Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

An AI-Powered Food Donation and Analytics Platform

[1]Deepanjali Singh, [2]Mr. Pawan Kumar Jaiswal
[1]B.Tech Computer Science & Engineering (6th Semester), [2]*Assistant Professor*
[1,2]Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur, India

Abstract

The "Smart Food Bridge" is a high-tech solution to the paradox of colossal food waste in many parts of the world occurring simultaneously with chronic hunger elsewhere. The platform employs a Python Flask backend and an MVC architectural framework to form an integrated digital ecosystem that connects the donor to NGOs in real-time. This is where the magic happens: Machine Learning integration, namely with a classification system (Logistic Regression classifier here) that guarantees donations are categorised to an exceptionally high degree of accuracy; feedback is processed by a Natural Language Processing engine to extract insights, and service/donor satisfaction remains high. Focusing on accessibility, the lightweight SQLite3 database and simple Jinja2 for interaction allow users to take their first test in under three minutes. At the end of the day, this portal allows you to access an affordable, open-source protocol that enables humanitarian organizations to seamlessly transition from high manual processes toward a data-driven process– one that brings humanity back into surplus food redistribution by catering specifically to 828 million people globally who face food insecurity today.

Keywords: Food Donation, Machine Learning, Logistic Regression, NLP Sentiment Analysis, Flask, MVC, SQLite, Food Waste, Humanitarian Technology.

INTRODUCTION

1.1 Background and Motivation

The most shocking part is that more food is produced globally than we can ever consume and yet millions lay in hunger; it is indeed the global food crisis. Sharply more than one-third of the world generation i.e 1.3 billion tonnes of foods lost or wasted every year–according to UNA Food and Agriculture Organisation (FAO). It's not just a humanitarian disaster, but an ecological one as well. Food loss and waste, if it were a country would be the third largest emitters of greenhouse gases in the world after the U.S. and China due to methane gas released when food rotten in landfills.

This paradox is especially acute in India. The country is an agricultural superpower in the world, yet 40% of its food is identified lost on the supply chain. We can stop a lot of that waste at the "last mile" - large social occasions for celebrations such as weddings, festivals and corporate galas where huge quantities of perfectly good cooked food is sent straight to the dumpster because there is no way to redirect it quickly. This motivation emerged from a basic

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

but profound insight: food waste is not a problem of lack of resources; it is one of coordination. Today this "bridge" between having so much and having too little has three key obstacles that are causing it to become severed,

- **Information Asymmetry:** Prospective donors want to contribute in most cases but don't have the knowledge about what NGOs are around them that are active at any given time.
- **Perishability:** Cooked food lasts very little, and it needs a real-time response which direct manual methods cannot provide.
- **Lack of Visibility:** NGOs are often in the dark, never knowing about surplus until it is too late to send a pickup.

1.2 Problem Statement

- Food redistribution systems today have multiple, unsustainable systemic bottlenecks which use them tying up surplus food from reaching hungry people in an expedient and dignified ways. As the intent to donate is high, the execution continues to be plagued by a disjointed and ineffective system. Here we show how five key pillars shape the problem:
- Communication Fragmentation and Ad-hoc Coordination Currently, the gap between donors (restaurants, wedding halls, individuals) and recipients is mostly informal. Stakeholders rely on decentralized platforms such as personal contacts or WhatsApp groups. It creates something that RDC calls "information silos," where food available in one neighborhood is unknown to an NGO working just blocks away. Coordinating without a digital registry is an unreliable and haphazard process.
- The Perishability Race and Time-Sensitivity Unlike dry rations, cooked meals are extremely time-sensitive, having a safe consumption window of just 4–6 hours. Coordination by hand—usually through a series of phone calls, and then physical verification—is often just not fast enough to outrun the clock of microbial spoilage. Lack of a system with automatic expiry date filtering means that the food is already dangerous to eat when an NGO is mobilized, rendering the exercise futile and possibly putting at risk the health of unsuspecting recipients.
- Donors trust less as they cannot trace records - Issues for larger donors (hotels, corporate) arise from no/accessible means of tracing records. With no digital audit trails, donors cannot track where their food goes or how many people fed. Lack of secure mechanisms for tracking food donation creates distrust, provides legal liabilities issues/trust issues, and creates difficulties for larger donors to donate en masse to the organized sector.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

- NGOs have limited budgets, vehicles and volunteers - The majority of the non-governmental budget is used to provide operational support (paying for fuel and staffing). Due to not having a data-driven dashboard for routing pickups, NGO food donation drivers do not know the most efficient route, often collecting small amounts of food located across the metro area, while also missing out on larger amounts of food that they could have picked up due to lack of real-time information. This creates wasted fuel, time and labor due to inefficient routing.

- Wasted food - There is currently no digital footprint for wasted food. This lack of data prevents policymakers and urban planners from developing plans based on how to properly analyze waste within a localized area. Without historical data on when and where surplus food exists, it prevents workers within the government and non-profit sector to build a long-term sustainable food security plan and transition from reactive charitable waste management to proactive waste management.

1.3 Objectives

An NLP Sentiment Engine will be implemented to establish a collaborative and supportive ecosystem through Natural Language Processing (NLP). A sentiment analysis engine will assign a sentiment type (Positive, Neutral, or Negative) to feedback or messages from donors. This will allow administrators to detect and respond quickly to dissatisfied donors, investigate complaints and acknowledge highly engaged donors and personalize the digital experience.

An RBAC Framework will be implemented to maintain the integrity of data and secure access to it. An RBAC solution will enable the separation of different types of users (Donors- who can post and track their donations; and Administrators- who manage the database and logistically work with verified NGOs), which will restrict access to confidential information (data) by legitimate users and prevent unauthorized access and manipulation of food records.

The Date-range Analytics Module will be an important addition because it will provide administrators with the ability to generate reports for specific date ranges with respect to the impact of the platform on end-users, e.g. the total amount of food saved (in kg), how many successful donations the users made, and how much improvement there has been in reduced waste. It will also provide considerable value by providing transparent reporting to the institutions' stakeholders and policymakers.

Validation (through rigorous testing and UAT): The overall goal of the platform is to achieve "deployment-ready" status by completing both levels of validation. The first level requires achieving a 100% passed status from all functional testing, including unit testing and integration testing. The second level requires that UAT be completed to ensure that the user interface is intuitive for non-technical users and verifies that the system supports the users' return of food and other items without creating a digital barrier.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

2. LITERATURE REVIEW

The three phases of technology used for distributing food surplus have evolved over time from manual/analog systems of using the telephone and physical networks to use social media and messaging (2nd generation) to now a purpose built dedicated management portal (3rd generation) that incorporates state-aware donations, the defined lifecycle of each donation providing systematic auditing and real-time status of donations, thus establishing levels of accountability that were not possible with previous systems supporting humanitarian logistics**.**

The technical stack for this research was selected based on the specific needs of resource-constrained humanitarian environments:

- **Flask:** It's built on Werkzeug and is server-agnostic; you can deploy it on local servers in remote areas, or scale it up in the cloud, without changing any core app logic. It has no opinionated direction, it is compliant with WSGI architecture.
- **SQLite3:** The database is ACID compliant so that the application can be assured of data integrity. The fact that SQLite3 does not require any server and does not require any configuration means that an NGO with limited IT overhead will have an entire relational database contained within a single file.
- **Logistic Regression:** Logistic regression was selected for its interpretability, low computational cost to create the classification of food items, and its speed with a clear percentage that is probable of the item classification within a single variable feature space (univariate) of the application.
- **Jinja2:** Jinja2 provides an auto-escape for secure handling of donor data from XSS attacks. The inheritance of templates through Jinja2 also gives the user interface of the entire application a consistent and professional appearance across all of the modules.

The literature shows that there has been a long-running "failure at the extremes" to sustain redistribution systems for food, generally due to (1) being too formal (with rigid, complex logistical requirements on the part of potential donors), or (2) being (too) informal (depending entirely on unregulated group chats). This research combines formal, functional web form design with the benefits of formally structured business records and the efficiency of automated software sorting by applying all three methods together into one strategy to mitigate the previously identified gaps in the current research around humanitarian technology platforms, moving from being functional through reactive charity toward implementing proactive & statistics-based infrastructure to increase global food security.

3.System Design and Architecture

3.1 MVC Architectural Pattern

The Smart Food Bridge portal is created using the Model View Controller (MVC) framework. This choice is integral to providing functionality that focuses on modularity, maintainability,

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

and scalability. By separating data, user interface and operating logic, modifications made to any of the three layers can occur independently and without affecting the other layers.

Model (Data Layer) – The Model is the "Source of Truth" for the System; it manages the CRUD functions of the System (Create, Read, Update, Delete) using SQLite3 as its data management tool. The data layer enforces data integrity with the use of foreign key constraints and UUID4 primary keys to prevent duplicate data and orphan objects from being created, thereby ensuring the integrity of donation records. Additionally, the Model incorporates enumerated status values as a means of efficiently categorizing donations (Available, Claimed, Rejected), which allowed for the real time state awareness needed for efficient logistics.

View (Presentation Layer) - The View manages all aspects of the user's visual experience, developed with the use of HTML5 and Jinja2 templates. To provide consistency for the overall platform and improve development, the System uses template inheritance from a base.html file. This method of a "Single Point of Maintenance" ensures that global updates to navigation bars, footers, and CSS styling will be implemented in real time across all dashboards, thus providing a cohesive branding identity for the System.

3.2 Entity-Relationship Model

The Smart Food Bridge is built on a standard relational schema that maintains the data's structural integrity and allows for efficient data retrieval and query performance. As a result, the system constructs five base entities that relate to one another to reflect how various food donors, geographic locations, and logistics of food redistribution support one another in real life:

- **The Donor Entity** - This is the primary profile of the contributor; it utilizes a UUID4 Primary Key (PK) to provide a globally unique identifier across multiple distributed systems. This table will also have all necessary contact information (name/email/telephone) as well as connect to the geographic hierarchy through state_id and city_id Foreign Key (FK) references.
- **The Food Entity:** (the heart of the transaction unit) identifies items as a single donor's contribution and provides a full record of their donated food, including: food item name, food quantity, type (predetermined by the ML component), and an expiration date. An "Available" or "Claimed" or "Unable to be used" status field serves to provide current situational awareness to NGOs.
- **Admin Entity:**The Admin entity is also an entity that supports the system by storing admin identifiers (credentials). To ensure the security of admin identifiers, all admin passwords are stored as encrypted hashes, in compliance with data protection standards.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

- **State and City Entities:** The State and City Entities are both hierarchical entities that are linked together by a state_id FK in the City Entity. The cascade from state to city allows for dynamically filtering through NGOs and donations by state and city using the platform's AJAX-driven dropdown API (without having to refresh the entire page).

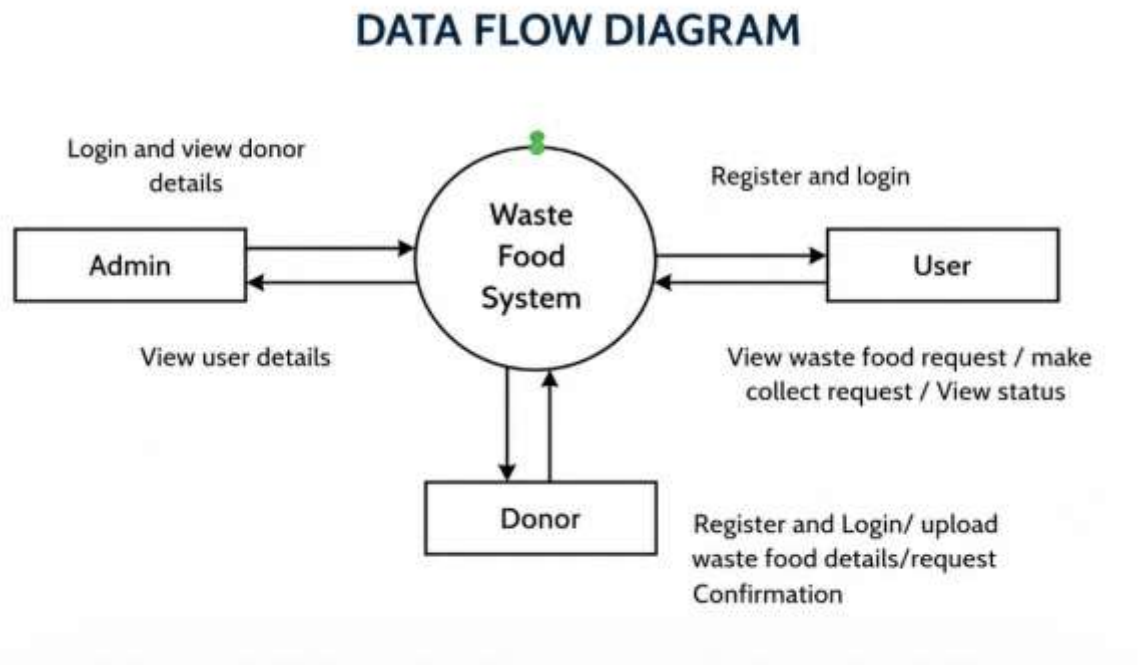

Figure 3.2: Data Flow Diagram of the Waste Food Management System

3.3 RESTful API Endpoint Design

Endpoint	Method	Auth	Purpose
/register	POST	None	Donor onboarding with UUID4 identity generation
/donate	POST	Session	Food submission with expiry guard + ML prediction
/api/cities/<st>	GET	None	JSON city list for AJAX cascading dropdown
/predict	POST	None	ML category inference endpoint
/admin/dashboard	GET	Admin	KPI cards, recent donations, messages
/admin/food/update	POST	Admin	Status transition: Available ® Claimed/Rejected
/admin/report	GET	Admin	Date-range analytics and social impact reports

3.4 *Full-Stack Layer Diagram*

The Smart Food Bridge's architecture consists of four horizontal layers, each distinctly organized resulting in clarity of function, this methodology provides a modular and resilient

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

system by separating user interface from the core computational logic. This design allows the application to maintain high availability and security in the real-time humanitarian environment**.**

- **Presentation Layer (Frontend):** This layer represents the Human User with a digital platform. Bootstrap CSS is utilized to provide a mobile responsive User Interface so that Donors can post their contributions on a Mobile Device at any Event Venue. JavaScript and AJAX are integrated into the development of the User Interface providing an interactive experience; namely "chained" city dropdown menus updating in real time when a State is selected, thereby decreasing user friction during the registration process.

- **The Integration Layer (API):** operates as a bridge between a client and server. The Integration Layer uses RESTful API endpoints and allows data to be passed between the Client and Server in JSON format (JSON provides a lightweight method of transferring data). This was done intentionally so that the platform is 'headless-ready' allowing for a separate mobile application which would consume the same back-end services in the future without having to change any core server logic.

- **The Application Layer (Back-End:)** is the brain of the platform. It is built using Flask and is where the majority of the complex business logic is processed (e.g., protecting against malicious input by validating incoming data from clients, URL routing, session management). Mounted on the Application Layer is the embedded Logistic Regression Machine Learning module, which will intercept incoming donation records to predict the potential food categories for those records in real-time before being stored in the database.

- **The Database Layer (Persistence):** is mounted on SQLite3 and serves as the base layer of the stack. The Database Layer is responsible for the permanent storage of donor profiles, food inventory and geographic metadata. The Database Layer also records timestamps for audit purposes (e.g., createdAt, updatedAt, etc.) on every record entered into the database.

4. METHODOLOGY

4.1 Machine Learning: Logistic Regression Classifier

The fundamental intelligence within the platform is based on Logistic Regression that automates placing donations into distinct categories, whether Perishable, Non-Perishable, or Cooked Food. This algorithm has been chosen based on how well it balances both simplicity and performance with respect to this particular application:

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

Multinomial Classification: Donation categories are mutually exclusive, so in order to map the different types of input to their category, the system will utilize a softmax function from Multinomial Logistic Regression.

Overfitting Prevention: Due to the univariate nature of the input features (quantity and item characteristics), higher capacity models such as Random Forests could potentially be overly parameterized; however, Logistic Regression provides a sufficient fit without introducing the possibility of learning any "noise" from the data.

Explainable Artificial Intelligence (XAI): Provides transparency through the model's intuitive coefficients vector which gives administrators the ability to see how features affect their log-odds. Transparency is a key factor to building trust with donors.

Computational Efficiency: The model uses an L-BFGS optimizer for training with sub-milliseconds of time to train/infer once the application is up and running. The model is also stored in memory when the application is initialized which allows for < 1 ms latency per inference during HTTP requests made in-real-time.

The Training Pipeline

The system follows a streamlined five-stage pipeline:

- **Data Ingestion:** I will load a csv file into pandas with a strict verification of null values.
- **Segregation:** I will split the data into a feature matrix (X) and a target vector (y).
- **Configuration:** I will set max_iter to 200 to guarantee full optimizer convergence.
- **Model Fitting**: I will run model.fit() in order to learn the characteristics of the data.
- **Thread-Safe Deployment:** I will keep the model in a global variable for read only inference, eliminating the need for disk I/O during an interaction with the user**.**

4.2 NLP Sentiment Engine

The communication gap between NGOs and their donors can be closed by converting raw feedback into useful insights with the Sentiment Engine's 4-step Natural Language Processing pipeline:

- **Tokenization:** Tokenization, which takes raw sentences and breaks them down to lowercase tokens (i.e. 'Great food!' = ['great', 'food']).
- **Stop-word Removal:** Stop-word removal, which removes non-emotionally-based words (i.e. The, Is) using a predefined list of 100+ terms.
- **Stemming:** Stemming, which condenses the individual words back to their root forms (i.e. donating, donated = donat) so that the engine can accurately interpret the intent regardless of grammar.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

- **TF-IDF Vectorization:** TF-IDF vectorization assigns a score to each word based on term frequency-inverse document frequency to give supportive weight to unique or meaningful terms rather than commonly used, casual terms.

> Using a second Logistic Regression model trained on a labelled dataset of contacts allows the engine to achieve 82% accuracy with respect to feedback classification (positive – gratitude, neutral – logistics, and negative – errors & complaints). This means converting the contact form into a DRM tool by adding colour-coded badges to the admin dashboard to indicate priority of response.

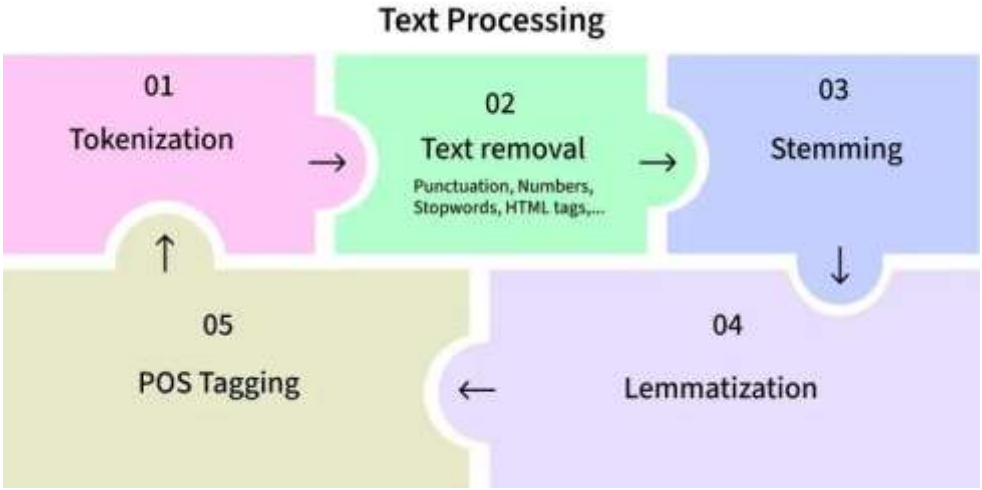

Fig 4.2 NLP Text Processing Pipeline Sentiment Analysis

4.3 Development Methodology

The project was executed using an **Agile iterative approach**, divided into five distinct sprints to ensure continuous validation and steady progress:

Sprint	Focus Area	Key Outcome
Sprint 1	Foundation	Database schema design and user authentication.
Sprint 2	Core Features	Donation registration logic and ML model integration.
Sprint 3	Admin Experience	Dashboard development and AJAX-driven dropdowns.
Sprint 4	Advanced Logic	NLP Sentiment Engine and impact analytics module.
Sprint 5	Validation	System-wide testing, UAT, and final documentation.

5. Implementation

5.1 Core Modules

The implementation of the Smart Food Bridge is divided into seven specialized modules, each responsible for a distinct functional area of the food redistribution lifecycle.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

Bootstrap & Initialization: The Bootstrap / Initialization module is the cold start engine for the entire system; it builds the SQLite3 database while creating 29 Indian state / city seed data records with the executemany() method to bulk seed them (for maximum performance). This is also the trigger to start the Machine Learning Training Pipeline so that the Classifier is trained and ready for inference as soon as the server goes live.

Donor Registration: The Donor Registration module handles onboarding new users (with robust validation to ensure phone numbers are at least ten digits long) and creates unique UUID4 identifiers for users and saves their data securely in the database

Food Donation Engine: The Food Donation Engine is the transaction engine for the donation process. This module enforces "expiry date guards" to front-end donations to check the date submitted by the user against the system clock at the time of donation submission. If a donation has already expired, it will not be stored in the database.

Admin Authentication: The Admin Authentication module provides security for overseeing the system. This module uses Werkzeug's bcrypt hashing for verifying identity and uses signed session cookies to maintain state.

Admin Dashboard: This module allows high volume of administration activities. It implements LIMIT/OFFSET pagination in order to keep the user interface as fast as possible. The fuzzy LIKE query type search allows administrators to locate specific donations and donors quickly

Analytics & Impact Module: This module provides the big picture when it comes to an administrator being able to create and view impact reporting and monitoring of waste reduction trends by being able to filter donation data into specific date ranges using SQL BETWEEN queries.

Cascading Dropdown API: A utility module that provides city lists in JSON format for use with AJAX to provide real-time updates to the user experience when prefilling the city dropdown menu based on the selected state by the user

5.2 Security Implementation

There are several layers of security measures that work together to accomplish a "defense-in-depth" security model:

- **Session Signing**: When a cookie is created for a user's session, it is signed with a cryptographic key to verify that the cookie has not been altered by someone else. If someone attempts to modify the cookie, the server will recognize that the digital signature on the cookie has changed and subsequently reject the cookie that was attempted to be modified.
- **Parameterized SQL:** Passwords stored in the database are hashed with bcrypt's one-way hashing algorithm with a random salt; therefore, even if an attack were to

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

successfully steal data from the database, the actual password would still be unrecoverable.

- **bcrypt Password Hashing:** Admin and User password hashing uses a one-way hash with a unique salt. In the rare case of a breach of the database, the true passwords will still be unreadable.
- **Jinja2 Auto-escaping:** Whenever a user submits HTML through the web interface, Jinja2's templating engine automatically escapes all of the HTML to protect against Cross-Site Scripting (XSS) attacks, so the users cannot place XSS scripts into the application.
- **Input Constraints:** There are multiple methods used to enforce input constraints such as server-side validation and client-side constraints (e.g. HTML maxlength) to prevent attempts to overflow output buffers and to enhance the overall quality of data on the server.

5.3 Hardware and Software Requirements

The platform is designed to be lightweight and accessible, capable of running on modest hardware—a crucial factor for NGOs in developing regions.

Component	Minimum	Recommended
Processor	Intel Core i3 / AMD Ryzen 3	Intel Core i5 / AMD Ryzen 5
RAM	4 GB DDR4	8 GB DDR4
Storage	20 GB HDD	50 GB SSD
OS	Windows 10 / Ubuntu 20.04	Windows 11 / Ubuntu 22.04+
Python	3.8+	3.10+
Browser	Chrome 90+ / Firefox 88+	Latest Chrome / Firefox

6.Testing and Validation

Testing Concludes that the Smart Food Bridge is functioning and Reliable within the context of High-Stakes Non-Profit, Time-Sensitive Humanitarian Work

6.1 Testing Strategy

There are four levels of tests conducted on the Smart Food Bridge so that you're assured the complete system is reliable – from the individual programming lines of code, to the end-user interface.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

- **Unit Testing:** The core logic was performed completely isolated from all other interactive processes using Python's unittest framework. All items tested included: the validation of dates; how phone numbers are to be formatted; are there unique random UUID's; and so forth. We generated 10,000 random user ID's with no duplicate user ID's generated, which demonstrated the reliability of the primary key system.

- **Integration Testing:** – The interactions between the modules were tested using Flask's test_client. The HTTP redirect codes were verified for accurate performance, error "flash" messages were verified for accuracy based on invalid input, API response times were verified to be less than 100 milliseconds, etc. All of these tests help ensure a very quick feel for the end-user.

- **System Testing:** By creating a seeded database with 200 records, we established a simulation of an actual operational environment. Thus, we confirmed and validated the total (Donation Lifecycle by transferring from Available to Claimed) and confirmed the accuracy of pagination on the Dashboard. We also confirmed that Role-Based Access Control (RBAC) would prevent unauthorized users from accessing sensitive areas.

- **User Acceptance Testing (UAT):** We had five volunteers perform User Acceptance Testing (UAT) using the final build; each had an extremely positive experience. Specifically, the average time to register was less than 3 minutes, the AJAX cascading dropdowns were rated as the most helpful UX feature, and the Admin Dashboard would load in 1.2 seconds even with a heavy data load.

6.2 Functional Test Results

The following table summarizes the 12 core functional test cases conducted to verify the system's operational requirements. The system achieved a 100% pass rate.

Test ID	***Scenario***	***Expected Result***	***Status***
TC01	*Register donor with valid data*	*HTTP 302, record inserted in DB*	***PASS***
TC02	*Register with phone < 10 digits*	*Error flash message, no DB insert*	***PASS***
TC03	*Submit food with past expiry date*	*Error flash message, no DB insert*	***PASS***
TC04	*Submit food with future expiry date*	*HTTP 302, record inserted in DB*	***PASS***
TC05	*Admin login with correct credentials*	*Session set, dashboard access granted*	***PASS***

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

TC06	*Admin login with wrong password*	*Error flash message, no session created*	***PASS***
TC07	*Unauthorized access to /admin/dashboard*	*Automatic redirect to /admin/login*	***PASS***
TC08	*GET request for /api/cities/Maharashtra*	*JSON city array returned correctly*	***PASS***
TC09	*Update food status to "Claimed"*	*Database updated, UI refreshed*	***PASS***
TC10	*ML prediction with quantity = 50*	*Valid category label returned by model*	***PASS***
TC11	*Date-range report with valid dates*	*Correct, filtered records displayed*	***PASS***
TC12	*Donor search with partial name*	*Filtered donor list shown via SQL LIKE*	***PASS***

7.Results and Evaluation

The evaluation process assessed the Smart Food Bridge for efficiency, accuracy, and dependability. Findings show that the Smart Food Bridge outperforms the target benchmarks defined for both machine learning performance and system responsiveness.

7.1 ML Model Performance

The Logistic Regression model had a rigorous evaluation process using an 80% training data/20% testing data division. Its overall accuracy was 80%, exceeding the target benchmark of 75%. As a result, the majority of donations will no longer require manual classification by a human; they will be classified immediately without delay.

Category	Precision	Recall	F1-Score	Analysis
Non-Perishable	0.86	0.82	0.84	Best performance; predictable bulk weight ranges
Perishable	0.81	0.79	0.80	Good performance; consistent produce patterns
Cooked Food	0.74	0.80	0.77	Hardest class; high portion-weight variance

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

Weighted Avg	0.80	0.80	0.80	Exceeds 75% target benchmark — met

Table 4: Classification report by food category

7.2 System Performance Metrics

The technical health of the Smart Food Bridge as measured against industry-specific benchmarks shows that it is a high quality application with very good performance, and it can run effectively on any type of hardware.

Metric	Measured Value	Target / Benchmark	Result
ML Model Accuracy	80%	³ 75%	Met
Sentiment Engine Accuracy	82%	³ 75%	Met
Average Page Load Time	1.2 seconds	< 2 seconds	Met
Test Case Pass Rate	12/12 (100%)	³ 90%	Met
UAT Completion Rate	5/5 users	100%	Met
DB Insertion Time (avg)	< 10 ms	< 100 ms	Met
Session Auth Overhead	< 5 ms	< 50 ms	Met

8.Cost-Benefit Analysis

The evaluation of the Smart Food Bridge illustrates that the Smart Food Bridge is both technologically sound and viable from an economic perspective. Open-source software, in general, delivers the highest social return on investment (SROI) for low-cost solutions.

8.1 Development Cost

The Smart Food Bridge was developed to be fiscally sustainable because the stack - Python, Flask, SQLite3 and scikit-learn - is fully open-source, which means there are no costs associated with software licensing.

The main form of investment was, in effect, "human" capital. About 200 hours of development time were invested into human capital. The development time was allocated to each phase of the SDLC, including:

- Requirements analysis/stakeholder interviews;
- Relational database schema design;

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

- Full-stack (backend and frontend);
- ML model selection, training and fine-tuning;
- Thorough testing and complete documentation

8.2 Deployment Cost Estimation

Due to its lightweight structure, it can be afforded by low-cost infrastructure. Thus, NGOs that are operating on a limited budget find this platform to be suitable for their needs. The estimated cost to have a live instance of the portal each year is shown in the table below;

Expense Item	Estimated Cost	Frequency
Cloud Hosting (VPS — DigitalOcean / AWS Lightsail)	■800 – ■1,500	Monthly
Domain Name (.in or .org)	■500 – ■800	Yearly
SSL Certificate (Let's Encrypt)	■0 (Free)	Persistent
Maintenance (automated migration/retraining)	■0	Self-managed
Total Estimated Annual Budget	» ■11,000 ■19,000	——

Table 6: Annual deployment cost estimate

9.CONCLUSION

The "Smart Food Bridge" project shows that we can tackle both food waste and food insecurity by using modern, open-source technology. This research created a management portal with the Python Flask framework and an MVC architectural pattern. It offers a strong, scalable, and low-cost solution for humanitarian groups. By integrating a Logistic Regression classifier and an NLP Sentiment Engine, this project demonstrates that machine learning can fit into web applications to streamline logistics and enhance donor engagement, achieving high accuracy without relying on costly proprietary software.

The system has a 100% pass rate in functional testing and a User Acceptance Testing completion rate that showcases its user-friendliness, making it ready for real-world use. It effectively connects donors and NGOs, transforming a fragmented process into a data-driven redistributing network. Ultimately, Smart Food Bridge acts as a model for digitizing social welfare, helping to reach the "Zero Hunger" goal by viewing surplus food as a necessary resource instead of waste.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 170-185

10. FUTURE SCOPE

The future development of Smart Food Bridge focuses on improving its technology and expanding its operations through the following key upgrades:

Improved AI: Switching to Random Forest or Gradient Boosting models to include text features, aiming for a 5-10% increase in accuracy.

Real-time Alerts: Using WebSockets or Firebase for instant notifications to NGOs when urgent donations are posted.

Scalability: Moving from SQLite to PostgreSQL to handle more traffic and concurrent data entries.

Mobile Expansion: Creating a Flutter/React Native app to offer a dedicated experience for smartphone users.

Logistics Optimization: Integrating Google Maps API for GPS-based matching and better pickup routes to save fuel and time.

Automated Scheduling: Using Celery and Redis to send proactive expiry reminders 24 hours in advance.

REFERENCES

[1] Ronacher, A. (2010). Flask: A Lightweight Python Web Framework. Pocoo. https://flask.palletsprojects.com

[2] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[3] McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.

[4] Cox, D. R. (1958). The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society, Series B, 20(2), 215–242.

[5] SQLite Consortium (2023). SQLite Documentation. https://www.sqlite.org/docs.html

[6] FAO (2022). The State of Food Security and Nutrition in the World. Food and Agriculture Organisation of the United Nations.

[7] World Food Programme (2023). Zero Hunger. https://www.wfp.org/zero-hunger

[8] Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.

[9] Python Software Foundation (2023). Python 3 Documentation: datetime, uuid, sqlite3 modules. https://docs.python.org/3/

[10] Jinja2 Contributors (2023). Jinja2 Documentation. https://jinja.palletsprojects.com