

Technical Architecture:

The deliverable includes the architectural diagram and component details as per the project requirements.

The “*To Supply Leftover Food to Poor*” system will be developed as a **web and mobile-based application** that connects **food donors** (restaurants, hotels, hostels) with **receivers** (NGOs and volunteers).

The system architecture is divided into three primary layers:

1. **Frontend Layer** – for user interaction and data input.
2. **Application Layer** – for managing logic, coordination, and automation.
3. **Database Layer** – for storing and retrieving donation, user, and delivery information.

The solution is cloud-hosted for better scalability and availability, using secure APIs for notifications and real-time tracking.

Reference:

<https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

Guidelines:

- Include all processes (food collection, verification, distribution) as application logic components.
- Indicate **cloud-based infrastructure** for database and storage.
- Integrate **external APIs** (Google Maps, Twilio) for location tracking and communication.
- Use a **centralized database** for all donor and NGO-related data.
- Optionally connect to **AI modules** for food demand prediction or waste reduction insights.

Table - 1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Donors, NGOs, and volunteers interact via web and mobile applications.	
2	Application Logic – 1	Handles donor registration and login functionalities.	
3	Application Logic – 2	Manages food listing, notifications, and volunteer assignment.	
4	Application Logic – 3	Processes food pickup confirmation and	

5	Database	updates. Stores donor, food, volunteer, and distribution data.	HTML5, CSS3, Bootstrap 5, React.js Node.js / Express.js Python (Flask) / REST API Django REST Framework MySQL / PostgreSQL
6	Cloud Database	Provides scalable and secure data storage.	AWS RDS / Firebase
7	File Storage	Maintains food images, feedback files, and reports.	AWS S3 / Google Cloud Storage
8	External API – 1	Sends SMS/email alerts for food pickup and delivery.	Twilio / SendGrid API
9	External API – 2	Integrates location services for tracking and mapping.	Google Maps API
10	Machine Learning Model	Predicts demand zones and optimal food collection routes.	TensorFlow / Scikit-learn
11	Infrastructure (Server)	Hosted and maintained on a scalable cloud	AWS EC2 / Azure App

Table – 2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Developed using free and open-source web technologies.	React, Node.js, Bootstrap
2	Security Implementations	Role-based access, encrypted user data, and HTTPS protocols. efficiently.	JWT Authentication, SSL
12			
13	Availability	High uptime ensured through cloud hosting and redundant servers.	
14		Optimized backend APIs and caching for real- time notifications.	
15	Performance Maintainability	Modular code structure for easy updates and bug fixes.	
16	Integration	Supports integration with third-party APIs for communication and logistics.	

