

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.	HTML5, CSS3, Bootstrap 5, React.js
2	Application Logic – 1	Handles donor registration and login functionalities.	Node.js / Express.js
3	Application Logic – 2	Manages food listing, notifications, and volunteer assignment.	Python (Flask) / REST API
4	Application Logic – 3	Processes food pickup confirmation and delivery updates.	Django REST Framework
5	Database	Stores donor, food, volunteer, and distribution data.	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 / Cloud)	Hosted and maintained on a scalable cloud environment.	AWS EC2 / Azure App Services

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.	JWT Authentication, SSL
3	Scalable Architecture	Can handle multiple users, donors, and transactions efficiently.	Cloud Auto-Scaling
4	Availability	High uptime ensured through cloud hosting and redundant servers.	AWS Cloud Hosting
5	Performance	Optimized backend APIs and caching for real-time notifications.	Redis, RESTful Services
6	Maintainability	Modular code structure for easy updates and bug fixes.	MVC Framework
7	Integration	Supports integration with third-party APIs for communication and logistics.	REST / JSON APIs