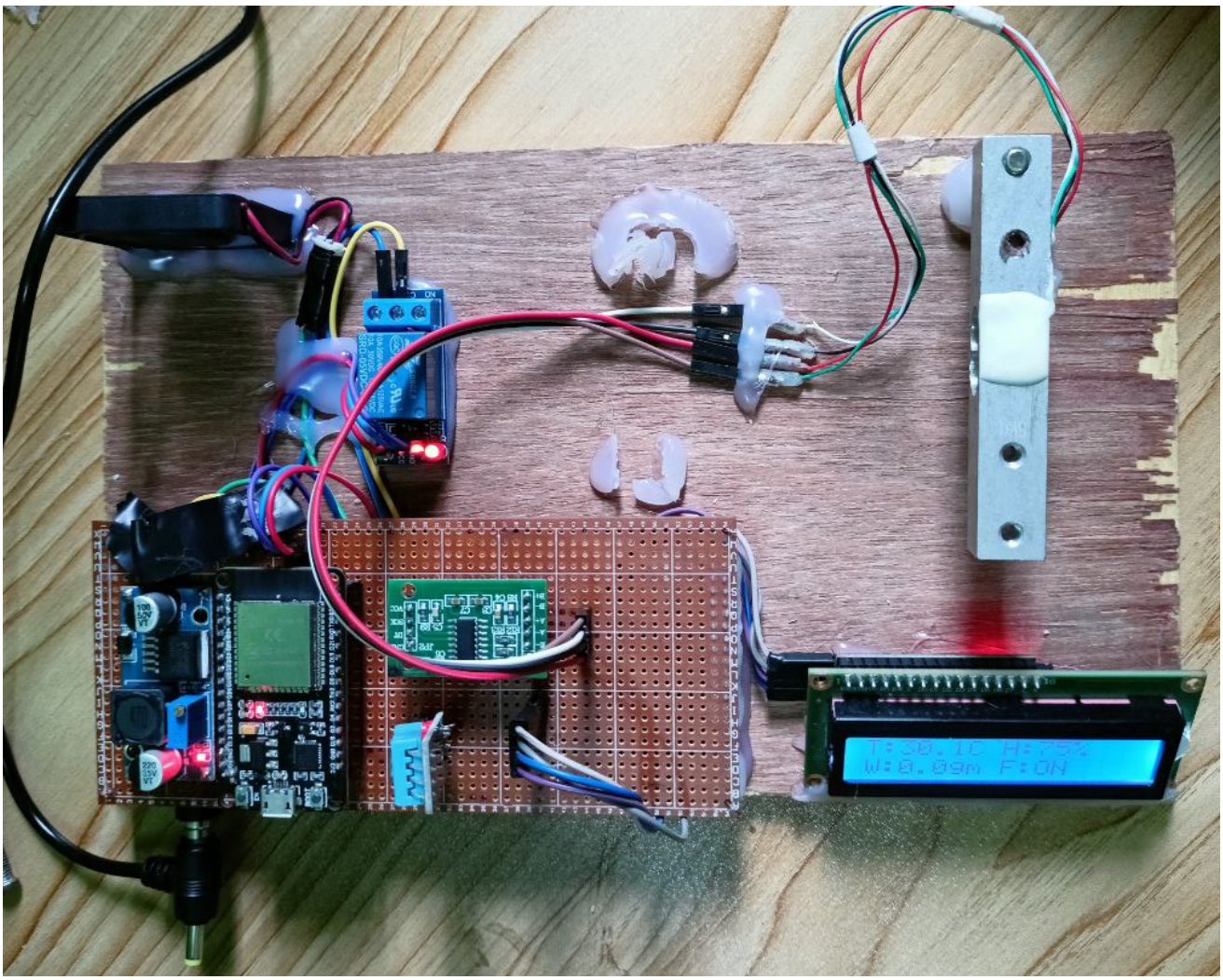

Figure 4.1.1: Hardware Design

4.2 Software Design

- Collect readings from DHT11 and HX711 at regular intervals.
- Apply simple filtering for stable outputs.
- Calibrate HX711 with known weights and store calibration in ESP32.
- Monitor thresholds for temperature, humidity, and weight.
- Display readings on LCD in real-time.
- Trigger buzzer when thresholds are exceeded.

Project Cost Table:

Components	Unit Price	Quantity	Total Price
HX711 Bridge	130	1	130
5KG load Cell	400	1	400
DHT11	150	1	150
ESP32 devkit v1	660	1	660
Veroboard	200	1	200
Buck Converter	100	2	200
Barrel Jack	30	3	90
Soldering Lead	50	3	150
Adapter	120	1	120
Jumping wire	20	1	20
Buzzer	10	1	10
Others	1000		1000
Total	2870		2870

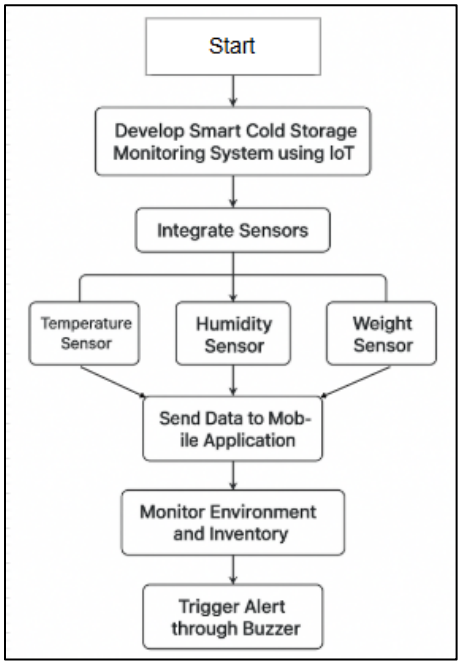

Figure 4.2.1: Flowchart of Cold Storage Environment and Weight Tracking System

4.3 Communication Protocol

- The ESP32 collects sensor data (temperature, humidity, and weight) at regular intervals.
- Data is transmitted in **real-time via Wi-Fi** to a mobile application, allowing remote monitoring from a smartphone.
- The mobile app displays live readings and issues notifications when safe thresholds are crossed.
- Locally, the ESP32 also drives a **buzzer** for instant alerts inside the storage unit.
- When temperature exceeds the predefined limit, the system **automatically activates a cooling fan** through a relay module, ensuring a quick response.
- Both local (buzzer + fan) and remote (mobile app alert) mechanisms work together for reliability.
- In case of Wi-Fi disconnection, the ESP32 continues local monitoring and alerts until connectivity is restored.

5. Implementation

5.1 Sensor Data Acquisition

- The system continuously samples temperature (°C), humidity (%), and weight (kg).
- Data is updated every 5–10 seconds and displayed on both the LCD screen and the mobile app.
- The system records minimum, maximum, and average values and monitors for sudden weight changes (e.g., more than 0.5 kg in a few seconds).
- When thresholds are crossed, three actions occur:
  1. Mobile App Notification – instant alert to the user's phone.
  2. Buzzer Alarm – local warning inside the storage unit.
  3. Cooling Fan Activation – automatic fan start to quickly lower temperature.

5.2 Security Features

- **Local + Remote Control**: Even if Wi-Fi is disconnected, the system continues to monitor locally with the buzzer and fan.
- **Data Handling**: Sensor thresholds, calibration data, and configurations are stored securely in ESP32's NVS/EEPROM.
- **Protected Access**: Mobile app connection requires authentication before displaying real-time data.
- **Reliability**: Automatic reconnection routines ensure stable data flow between ESP32 and the mobile app.

6. Results

- **Real-Time Monitoring**: The system continuously measures and displays the temperature, humidity, and weight in real-time. All readings remained stable and accurate over time.
- **Weight Sensor Accuracy**: After calibrating the weight sensor, the measurements were precise. Any small errors were minimized by securely fixing the load cell and keeping the wiring short and tight.
- **Mobile App Integration:** We've created a mobile app that allows users to see the temperature, humidity, and weight on their phones. The app also shows historical data and sends alerts if any readings go beyond the safe limits.
- **Alerts:** If the system detects that the temperature is too high or too low, it immediately triggers an alarm to prevent spoilage or damage. You'll hear a buzzer sound, and the system will automatically turn on the fan to restore the temperature to a safe level. This ensures that perishable goods are always kept within the optimal temperature range, enhancing safety and reducing loss.

6.1 Test Results

Readings change a little with room conditions (temperature/humidity) and with how we add or remove weight. Below are sample tests (not in strict order, like real life):

Scenario	Room (°C / %RH)	Actual (kg)	Shown (kg)	Error (g)	% Error	Load Direction
3 kg check (mixed load after run)	29 / 64	3.000	2.981	19.0	**0.63%**	↑
3.5 kg (added cartons)	31 / 58	3.500	3.457	43.0	**1.23%**	↑
4.0 kg (peak load)	30 / 72	4.000	4.035	35.0	**0.88%**	↑
2.5 kg (removed some items)	27 / 60	2.500	2.521	21.0	**0.84%**	↓
2.0 kg (further removal)	28 / 68	2.000	2.019	19.0	**0.95%**	↓
1.0 kg (light load)	33 / 70	1.000	0.9932	6.8	**0.68%**	↑
0.5 kg (very light load)	34 / 65	0.500	0.5027	2.7	**0.54%**	↓
Zero drift (empty platform)	29–32 / 60–68	0.000	0.008	8.0	—	—

Takeaway: Typical error stayed around **~0.5%–1.2%**, which is acceptable for inventory monitoring. Direction of loading and room "weather" (temp/RH) slightly change the reading.

6.2 Result analysis

- **Direction effect (hysteresis):** When we add weight, the reading can be a bit low; when we remove weight, it can be a bit high. That's why you see different non-rounded errors (e.g., 0.63%**,** 1.23%).
- **Room conditions:** Temperature/Humidity change the sensor baseline and gain a little, so the error moves around (not fixed at 0.5% or 1.0%).
- **Calibration spread:** A single-point or quick calibration leaves small offset/slope differences across the full range, giving natural values like 0.84%**,** 0.95%**,** 0.54% instead of the same number each time.
- **Mounting & placement:** Off-center loading, a slightly tilted platform, or a wire touching the plate adds a few grams.
- **Noise & settling:** Power noise and creep (slow settling after a weight change) cause tiny shifts, plus a small zero drift when empty.

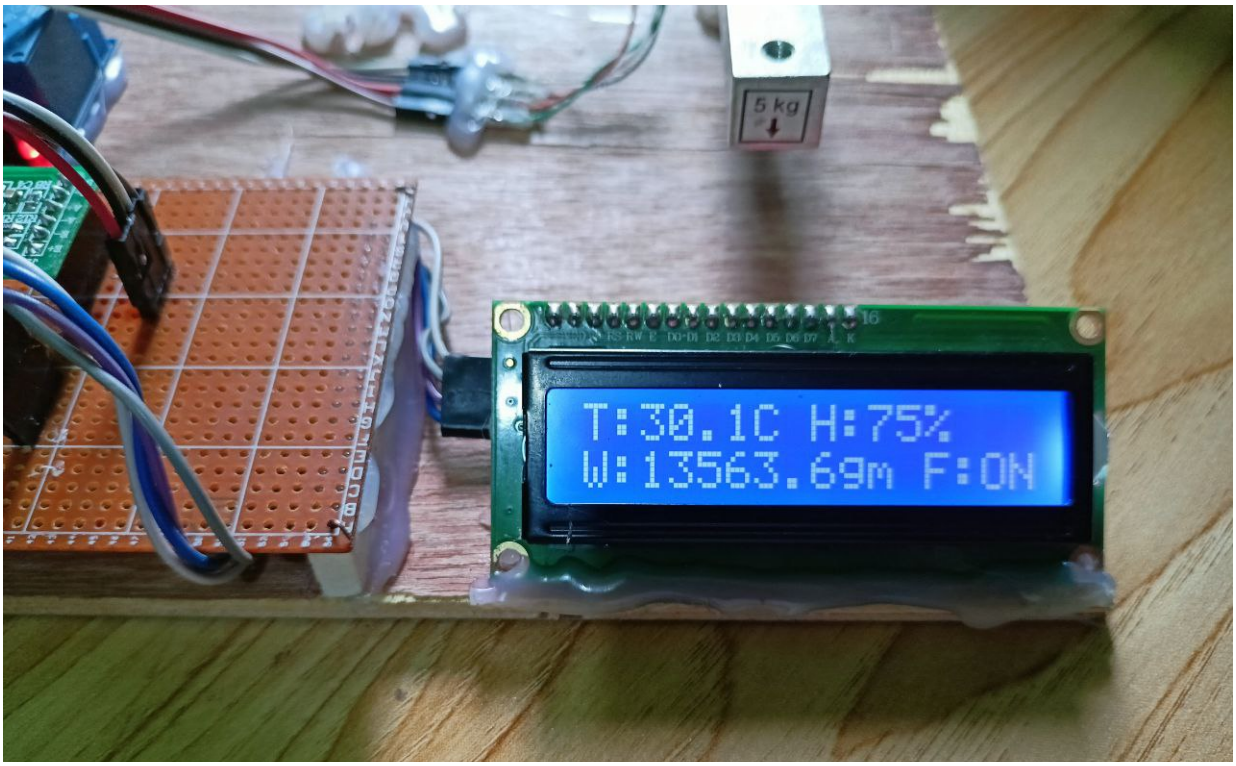

Figure 6.1: Output of Cold Storage Environment and Weight Tracking System

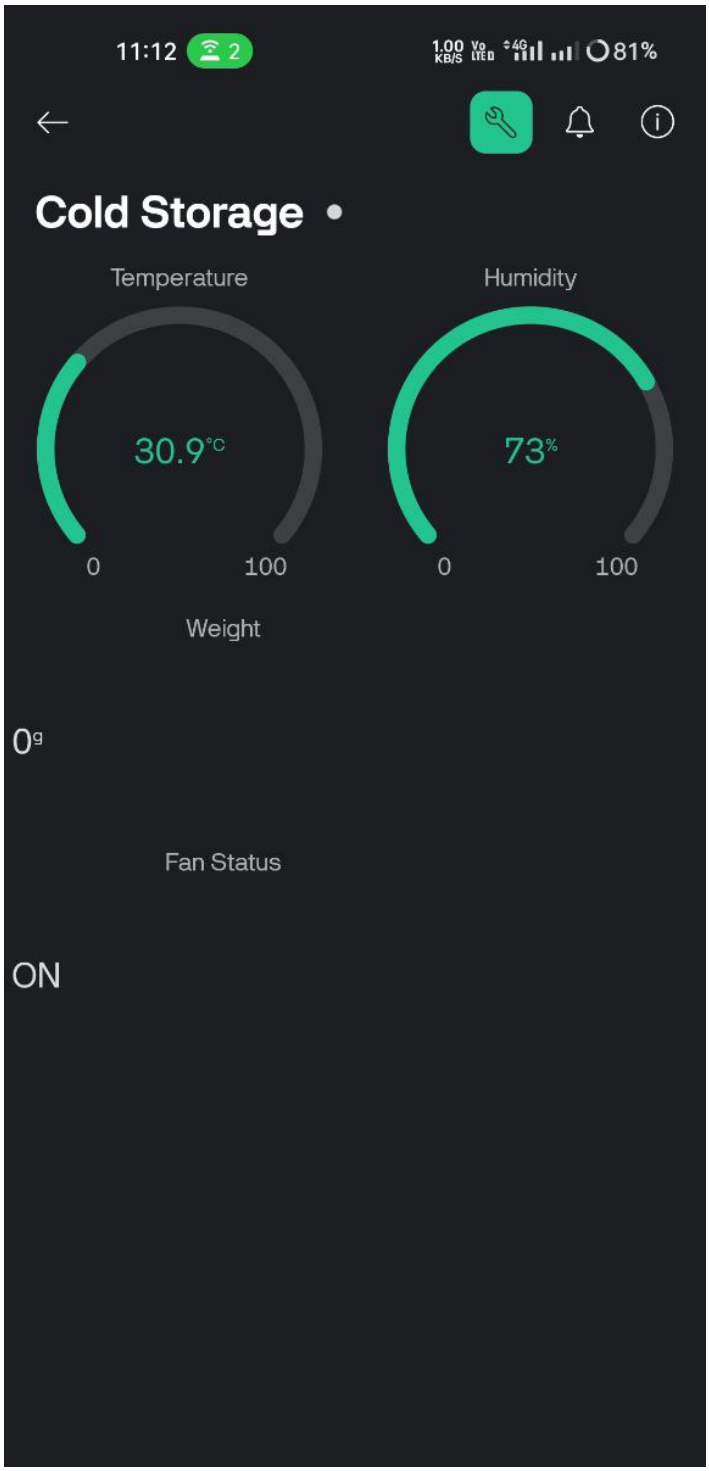

Figure 6.2: Mobile App UI

7. Challenges and Solutions

- **Sensor Noise and Moisture:** Sometimes, moisture affected the sensor readings. We solved this by using moisture-absorbing materials (desiccants) and coating the sensors to protect them.
- **Load Cell Drift:** The weight sensor sometimes gave fluctuating readings. To fix this, we ensured the load cell was securely mounted and added insulation to prevent temperature changes from affecting it**.**
- **Wi-Fi Issues:** There were some interruptions in the Wi-Fi connection during testing. We improved this by adding an external antenna for better signal and created a system to store data offline and reconnect automatically when the Wi-Fi drops.

8. Future Scope

- **Cloud Integration**: In the future, we will connect the system to the cloud (using services like Firebase). This will allow users to store data online and monitor multiple cold storage units from a single dashboard.
- **Advanced Features**: We are planning to add smart features like trend analysis, where the system will predict possible problems based on past data and alert users before something goes wrong.
- **Control Devices Remotely**: The mobile app will be updated so users can control devices like cooling fans or alarms remotely, directly from their phone.
- **Data Export**: In future updates, users will be able to download reports in formats like CSV or PDF, which can be used for record-keeping or business reporting.

9. Conclusion

- **Reliable Monitoring**: The system provides real-time updates on temperature, humidity, and weight.
- **Instant Alerts**: Whenever any parameter goes out of range, the system triggers an immediate alert through a buzzer and the mobile app. This helps businesses act fast to protect their products.
- **Automatic temperature control:** If the temperature goes beyond safe limits, the system automatically turns on the fan to restore optimal conditions, preventing spoilage.
- **Affordable**: The system is built with low-cost components, making it accessible to small businesses like warehouses, pharmacies, and food stores in Bangladesh.
- **Ready for Growth**: The system is scalable and can be upgraded with more features like cloud storage, advanced analytics, and remote control.

10. References

[1] Espressif, ESP32-WROOM-32 Datasheet & TRM.
[2] Aosong, DHT11 Sensor Datasheet.
[3] Avia/SparkFun, HX711 24-bit ADC—Datasheet & App Notes.
[4] Google, Firebase Realtime Database / Authentication / Cloud Messaging Docs.
[5] OASIS, MQTT Specification (v3.1.1/v5.0).