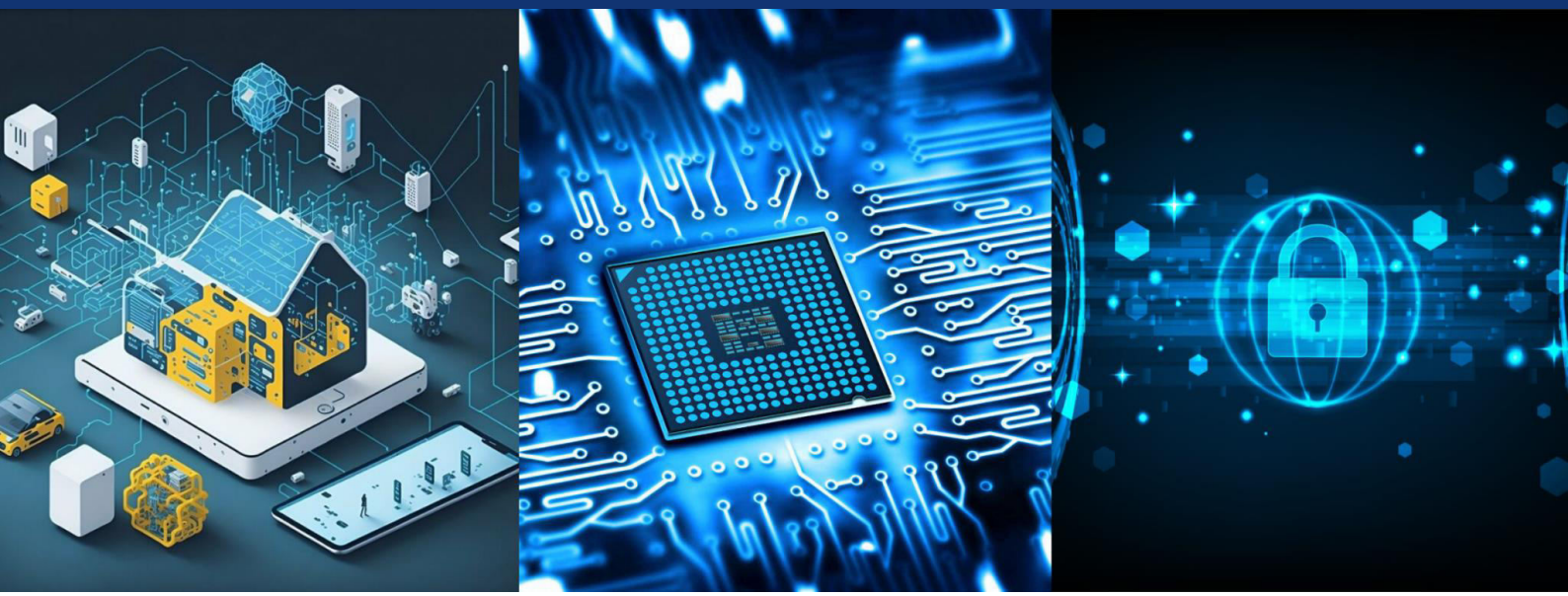
IJIRCCE

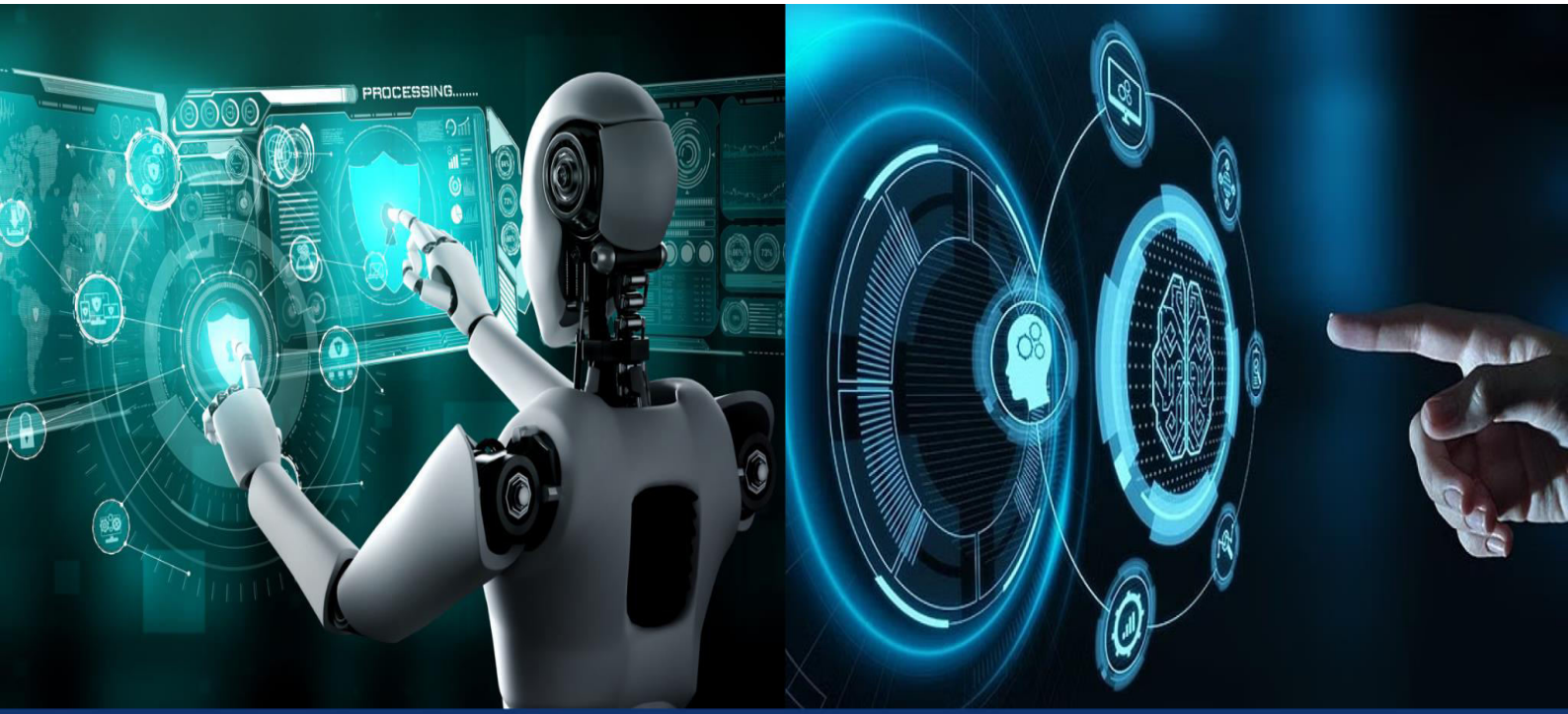
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 10, October 2026

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IoT-Based Smart Warehouse Automation System with RFID Grain Sorting, Keypad Access Control and Cloud Alerts

Darshan N S[1], Darshan S K[2], Ganesh T M[3], Gowri H K[4], Suma K G[5]

B.E. Students, Department of Electronics & Communication Engineering, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India[1,2,3,4]

Assistant Professor and Project Guide, Department of Electronics & Communication Engineering, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India[5]

ABSTRACT: Small grain stores in India are still run on paper registers, a lock and the judgement of whoever is on duty. This paper reports a bench prototype that replaces those habits with sensing, authentication and remote reporting for roughly ₹5,000 in parts. A NodeMCU (ESP8266) drives the DHT11, the MQ-2 module, two SG90 servos, a fan relay, a buzzer and an SSD1306 display, and it is the only board that talks to the cloud. An Arduino Uno reads a 4×3 keypad and an RC522 reader and forwards short serial messages to it; a second NodeMCU is set aside for an HX711 load-cell amplifier. A four-digit PIN that the owner can change in Firebase opens the door for five seconds. One of two registered RFID cards swings a sorting gate to the rice or wheat position for ten seconds. The fan starts above 45 °C and the buzzer follows the MQ-2 digital output. State changes are written to Firebase, shown on a login-protected React dashboard and relayed to Telegram. Bench tests showed every function responding as designed. We also record the weaknesses we found, among them the lack of fan hysteresis and the placeholder weight values, so that later work can address them.

KEYWORDS: Internet of Things, ESP8266, RFID, grain sorting, Firebase, Telegram, access control, warehouse monitoring.

I. INTRODUCTION

A warehouse is where a delay or a mistake in production turns into a loss for the consumer. In small and medium stores the routine is still manual. Stock goes into a register, someone walks in to feel whether the room is too hot, and a lock plus a watchman handle security. That routine breaks down at night, on holidays and whenever the person on duty is busy. For grain the cost is easy to see: a few warm, damp days can start mould, and a smouldering sack or a gas leak can ruin an entire consignment before anyone notices.

Cheap Wi-Fi microcontrollers and hobby-grade sensors have made it affordable to automate part of this. Most published prototypes watch temperature and humidity and stop there. We wanted one low-cost build that also decides who may enter, tells two grain types apart, and tells the owner what happened without the owner being present. The prototype therefore does four things: it checks a PIN before the door moves, it routes grain by RFID card, it watches temperature, humidity and combustible gas and acts on them, and it publishes every change to a dashboard and a Telegram chat. Section II reviews earlier work, Section III states the problem, Section IV explains the design and Section V the build and its cost. Section VI discusses what we observed on the bench, including shortcomings, and Section VII closes the paper with future work.

II. RELATED WORK

Warehouse monitoring with IoT has been tried many times, usually with an Arduino-class board, a few environmental sensors and a free cloud dashboard. Mohanraj et al. [1] fitted a grain store with vibration, humidity, temperature and fire sensors, and their main claim was that logging the readings in the cloud beats the way granaries have traditionally been managed. Rao et al. [2] began from the weaknesses of what many stores already own: periodic walk-through

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

checks, a smoke detector, and separate temperature or humidity gadgets that neither trigger anything nor feed a common record. Their answer was a single monitoring platform.

Kapile et al. [3] paired a DHT11, an MQ-2 and flame sensors with an Arduino, plotted the values on ThingSpeak and let the controller start fans and extinguishers. Borwankar et al. [5] followed a similar sensor set and used relays for fans and lights and a buzzer for warnings. Arasavalli et al. [4] described the general chain of acquisition, processing, communication and decision, carried over Wi-Fi or LoRa to services such as Blynk or ThingSpeak. For wider background we relied on the IoT surveys of Gubbi et al. [6] and Atzori et al. [7] and on Want's overview of RFID [8].

What these papers share is a focus on climate and fire. None of them, as we read them, controls who may enter, identifies the goods themselves, or moves them, and alerts rarely go beyond a web page. Table I sets the gaps next to this work.

Feature	[1], [2]	[3]–[5]	This work
Temp./humidity/gas sensing	Yes	Yes	Yes
Cloud logging	Yes	Yes	Yes
Automatic actuation	No	Yes	Yes
Phone messaging	No	No	Yes
PIN-controlled door	No	No	Yes
RFID-based routing	No	No	Yes
Load-cell hardware	No	No	Fitted*

TABLE I. Features Reported in Surveyed Work and in This Prototype.

No = not described in the summaries we reviewed. * Live weight values are not yet merged into the cloud record; see Section VI-D.

III. PROBLEM STATEMENT AND OBJECTIVES

The faults we set out to reduce are the ones a visit to a small store makes obvious. Stock figures depend on handwriting. Nobody measures temperature, humidity or air quality between visits. The door is protected by a key that anyone can copy or borrow. The owner has to be on the premises to know anything, and repairs happen after something fails rather than before. For this prototype we narrowed the work to three objectives:

1) limit entry to people who know a PIN, and make a failed attempt audible and visible;
2) remove hand-sorting mistakes between two grain types by letting an RFID card decide the position of a sorting gate; and
3) limit spoilage and accidents by watching temperature, humidity and combustible gas, reacting locally and reporting remotely.

IV. SYSTEM DESIGN

A. Architecture

The block diagram is given in Fig. 1. Three boards share the work. The main NodeMCU reads the DHT11 and the MQ-2, drives the door servo, the sorting servo, the fan relay and the buzzer, updates the OLED, and handles Wi-Fi and Firebase. An Arduino Uno looks after the 4×3 keypad and the RC522 reader. The reason for the split is pin count: the keypad matrix needs seven lines and the SPI reader about six more, which would have used up most of the eleven GPIOs on a NodeMCU. A second NodeMCU is dedicated to the HX711 so that the load-cell readout does not compete with the Wi-Fi stack. A 12 V adapter and a speed controller supply the vibration motor that keeps grain moving in the chute.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

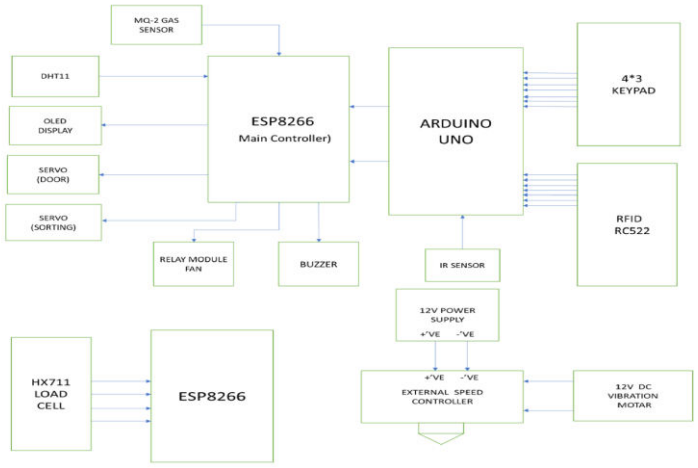


Fig. 1. Block diagram of the proposed system.

Table II lists the pin allocation taken from the firmware, and Fig. 2 shows the wiring. The two microcontrollers exchange data over a single serial line at 9600 baud. The Uno sends one text line per event, and the NodeMCU tells the two kinds apart by length, as shown in Table III.

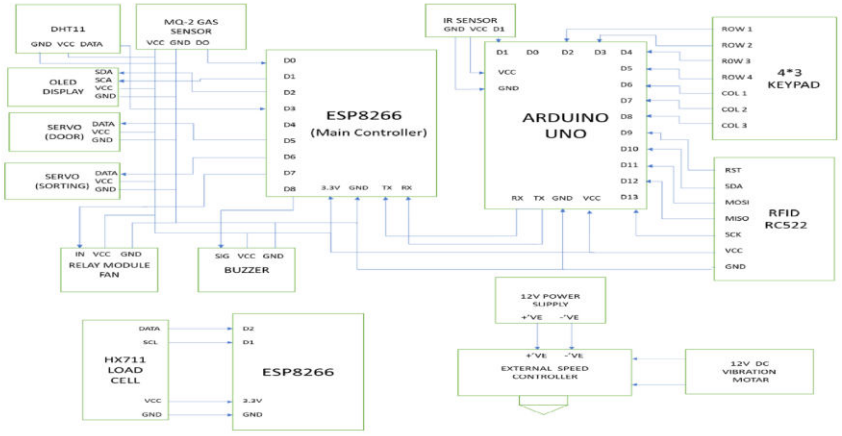


Fig. 2. Schematic diagram of the prototype.

Signal	Board	Pin
Door servo	NodeMCU	D5
Sorting servo	NodeMCU	D6
Fan relay	NodeMCU	D7
Buzzer	NodeMCU	D8
DHT11 data	NodeMCU	D3
MQ-2 digital out	NodeMCU	D0
OLED (I²C, 0x3C)	NodeMCU	SDA/SCL
Keypad rows / columns	Uno	2–5 / 6–8
RC522 SS / RST	Uno	10 / 9

TABLE II. Pin Allocation in the Firmware

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Line received	Meaning	Action
Four digits	PIN attempt	Compare with stored PIN
1	Card 1 read	Sorter to 0° (rice)
2	Card 2 read	Sorter to 180° (wheat)

TABLE III. Serial Messages from the Uno to the NodeMCU

B. Control Logic

Fig. 3 gives the flow chart. The firmware is a single non-blocking loop, so the routines below run side by side and use millis() timers rather than delays.

Entry. The visitor types four digits on the keypad and presses #; * deletes the last digit. The Uno sends the digits to the NodeMCU, which compares them with the PIN it last fetched from Firebase (the default is 0000, and the value is refreshed every 5 s, so the owner can change it from the cloud). A match moves the door servo to 90° for 5 s and then back to 0°. A mismatch leaves the door shut and sounds the buzzer for half a second. The door node in the database reads 1 while open, 2 after a rejected PIN and 0 otherwise.

Sorting. The sorting servo rests at 90°. A read of card 1 sends it to 0° and card 2 to 180°, where it stays for 10 s before returning. Further card reads are ignored while the gate is away from rest, which stops a second tap from disturbing a grain stream in progress. The vibration motor keeps grain from bridging in the chute.

Climate. The DHT11 is read every 3 s. A failed read (NaN) is discarded. When the temperature is above 45 °C the relay closes and the fan runs; at 45 °C or below the relay opens again.

Gas. The MQ-2 digital output is sampled every second. The module pulls the line low when the preset threshold is crossed, and the buzzer then sounds for as long as the line stays low.

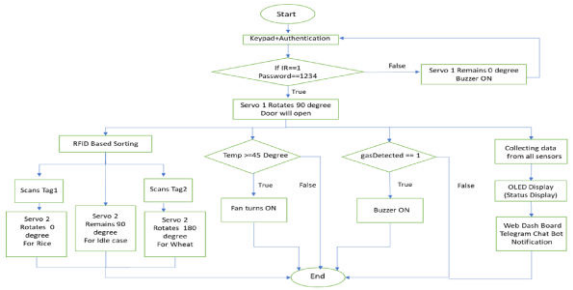


Fig. 3. Flow chart of the control logic

C. Cloud and User Interface

The NodeMCU authenticates to Firebase Realtime Database with an e-mail account and writes the values under one parent node (Table IV). The dashboard is a React page that listens to those nodes. It asks for the owner's e-mail address and password, and its login button stays disabled until both fields contain text. Each card on the page shows alerts, gas, humidity, temperature, door, PIN and the state of the fan and dispenser.

Telegram is handled in the same page. When the dispenser node changes from idle to 1 or 2, the page sets an alert node to 1 and sends one summary message to the owner's chat; the flag is cleared when the node returns to 0, so a single sorting event produces a single message. The bot also answers one-word queries such as temperature, gas, humidity, door, fan or pin with the current value.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Node	Type	Values
Temperature	float	°C
Humidity	float	% RH
Fan	string	0 off, 1 on
Gas	int	0 clear, 1 detected
Door	string	0 shut, 1 open, 2 rejected
Dispenser	string	0 idle, 1 rice, 2 wheat
Weight1, Weight2	int	weight readout
Pin	string	current PIN (read by device)

TABLE IV. Database Nodes Under /WareHouse

V. IMPLEMENTATION

A. Hardware

The main parts and their roles are in Table V. A few datasheet figures from the project report shaped the design. The RC522 works at 13.56 MHz and reads at 2–5 cm, which suits a station where a bag or card is held to the reader. The SG90 servos take a 50 Hz PWM signal and swing about 160–180°. The DHT11 covers 0–50 °C at ±2 °C and 20–90 % RH at ±5 %, with one sample per second at best. The MQ-2 heater draws about 850 mW, so it is the largest single load on the 5 V rail besides the servos.

B. Software

Both boards were programmed from the Arduino IDE; the ESP8266 core was added through the Boards Manager. The Uno sketch uses the Keypad and MFRC522 libraries, and the card identifiers are stored in the sketch as byte arrays. The NodeMCU sketch uses the Firebase ESP Client, Servo, DHT and Adafruit SSD1306 libraries. The dashboard was written in Visual Studio Code and talks to Firebase and to the Telegram Bot API.

Component	Role
NodeMCU ESP8266 (main)	Control, Wi-Fi, actuators
NodeMCU ESP8266 (second)	HX711 acquisition
Arduino Uno	Keypad and RFID input
RC522 RFID reader	Card UID check
4×3 keypad	PIN entry
HX711 + load cell	Weight sensing
SG90 servo × 2	Door; sorting gate
DHT11	Temperature, humidity
MQ-2	Smoke, combustible gas
IR sensor	Presence at entrance
Relay + fan	Cooling
Buzzer, SSD1306 OLED	Local alarm, status
12 V vibration motor	Grain flow

C. Cost

Table VI groups the bill of materials from the report. The bill of ₹4,738 (which already contains ₹500 for the prototype stand) becomes ₹5,012 after ₹274 of tax.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Group	Items	Amount
Controllers	Uno, 3 × ESP8266	1,069
Sensors	PIR/IR, RFID, DHT11, MQ-2, load cells, vibration	1,160
Outputs	Servos, OLED, relay, fan, buzzers, motor	990
Input	Keypads	380
Cables, power, stand	USB, jumpers, adapter, prototype	1,139
Subtotal		4,738
Tax		274
Total		5,012

TABLE VI. Cost of the Prototype (₹).

VI. RESULTS AND DISCUSSION

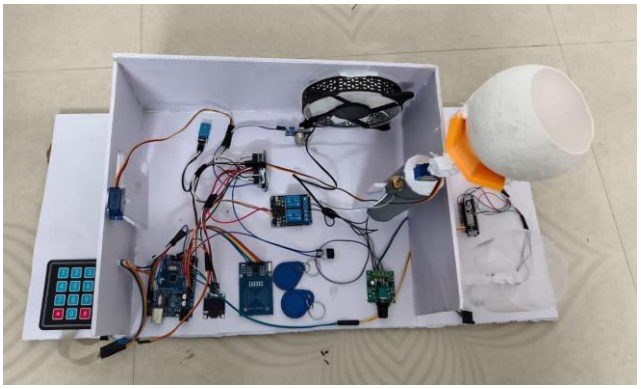


Fig. 4. Working prototype of the warehouse system.

The prototype (Fig. 4) was tested one module at a time and then as a whole. The checks were functional: we set a condition, watched the response and compared it with the design. We did not run repeated trials, so no accuracy or response-time statistics are claimed. Table VII summarises the outcome.

Module	Condition applied	Observed
Keypad + door	Correct PIN	Door opened, closed after 5 s
Keypad + door	Wrong PIN	Door shut, buzzer, node set to 2
RFID + sorter	Card 1 / card 2	Gate at 0° / 180°, then rest
DHT11 + relay	Temperature over 45 °C	Fan switched on
MQ-2 + buzzer	Smoke or LPG	Buzzer on, OLED and cloud updated
Dashboard	Wrong login	Rejected with error message
Telegram	Sorting event	One message with all values

TABLE VII. Functional Check of Each Module

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410039

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A. Observations

Climate. The DHT11 readings changed when the door was opened, when the fan ran and when the load changed, and they stayed steady over the slow drifts normal in a closed room. Once the reading passed 45 °C the relay closed. Because the sensor is rated only to 50 °C with ±2 °C error, the real room temperature at the moment of switching could lie anywhere from about 43 to 47 °C, which is acceptable for a fan but too coarse for a tighter limit.

Gas. The MQ-2 output went active for smoke and LPG, and the buzzer and OLED responded at once. The report notes that dust and humidity shift the sensor baseline, so the potentiometer threshold needs periodic resetting.

Access and sorting. The correct PIN opened the door and wrong entries did not. The first card turned the gate to the rice position and the second to the wheat position. The vibration motor prevented grain from sticking in the chute.

Remote view. The login page admitted only the stored owner account (Fig. 5). The dashboard showed live values (Fig. 6), and the Telegram bot delivered a time-stamped message with the readings (Fig. 7).



Fig. 5. Warehouse owner login page.

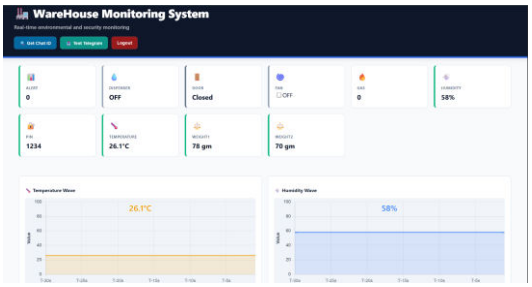


Fig. 6. Real-time monitoring dashboard.



Fig. 7. Telegram bot alert message.

B. Timing

The delays built into the firmware set how quickly each event can reach the owner (Table VIII). Gas is the quickest because it is polled once a second and the buzzer is driven directly, with no cloud round trip. A PIN changed in Firebase can take up to five seconds to take effect on the device.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Parameter	Value
DHT11 read interval	3 s
MQ-2 poll interval	1 s
PIN refresh from Firebase	5 s
Door-open duration	5 s
Sorting-gate hold	10 s
Buzzer on wrong PIN	0.5 s
Fan threshold	> 45 °C

TABLE VIII. Timing Parameters Set in the Firmware

C. Benefits

Compared with the manual routine in Section III, the prototype removes the need for routine walk-through checks, leaves a time-stamped record, layers a PIN, a presence sensor and an alert on the door, and lets the owner look in from anywhere. Everything is built from parts sold in hobby shops, so adding another sensor or actuator is cheap.

D. Limitations

Several shortcomings showed up during the work and are listed here so they can be fixed. (i) The weight values written to the cloud in the present firmware are generated placeholders; the HX711 path on the second NodeMCU is not yet merged into the record. (ii) The fan has no hysteresis, so a reading that hovers around 45 °C will make the relay chatter. (iii) The PIN is four digits, travels as plain text in the database and has no retry limit. (iv) The two card identifiers are hard-coded and UID matching alone can be cloned. (v) Wi-Fi, Firebase and bot credentials are written into the source and should be moved out before any real deployment. (vi) Cloud logging and Telegram need the internet, although the OLED and the buzzer keep working without it. (vii) Only two grain types are supported. (viii) The DHT11 has modest accuracy.

E. Applications

The design suits grain and seed stores, storerooms that need restricted entry, small cold or cool rooms where a fan is enough, and any unattended store where the owner wants a message when something changes.

VII. CONCLUSION AND FUTURE SCOPE

We built and tested a low-cost warehouse prototype that combines PIN-based entry, RFID grain routing, temperature and gas watching, automatic cooling and phone alerts on three small boards and about ₹5,000 of parts. Each function behaved as designed on the bench, and the owner could follow the state of the store from a browser or from Telegram. The prototype is not production-ready, and the limits listed in Section VI-D are the first items to address.

Next steps are to merge the live HX711 readings into the database, add hysteresis to the fan and a retry lock to the keypad, replace the DHT11 with a more accurate sensor, move credentials out of the source, register more card types, and use the logged data to forecast temperature rise. A mobile application, several sites on one cloud project and solar power are longer-term options.

VIII. ACKNOWLEDGMENT

The authors thank the Department of Electronics & Communication Engineering, Bapuji Institute of Engineering and Technology, Davangere, for laboratory facilities and guidance.

REFERENCES

[1] K. Mohanraj, S. Vijayalakshmi, N. Balaji, R. Chithrakkannan and R. Karthikeyan, “Smart warehouse monitoring using IoT,” Int. J. Eng. Adv. Technol., vol. 8, no. 6, Aug. 2019.

[2] N. S. Rao, K. H. Sri, B. J. S. Lakshmi, R. R. Sastry and D. Eswar, “The smart warehouse monitoring system using IoT,” vol. 10, no. 2, 2024.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[3] V. S. Kapile, G. S. Shilawat, V. S. Jadhav, S. V. Datir and G. K. Gaikwad, "Smart warehouse using IoT," vol. 4, no. 4, pp. 4005–4009, Apr. 2023.
[4] N. Arasavalli et al., "Development of Internet of Things-based smart warehouse monitoring with efficient cloud architectures," vol. 11, no. 3, Mar. 2023.
[5] J. Borwankar, S. Pandit, V. Patel and J. Nirmal, "IoT based smart warehouse monitoring system," no. 5, May 2019.
[6] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.
[7] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, 2010.
[8] R. Want, "An introduction to RFID technology," IEEE Pervasive Comput., vol. 5, no. 1, pp. 25–33, Jan.–Mar. 2006.
[9] Espressif Systems, "ESP8266EX datasheet," manufacturer documentation.
[10] NXP Semiconductors, "MFRC522 standard 13.56 MHz contactless reader IC," data sheet.
[11] Google, "Firebase Realtime Database documentation," and Telegram, "Bot API documentation," online documentation.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details