

## Project Design Phase-II Technology Stack (Architecture & Stack)

Date	01 November 2025
Team ID	NM2025TMID07455
Project Name	To Supply Leftover Food to Poor
Maximum Marks	4 Marks

### Technical Architecture:

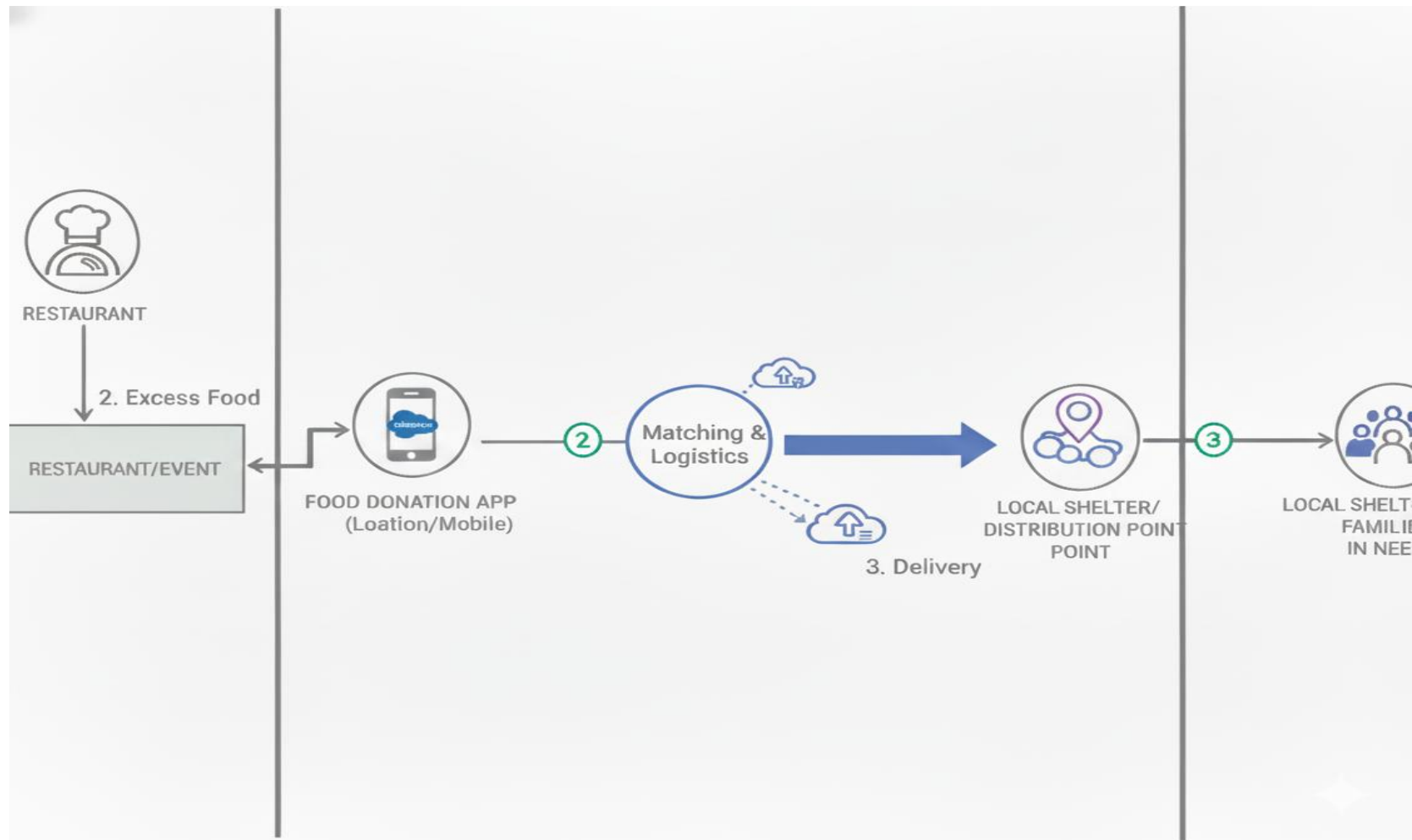
The deliverable includes the **architectural design** of the food redistribution system, showing all components — from donor registration to food delivery.

**Example: Food Collection and Distribution Management System for Social Welfare.**

**Reference:** <https://www.createely.com/diagram/example/io5k80v01/food-distribution-management-system>

### Guidelines:

- Include all processes such as donor registration, food listing, volunteer assignment, and delivery tracking.
- Provide clear separation of components between local device (user access) and cloud services (database, APIs).
- Indicate external interfaces (Google Maps API, NGO verification systems, etc.).
- Specify data storage components (MySQL or Firebase).



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

S.No	Component	Description	Technology
1.	User Interface	Donors, volunteers, and admins interact through mobile and web applications.	HTML, CSS, JavaScript, Flutter
2.	Application Logic-1	Handles donor registration and food listing workflow.	Python (Django / Flask)
3.	Application Logic-2	Matches donors with nearby NGOs or volunteers using location services.	Google Maps API Integration
4.	Application Logic-3	Coordinates pickup and delivery schedules; sends status notifications.	Firebase Cloud Messaging
5.	Database	Stores user details, food listings, and delivery data.	MySQL / Firebase Realtime Database
6.	Cloud Database	Maintains backup of food records and NGO networks on cloud servers.	Google Cloud / AWS Database
7.	File Storage	Stores images of food and donor receipts.	Firebase Storage / Cloud Storage
8.	External API-1	Uses Google Maps for live location and routing.	Google Maps API
9.	External API-2	Integrates with NGO verification or government food safety API.	REST API
10.	Machine Learning Model	Predicts high food waste areas and optimal delivery routes.	Python (scikit-learn / TensorFlow)
11.	Infrastructure (Server / Cloud)	Hosted and managed on a cloud-based platform for 24/7 service.	AWS / Firebase Cloud Hosting

**Table-2: Application Characteristics:**

<b>S.No</b>	<b>Characteristics</b>	<b>Description</b>	<b>Technology</b>
1.	Open-Source Frameworks	Uses open-source tools for development and deployment.	Django, Flask, Flutter
2.	Security Implementations	Ensures donor and recipient data privacy and secure access.	Authentication, SSL Encryption
3.	Scalable Architecture	Cloud-based system that scales with more users and donors.	Firebase / AWS Auto Scaling
4.	Availability	Highly available and operational at all times for emergency requests.	Load-balanced Cloud Servers
5.	Performance	Optimized for fast data access and low-latency API responses.	Indexed Database Queries, RESTful APIs