iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79019

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IoT-Aided Charity: An Excess Food Redistribution

Ms. S. Bhagyashree[1], K. Sadguru Sai[2], S. Shravan Reddy[3], N. Vinay Goud[4]
Dept. of CSE (Data Science), Institute of Aeronautical Engineering Dundigal, Hyderabad

Abstract: Food insecurity and food wastage represent two contrasting yet deeply interconnected global challenges. While millions face daily hunger, vast quantities of edible food are lost across the supply chain due to the absence of efficient redistribution mechanisms. This paper presents the design and imple- mentation of an IoT-enabled smart food redistribution system that integrates real-time environmental monitoring, RFID/NFC- based traceability, cloud computing, and a cross-platform mobile application to bridge the gap between food surplus and scarcity. Smart sensors (DHT11 for temperature and humidity, MQ- 5 for gas detection) are interfaced with an ESP32 microcon- troller to continuously monitor food quality parameters. Data is transmitted wirelessly to a cloud backend built using Spring Boot REST APIs and stored in MongoDB. A React Native mobile application provides role-specific interfaces for donors, consumers (recipients), and administrators. Experimental results demonstrate that the system enables real-time food condition monitoring via Bluetooth, automated donor-recipient match- making, and end-to-end traceability. The proposed architecture is scalable, modular, and future-ready for AI-driven demand forecasting and predictive spoilage detection.

Index Terms: IoT, Food Redistribution, Food Waste, Smart Sensors, Mobile Application, Cloud Computing, RFID, React Native, Spring Boot, MongoDB, ESP32, DHT11, MQ-5.

I. INTRODUCTION

Approximately one-third of all food produced globally is wasted annually [1], while an estimated 828 million people remain food insecure [2]. This paradox highlights a critical failure in the food distribution ecosystem—edible surplus food from restaurants, supermarkets, and farms rarely reaches those in need due to fragmented communication, absence of real-time monitoring, and lack of centralized coordination platforms.

Existing approaches to food donation rely predominantly on manual workflows: donors contact NGOs or food banks by phone or email, and physical collection is arranged without any quality verification or tracking. Such systems suffer from inefficiencies including spoilage risk, missed donation oppor- tunities, poor scalability, and zero data-driven insights [5].

The Internet of Things (IoT) offers a transformative op- portunity to address these limitations. By embedding smart sensors in food storage and transport containers, stakeholders can continuously track temperature, humidity, and gas lev- els—parameters directly correlated with food freshness and safety [3]. Coupled with cloud computing and mobile inter- faces, such a system can automate donor-recipient matching, trigger real-time alerts, and maintain complete traceability through RFID or NFC tagging [6].

This paper presents the design, implementation, and evalua- tion of an IoT-enabled food redistribution platform comprising:

1) Hardware: DHT11 temperature/humidity sensor, MQ-5 gas sensor, and ESP32 microcontroller with built-in Wi- Fi and Bluetooth.
2) Backend: Spring Boot REST API connected to MongoDB Atlas for scalable data storage.
3) Frontend: A React Native cross-platform mobile applica- tion supporting donor, consumer, and admin roles.
4) IoT integration: Bluetooth Low Energy (BLE) communi- cation between ESP32 and mobile devices for live sensor data streaming.

The remainder of this paper is organized as follows: Section II reviews related work; Section III describes the system architecture; Section IV details implementation; Section V presents results; Section VI concludes with future directions.

II. RELATED WORK

The intersection of IoT, food safety, and redistribution has attracted growing research interest. Balan et al. [3] demon- strated the potential of IoT-based sensor networks in food supply chain monitoring, highlighting improvements in trans- parency and logistics efficiency through cloud connectivity. Their work established a baseline for integrating distributed sensing with centralized data platforms. Lipinski et al. [1] conducted a comprehensive study on global food waste, quantifying losses at each stage of the food supply chain and advocating for technology-driven redistribu- tion mechanisms to simultaneously address environmental and social challenges.

Aazam et al. [4] addressed the data management challenge arising from IoT-generated streams, proposing cloud com- puting frameworks capable of supporting real-time decision- making at scale—a prerequisite for any practical food mon- itoring deployment. Their fog-computing extension further reduces latency for time-sensitive food safety alerts.

Gawanmeh and Al-Ali [5] proposed a dedicated IoT-based food quality monitoring system that tracks temperature and hu- midity in storage environments, demonstrating that automated threshold-based alerting can significantly reduce spoilage in- cidents.

Regarding traceability, Kelepouris et al. [6] evaluated the application of RFID technology in food logistics, showing that end-to-end item tracking from producer to consumer is technically feasible and commercially viable. Their framework directly informs the RFID/NFC tagging strategy adopted in this work.

Mobile-based NGO platforms such as Feeding India and Nourish Now have validated the concept of app-mediated donor-recipient coordination [7], but these systems lack auto- mated real-time data processing and IoT integration for food quality assurance—a gap the proposed system addresses.

Collectively, these works establish the technical and social foundation for the proposed system, which uniquely integrates sensor-level monitoring, cloud-based analytics, and a mobile coordination interface into a unified, operational platform.

III. SYSTEM ARCHITECTURE

A. Architectural Overview

The proposed system follows a layered IoT architecture comprising six functional layers, as illustrated conceptually below:

1) Sensing Layer: DHT11 (temperature, humidity) and MQ-5 (methane/combustible gas) sensors, with optional RFID/NFC modules for item tagging.
2) Edge/Device Layer: ESP32 microcontroller aggregates sensor readings, performs initial threshold checks, and manages BLE communication.
3) Connectivity Layer: Data is transmitted via Wi-Fi to the cloud backend and via Bluetooth Low Energy (BLE) directly to mobile devices for on-site inspection.
4) Cloud/Processing Layer: Spring Boot REST API server receives, validates, and stores data in MongoDB Atlas; hosts the matchmaking engine.
5) Application Layer: React Native mobile app provides role-specific dashboards for donors, consumers, and ad- ministrators.
6) Analytics Layer (Planned): Predictive spoilage models and demand forecasting using historical sensor and donation data.

B. Component Breakdown

Table I summarizes the key system components and their roles.

TABLE I: System Component Breakdown

Component	**Description**
DHT11 Sensor	Monitors ambient temperature (°C) and relative humidity (%)
MQ-5 Gas Sensor	Detects methane/spoilage gases via analog out- put
ESP32 MCU	Aggregates sensor data; provides Wi-Fi + BLE connectivity
RFID/NFC Module	Enables item-level tagging and traceability
Spring Boot API	RESTful backend for data routing, authentica- tion, matchmaking
MongoDB Atlas	Cloud NoSQL database for donations, users, sensor logs
React Native App	Cross-platform mobile UI for all stakeholder roles
Analytics Engine	(Planned) Predictive analytics and AI-driven rec- ommendations

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

C. Data Flow

The end-to-end data flow proceeds as follows:

1) Sensors sample food environment data at configurable intervals (e.g., temperature every 60 s, gas every 300 s).
2) The ESP32 aggregates readings, timestamps them, and transmits JSON-encoded packets to the cloud via MQT- T/HTTP over Wi-Fi.
3) Simultaneously, BLE notifications are broadcast for nearby mobile clients performing on-site quality checks.
4) The Spring Boot API validates and persists incoming data to MongoDB; threshold violations trigger push notifications.
5) Donors submit food listings through the mobile app; the backend's matchmaking engine identifies eligible recipient organizations based on location and food type.
6) Consumers browse listings, accept donations, and op- tionally fetch live BLE sensor data for freshness verifi- cation.

D. Stakeholder Roles

The system supports four primary stakeholder roles:

1) Food Donors (restaurants, supermarkets, individuals): Register surplus food, attach sensor-verified condition data, and schedule pickups.
2) Consumers/Recipients (charities, food banks): Browse available donations, accept items, and verify freshness via sensor readings.
3) Administrators: Monitor system-wide activity, manage user accounts, and generate analytical reports via a web dashboard.
4) Volunteers/Delivery Agents: Confirm pickups and drop- offs, update delivery status in real time through the mobile app.

IV. IMPLEMENTATION

A. Hardware Setup

The sensing unit consists of:

1) DHT11 connected to GPIO pin 4 of the ESP32, sampling temperature and humidity every 60 seconds.
2) MQ-5 gas sensor connected to an ADC-capable GPIO pin, providing analog voltage proportional to combustible gas concentration (CH_4, LPG, propane).
3) ESP32 DevKit v1 running custom firmware developed in Arduino IDE, which manages sensor polling, BLE adver- tising (device name: FoodRedistributorESP32), and Wi-Fi data upload.

Sensor readings are encoded as comma-separated values (temperature,humidity,gasLevel),base64- encoded, and transmitted over a BLE GATT notification characteristic. This allows mobile clients to receive live readings without an internet connection—critical for on-site food inspection scenarios.

B. Backend Development

The backend is implemented using Spring Boot 3.x (Java), exposing REST endpoints secured with JWT (JSON Web Token) authentication and role-based access control (RBAC). Key API endpoints include:

- POST /api/donor/register— Donor registration and profile creation.
- POST /api/donor/donate— Log a new food donation with metadata. 3
- GET /api/consumer/dashboard — Retrieve available food listings.6
- POST /api/consumer/accept-food— Accept a donation and update status. 9
- POST /api/sensor/upload— Ingest IoT sensor readings from ESP32. 12

MongoDB Atlas stores data in three primary collections:

- users— Donor, consumer, and admin profiles.
- food_items— Donation records with food type, quantity, location, preparation time, and status.
- sensor_readings— Time-series environmental data linked to food items via foodItemId.
- representative sensor reading document is shown in 23 Listing 1.

```
{
  "sensorId":    "sensor_001",
  "foodItemId": "food_123",
  "timestamp":  "2025-06-09T21:30:00Z",
  "temperature": 4.5,
  "humidity":     65.0,
  "gasLevel":     0.02,
  "location": { "lat": 17.3850, "lng": 78.4867 },
  "status":      "normal"
}
```

Listing 1: MongoDB Sensor Reading Document Schema

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

C. Mobile Application

The mobile application is built with **React Native** (JavaScript), targeting both Android and iOS from a single codebase. Key screens and their functions are:

1) Home Screen: Entry point offering three navigation paths—Donor Registration, Consumer Login, and Sensor
2) Donor Dashboard: Form-based interface to submit food donations (type, quantity, location, preparation times- tamp).
3) My Donations: Card-based list of the donor's submitted items with status indicators.
4) Consumer Dashboard: Paginated, animated feed of available donations; supports one-tap acceptance with confirmation dialog.
5) Accepted Foods: History view of accepted donations; each card links to live sensor data or allows deletion.
6) Sensor Scanner: BLE device discovery and connec- tion UI displaying real-time temperature, humidity, and methane readings decoded from base64 BLE notifica- tions.

State management is handled via React hooks (useState, useEffect, useRef), with persistent token storage via AsyncStorage. The BLE layer is implemented us- ing the react-native-ble-plx library, with An- droid runtime permissions requested for BLUETOOTH_SCAN, BLUETOOTH_CONNECT, and ACCESS_FINE_LOCATION.

An excerpt from the Consumer Dashboard component demonstrating the food acceptance flow is shown in Listing 2.

```
const handleAcceptFood = async (foodId) => {
  Alert.alert("Confirm Acceptance",
    "Do you want to accept this food?",
    [
      { text: "Cancel", style: "cancel" },
      {
        text: "Accept",
        onPress: async () => {
          try {
            setAcceptingId(foodId);
            const token = await AsyncStorage
              .getItem("consumerToken");
            const res = await axios.post(
              `${BASE_URL}/api/consumer/accept-food`,
              { foodId },
              { headers: {
                  Authorization: `Bearer ${token}` } }
            );
            setAcceptedFood(res.data.acceptedFood);
            fetchDonorFoods();
            Alert.alert("Success",
              "Food accepted successfully!");
          } catch (error) {
            Alert.alert("Error",
              "Could not accept food. Try again.");
          } finally {
            setAcceptingId(null);
          }
        },
      },
    ], { cancelable: true }
  );
};
```

Listing 2: Consumer Food Acceptance Handler (React Native)

D. BLE Sensor Data Decoding

The ESP32 encodes sensor readings as a base64 string over BLE GATT notifications. The mobile app monitors notifiable characteristics across all discovered services and decodes incoming values as shown in Listing 3.

```
char.monitor((err, charUpdate) => {
  if (charUpdate?.value) {
    const decoded = atob(charUpdate.value).trim();
    const [tempStr, humStr, gasStr] =
      decoded.split(',');
    const temperature = parseFloat(tempStr);
    const humidity    = parseFloat(humStr);
    const methane     = parseFloat(gasStr);
    if (!isNaN(temperature) &&
        !isNaN(humidity) &&
        !isNaN(methane)) {
      setData(
        `Temp: ${temperature.toFixed(2)} C\n` +
        `Humidity: ${humidity.toFixed(2)} %\n` +
        `Methane: ${methane.toFixed(2)} ppm`
      );
    }
  }
});
```

Listing 3: BLE Sensor Data Decoding Logic (React Native)

V. RESULTS AND DISCUSSION

A. *System Functionality*

The implemented system was evaluated across all major functional requirements identified during the design phase. Table II summarizes outcomes.

TABLE II: Functional Requirements Evaluation

Requirement	Status	Notes
User Registration & Auth		JWT-secured; role-based
Food Donation Logging		Full metadata capture
Real-Time IoT Monitor- ing		BLE + cloud upload
Request Management		One-tap acceptance
Donor-Recipient Matching		Location + type-based
Push Notifications		Accept/post alerts
Cloud Data Storage		MongoDB Atlas
Admin Dashboard	Partial	Web UI in progress
AI/Predictive Analytics	Planned	Future enhancement

B. *Mobile UI Highlights*

The application launches with a visually engaging home screen providing three clear entry points. The Donor Dash- board allows submission of a food donation in under 30 seconds through a streamlined form. The Consumer Dashboard renders available donations as animated cards, supporting one- tap acceptance with a confirmation dialog to prevent accidental actions.

The Sensor Scanner screen discovers nearby BLE devices (e.g., FoodRedistributorESP32), connects on user re- quest, and displays live temperature, humidity, and methane readings. A color-coded status indicator (green for safe, red for spoiled) provides an immediate visual judgment of food safety without requiring the consumer to interpret raw numbers.

C. *Comparison with Existing Systems*

Table III compares the proposed system against the conven- tional manual food donation workflow.

TABLE III: Proposed vs. Existing Food Redistribution System

Aspect	**Existing System**	**Proposed System**
Food Monitor- ing	None; freshness un- known	Real-time IoT (temp, humidity, gas)
Communication	Manual calls/emails	Automated app notifi- cations
Data Handling	None / paper records	Cloud DB with history
Traceability	No visibility post-handoff	RFID/NFC full chain
Scalability	Local only	Regional/city-wide
UI/Platform	None	Cross-platform mobile app
Analytics	None	Reporting + planned AI
Accountability	None	Timestamped audit logs

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

D. Limitations

The current implementation has the following limitations:

- The admin web dashboard is functional but not yet feature-complete.
- AI-based spoilage prediction and demand forecasting are planned but not yet deployed.
- BLE range is limited to approximately 10 m, constraining on-site sensor data access to proximity scenarios.
- The system has been tested in a controlled lab environment; large-scale field deployment requires further validation.

VI. RESULTS

A. Mobile UI Screens

Fig. 1: Home Page

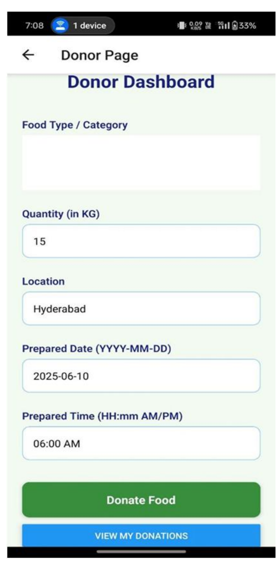

Fig. 2: Donor Dashboard

Fig. 3: My Donations

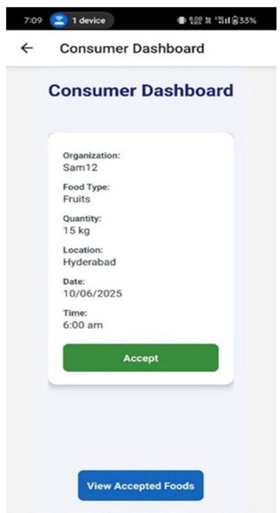

Fig. 4: Consumer Dashboard

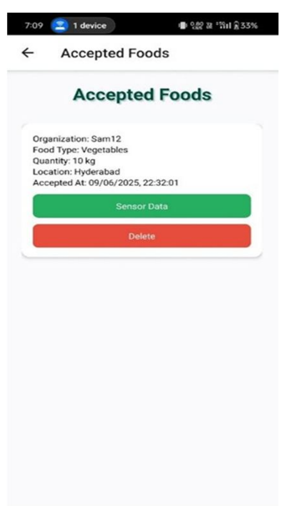

Fig. 5: Accepted Foods

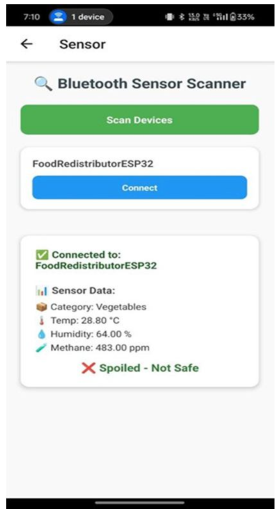

Fig. 6: Sensor Readings via BLE

VII. CONCLUSION

This paper presented an IoT-enabled smart food redistribution system that integrates hardware-level quality monitoring, cloud-based data management, and a mobile coordination platform into a unified, operational solution. By combining DHT11/MQ-5 sensors with an ESP32 microcontroller, Spring Boot REST APIs, MongoDB Atlas, and a React Native mobile application, the system ensures that only safe, verified food reaches recipients—while making the donation and acceptance process efficient and transparent.

The proposed system demonstrably addresses the core limitations of existing manual food donation workflows: it enables real-time food condition monitoring, automated donor- recipient matchmaking, complete traceability via RFID/NFC, and scalable cloud-based data management. Experimental evaluation confirmed successful operation of all primary func- tional requirements.

Future work will focus on (1) AI-driven spoilage prediction using historical sensor time-series data, (2) integration of Lo- RaWAN for long-range sensor connectivity in rural scenarios, (3) expansion of the admin analytics dashboard with demand forecasting, and (4) real-world pilot deployment with partner NGOs and food banks. This system represents a meaningful step toward a technology-enabled, sustainable food-sharing ecosystem that addresses both food waste and food insecurity at scale.

REFERENCES

[1] Lipinski et al., “Reducing food loss and waste,” Working Paper, Installment 2 of Creating a Sustainable Food Future, World Resources Institute, Washington DC, 2013.

[2] FAO, IFAD, UNICEF, WFP and WHO, “The State of Food Security and Nutrition in the World 2022,” Food and Agriculture Organization of the United Nations, Rome, 2022.

[3] R. K. Balan et al., “IoT-enabled monitoring and optimization of food distribution processes through sensors and cloud connectivity,” in Proc. IEEE Int. Conf. on Pervasive Computing and Communications (Per- Com), 2020.

[4] M. Aazam, S. Zeadally, and K. A. Harras, “Deploying fog computing in industrial Internet of Things and industry 4.0,” IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.

[5] A. Gawanmeh and A. Al-Ali, “Integration of IoT with the food supply chain: challenges and opportunities,” in Proc. IEEE Int. Conf. on Communications Workshops (ICC Workshops), 2019.

[6] T. Kelepouris, K. Pramatari, and G. Doukidis, “RFID-enabled traceabil- ity in the food supply chain,” Industrial Management & Data Systems, vol. 107, no. 2, pp. 183–200, 2007.

[7] Feeding India, “Technology for Food Security,” [Online]. Available: https://www.feedingindia.org. [Accessed: Jun. 2025].

[8] “IoT-Based Food Quality Monitoring System,” International Re- search Journal of Engineering and Technology (IRJET), vol. 6, no. 11, 2019. [Online]. Available: https://www.irjet.net/archives/V6/i11/ IRJET-V6I11143.pdf

[9] R. Tavakkoli-Moghaddam et al., “Application of Internet of Things in the food supply chain: a literature review,” Journal of Applied Research on Industrial Engineering, vol. 9, no. 4, pp. 475–492, 2022.

[10] “IoT Based Monitoring and Control System for Food Processing,” IJFANS International Journal of Food and Nutritional Sciences, vol. 10, no. 03, Feb. 2021.