**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| **Date** | **2 NOV 2025** |
| **Team ID** | NM2025TMID02631 |
| **Project Name** | Supply Leftover Food to Poor |
| **Maximum Marks** | 4 Marks |

---

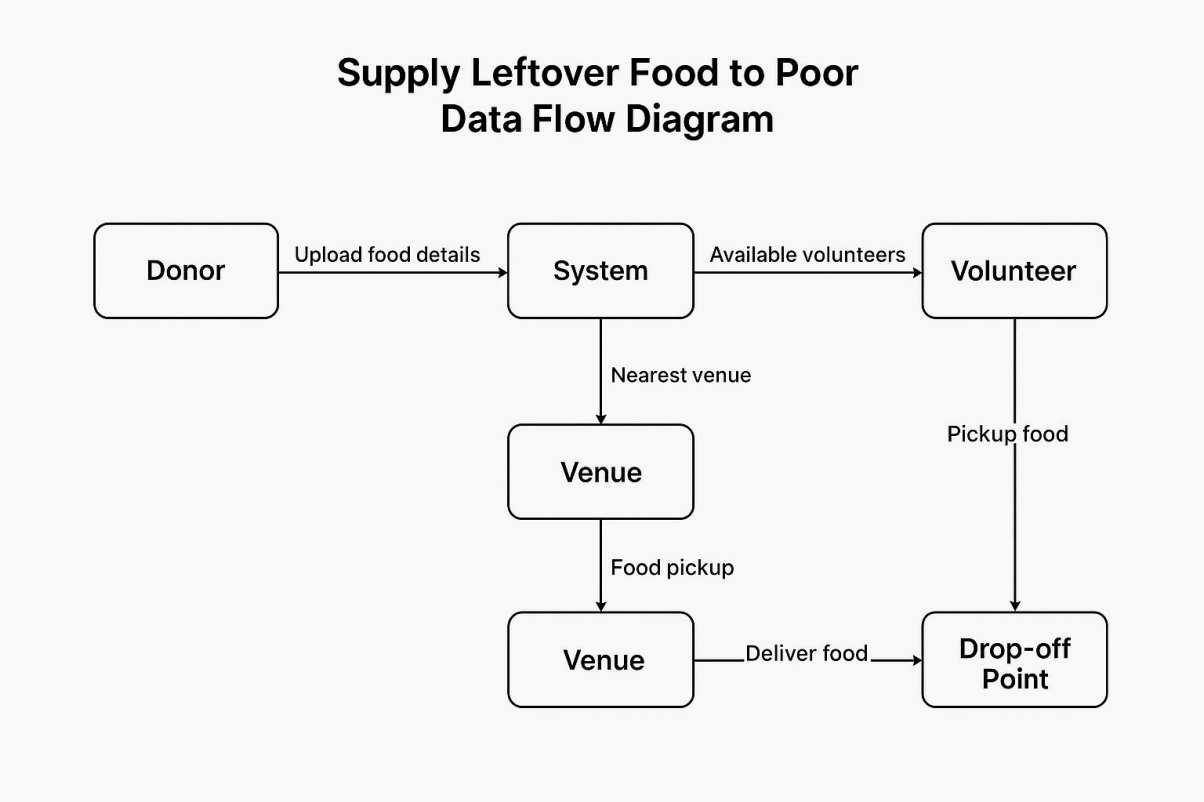
**Technical Architecture:**

The deliverable shall include the architectural diagram (to be created later) and the information as per Table 1 & Table 2.

**Example:** Food collection and distribution management system for surplus food donation.

**Reference:** https://dns00000ems3u2ac-dev-ed.develop.lightning.force.com/lightning/page/home

---




**Guidelines:**

- Include all processes (Application Logic / Technology Stack)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (Third-party APIs, Maps, etc.)

- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

---

**Table 1: Components & Technologies**

| S.No | Component | Description | Technology |
|---|---|---|---|
| **1.** | **User Interface** | Donor and volunteer interact through a web or mobile dashboard. | React Native (Expo) / Web UI |
| **2.** | **Application Logic-1** | Handles donor food upload forms, images, and location details. | Node.js / Express.js |
| **3.** | **Application Logic-2** | Assigns volunteers based on nearest location and available slots. | Firebase Cloud Functions |
| **4.** | **Application Logic-3** | Sends real-time notifications for pickup, drop-off, and delivery confirmation. | Firebase Cloud Messaging (FCM) |
| **5.** | **Database** | Stores donor details, volunteer tasks, food info, and drop-off locations. | Firebase Firestore / Realtime Database |
| **6.** | **Cloud Database** | Stores backup of all transactions and images securely. | Google Cloud Firestore Backup |
| **7.** | **File Storage** | Stores uploaded food images and related files. | Firebase Storage |
| **8.** | **External API-1** | Google Maps API for distance and venue location tracking. | Google Maps API |
| **9.** | **External API-2** | Optional – NGO / Food Shelter API for direct delivery requests. | REST API Integration |
| **10.** | **Machine Learning Model** | Suggests optimal routes and identifies food wastage patterns. | TensorFlow Lite (optional) |
| **11.** | **Infrastructure (Server / Cloud)** | Hosted and managed on Firebase / Google Cloud Platform (GCP). | Firebase Hosting / GCP |

**Table-2: Application Characteristics**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | **Open Source Frameworks** | Uses open-source and cross-platform technologies for cost-effective and scalable deployment. | React Native, Node.js, Firebase SDK |
| 2. | **Security Implementations** | Secure user authentication and data protection with encrypted database access and role-based permissions. | Firebase Authentication, HTTPS, Firestore Security Rules |
| 3. | **Scalable Architecture** | Cloud-based, horizontally scalable structure capable of handling multiple donors, volunteers, and NGOs. | Google Cloud Platform (GCP), Firebase Cloud Functions |
| 4. | **Availability** | Highly available and accessible 24/7 through cloud hosting for all registered users. | Firebase Hosting, Cloud Backup |
| 5. | **Performance** | Optimized for real-time synchronization and minimal response delay in data updates and notifications. | Firebase Realtime Database, Cloud Messaging |