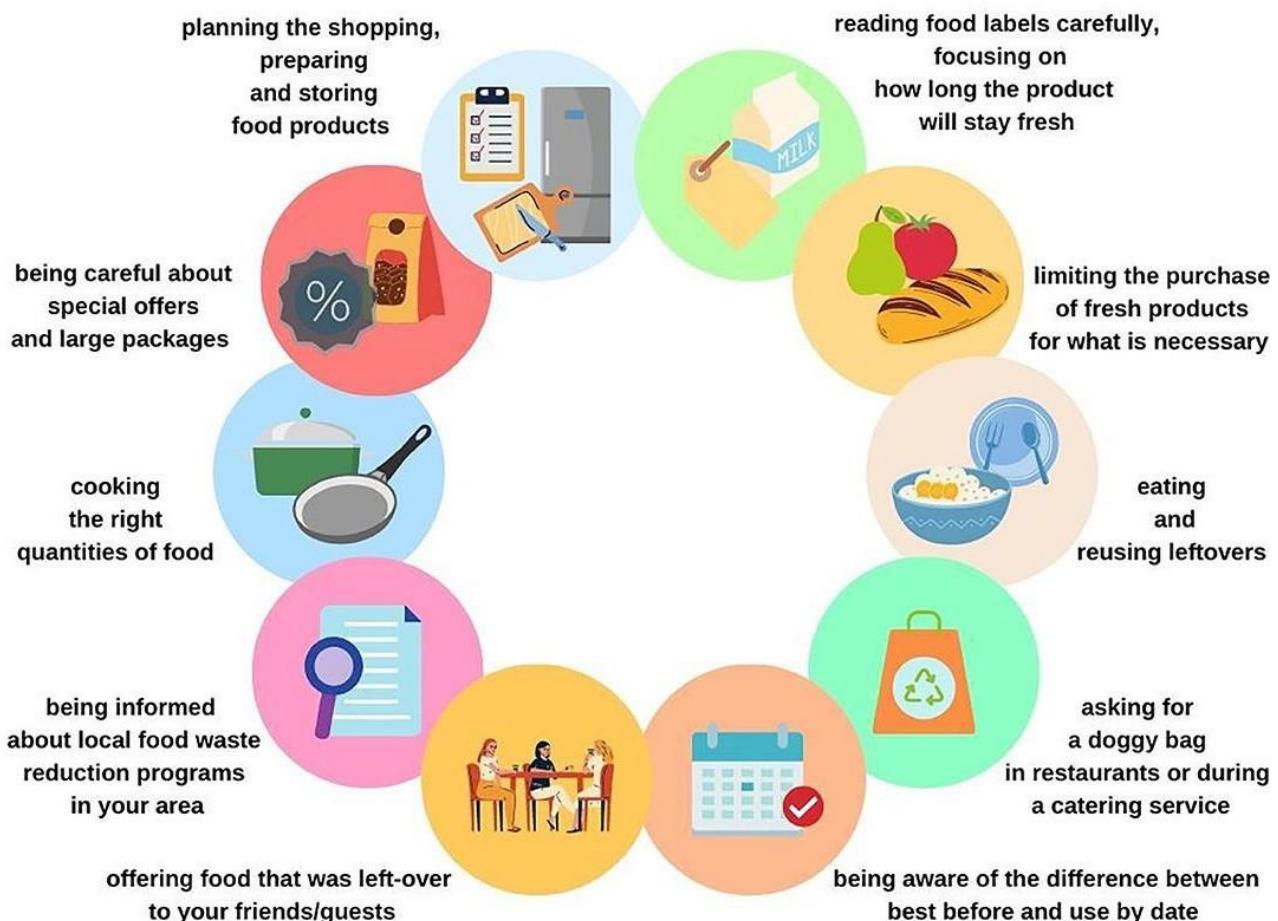


## TECHNOLOGY STACK

<b>Date</b>	04 november 2025
<b>Team ID</b>	NM2025TMID04293
<b>Project Name</b>	To supply leftover food to poor
<b>Maximum Marks</b>	5 Marks

### Technical Architecture :

A scalable leftover-food redistribution platform should consist of mobile apps for donors and volunteers, and a web portal for NGO administrators, all routed through an API gateway and load balancer to stateless backend services. Core backend components include authentication & RBAC, a task-matching service that pairs available food pickups with volunteers, a routing engine (maps + route optimization), and a realtime task queue (Pub/Sub or message broker) for reliable delivery workflows. Persistent storage combines an RDBMS for transactional records and a geo-enabled datastore (PostGIS or similar) for pickup/delivery locations; audit logs, telemetry, and analytics feed an operations dashboard. Integrations include SMS/notifications, mapping APIs, and optional payment or compliance services.



### **Table-1 : Components & Technologies :**

The system will be structured around three primary components to handle the defined functional requirements:

#### **1. Operational Layer(Front-End/User Interaction):**

- **Function:** Handles all daily user-facing activities such as posting leftover food availability, assigning volunteers for pickup, tracking delivery to beneficiaries, and confirming successful distribution. This layer is used by Donors, Volunteers, and NGO/Admin staff.
- **Components:** Donor App (Food Submission, Pickup Request)

#### **2. Analytical Layer (Data/Reporting):**

- **Function:** Analyzes operational data to generate insights on donation patterns, volunteer performance, delivery efficiency, food wastage reduction, and community impact. Helps NGOs make data-driven decisions.
- **Components:** Reporting Dashboard (Daily / Monthly Donation & Delivery Reports)

#### **3. Integration Layer (Middleware/Back-End):**

- **Function:** Ensures smooth end-to-end communication across mobile apps, database, maps APIs, and optional IoT sensors. Handles authentication, routing, and secure communication between modules.
- **Components:** API Gateway (Request Routing, Rate Limiting)

### **Table-2: Application Characteristics:**

#### **1. High Security and Data Protection**

- **Characteristic:** The application handles sensitive donor, volunteer, and beneficiary information including contact details, location data, food safety status, and delivery records.
- **Impact:** Requires secure authentication, encrypted data transmission, role-based access control, and continuous monitoring to protect user identity and prevent data misuse.

#### **2. Integration Capability**

- **Characteristic:** The system integrates with external services including maps API (for routing), messaging/notification services, and optional IoT modules for food temperature

- **Impact:** Needs API gateway, secure integration protocols, and reliable sync mechanisms for accurate location tracking, timely communication, and food-safety compliance.

### 3. Real-Time Operations

- **Characteristic:** The entire workflow depends on real-time coordination between donors (food availability), volunteers (pickup & delivery), and NGO admins (task assignment & monitoring).
- **Impact:** System must support instant notifications, live location tracking, real-time task updates, and low-latency responses to ensure food reaches beneficiaries on time.

### 4. Scalability & High Availability

- **Characteristic:** The platform is expected to handle large numbers of users, peak-time donation requests, volunteer assignments, and city-wide operations.
- **Impact:** Requires scalable cloud infrastructure, load balancing, and database optimization to ensure continuous availability without system slowdowns or downtime.

### 5. User-Centric & Accessible

- **Characteristic:** Designed for users from different backgrounds — restaurant owners, volunteers, NGOs — requiring simple and multilingual UX.
- **Impact:** Provides intuitive UI, step-by-step workflows, accessibility support, and multi-device compatibility to enable easy adoption and efficient operations.