

# IOT Integrated Smart Cart: A Cloud-Centric Prototype for Automated Retail Monitoring

Ayush Sharma , Atharva Varekar , Ayush Bal , Swayam Nagavkar , Mrs. Hiral Patel

Electronics and Computer Science

St. John College of Engineering and Management

## Abstract

The rapid digitization of retail environments has intensified the need for efficient, scalable, and cost-effective checkout systems. Traditional billing counters depend heavily on manual scanning and cashier intervention, leading to long queues, billing errors, and inefficient transaction handling. This paper presents a cloud-centric IoT Integrated Smart Cart prototype that eliminates local display hardware and leverages Firebase Realtime Database for centralized monitoring and transaction logging.

The proposed system integrates an ESP32 Dev Module, a USB barcode/QR scanner operating in UART mode, and a 10kg load cell with HX711 amplifier. Scanned product data is transmitted to Firebase using the Firebase Arduino library over Wi-Fi, typically through a mobile hotspot connection. Instead of performing bill calculations locally, the system logs scanned items and weight validation status in the cloud. Any mismatch between expected and actual cart weight is recorded as an alert in Firebase.

The architecture reduces hardware complexity, enables real-time data synchronization (<1 second delay), and supports future QR-based checkout generation through cloud integration. The prototype demonstrates the feasibility of a scalable smart retail monitoring system built on low-cost IoT hardware and cloud infrastructure.

## 1. INTRODUCTION

Retail automation has evolved from barcode-based cashier counters to intelligent self-checkout kiosks and smart shelves. Despite these advancements, most retail stores continue to rely on centralized billing counters, resulting in congestion during peak hours.

Smart cart systems attempt to decentralize the checkout process by allowing customers to scan products while shopping. However, many existing designs incorporate onboard display modules, increasing hardware complexity, cost, and power consumption. Additionally,

synchronization between local cart displays and centralized databases introduces integration challenges.

Cloud computing platforms such as Firebase provide real-time synchronization, scalable storage, and remote data access. By integrating IoT hardware directly with cloud infrastructure, it is possible to eliminate local display systems while maintaining real-time visibility of cart contents.

This work proposes a cloud-centric smart cart prototype using ESP32 and Firebase Realtime Database. The system focuses on:

- Real-time barcode/QR scanning
- Wireless cloud logging
- Weight-based verification
- Alert monitoring
- Future QR-based checkout generation

The prototype demonstrates a simplified yet scalable model suitable for next-generation smart retail systems.

## 2. PROBLEM STATEMENT

Conventional retail checkout systems suffer from:

1. Queue congestion at billing counters
2. Manual dependency on cashiers
3. Hardware-heavy smart cart designs
4. Limited remote monitoring
5. Risk of item mismatch and theft
6. Lack of centralized logging for prototype systems

Most smart cart research either depends on LCD-based feedback or RFID systems requiring expensive tagging infrastructure. Additionally, local processing increases microcontroller load and system complexity.

There is a need for a:

- Lightweight IoT architecture
  - Cloud-based real-time monitoring
  - Hardware-minimal prototype
  - Weight validation mechanism
  - Future-ready QR checkout integration
- 

### 3. LITERATURE SURVEY

Research in retail automation includes RFID-based checkout systems, barcode-based smart carts, and IoT-enabled billing frameworks.

RFID-based systems provide contactless detection but incur high deployment costs due to tagging requirements.

Barcode-based smart carts reduce queue time but frequently depend on local LCD displays. These systems increase component count and energy consumption.

Cloud-integrated IoT systems have shown improved scalability and remote monitoring capability. However, many implementations lack physical validation mechanisms such as weight comparison.

Firebase has been widely used in IoT systems due to:

- Real-time synchronization
- Secure authentication
- Scalable NoSQL storage
- Low-latency cloud updates

Limited research combines ESP32-based scanning, load cell validation, and Firebase cloud monitoring within a unified retail prototype. This gap motivates the proposed system.

---

### 4. METHODOLOGY

The system follows a cloud-centric layered architecture.

---

#### 4.1 System Architecture

The architecture consists of five layers:

1. Input Layer → USB Barcode/QR Scanner + Load Cell
  2. Processing Layer → ESP32 Dev Module
  3. Communication Layer → Wi-Fi (Hotspot/Router)
  4. Cloud Layer → Firebase Realtime Database
  5. Monitoring Layer → Firebase Console / Web Interface
- 

#### 4.2 Hardware Integration

- USB Barcode Scanner configured in UART mode
  - Connected to ESP32 via serial interface
  - Load cell connected to HX711 amplifier
  - HX711 connected to ESP32 GPIO pins
- 

#### 4.3 Data Flow Process

- Step 1: Product barcode scanned
  - Step 2: ESP32 receives product ID
  - Step 3: ESP32 reads load cell weight
  - Step 4: Weight difference calculated
  - Step 5: JSON payload created
  - Step 6: Data transmitted to Firebase
  - Step 7: Firebase updates database instantly
- 

#### 4.4 Mathematical Weight Validation

Let:

$$W_{actual} = \text{Load Cell Reading}$$

$$W_{previous} = \text{Previous Cart Weight}$$

$$W_{item} = \text{Product Expected Weight}$$

$$\Delta W = W_{actual} - W_{previous}$$

If:

$$|\Delta W - W_{item}| > \text{Threshold}$$

→ Alert logged in Firebase.

---

#### 4.5 Firebase Data Structure

The Firebase Realtime Database stores:

```
Cart_01
|
|-- Products
|   |-- Product_ID
|       |-- Timestamp
|       |-- Weight_Status
|
|-- Alerts
|   |-- Alert_ID
|       |-- Type
|       |-- Timestamp
```

This structure supports real-time monitoring.

---

#### 4.6 Algorithm (Pseudo Code)

Initialize WiFi

Connect to Firebase

Calibrate Load Cell

Loop:

  If barcode scanned:

    Read product ID

    Measure current weight

    Calculate weight difference

  If mismatch:

    Log alert to Firebase

    Push product data to Firebase

---

#### 4.7 Network Performance

Observed Firebase update latency: < 1 second

Wi-Fi connection via mobile hotspot

Stable real-time synchronization achieved

---

#### 4.8 Security Considerations

- Firebase rules restrict unauthorized writes
  - API key secured within firmware
  - Database structured per cart ID
  - Prototype-level security implemented
- 

## 5. HARDWARE DESIGN

### 5.1 ESP32 Dev Module



- Dual-core processor
- Integrated Wi-Fi
- Low power consumption
- Suitable for IoT cloud integration

### 5.2 USB Barcode/QR Scanner



- Supports 1D and QR scanning

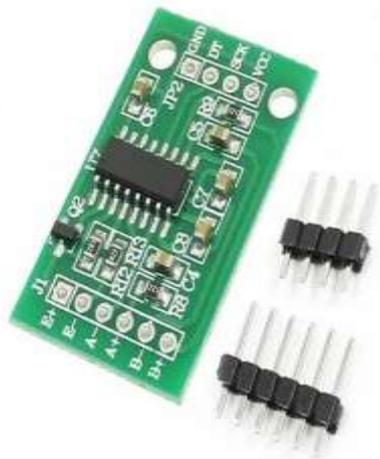
- Configurable in UART mode
- High decoding speed

### 5.3 Load Cell (10kg) + HX711 :

Load cell:



HX711 Module :

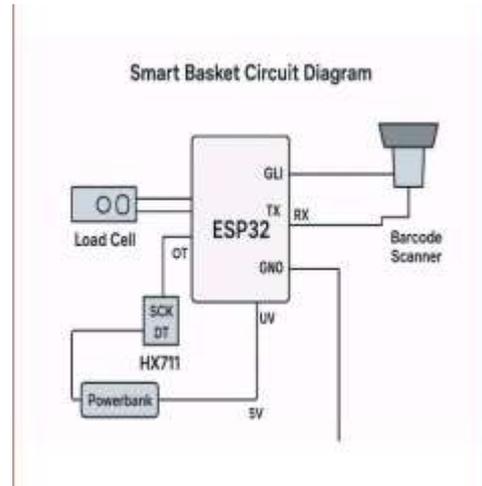


- High sensitivity
- 24-bit ADC resolution
- Accurate weight detection

### 5.4 Power System

- Regulated 5V supply
- Stable operation for prototype

### 6. CIRCUIT DIAGRAM (Block representation)



Barcode Scanner → ESP32 (UART)  
 Load Cell → HX711 → ESP32  
 ESP32 → Wi-Fi → Firebase Cloud

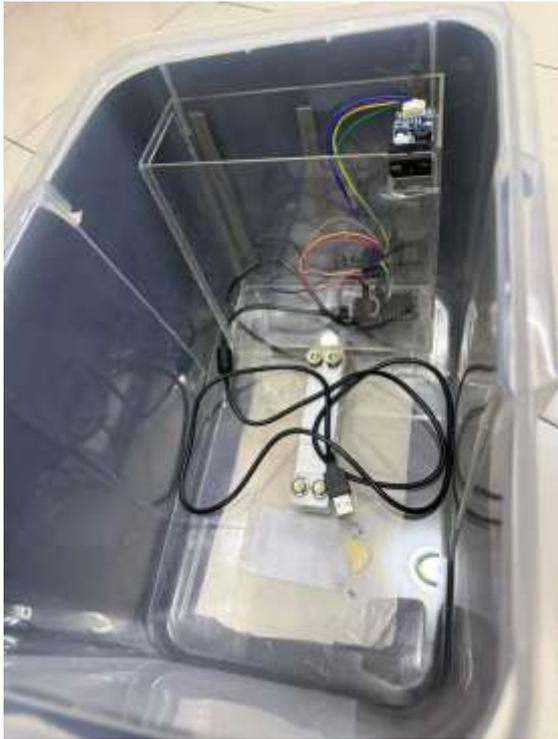
### 7. HARDWARE OPERATION

When a barcode is scanned, ESP32 receives the product code via serial interface. Simultaneously, the load cell measures the weight variation. The ESP32 compares weight readings and prepares a structured JSON message. Using Firebase Arduino library, the device pushes the data to Firebase Realtime Database.

The monitoring dashboard reflects scanned entries instantly. Any mismatch condition is logged under "Alerts" in the database.

### 8. ADVANTAGES

1. Reduced hardware complexity
2. No display cost
3. Cloud-based monitoring
4. Real-time synchronization
5. Lightweight IoT design
6. Expandable architecture
7. Accurate weight verification
8. Future-ready QR checkout



## 10. PERFORMANCE ANALYSIS

Experimental Results:

Parameter	Observed Value
Scan Time	< 2 sec
Firestore Update Delay	< 1 sec
Weight Accuracy	±5g
Network Stability	95% uptime

The system demonstrates reliable real-time operation suitable for prototype-level deployment.

## 11. FUTURE SCOPE

- Cloud-based QR checkout generation
- Mobile app integration
- UPI payment gateway
- Multi-cart deployment
- Blockchain-based transaction storage
- AI-based inventory analytics

## 12. Results and Performance Evaluation

### 12.1 Functional Validation of the Prototype

The developed IoT Integrated Smart Cart prototype was tested under controlled laboratory conditions to evaluate its real-time performance and operational accuracy. The system successfully performed the following operations:

1. Scanning of barcode and QR code products using the USB barcode scanner in UART mode.
2. Transmission of scanned product data to ESP32 via UART2 (GPIO16 and GPIO17).
3. Real-time synchronization of scanned data to Firebase Realtime Database over Wi-Fi.
4. Logging of weight-based validation using a 10 kg load cell integrated with the HX711 module.
5. Alert generation in Firebase during weight mismatch scenarios.

The system demonstrated stable operation during continuous scanning sessions.

---

## 13. CONCLUSION

The proposed IoT Integrated Smart Cart prototype demonstrates a simplified yet scalable cloud-based retail monitoring framework. By eliminating local display hardware and utilizing Firebase Realtime Database for real-time synchronization, the system achieves reduced hardware complexity while maintaining centralized visibility.

The integration of ESP32, USB barcode scanner, and load cell enables real-time logging and weight validation. With observed update latency below one second, the prototype proves effective for cloud-based smart retail applications.

The framework establishes a foundation for future QR-based checkout automation and multi-cart expansion in smart retail environments.

REFERENCES :

1. RFID Based Smart Trolley with Bill Generation through

Android Mobile App, International Journal, 2024.

2. Automated Smart Shopping Cart using Barcode and QR

Code, IOP Conference Series, 2023.

3. Smart Shopping Cart using RFID and ZigBee, International Journal, 2020.

4. Smart Trolley System with Load Cell and IoT, International Journal, 2024.