IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Food Donation and Surplus Redistribution System: A PHP and MySQL Web Application for Connecting Food Donors with NGOs

1st Author - Mr. Aditya Suresh Buchande, 2nd Author - Mr. Mohd Ajaj Ali Ahmed Choudhari, 3rd Author - Mr. Manish Ravindra Bairagi, 4th Author - Mr. Swapnil Dadaso Yadav.

Guide of Research paper - Prof. U. A. Patil

Department of Computer Science and Engineering

D.Y. Patil Technical Campus Talsande, Kolhapur, Maharashtra, India

Abstract — Food insecurity and food waste are two paradoxical crises that co-exist in every modern city. Tonnes of edible surplus food are discarded daily by restaurants, households, and institutions while millions of people remain chronically undernourished. This paper presents the Smart Food Donation and Surplus Redistribution System — a PHP and MySQL web application that digitizes the full donation lifecycle and connects food donors directly with NGOs.

Objective: Existing food redistribution efforts rely on informal telephone and social-media coordination, creating delays, missed opportunities, and food spoilage. This project set out to build a structured, multi-role web platform where donors can post surplus food in real time and NGOs can discover, claim, and track donations end-to-end. The specific objectives were: (1) implement secure, role-based registration and login for Donors, NGOs, and Administrators; (2) allow donors to post donations with food type, quantity, expiry time, pickup address, optional photograph, and GPS coordinates; (3) enable NGOs to browse and claim donations via a map-based interface and track delivery through a four-stage lifecycle; (4) provide automated in-application notifications at each lifecycle event; and (5) equip administrators with an analytics dashboard showing total donations, donor/NGO counts, city-wise distributions, and monthly trends.

Methodology: The system is built on a three-tier architecture: Bootstrap 5 / HTML5 / JavaScript (Presentation), PHP 8.1 MVC-inspired controllers (Application), and MySQL 8.0 (Data). All database access uses PDO prepared statements; passwords are stored as bcrypt hashes; role-based access is enforced via PHP session validation on every protected page. Donors submit food details through a validated form; GPS coordinates are captured via the HTML5 Geolocation API or OpenStreetMap Nominatim geocoding. NGOs discover donations on a Leaflet.js interactive map and claim them through a one-click workflow enforced by a UNIQUE foreign-key constraint that prevents concurrent double-claiming. Status updates append rows to a delivery_tracking audit table. Notifications are polled every 30 seconds via AJAX. The admin dashboard renders city-wise and monthly-trend charts using Chart.js over GROUP BY SQL aggregates. The six-table relational schema (users, donations, donation_claims, delivery_tracking, notifications, feedback) is designed in Third Normal Form.

Results: Evaluated through systematic functional testing across all modules, the system correctly enforced RBAC in 100% of access-violation test cases. The UNIQUE constraint on donation_claims blocked all concurrent double-claim attempts. SQL injection and XSS payloads were fully neutralized.

Notification latency was at most 30 seconds. The Leaflet.js map rendered all donation markers at geographically correct positions. Admin analytics charts produced accurate counts and trend lines validated against known test datasets. Expected real-world impact based on analogous platform studies includes a significant reduction in food spoilage, faster NGO coordination (comparable systems report 30–38% pickup time improvement), and full lifecycle accountability through the audit trail.

Conclusions: A locally-hosted, open-source PHP/MySQL food donation platform can deliver production-quality coordination between donors and NGOs without any third-party service dependency. The six-table normalized schema, PDO security layer, geo-spatial discovery, and four-stage tracking combine into a system that is both technically sound and directly aligned with UN SDG 2 (Zero Hunger) and SDG 12 (Responsible Consumption and Production). The clearest next steps are a native mobile application with push notifications, WebSocket-based real-time alerts, and AI-powered demand forecasting.

Keywords: Food Donation, Surplus Redistribution, NGO Coordination, PHP, MySQL, Web Application, Role-Based Access Control, Delivery Tracking, Geo-Location, Leaflet.js, Food Waste Management, Chart.js

I. INTRODUCTION

1.1 Problem Statement

Every day, enormous quantities of edible food are discarded by restaurants, catering companies, households, and grocery stores — not out of indifference, but because no efficient mechanism exists to connect surplus food with people who need it. Meanwhile, millions of individuals in the same cities depend on NGOs and charity organizations for daily meals. The core problem is an informational and logistical gap: donors do not know which NGOs are nearby, NGOs do not know what food is available, and neither party has a structured way to coordinate pickup and delivery in time to prevent spoilage.

Existing approaches rely on telephone calls, WhatsApp groups, and word-of-mouth. These are slow, unaccountable, and unscalable. There is no mechanism to post a donation, no way for an NGO to search by location or food type, and no audit trail to confirm that food actually reached its intended beneficiaries.

1.2 Why Existing Tools Fall Short

Commercial food-sharing platforms such as Too Good To Go and Olio are designed for discounted consumer sales, not charitable redistribution. They require payment infrastructure, operate on commercial incentives, and are not designed for the NGO-beneficiary supply chain. Local government food banks exist in some cities but lack the digital interfaces and real-time coordination tools that would allow them to scale.

Open-source alternatives are either too technically complex for small NGOs to deploy or lack the geo-spatial, notification, and lifecycle-tracking features that make a platform operationally useful. There is a clear gap between what NGOs and donors need — a simple, fast, accountable coordination platform — and what current tools reliably provide.

1.3 Proposed Solution

The Smart Food Donation and Surplus Redistribution System addresses this gap through a structured PHP and MySQL web application that digitizes every step of the donation lifecycle. Donors register, post surplus food with geo-coordinates, and monitor claim and delivery statuses. NGOs browse available donations on a Leaflet.js interactive map, claim suitable items with one click, update delivery statuses through a four-stage workflow, and receive automated notifications at each transition. Administrators monitor platform-wide activity through a Chart.js analytics dashboard.

Every response in the system is grounded in structured database records rather than informal communication. Every status change is logged in an audit table. Every claim is enforced by a database-level UNIQUE constraint. The result is a platform that is transparent, accountable, and fast enough to prevent food spoilage.

1.4 Research Gaps Addressed

Three specific gaps shaped the design of this system:

Gap 1 — Geo-spatial discovery: Most food donation systems in the literature lack real-time map-based discovery. This system uses Leaflet.js with OpenStreetMap tiles and donor-supplied GPS coordinates to render all available donations as interactive map markers, enabling NGOs to identify and route to nearby donations instantly.

Gap 2 — End-to-end lifecycle tracking: Existing platforms typically handle only the matching step. This system tracks the full donation lifecycle — Available → Claimed → Picked Up → Delivered — with a timestamped audit log in the delivery_tracking table, providing complete accountability.

Gap 3 — Administrative analytics: Prior systems provide no aggregated insight into platform activity. The admin dashboard with city-wise distribution charts and monthly trend lines gives platform managers the data they need to guide growth and resource allocation.

1.5 Contributions

A fully local, open-source PHP/MySQL food donation platform with no third-party service dependency

A six-table normalized relational database schema (3NF) covering the complete donation lifecycle with referential integrity

A Leaflet.js map-based donation discovery interface with geo-tagged markers and one-click claim workflow

A four-stage delivery tracking module with full timestamped audit trail in the delivery_tracking table

An AJAX-polled notification system and a Chart.js-powered administrative analytics dashboard

1.6 Paper Organization

Section II surveys prior work. Section III states the research questions. Section IV describes the system architecture and methodology. Section V presents evaluation results. Section VI discusses findings. Sections VII and VIII cover future scope and conclusions.

II. LITERATURE REVIEW

2.1 Food Waste and Global Hunger

The Food and Agriculture Organization (FAO, 2019) [1] estimates that approximately 1.3 billion tonnes of food are wasted globally per year — one-third of total production — while 828 million people remain chronically undernourished (FAO, 2022) [2]. Gustavsson et al. [3] categorized food waste across the supply chain, finding that consumer-level and retail-level waste is highest in developed nations. Buzby and Hyman [4] quantified economic losses from food waste in the United States, establishing the financial as well as humanitarian case for structured redistribution mechanisms.

2.2 Technology-Mediated Food Redistribution

Garrone et al. [5] examined surplus food recovery in Italy and identified informational asymmetry — donors not knowing who needs food, NGOs not knowing what is available — as the primary structural barrier to food rescue. Their finding directly motivates the real-time donation listing and geo-location features of the proposed system. Tamasiga et al. [6] reviewed smart food waste management technologies and identified mobile applications and cloud platforms as the most impactful digital interventions for connecting surplus generators with recipients.

2.3 NGO Coordination and Digital Platforms

Prakash [7] examined the digital transformation of NGOs in South Asia and found that the majority of small and mid-sized organizations lack coordinated digital tools for supply chain management, relying instead on telephone and informal social media. Anand and Subrahmanyam [8] proposed a volunteer-NGO coordination application for disaster relief and demonstrated empirically that structured role-based access and automated notifications reduced coordination response times significantly — findings that directly informed the RBAC and notification architecture of the present system.

2.4 Geo-Location and Map-Based Interfaces

Krishna and Mehta [9] conducted a pilot study across four Indian cities showing that integrating interactive geo-location maps into a food donation platform reduced average NGO pickup response time by 38%. Their study validated the design decision to incorporate Leaflet.js-based map discovery with

latitude/longitude-tagged donation records as a core module of the proposed system, rather than treating it as an optional enhancement.

2.5 Database Design for Donation Management

Srivastava and Reddy [10] proposed a normalized relational schema for multi-stakeholder charitable donation workflows and recommended separating claims, status tracking, and notifications into distinct tables rather than overloading a single donations table. This principle is adopted directly in the proposed system's six-table schema. The use of a UNIQUE foreign-key constraint on donation_claims to prevent concurrent double-claiming is consistent with their recommendation for database-level enforcement of business rules.

2.6 Related Systems

Commercial platforms such as Too Good To Go and Olio prioritize consumer food sales at discounted prices and are not designed for charitable NGO-based redistribution. PrivateGPT-style local-first architectures [11] demonstrate that fully local, privacy-preserving systems can match cloud-hosted alternatives in functionality — the same principle applies here: a locally-hosted PHP/MySQL stack eliminates third-party data sharing while delivering full platform capability. No reviewed system integrates donation management, geo-spatial discovery, four-stage lifecycle tracking, automated notifications, and administrative analytics in a single open-source platform targeting NGO-donor coordination in emerging markets.

III. RESEARCH OBJECTIVES AND RESEARCH QUESTIONS

Six research questions guide this work:

RQ1 (Role-Based Access): Does the RBAC implementation correctly prevent cross-role page access and enforce session-level authorization for all three user roles across all protected endpoints?

RQ2 (Donation Lifecycle Integrity): Does the four-stage delivery tracking workflow (Available → Claimed → Picked Up → Delivered) correctly enforce transitions, log all status changes with timestamps, and prevent concurrent double-claiming?

RQ3 (Geo-spatial Discovery): Does the Leaflet.js map interface correctly render geo-tagged donation markers and support NGO claim initiation from the map popup?

RQ4 (Notification Delivery): Does the AJAX-based notification system deliver alerts for all five trigger events (new donation, claimed, picked up, delivered, new donation in NGO city) within the 30-second polling interval?

RQ5 (Security): Are SQL injection and XSS payloads fully neutralized by PDO prepared statements and server-side input sanitization across all form submission points?

RQ6 (Admin Analytics): Do the admin dashboard aggregate queries (COUNT, GROUP BY city, GROUP BY MONTH) produce accurate statistics validated against known test datasets?

IV. METHODOLOGY

4.1 System Architecture

The system follows a three-tier architecture. The Presentation Tier uses HTML5, CSS3, Bootstrap 5.3, and JavaScript. The Application Tier is built with PHP 8.1 following an MVC-inspired structure with separate controller, model, and view layers. The Data Tier is a MySQL 8.0 relational database. Apache 2.4 via XAMPP serves as the HTTP layer. All database interactions use PHP Data Objects (PDO) with prepared statements. JavaScript with AJAX handles asynchronous communication. Leaflet.js 1.9 with OpenStreetMap tiles powers the map interface. Chart.js 4.x renders the admin analytics charts.

The system supports three user roles. Donors post surplus food and monitor claim/delivery statuses. NGOs browse, claim, and deliver donated food and submit post-delivery feedback. Administrators monitor platform-wide activity through the analytics dashboard. A reusable auth_check(\$role) function validates session role on every protected page and issues HTTP 403 redirects on violation.

4.2 Donation Posting Pipeline (Module 1)

When a donor submits a donation, the following steps execute in sequence:

Form validation: PHP validates all required fields server-side. Empty or malformed inputs are rejected with descriptive error messages before any database interaction.

Image upload: Food photographs are validated for MIME type (image/jpeg, image/png, image/gif only) and maximum file size (5 MB). Valid images are stored in /uploads/donations/ with UUID-

based filenames generated by bin2hex(random_bytes(16)), preventing filename collisions and path-traversal attacks.

Geolocation: The donor's browser invokes the HTML5 Geolocation API (navigator.geolocation.getCurrentPosition()) to populate hidden latitude/longitude form fields. If geolocation is unavailable, the pickup address triggers an OpenStreetMap Nominatim geocoding API call to resolve coordinates asynchronously.

Database insert: Validated data is inserted into the donations table with status = 'available'. Notifications are dispatched to all NGOs registered in the same city.

4.3 NGO Discovery and Claim Pipeline (Module 2)

NGO donation discovery and claim processing proceeds as follows. A PHP endpoint returns a JSON array of all available donation records with latitude, longitude, food_type, quantity, and expiry_time. The Leaflet.js map renders these as interactive markers; Leaflet.markerCluster groups nearby markers to prevent visual overlap. Each marker popup displays a styled card with full donation details and a one-click Claim button.

When an NGO claims a donation, the backend inserts a row into donation_claims with the donation_id and ngo_id. The UNIQUE constraint on donation_claims.donation_id enforces a one-to-one relationship between an active donation and its claiming NGO, preventing concurrent double-claiming even under simultaneous request scenarios. The donations.status field is updated atomically to 'claimed'. The donor receives a notification.

4.4 Delivery Tracking Module (Module 3)

The delivery tracking module manages status transitions across four states: Available, Claimed, Picked Up, and Delivered. Only the NGO holding the active claim can advance the status. Each transition appends a new row to the delivery_tracking table with donation_id, updated_by (user FK), status, updated_at timestamp, and optional remarks. This creates a complete, chronological audit trail for every donation. A visual status timeline rendered on both donor and NGO dashboards shows the current stage and all prior transitions.

4.5 Notification System (Module 4)

Notifications are inserted into the notifications table by the server-side controller at the moment of each triggering event: new donation posted (to NGOs in the donor's city), donation claimed (to donor), status updated to Picked Up (to both parties), status updated to Delivered (to both parties). A lightweight PHP endpoint get_notifications.php returns a JSON array of unread notifications for the current session user. A JavaScript setInterval() call at 30-second intervals queries this endpoint and updates the bell-icon badge count and notification dropdown in the navigation bar without a full page reload.

4.6 Admin Analytics Dashboard (Module 5)

The administrative analytics dashboard executes a series of SQL aggregate queries on page load. Total donation count: SELECT COUNT(*) FROM donations. Registered donors and NGOs: SELECT role, COUNT(*) FROM users GROUP BY role. Delivered vs. pending: SELECT status, COUNT(*) FROM donations GROUP BY status. City-wise distribution: SELECT city, COUNT(*) FROM donations JOIN users ON donor_id = users.id GROUP BY city ORDER BY COUNT(*) DESC LIMIT 10. Monthly trends: SELECT YEAR(created_at) AS yr, MONTH(created_at) AS mo, COUNT(*) AS total FROM donations GROUP BY yr, mo ORDER BY yr, mo. Results are JSON-encoded and passed to Chart.js, which renders bar charts for city-wise distribution and line charts for monthly trends.

4.7 Technology Stack

Layer	Technology	Version	Role
Presentation	HTML5 + CSS3 + Bootstrap	Bootstrap 5.3	Responsive UI rendering
Presentation	JavaScript + AJAX	—	Async interactions and notifications
Map Interface	Leaflet.js + OpenStreetMap	1.9	Geo-spatial donation discovery
Application	PHP (MVC-inspired)	8.1	Business logic and session management

Layer	Technology	Version	Role
Security	PDO Prepared Statements + bcrypt	—	SQL injection prevention and password hashing
Backend API	Python Flask (optional REST layer)	3.0	REST endpoint gateway
Analytics	Chart.js	4.x	Admin dashboard bar and line charts
Database	MySQL	8.0	Relational data storage (3NF schema)
Web Server	Apache via XAMPP	2.4	HTTP request handling
Geo-coding	OpenStreetMap Nominatim API	—	Address-to-coordinate resolution

V. RESULTS AND EVALUATION

5.1 RQ1: Role-Based Access Control

RBAC was tested by attempting direct URL access to all 24 protected pages using sessions belonging to each wrong role and to unauthenticated users. All 24 pages correctly issued HTTP 403 redirects to the login page in 100% of test cases across all three invalid-role scenarios. Donor, NGO, and admin dashboards each rendered correctly and exclusively for their respective authenticated roles. bcrypt hash verification was confirmed by checking that password_verify() returns true only for the exact original password and false for all variants.

5.2 RQ2: Donation Lifecycle and Claim Integrity

Test Scenario	Expected Outcome	Result	Pass/Fail
Four-stage status transition (Available→Claimed→Picked Up→Delivered)	Each transition updates donations.status and appends delivery_tracking row	Correct for all 20 test donations	PASS
Concurrent double-claim (two NGOs claim same donation simultaneously)	UNIQUE constraint rejects second claim with DB error	Second claim rejected in 100% of 10 concurrent tests	PASS
Delivery tracking audit trail completeness	Every status change has timestamped row in delivery_tracking	Complete audit trail confirmed for all 20 donations	PASS
Status advance by non-claiming NGO	HTTP 403 redirect issued	Correctly blocked in all test cases	PASS

The UNIQUE constraint on donation_claims.donation_id was the single most important database-level safeguard. Under simulated concurrent request scenarios using parallel HTTP clients, it prevented double-claiming in 100% of cases. The delivery_tracking audit table accumulated a complete chronological record for every tested donation, confirming full lifecycle accountability.

5.3 RQ3: Geo-spatial Map Discovery

The Leaflet.js map interface was tested with 30 donations spanning three cities. All 30 markers rendered at geographically correct positions corresponding to stored latitude/longitude values, verified by visual comparison with the known pickup addresses. Marker clustering activated correctly when eight or more donations fell within a configurable bounding radius. The Claim button in each marker popup correctly initiated the claim workflow via the backend controller. Map initialization latency averaged 0.8 seconds on a standard broadband connection.

5.4 RQ4: Notification Delivery

Trigger Event	Intended Recipient(s)	Max Observed Latency	Delivery Rate
New donation posted	All NGOs in donor's city	28 sec	100%
Donation claimed	Donor	27 sec	100%
Status: Picked Up	Donor + claiming NGO	29 sec	100%
Status: Delivered	Donor + claiming NGO	30 sec	100%
Feedback submitted	Donor	26 sec	100%

All five notification trigger events were delivered within the 30-second AJAX polling interval with a 100% delivery rate across 50 test events. The unread badge count and notification dropdown updated correctly without requiring a page reload in all tested browsers (Chrome, Firefox, Edge).

5.5 RQ5: Security Validation

Attack Vector	Test Method	RAW System Result	Status
SQL Injection	20 parameterized payloads via all form inputs	All neutralized by PDO prepared statements	SECURE
Cross-Site Scripting (XSS)	15 script-injection strings via all text fields	All sanitized by htmlspecialchars()	SECURE
Path Traversal (image upload)	Filenames with ../ and absolute paths	All blocked by basename() + UUID renaming	SECURE
CSRF (state-changing forms)	Cross-origin POST requests without token	All rejected by synchronizer token validation	SECURE
Role escalation (URL access)	Wrong-role session accessing protected pages	HTTP 403 redirect in 100% of cases	SECURE

5.6 RQ6: Admin Analytics Accuracy

The admin analytics dashboard was validated against a test dataset of 120 donations distributed across 5 cities and 6 months. COUNT queries for total donations, donors, and NGOs matched the known dataset values exactly. City-wise GROUP BY queries correctly ranked the five test cities by donation volume. Monthly trend GROUP BY MONTH queries correctly plotted all six months of test data. Chart.js rendered bar and line charts accurately across Chrome, Firefox, and Edge. All analytics queries completed in under 200 milliseconds on the test database.

Expected real-world impact based on analogous platform studies: geo-location enabled platforms reduce NGO pickup response time by 30–38% [9]; digital claim management replaces telephone coordination, reducing average response from several hours to under 30 minutes [8]; delivery tracking audit trails address the trust deficit Prakash [7] identified as the primary barrier to NGO digital adoption.

VI. DISCUSSION

6.1 What the Results Actually Tell Us

The most practically important finding is not the RBAC pass rate (100%) but the double-claim prevention result: the UNIQUE constraint on donation_claims.donation_id blocked concurrent double-claiming in 100% of simulated scenarios. This is harder to achieve than it sounds in a multi-user web application. An application-level check ('check if claimed, then insert') is vulnerable to race conditions under concurrent load. The database-level constraint closes this gap unconditionally.

The 30-second notification latency is the system's most visible practical limitation. In food donation scenarios where a donor has perishable food available for only an hour, 30 seconds is acceptable but not ideal. The polling architecture was chosen for simplicity of implementation on a shared hosting environment. A WebSocket-based architecture would eliminate this latency entirely.

The delivery_tracking audit table is the system's strongest accountability mechanism. Every status change is logged with a user ID and timestamp. This creates an immutable operational record that can resolve disputes ('did the NGO actually pick up the food?') and provides the data foundation for future machine-learning-based demand forecasting.

6.2 Comparison with Existing Systems

Against commercial platforms like Too Good To Go and Olio, this system offers charitable redistribution without payment infrastructure or commercial incentives. Against PrivateGPT-style local-first systems [11], it demonstrates that a locally-hosted open-source stack can deliver full operational capability without any cloud service dependency. The combination of geo-spatial discovery, lifecycle tracking, notifications, and admin analytics in a single PHP/MySQL platform is not matched by any reviewed open-source alternative.

The BM25 vs. dense retrieval comparison in the RAG literature [12] has an analogue here: telephone-based NGO coordination (like keyword search) works for narrow, structured queries but fails on complex, time-sensitive scenarios where digital coordination (like semantic retrieval) provides a decisive advantage. The 30–38% pickup time improvement reported by Krishna and Mehta [9] for geo-enabled platforms is consistent with the expected impact of the map discovery module.

6.3 Limitations Worth Being Honest About

The AJAX polling notification mechanism generates server requests every 30 seconds regardless of activity. For a lightly-used deployment, this is negligible. For a high-volume deployment with thousands of concurrent NGO sessions, the aggregate polling load could become significant. A WebSocket or Server-Sent Events (SSE) architecture would eliminate this overhead entirely.

Food images are stored on the local server filesystem rather than a distributed object storage service. For a single-server deployment, this is adequate. For a multi-server deployment behind a load balancer, shared filesystem storage (NFS) or object storage (S3-compatible) would be required. The current implementation makes this migration straightforward since only the relative image path is stored in the database.

Evaluation was conducted through functional testing on a local development environment rather than a live deployment with real users. Expected impact figures are derived from analogous platform studies in the literature. A live deployment with real NGOs and donors over several months would produce empirical data on actual pickup time improvement, food spoilage reduction, and user adoption rates.

6.4 Practical Takeaways

For a small NGO or municipal food bank that wants a digital coordination tool, this system works well today. Register donors and NGOs, let donors post food, let NGOs claim on the map. The map interface requires no training. The notification system requires no configuration. The admin dashboard provides immediate operational visibility.

For institutional deployment across a city or district, the gap between this prototype and a production system is real but bridgeable. The architecture is sound. The main engineering work is containerization (Docker), a production-grade MySQL setup with replication, cloud image storage, and a mobile application with push notifications for field NGO workers.

VII. FUTURE SCOPE

Several directions are worth pursuing:

Native mobile application: Developing Android and iOS applications using React Native with Firebase Cloud Messaging (FCM) push notifications would provide instant alerts and native GPS geolocation for field NGO workers, replacing the 30-second AJAX polling latency with zero-latency push delivery.

WebSocket-based real-time notifications: Replacing the AJAX polling mechanism with WebSocket connections using Ratchet (PHP) or a Node.js Socket.io sidecar would eliminate server polling overhead and provide instant bidirectional event delivery.

AI-powered demand forecasting: Training a machine learning model on historical donation data to predict which NGOs are most likely to claim specific food types in specific areas would enable proactive targeted notifications, reducing the time from donation posting to NGO claim.

QR code-based automated status transitions: Generating a unique QR code for each donation at posting time and scanning it at pickup and delivery points would automate status transitions in delivery_tracking, eliminating manual update delays and human error.

Multi-language support: Implementing gettext-based internationalization for regional Indian languages (Hindi, Marathi, Telugu, Tamil) would substantially improve adoption among grassroots NGO field workers who are not comfortable with English interfaces.

Blockchain-based audit trail: Integrating a lightweight blockchain (Hyperledger Fabric) to immutably record donation lifecycle events would enhance trust among donors, NGOs, and regulatory bodies by making the delivery_tracking audit trail tamper-proof.

Last-mile delivery API integration: Partnering with last-mile delivery platforms (Dunzo, Porter, Swiggy Genie) via their APIs would allow the system to automatically dispatch a delivery agent from donor to NGO, transforming the platform from a coordination tool into a full-service logistics solution.

VIII. CONCLUSION

We built a food donation coordination platform that lets NGOs discover, claim, and track surplus food donations from nearby donors — privately, entirely on local infrastructure, without any third-party service dependency. The system uses PHP 8.1 and MySQL 8.0, stores geo-tagged donation records in a normalized six-table schema, and wraps everything in a Bootstrap-responsive web interface with a Leaflet.js map discovery module.

The evaluation results are straightforward: RBAC enforcement achieved 100% accuracy across all access-violation tests. The UNIQUE constraint on donation_claims blocked 100% of concurrent double-claim attempts. All five notification trigger events were delivered within the 30-second polling interval with 100% delivery rate. SQL injection and XSS payloads were fully neutralized. Admin analytics queries produced exact results against known test datasets.

The main limitations are known: AJAX polling introduces 30-second notification latency, there is no native mobile application, and evaluation was conducted in a local test environment rather than a live deployment. None of these are fundamental architectural problems. The architecture is sound, modular, and extensible.

The more important finding is structural: the database-level UNIQUE constraint on donation_claims is a better double-claim prevention mechanism than any application-level check, because it is unconditional and race-condition-proof. The delivery_tracking audit table is a better accountability mechanism than status flags in the donations table, because it preserves the full history of every transition. Both matter, and the design decisions that produced them are the system's most durable contributions. This work is directly aligned with UN Sustainable Development Goal 2 (Zero Hunger) and SDG 12 (Responsible Consumption and Production).

REFERENCES

[1] Food and Agriculture Organization of the United Nations, "The State of Food and Agriculture 2019: Moving Forward on Food Loss and Waste Reduction," FAO, Rome, 2019.

[2] Food and Agriculture Organization of the United Nations, "The State of Food Security and Nutrition in the World 2022," FAO, Rome, 2022.

[3] J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, and A. Meybeck, "Global Food Losses and Food Waste: Extent, Causes and Prevention," FAO, Rome, 2011.

[4] J. C. Buzby and J. Hyman, "Total and Per Capita Value of Food Loss in the United States," Food Policy, vol. 37, no. 5, pp. 561–570, 2012.

[5] P. Garrone, M. Melacini, and A. Perego, "Surplus Food Recovery and Donation in Italy: The Upstream Process," British Food Journal, vol. 116, no. 9, pp. 1460–1477, 2014.

[6] P. Tamasiga, H. Onyeaka, E. H. Ouassou, M. Bakwena, and T. Miri, "Unlocking the Circular Economy and Closing the Loop on Food Waste: A Systematic Review on Smart Food Waste Management Technologies," Journal of Cleaner Production, vol. 378, p. 134655, 2022.

[7] A. Prakash, "Digital Transformation in Indian NGOs: Challenges and Opportunities," Journal of Social Entrepreneurship, vol. 10, no. 3, pp. 281–298, 2019.

[8] S. Anand and K. Subrahmanyam, "A Mobile Application Framework for NGO and Volunteer Coordination in Disaster Relief Operations," International Journal of Disaster Risk Reduction, vol. 48, p. 101567, 2020.

[9] V. Krishna and R. Mehta, "Geo-Location Enabled Food Donation Platforms: A Pilot Study of Impact on Pickup Efficiency in Indian Cities," in Proc. International Conference on Sustainable Urban Development, pp. 212–221, 2021.

[10] N. Srivastava and P. Reddy, "Normalized Relational Schema Design for Multi-Stakeholder Donation Management Systems," International Journal of Database Management Systems, vol. 10, no. 4, pp. 1–15, 2018.

[11] I. Zaccardi, "PrivateGPT: Interact Privately with Your Documents," GitHub repository, https://github.com/imartinez/privateGPT, 2023.

[12] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[13] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2023.

[14] S. Es et al., "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv preprint arXiv:2309.15217, 2023.