## Project Design Phase-II

Technology Stack (Architecture & Stack)

Date: 1 November 2025

Team ID: NM2025TMID02077

Project Name: Supply Leftover Food for Poor People

Maximum Marks: 4 Marks

### Technical Architecture:

This project aims to create a web-based platform where restaurants, event organizers, and individuals can donate their leftover food to poor and needy people. The system enables donors to upload food details, and receivers can view and request available food. Admins monitor donations and ensure food safety compliance.

Architecture Overview:
- Frontend: HTML, CSS, JavaScript (User Interface for Donors and Receivers)
- Backend: PHP (Handles business logic and API communication)
- Database: MySQL (Stores user details, donations, and requests)
- Hosting: Local/Cloud-based PHP Server
- External Interfaces: Google Maps API for location tracking
- Cloud Storage: Stores uploaded food images

### Table-1 : Components & Technologies

| S.No | Component Description | Technology |
|---|---|---|
| 1 | User Interface | HTML, CSS, JavaScript (Frontend) |
| 2 | Application Logic-1 | PHP backend for donor and receiver modules |
| 3 | Application Logic-2 | Handles donation and request validation using PHP |
| 4 | Application Logic-3 | Admin approval and |

| | | |
|---|---|---|
| | | notifications system |
| 5 | Database | MySQL database for storing users, donations, and requests |
| 6 | Cloud Database | MySQL hosted on cloud/localhost |
| 7 | File Storage | Stores images of donated food items |
| 8 | External API-1 | Google Maps API for donor and receiver location tracking |
| 9 | External API-2 | Optional email notification service |
| 10 | Machine Learning Model | Future enhancement for predicting food demand areas |
| 11 | Infrastructure | Hosted on Local/Cloud PHP Server |

### Table-2 : Application Characteristics

| S.No | Characteristics | Technology |
|---|---|---|
| 1 | Open-Source Frameworks | PHP, MySQL (Open Source) |
| 2 | Security Implementations | User authentication, form validation, and admin access control |
| 3 | Scalable Architecture | Can be deployed on local or cloud servers with database scaling |
| 4 | Availability | 24/7 web access via hosting server |
| 5 | Performance | Optimized with PHP caching and database indexing |