NAAN MUDHALVAN PROJECT REPORT

SB8067- SALESFORCE DEVELOPER

“APPLY LEFTOVER FOOD TO POOR“

Submitted by:

S.CHELLAPPAN(REG.NO:912422104009)

M.DEVENDIRAN(REG.NO:912422104010)

P.GAJENDRAN(REG.NO: 912422104012)

S.GOKUL(REG.NO:912422104013)

SHANMUGANATHAN ENGINEERING COLLEGE, ARASAMPATTI – 622 507

ANNA UNIVERSITY: CHENNAI - 600 025 NOV-DEC 2025

3. Requirement Analysis Phase — Apply Leftover Food to Poor

3.1 Introduction

Defines functional and non-functional requirements for the NGO-centered food-recovery system.

3.2 Functional Requirements

User Management: Secure registration/login for Admin, Donor, NGO, Volunteer roles.

Donation Posting: Donors create donation records (type, quantity, pickup window, location, photos).

Matching & Scheduling: NGOs/volunteers can view and accept donations; schedule pickups.

Pickup Tracking: Update statuses (Posted → Accepted → Picked-up → Distributed).

Safety Checklist: Donor and NGO verification steps (temperature, packaging, time since cooking).

Routing Assistance: Provide suggested pickup routes for volunteers.

Notifications: Real-time alerts (SMS/email/push) for new donations and status changes.

Reporting: Analytics on donations collected, beneficiaries reached, and waste reduced.

3.3 Non-Functional Requirements

Performance: Support concurrent users (donors and NGO users) with low latency.

Security: Authentication, role-based access, data encryption, minimal PII storage.

Reliability: High availability during meal-times and events.

Usability: Simple UI for low-tech NGO staff and volunteers.

Scalability: Able to add new cities/NGOs and integrate SMS gateways.

3.4 System Requirements

Hardware: Servers or cloud VM, volunteers' mobile devices for pickup confirmations. Software: Web browser for donors/NGOs, Node.js/Django, React, MySQL/Postgres, mapping API.

3.5 User Requirements

Donor: Quickly post surplus food and indicate pickup constraints.

NGO/Volunteer: Browse donations, accept, schedule pickups, and mark distribution.

Admin: Verify NGO registrations, monitor operations, and manage guidelines.

3.6 Feasibility

Technical: Uses proven web stack; mapping APIs and SMS gateways available.

Operational: NGOs can be onboarded with training materials; volunteers use mobile confirmations.

Economic: Low-cost open-source stack minimizes operational costs.