|  |  |
| --- | --- |
| Date | 29-05-2025 |
| Team ID | LTVIP2025TMID28821 |
| Project Name | TO SUPPLY LEFTOVER FOOD TO POOR |
| Maximum Marks | 4 Marks |

**REQUIREMENT ANALYSIS**

**TECHNOLOGY STACK**

**1. Introduction**

The implementation of *“To Supply Leftover Food to the Poor”* relies on an intelligently selected technology stack that supports a robust, scalable, and efficient cloud-based solution. The technological foundation of this project—NourishBridge—is built primarily on Salesforce, enabling powerful CRM, automation, mobile accessibility, and secure user portals.

The technology stack not only ensures seamless collaboration between food donors, NGOs, and volunteers, but also delivers automation, real-time reporting, and high-level security. This section outlines the complete ecosystem of tools and platforms that make NourishBridge technically sound and operationally sustainable**.**

**2. Core Platform: Salesforce**

Salesforce serves as the primary foundation for all backend, automation, UI, and data management activities. It was chosen for the following reasons:

* Low-Code Development: Enables rapid deployment using clicks over code.
* Security & Compliance: Trusted infrastructure with built-in access control and data protection.
* Scalability: Supports growing user bases and data volumes.
* Cloud-Based Accessibility: Operates on any device, anywhere, with real-time synchronization.

**3. Components of the Technology Stack**

**3.1 Salesforce Platform License**

* Hosts custom objects such as Donation\_\_c, NGO\_Profile\_\_c, and volunteer assignment data.
* Standard objects like Account and Contact are reused for restaurants and NGO profiles.
* Data model is highly relational and optimized for donation tracking and user segmentation.

**3.2 Salesforce Experience Cloud**

* Creates custom portals for three user groups:
  + Donor Portal: For posting donations and tracking impact.
  + NGO Portal: For accepting donations and managing pickups.
  + Admin Panel: For verification, analytics, and system monitoring.
* These portals provide a branded, mobile-responsive, and secure environment.

**3.3 Lightning Web Components (LWC)**

* Used to build the user interface with modern and responsive web elements.
* Provides fast load times and smooth interactions for donation forms, dashboards, and real-time updates.

**3.4 Salesforce Flows (Automation Layer)**

* Handles backend logic like:
  + Auto-matching donations with nearby NGOs.
  + Sending alerts and reminders.
  + Updating donation status across user portals.
* No-code/low-code flows accelerate development and reduce bugs.

**3.5 Apex Programming**

* Used when business logic exceeds the capabilities of Flows.
* Examples include:
* Preventing duplicate claims on a donation.
* Integrating with external APIs such as SMS gateways.
* Validating user permissions and workflow consistency.

**4. Integration Tools**

**4.1 Twilio API**

* Sends SMS alerts to NGOs when a new donation is posted.
* Enables real-time communication and ensures donors get immediate responses.

**4.2 Salesforce Maps (Future Scope Integration)**

* Planned integration to visualize donation sources, NGO service zones, and volunteer routes.
* Helps NGOs optimize logistics and minimize delivery time.

**5. Analytics and Reporting**

**5.1 Salesforce Reports & Dashboards**

* Out-of-the-box reporting features track:
  + Total meals served.
  + Quantity of food saved.
  + Volunteer activity.
  + City-wise donor engagement.
* Admins, NGOs, and donors each have customized dashboards for impact tracking.

**5.2 Export & Documentation**

* Supports generating monthly or quarterly reports in PDF/CSV format.
* Beneficial for recordkeeping, donor recognition, and funding audits.

**6. Data and Security Architecture**

**6.1 Data Storage**

* All data resides within Salesforce’s encrypted cloud database.
* Custom and standard objects are used with lookups, validation rules, and record types for clean data handling.

**6.2 Security Features**

* Role-based access ensures only authorized users can view or modify data.
* Multi-factor authentication and session control add additional protection.
* Audit trails track every user activity for compliance and transparency.

**7. Scalability and Future Readiness**

* Cloud-Native: The entire stack runs on Salesforce’s cloud infrastructure, enabling easy upgrades and global accessibility.
* API-First: Future integrations can add support for food safety tracking, route optimization, multilingual support, or IoT-based collection bins.
* Mobile-Friendly: Experience Cloud and LWC ensure responsive design across all devices; a native mobile app may be developed using Salesforce Mobile Publisher.

**8. Conclusion**

The technology stack powering *“To Supply Leftover Food to the Poor”* is a well-integrated, cloud-based ecosystem that blends performance, scalability, and usability. By harnessing the full potential of Salesforce and its extensions—such as Apex, Flows, and Twilio APIs—this solution automates critical workflows, enhances stakeholder trust, and empowers communities.

Technology isn’t just the backbone of this project—it’s the bridge that transforms compassion into action. The chosen stack ensures that surplus food doesn’t end in waste, but in hope, dignity, and sustenance.