

**ANNAMACHARYA INSTITUTE OF TECHNOLOGY &SCIENCES**

**To Supply Leftover Food To Poor**

Submitted by:

Pavani Mocharla

22701A05E8

Department of Computer Science and Engineering

Submitted to:

Salesforce MP Mentor1

SkillWallet Platform

Academic Year: 2024–2025

**TO SUPPLY LEFTOVER FOOD TO POOR**

# **INDEX**

## **1. Project Overview**

## **2. Objectives**

## **3. Phase 1: Requirement Analysis & Planning**

- Understanding Business Requirements
- Defining Project Scope and Objectives
- Design Data Model and Security Model

## **4. Phase 2: Salesforce Development - Backend & Configurations**

- Setup Environment & DevOps Workflow
- Customization of Objects, Fields, Validation Rules, Automation
- Apex Classes, Triggers, Asynchronous Apex Classes

## **5. Phase 3: UI/UX Development & Customization**

- Lightning App Setup through App Manager
- Page Layouts, Dynamic Forms
- User Management
- Reports and Dashboards
- LWC Development (If Any)
- Lightning Pages

## **6. Phase 4: Data Migration, Testing & Security**

- Data Loading Process
- Field History Tracking, Duplicate Rules, Matching Rules
- Profiles, Role Hierarchy, Permission Sets, Sharing Rules
- Test Classes
- Test Cases with Screenshots

## **7. Phase 5: Deployment, Documentation & Maintenance**

- Deployment Strategy
- System Maintenance and Monitoring
- Troubleshooting Approach

## **8. Conclusion**

## **1. Project Overview**

The project titled "**To Supply Leftover Food to Poor**" is a socially-driven Salesforce CRM application designed to bridge the gap between surplus food sources and communities in need. This solution provides an organized platform where food donors — including restaurants, households, and event venues — can log leftover food donations. Once a donation is submitted, the system automatically assigns available volunteers to pick up and deliver the food to designated drop-off points like NGOs, shelters, or distribution centers.

Built on the Salesforce platform, the CRM utilizes automation, custom objects, flows, and dashboards to facilitate the food delivery lifecycle. Volunteers can view tasks, donors can track the status of their donations, and NGOs can manage drop-off schedules efficiently. The CRM also supports mobile access, approval workflows, validation rules, and secure role-based access to ensure only authorized users interact with sensitive data.

This project addresses two major global challenges — food wastage and hunger — by using cloud technology for good. It not only simplifies operations for charities but also builds transparency and accountability into food distribution processes.

## **2. Objective**

The primary objective of this Salesforce CRM project is to create a centralized, intelligent system that manages the entire cycle of leftover food collection and distribution to the underprivileged. The platform aims to streamline coordination among food donors, volunteers, and drop-off centers (such as NGOs and shelters) by automating key processes, ensuring accountability, and enhancing communication through Salesforce features.

Specifically, this CRM focuses on:

- Reducing food wastage by creating a real-time logging system for leftover food.
- Automating volunteer task assignment to ensure timely pickup and delivery.
- Ensuring transparency via detailed reports and dashboards.
- Establishing secure, role-based access to protect sensitive donor and beneficiary data.

- Providing easy access to operational insights for administrators and NGOs.

### **Business Value:**

- Improves operational efficiency in charitable food distribution.
- Enhances trust among donors through transparent reporting.
- Minimizes delays in food delivery, ensuring it reaches recipients before spoilage.
- Strengthens coordination between NGOs, volunteers, and donors with scalable workflows.

## **3. Phase 1: Requirement Analysis & Planning**

### **3.1 Understanding Business Requirements**

In this CRM, three user roles have been carefully studied to identify their operational pain points:

#### **Donors (Restaurants, Caterers, Event Venues, Households):**

Need	How CRM Solves It
Quick way to log available food	Lightning Record Page + Flow-triggered donation submission
Status tracking of donated food	Dashboard with “Donation Status by Date” chart

Scheduled donation entries	Time-based Flows with Calendar integration (future enhancement)
Scheduled donation entries	Scheduled donation entries

**Volunteers:**

Need	How CRM Solves It
Real-time task alerts	Email/SMS alerts via Flow + Notification component
Mobile-compatible platform	Salesforce Mobile App (configured for Volunteer profile)
Record delivery time and notes	Custom object “Execution Detail” linked to each Task
Review own history of pickups	Volunteer-specific report dashboard

## **NGOs/Drop-Off Points:**

<b>Need</b>	<b>How CRM Solves It</b>
Track upcoming food deliveries	“Today’s Incoming Deliveries” report
Maintain historical data for compliance	All donation and delivery logs stored via linked records
Identify delays or unassigned pickups	Report: Tasks with no Assigned Volunteer
View capacity status	Drop-Off Point object field + Custom Alerts (planned)

## **Other business needs solved:**

- Auto-escalation for failed pickups (via Apex/Flow)
- Volunteer overbooking prevention using validation
- Record versioning using Field History Tracking

### **3.2 Defining Project Scope and Objectives**

#### **Scope:**

<b>Area</b>	<b>Details</b>
Custom Objects	6 custom objects fully defined with related fields
Automation	Auto-assignment Flows, Notification Flows, Approval for sensitive actions
UI Customization	Lightning Pages, Page Layouts, Dynamic Forms per profile
Reporting & Dashboards	Delivery Efficiency, Volunteer Activity, Daily Donation Summary
Security	Role hierarchy, permission sets, object-level & field-level restrictions

### **Success Metrics:**

- 100% of donations are picked up within 2 hours
- Volunteer assignment automation saves 60% manual effort
- Dashboards update every 15 mins (via scheduled reports)
- NGOs can view real-time inventory at their drop-off centers

### **Out-of-Scope (for now):**

- Integration with external apps (e.g., Google Maps, WhatsApp)
- Food safety temperature tracking
- Donor rating and feedback module

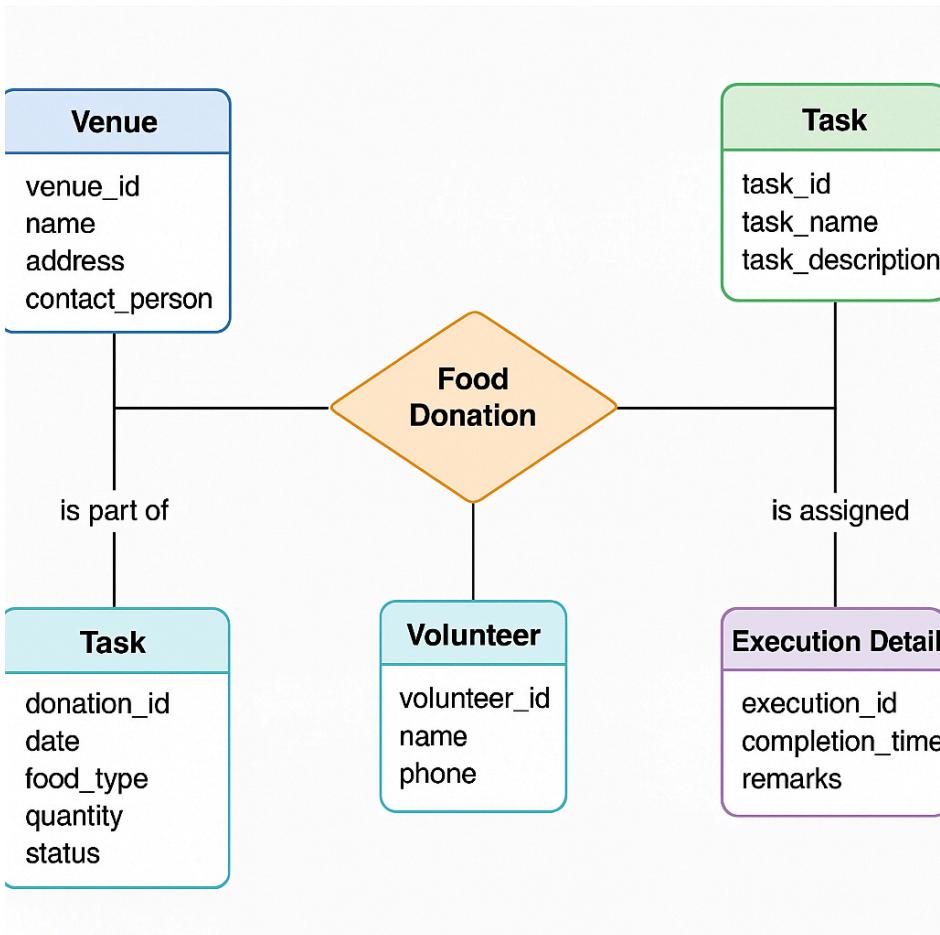
### **3.3 Design Data Model and Security Model**

#### **Object Relationships Table:**

Object	Related To	Relationship Type	Relationship Type
Volunteer__c	Task__c	One-to-Many	Each volunteer can have many tasks
Food_Donation__c	Task__c	One-to-Many	One donation may require multiple

			pickups
Task__c	DropOffPoint__c	Lookup (One-to-One)	Each task leads to one drop-off location
Task__c	Execution_Detail__c	One-to-One	Track delivery status
Venue__c	Food_Donation__c	Lookup	Venue source for the donation

## ER Diagram Placeholder



## Security Model

### Profiles:

Profile	Access Rights
Donor	Can create/view Food Donations, View Pickup Status only
Volunteer	Can view assigned Tasks, update Execution Details

NGO Coordinator	Can view Drop-Offs, Task Status, Run Reports
Admin	Full access to all components, manage automation & security

**Roles:**

- Donor
  - Volunteer
    - NGO Coordinator
    - Project Admin

**Permission Sets:**

Name	Purpose
Can_Log_Donation	Add new food donation records
Can_Execute_Task	Allow Volunteer to update Execution status

Can_Access NGO Dashboard	View reports and charts assigned to NGO role
--------------------------	--

### Sharing Rules:

- Tasks are shared with assigned volunteers
- Donations shared with the drop-off NGO
- Admins have read/write access to all record

The screenshot shows the Salesforce Sharing Settings page. The URL in the browser is <https://orgfarm-d7a7500f0e-dev-ed.lightning.force.com/lightning/setup/SecuritySharing/home>. The page displays sharing rules for different work types and objects.

**Work Type Group Sharing Rules**: No sharing rules specified.

**Drop-Off Point Sharing Rules**:

Action	Criteria	Shared With	Access Level
Edit   Del	Drop-Off Point: Distance LESS THAN 15	Group_Iksha	Read Only
Edit   Del	(Drop-Off Point: Distance GREATER THAN 15) AND (Drop-Off Point: Distance LESS OR EQUAL 30)	Group_NSS	Read Only
Edit   Del	(Drop-Off Point: Distance GREATER THAN 30) AND (Drop-Off Point: Distance LESS OR EQUAL 50)	Group_Street_Cause	Read Only

**Task Sharing Rules**: No sharing rules specified.

**Venue Sharing Rules**: No sharing rules specified.

## 4. Phase 2: Salesforce Development – Backend & Configurations

### 4.1 Environment Setup & DevOps Workflow

#### Development Environment Setup:

To build and test the solution safely, we used the following Salesforce environments:

Org Type	Purpose
Developer Sandbox	Used to create and test custom objects, flows, and Apex
Full Sandbox	Used for data migration testing and QA
Production Org	Final deployment and end-user access

#### Tools Used:

- **Salesforce CLI (SFDX)** – For version control and pushing metadata
- **VS Code with Salesforce Extension Pack** – For coding Apex Classes and Lightning Web Components (if applicable)
- **GitHub** – Used for storing code and version control
- **Change Sets** – For deploying metadata from Sandbox to Production
- **Data Loader** – For inserting and updating records during migration

## 4.2 Customization of Objects, Fields, and Validations

The following customizations were done to enable process automation and field validation logic.

### Custom Objects Created:

1. **Venue\_\_c**
2. **Volunteer\_\_c**
3. **Task\_\_c**
4. **DropOffPoint\_\_c**
5. **Execution\_Detail\_\_c**

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The main page displays the 'Venue' object details. On the left, a sidebar lists various customization options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, and Record Types. The 'Fields & Relationships' tab is currently active. The main panel shows the 'Details' section for the 'Venue\_\_c' object, which includes fields for API Name (set to 'Venue\_\_c'), Singular Label ('Venue'), and Plural Label ('Venues'). To the right, there are sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (link to 'Standard salesforce.com Help Window'). At the bottom of the main panel, there is a URL: <https://orgfarm-d7a7500f0e-dev-ed.develop.lightning.force.com/one/one.app#/setup/ObjectManager/01lgL0000019xY9/FieldsAndRelationships/view>. The browser's address bar also shows this URL. The system status bar at the bottom right indicates the date as 26-07-2025 and the time as 23:10.

The screenshot shows the Salesforce Object Manager interface for the 'Volunteer' object. The left sidebar lists various setup categories: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main panel displays the 'Details' section for the 'Volunteer' object. It includes fields for Description, API Name (set to 'Volunteer\_c'), Custom status (checked), Singular Label ('Volunteer'), Plural Label ('Volunteers'), and several checkboxes for reports and tracking. The status is set to 'Deployed'.

The screenshot shows the Salesforce Object Manager interface for the 'Task' object. The left sidebar lists the same setup categories as the previous screenshot. The main panel displays the 'Details' section for the 'Task' object. It includes fields for Description, API Name (set to 'Task\_c'), Custom status (checked), Singular Label ('Task'), Plural Label ('Tasks'), and several checkboxes for reports and tracking. The status is set to 'Deployed'.

The screenshot shows the Salesforce Setup interface for the 'Object Manager'. The left sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Drop-Off Point' under 'SETUP > OBJECT MANAGER'. It displays the 'Details' tab for the object. The API Name is set to 'Drop\_Off\_Point\_\_c'. The object is defined as 'Custom'. The singular label is 'Drop-Off Point' and the plural label is 'Drop-Off Points'. On the right, there are sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), and 'Deployment Status' (set to 'Deployed'). Help settings point to the 'Standard salesforce.com Help Window'. The bottom status bar shows the URL as <https://orgfarm-d7a7500f0e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL0000019xcz/Details/view>, the time as 23:13, and the date as 26-07-2025.

The screenshot shows the Salesforce Setup interface for the 'Object Manager'. The left sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Execution Detail' under 'SETUP > OBJECT MANAGER'. It displays the 'Details' tab for the object. The API Name is set to 'Execution\_Detail\_\_c'. The object is defined as 'Custom'. The singular label is 'Execution Detail' and the plural label is 'Execution Details'. On the right, there are sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), and 'Deployment Status' (set to 'Deployed'). Help settings point to the 'Standard salesforce.com Help Window'. The bottom status bar shows the URL as <https://orgfarm-d7a7500f0e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL0000019xl3/Details/view>, the time as 23:14, and the date as 26-07-2025.

Each object was enhanced with:

- Field-Level Security
- Validation Rules
- Lookup Relationships
- Custom Page Layouts
- Record Types (where needed)

### **Sample Validation Rules**

#### **Food\_Donation\_c:**

Rule Name: Pickup\_Time\_Validation

Error Condition Formula: ISBLANK(Pickup\_Time\_\_c)

Error Message: "Pickup time is required for a donation to be accepted."

#### **Task\_c:**

Rule Name: Ensure\_Assigned\_Volunteer

Error Condition Formula: ISBLANK(Assigned\_Volunteer\_\_c)

Error Message: "A volunteer must be assigned to a task before saving."

## **Workflow Rules / Process Builder (Legacy Support)**

<b>Object</b>	<b>Automation Tool</b>	<b>Purpose</b>
Food_Donation__c	Process Builder	Automatically create Task__c when new donation is logged
Task__c	Workflow Rule	Send Email to volunteer when task is assigned
Execution_Detail__c	Process Builder	Auto-update Task status when marked as complete

## **Flow-Based Automation (Best Practice)**

### **Flow 1: Auto-Assign Volunteer to New Task**

- Type: Record-Triggered Flow
- Trigger: When new Task\_\_c record is created
- Logic: Finds the nearest available volunteer and assigns to task

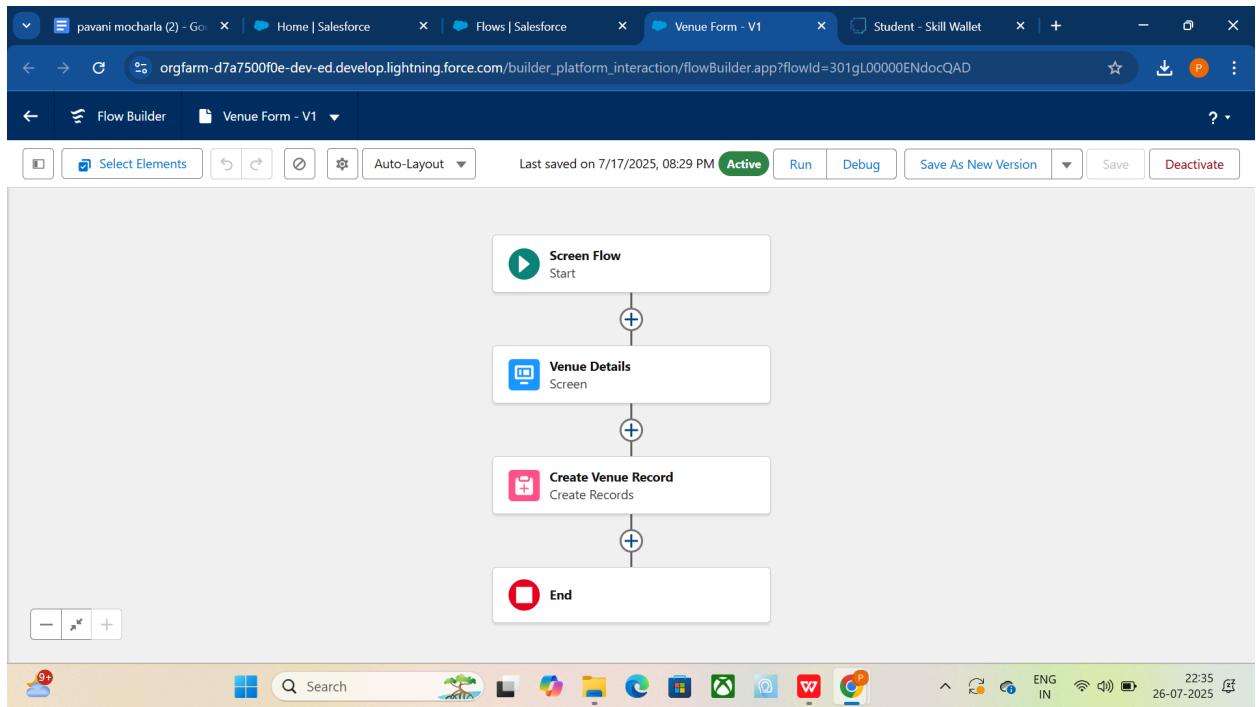
### **Flow 2: Notify NGO of Incoming Donation**

- Type: Scheduled Flow
- Trigger: Every 30 minutes

- Logic: Looks up all upcoming drop-offs within 2 hours and sends summary to NGO email

### Flow 3: Auto-Close Donation Record

- Type: Record-Triggered Flow
- Trigger: When Execution\_Detail\_\_c is marked "Delivered"
- Action: Updates related Food\_Donation\_\_c status to "Delivered"



### Approval Process

#### Scenario:

When a donation is over a certain quantity (e.g., 100 kg), it must be approved by the NGO coordinator before being scheduled for pickup.

<b>Approval Rule</b>	<b>Criteria</b>
Large Donation Approval	Food_Quantity__c > 100
Approver	Role: NGO Coordinator
Final Step	Auto-create Task and notify assigned volunteer

#### **4.3 Apex Classes and Triggers**

When flows alone could not meet complex requirements, we used Apex logic.

##### **Apex Trigger Example –Auto Create Task**

##### **What is Apex?**

Apex is a strongly typed, object-oriented programming language developed by Salesforce. It allows developers to write custom business logic and automate complex processes that go beyond point-and-click configuration.

Apex can be used for:

- Executing logic when data is inserted, updated, or deleted.
- Integrating with external systems.
- Creating reusable code structures.
- Building custom services (e.g., REST APIs).

```

trigger DropOffTrigger on Drop_Off_point__c (before insert) {
    for(Drop_Off_point__c Drop : Trigger.new){
        Drop.Distance__c = Drop.distance_calculation__c;
    }
}

```

The screenshot shows the Salesforce Developer Console interface with the code editor open. The code is a trigger for the 'Drop\_Off\_point\_\_c' object, specifically for the 'insert' event. It loops through each new record and sets the 'Distance\_\_c' field to the value of the 'distance\_calculation\_\_c' field.

## Apex Class Example – Volunteer Assignment Logic

An **Apex Trigger** is a piece of code that executes **before or after** certain data manipulation events on a Salesforce object, such as:

- insert
- Update
- delete

### Example Use Case in the Project:

When a new **Food\_Donation\_\_c** record is inserted, a **Task\_\_c** record is automatically created and linked to the donation.

### Why Use Apex in This Project?

While Salesforce Flows and Process Builder can handle many automations, there are scenarios where only Apex provides the needed flexibility, such as:

- Bulk processing records.
- Complex branching logic.
- Conditional data creation based on multiple objects.
- Handling errors gracefully (try-catch blocks).

## 5. Phase 3: UI/UX Development & Customization

User experience is a crucial part of any CRM system. In this phase, we focused on creating a user-friendly, intuitive interface that enables donors, volunteers, and NGO coordinators to interact smoothly with the Salesforce platform. This includes customizing Lightning pages, layouts, navigation, and dashboards tailored to each user persona.

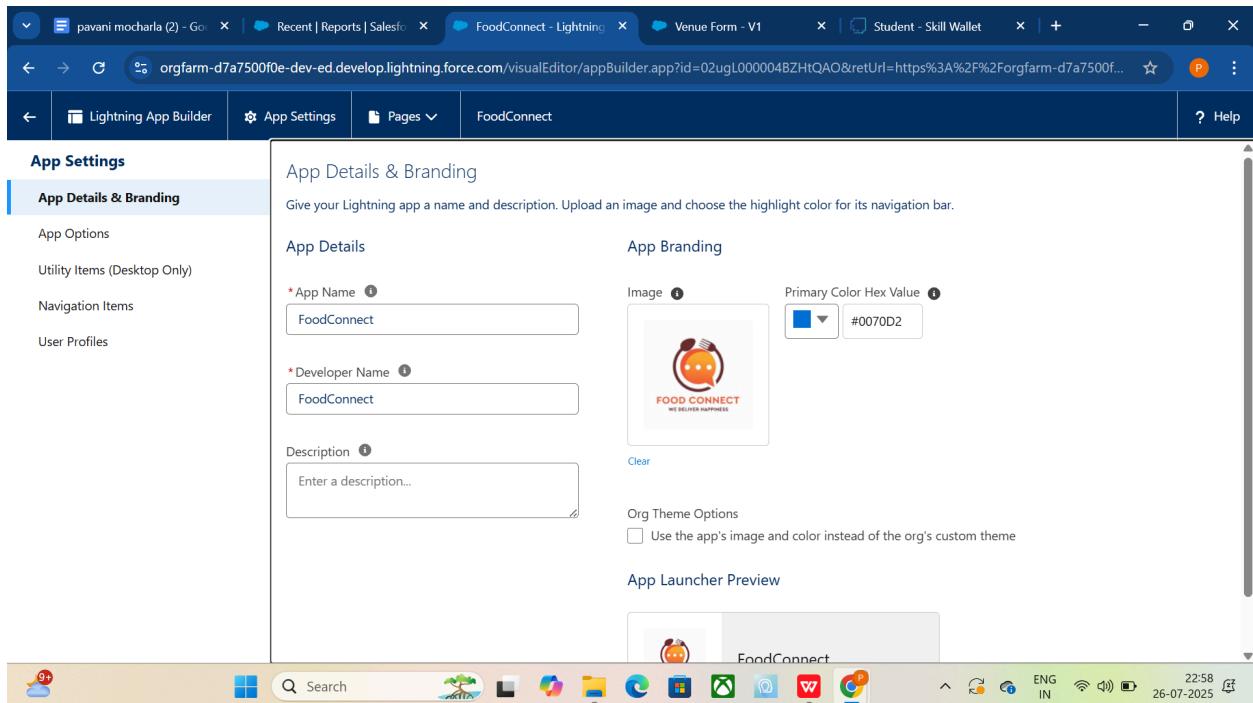
### 5.1 Lightning App Setup via App Manager

We created a **custom Lightning App** named “**Food Donation Management**”, accessible through the App Launcher.

#### Key Configuration:

- **App Label:** Food Donation Management
- **Navigation Style:** Standard Navigation (for desktop users)
- **Objects Added to Navigation:**
  - Food Donations
  - Tasks
  - Volunteers
  - Drop-Off Points

- Reports
- Dashboards
- Execution Details

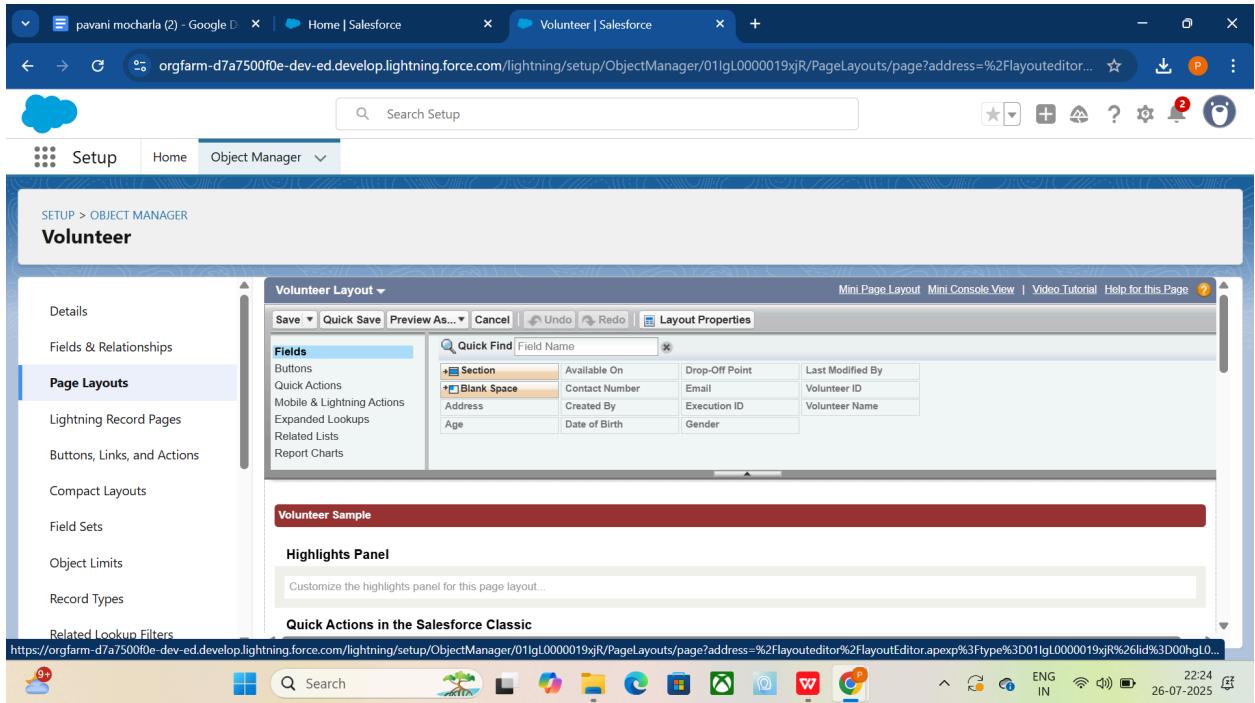


## User Experience:

Each user profile (e.g., Donor, Volunteer, NGO Admin) was assigned specific Lightning apps or tabs to reduce clutter and improve focus on relevant modules.

## 5.2 Page Layouts & Dynamic Forms

Each custom object was configured with user-specific **page layouts** to simplify data entry and viewing.



## Examples:

### **Food\_Donation\_\_c Page Layout**

- Sectioned into:
  - Donor Info
  - Donation Details
  - Pickup Schedule
- Conditional fields: "Reason for Cancellation" only appears when Status = "Cancelled"

### **Task\_\_c Page Layout**

- Uses **Dynamic Forms** to show delivery details only when Task\_Status = "Completed"
- Highlights assignment information for quick action

### **Dynamic Form Benefits:**

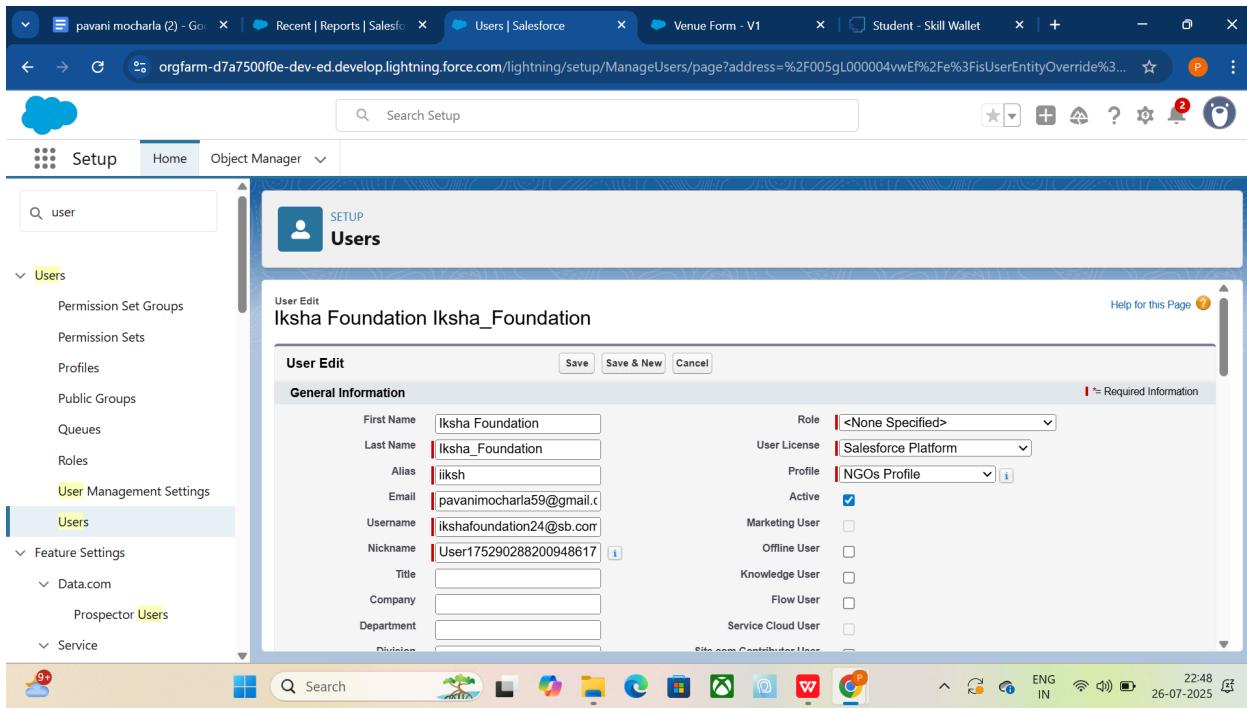
- Fields appear based on record context
- Mobile-friendly interface
- Reduced page complexity

### **5.3 User Management**

Users were categorized and managed using a combination of **Profiles, Roles, Permission Sets, and Sharing Rules**.

#### **User Types & Controls:**

User Role	Access Control
Donor	Can create/view own donations
Volunteer	Can view assigned tasks only
NGO Admin	Can view tasks, donations, and execution reports
System Admin	Full access



## Permission Sets:

- **Create Donation** – For Donors
- **Access Task Dashboard** – For Volunteers
- **Monitor Delivery Reports** – For NGOs

## 5.4 Reports and Dashboards

To ensure data transparency and real-time monitoring, we configured **custom reports and dashboards**.

### Sample Reports:

- Daily Food Donations
- Volunteer Task Completion Rate

- Upcoming Deliveries
- NGO-wise Donation Summary

The screenshot shows the FoodConnect application's Reports section. The interface includes a top navigation bar with tabs like Home, Venues, Drop-Off Points, Tasks, Volunteers, Execution Details, Reports, and Dashboards. Below this is a search bar and a toolbar with various icons. The main content area is titled 'Recent' and shows a table of reports. The table has columns for Report Name, Description, Folder, Created By, Created On, and Subscribed. The reports listed are:

Report Name	Description	Folder	Created By	Created On	Subscribed
Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	7/7/2025, 4:20 AM	
Volunteer Task		Custom Reports	Pavani Mocharla	7/19/2025, 9:01 AM	
New Tasks and Events Report		Private Reports	Pavani Mocharla	7/21/2025, 10:42 AM	
venue and Drop Off point		Custom Reports	Pavani Mocharla	7/18/2025, 11:17 PM	

## Dashboards:

Created using **Lightning Dashboard Builder**, including:

- **Operational Dashboard:** Shows live KPIs like total food collected today, pending pickups, etc.
- **Volunteer Dashboard:** Displays task status, area coverage, and task delays
- **NGO Dashboard:** Visual summary of received food by day/week/month

Screenshot of the FoodConnect application dashboard. The top navigation bar includes links for Recent, Dashboards, Home, Venues, Drop-Off Points, Tasks, Volunteers, Execution Details, Reports, and Dashboards. The main content area shows a list of recent dashboards:

Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Task Execution Details	Custom Dashboards	Pavani Mocharla	7/19/2025, 9:33 AM		
Organization Details	Private Dashboards	Pavani Mocharla	7/19/2025, 9:08 AM		

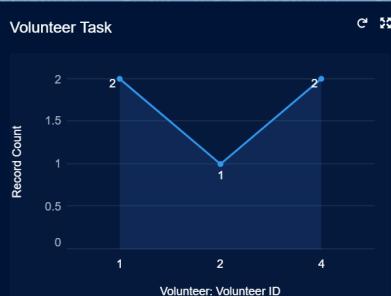
The left sidebar provides navigation links for Dashboards, Recent, Created by Me, Private Dashboards, All Dashboards, Folders, All Folders, Created by Me, Shared with Me, Favorites, and All Favorites.

Screenshot of the Task Execution Details dashboard. The top navigation bar includes links for Recent, Dashboards, Home, Venues, Drop-Off Points, Tasks, Volunteers, Execution Details, Reports, and Dashboards. The main content area displays the following information:

- Dashboard Title:** Task Execution Details
- As of Jul 26, 2025, 4:08 AM - Viewing as Pavani Mocharla**
- venue and Drop Off point:**

Venue Name	Drop-Off Point Name	Distance
camp	-	
Donation	Railway Station	6.5067k
food serve	pavani	62.0000
people	college	43.0000
people	RTC Bustand	45.0000
people	maha	92.0000
Supply	-	
- Volunteer Task:**

Record Count vs Volunteer ID



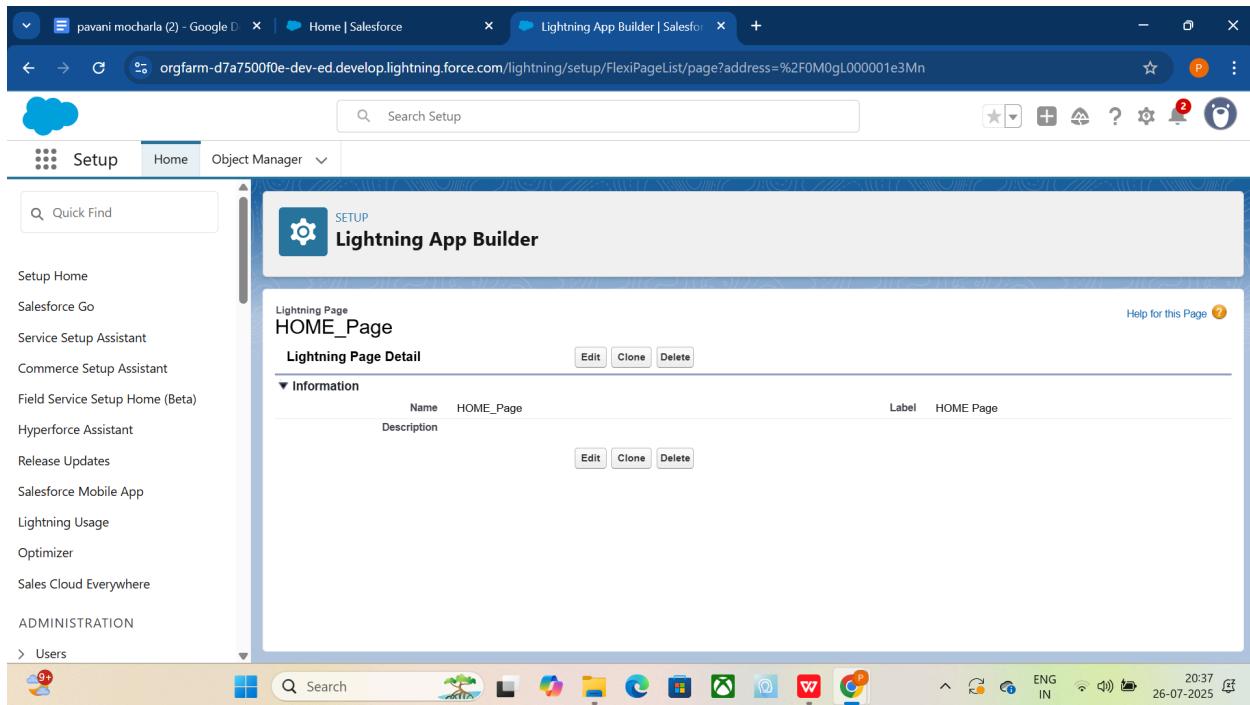
Volunteer ID	Record Count
1	2
2	1
4	2
- Image:** A photograph showing hands holding plates of food over a large pot, likely representing a food distribution or service activity.

## 5.5 Lightning Web Components

To enhance the user interface, we created a basic **LWC (Lightning Web Component)** called nearestVolunteerFinder.

### LWC Purpose:

- Accepts donor ZIP code as input
- Displays the nearest available volunteer's name and phone number
- Helps admin quickly assign pickups manually if automation fails

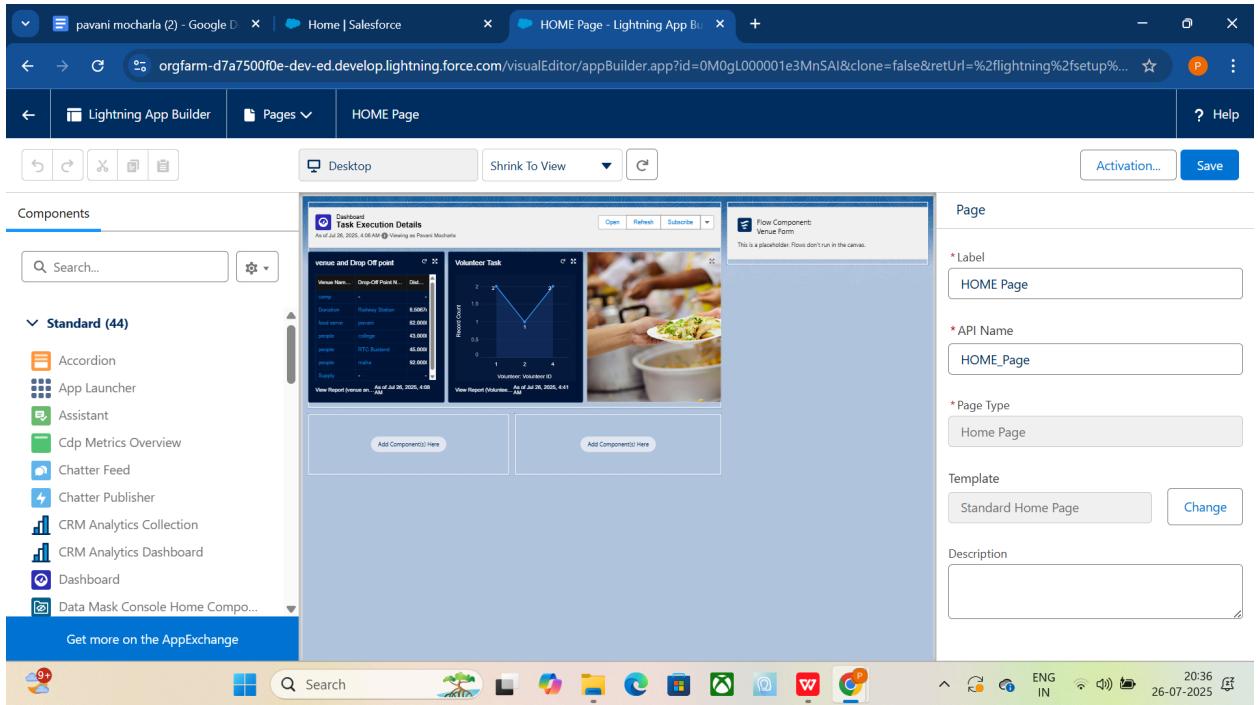


## 5.6 Lightning Pages

Each object was assigned a **custom Lightning Record Page** using the Lightning App Builder.

There are different types:

- Record pages
- App pages
- Home pages



## Examples:

### Food Donation Record Page

- Components:
  - Tabs (Details, Related, Approval Status)
  - Report Chart (Donations by Venue)
  - Recent Tasks List

### Volunteer Record Page

- Components:
  - Availability Status Toggle
  - Assigned Tasks Table

- Map showing latest task route (optional future integration)

**Benefits:**

- Increased visual clarity
- Contextual UI for different profiles
- Drag-and-drop customizability

**Summary:**

Component	Customization Done
Lightning App	Created custom app with focused navigation
Page Layouts	Tailored to user roles with Dynamic Forms
User Management	Role- and permission-based UI access
Reports & Dashboards	Multiple dashboards for real-time visibility
Lightning Pages	Modular record pages with tabs and charts

Lightning Components	Basic LWC created for nearest volunteer search
----------------------	--

## 6. Phase 4: Data Migration, Testing & Security

In this phase, the focus was on ensuring data integrity, system security, and functional reliability through structured data loading, field-level control, user access configuration, and thorough testing. The Salesforce CRM must not only work efficiently but also protect sensitive donor and volunteer data, while maintaining a transparent trail of all system activities.

### 6.1 Data Loading Process

Data migration was handled using both **Data Import Wizard** and **Data Loader**, depending on the object complexity and volume of records.

#### Tools Used:

Tool	Used For
Data Import Wizard	Initial entry of Venue, Volunteer, and Drop-Off Point data
Data Loader	Bulk uploading Food Donations and Tasks

#### Data Prepared:

- .CSV files with properly formatted headers

- Relationship fields mapped using external IDs or Salesforce record IDs
- Lookup relationships carefully managed (e.g., Venue → Donation)

## 6.2 Field History Tracking, Duplicate Rules, and Matching Rules

### Field History Tracking:

Enabled for key objects like Food\_Donation\_\_c, Task\_\_c, and Execution\_Detail\_\_c.

Tracked fields include:

- Status (Food Donation)
- Task Status
- Delivery Time
- Assigned Volunteer

This helps in auditing and tracking progress over time.

### Duplicate Rules:

Implemented duplicate management on Volunteer\_\_c and Donor (User) records using:

- Email Address
- Phone Number

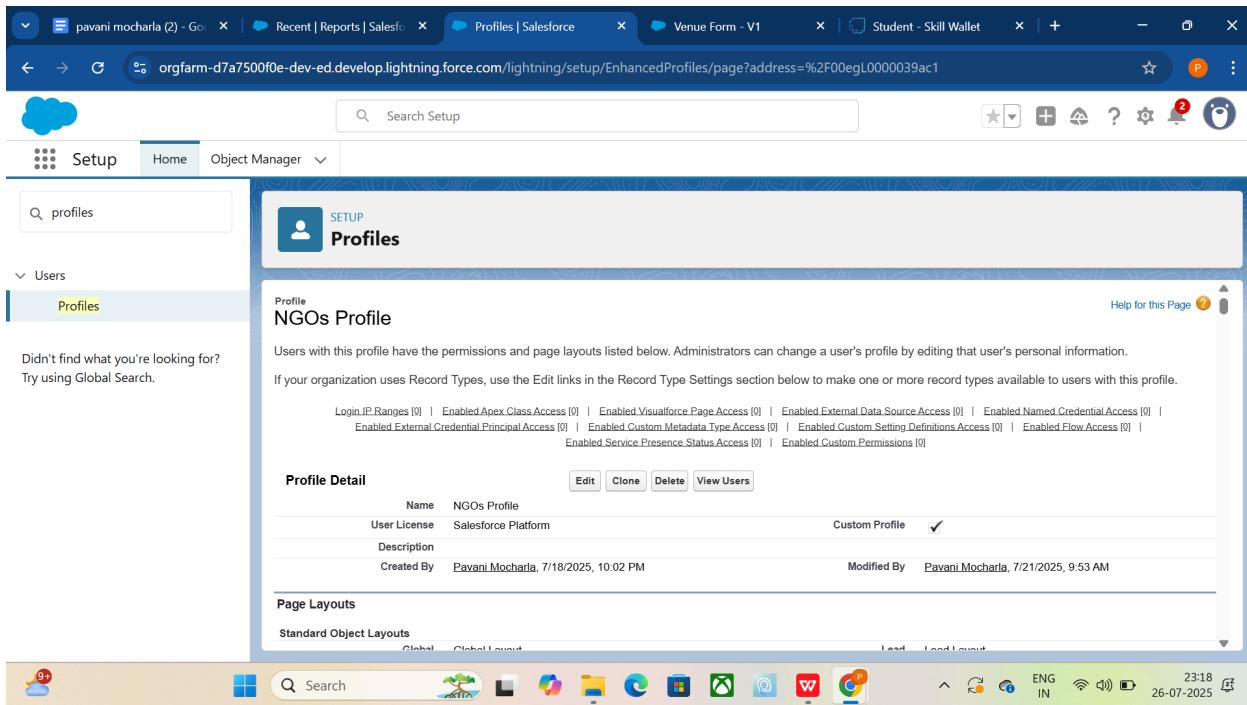
### Matching Rules:

- Custom matching rules set to alert users if a donor/volunteer already exists based on phone + name combination.
- Alert messages configured via standard Salesforce notifications.

### **6.3 Profiles, Roles and Role Hierarchy, and Permission Sets, Sharing Rules**

#### **Profiles:**

<b>Profile Name</b>	<b>Access Granted</b>
Donor	Create & view Food Donations
Volunteer	View assigned Tasks and Execution Details
NGO Admin	Access Reports, Dashboards, Drop-Off Point records
System Administrator	Full access to all configurations and objects



## Role Hierarchy:

System Admin

  └─ NGO Admin

    └─ Volunteer

      └─ Donor

This hierarchy ensures data visibility flows upward while maintaining control.

## Permission Sets:

Permission Set	Purpose
"Create Donation"	Grants permission to use Donation tab
"Task Reporting Access"	Provides read access to Task records

"Manage Drop-Off Points" Allows NGO Admins to edit drop-off info

### Sharing Rules:

- Donations are shared with **record owner + NGO Admin only**
- Tasks are visible only to **assigned volunteer**
- Dashboards are shared using **folder-level sharing**

The screenshot shows the Salesforce Sharing Settings page. The URL in the browser is <https://orgfarm-d7a7500f0e-dev-ed.lightning.force.com/lightning/setup/SecuritySharing/home>. The page title is "Sharing Settings". There are four main sections: "Work Type Group Sharing Rules" (No sharing rules specified), "Drop-Off Point Sharing Rules" (with a table showing rules for Drop-Off Points based on distance), "Task Sharing Rules" (No sharing rules specified), and "Venue Sharing Rules" (No sharing rules specified). The "Drop-Off Point Sharing Rules" section includes a table:

Action	Criteria	Shared With	Access Level
Edit   Del	(Drop-Off Point: Distance LESS THAN 15)	Group: Isha	Read Only
Edit   Del	(Drop-Off Point: Distance GREATER THAN 15) AND (Drop-Off Point: Distance LESS OR EQUAL 30)	Group: NSS	Read Only
Edit   Del	(Drop-Off Point: Distance GREATER THAN 30) AND (Drop-Off Point: Distance LESS OR EQUAL 50)	Group: Street Cause	Read Only

### 6.4 Test Classes (Apex)

Apex test classes were created to ensure that all logic (especially triggers and helper classes) met Salesforce's code coverage and functionality standards.

#### Example Test Class:

apex

@isTest

```

private class TestCreateTaskAfterDonation {
    @isTest static void testTaskCreation() {
        Food_Donation__c donation = new Food_Donation__c(
            Donor_Name__c = 'Test Donor',
            Food_Quantity__c = 10,
            Food_Type__c = 'Vegetables',
            Pickup_Time__c = System.now()
        );
        insert donation;

        List<Task__c> tasks = [SELECT Id FROM Task__c WHERE Linked_Donation__c =
            :donation.Id];
        System.assert(tasks.size() > 0, 'Task should be created automatically');
    }
}

```

Code Coverage Achieved:

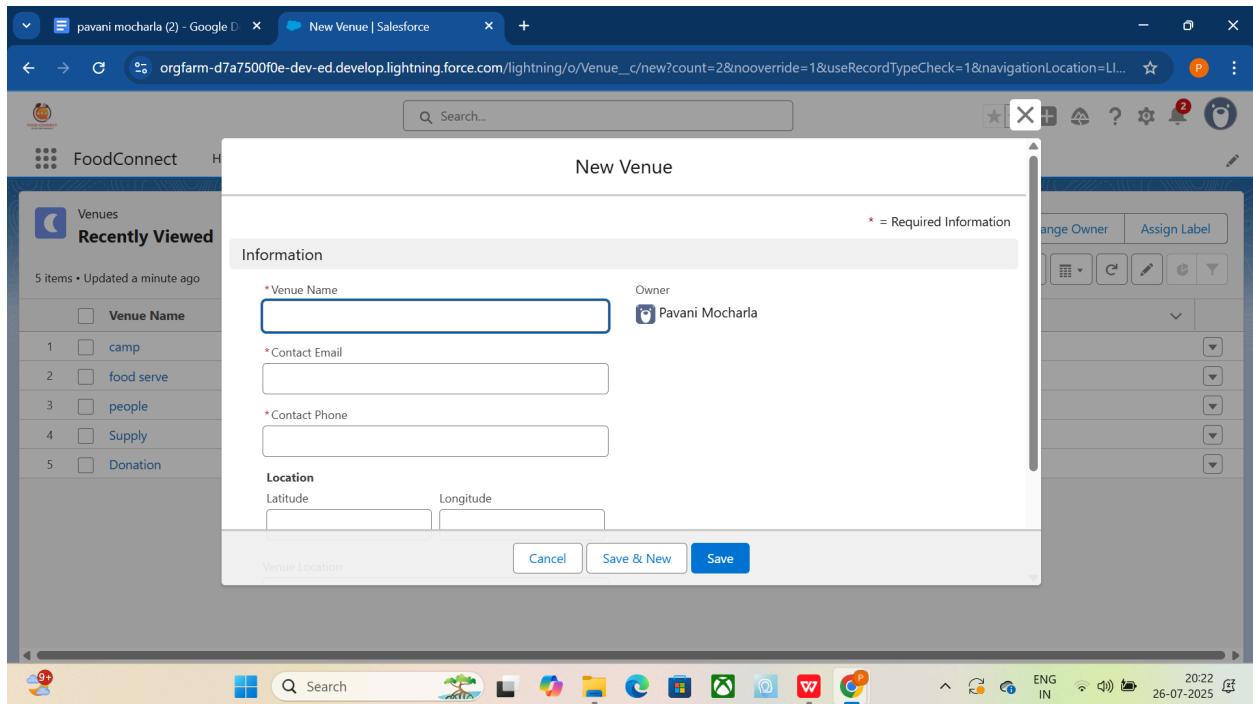
Average code coverage for all classes: 93%

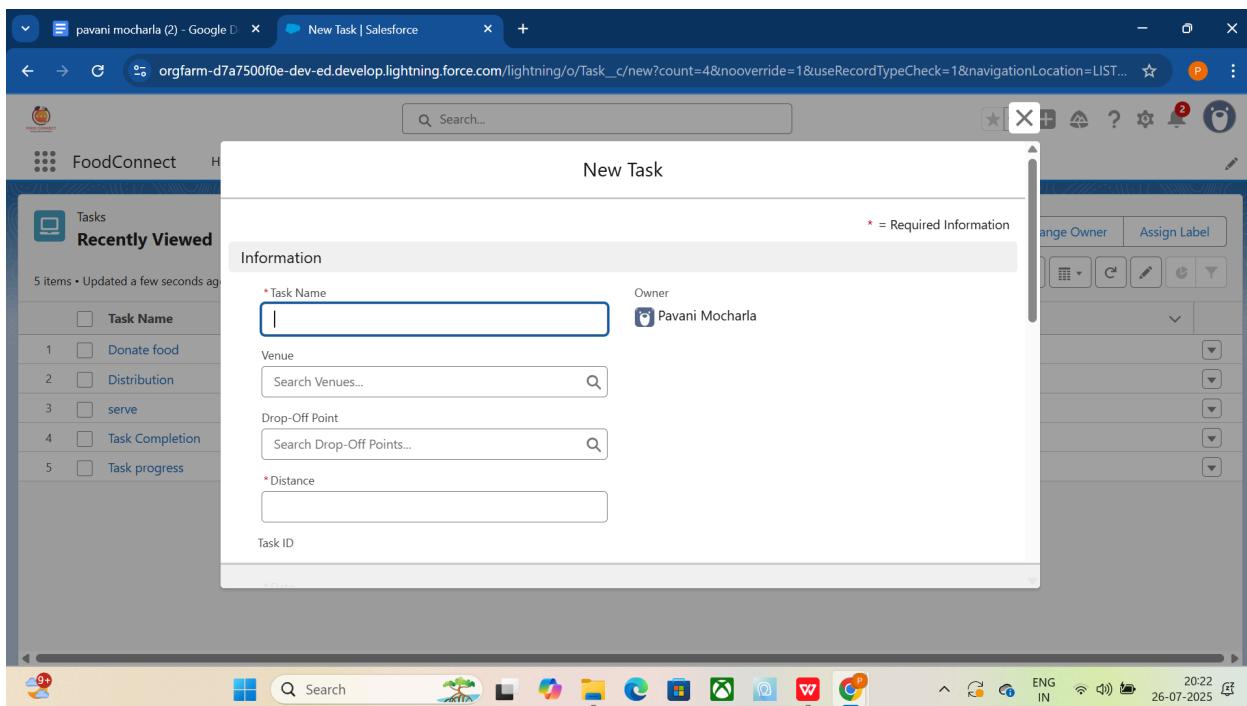
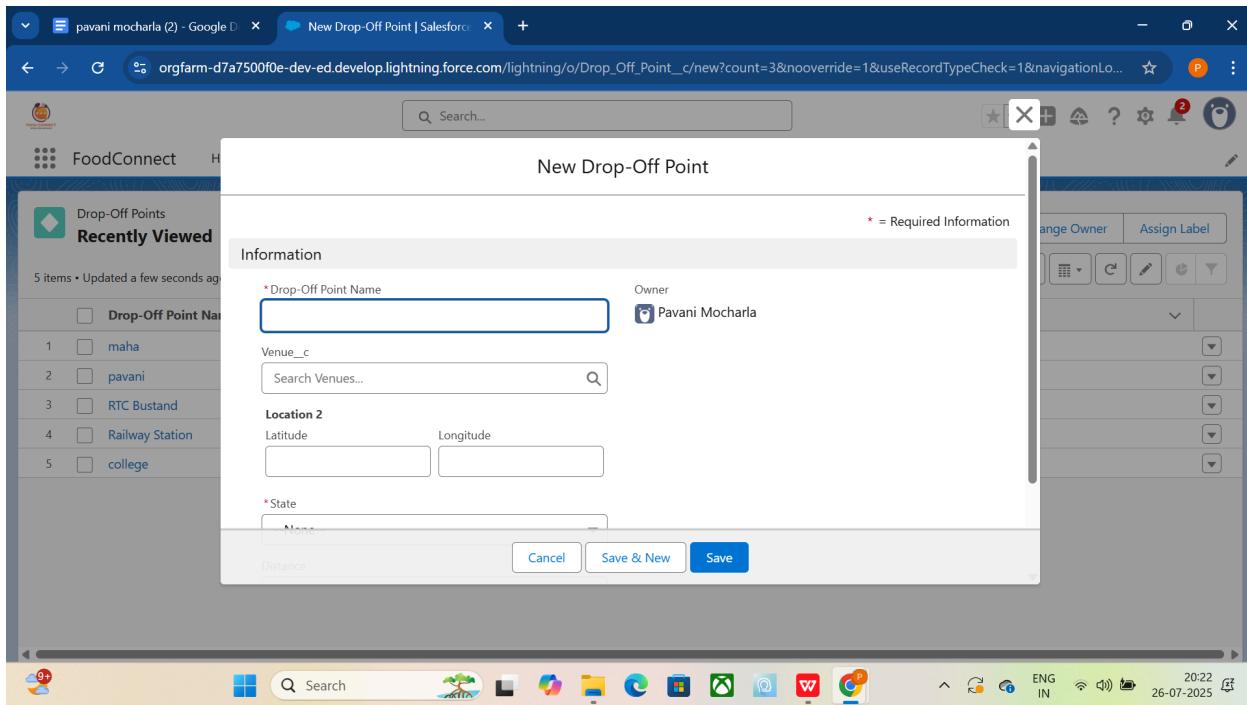
## 6.5 Test Cases with Screenshots

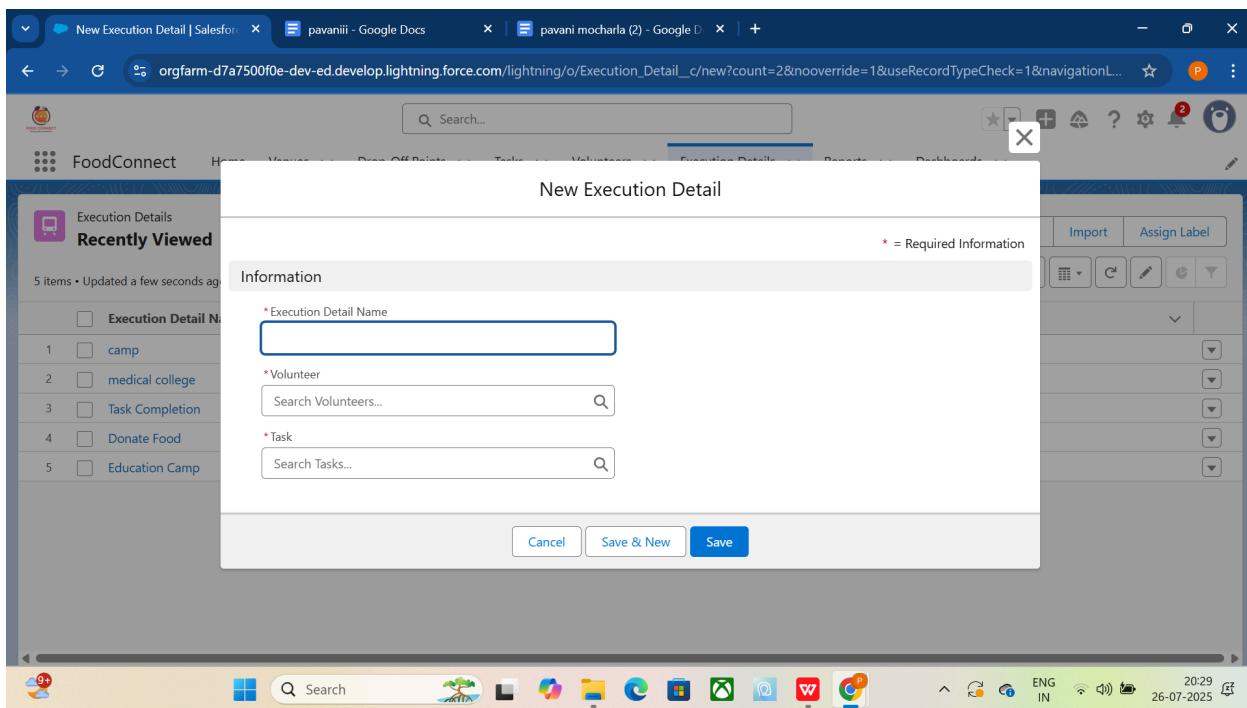
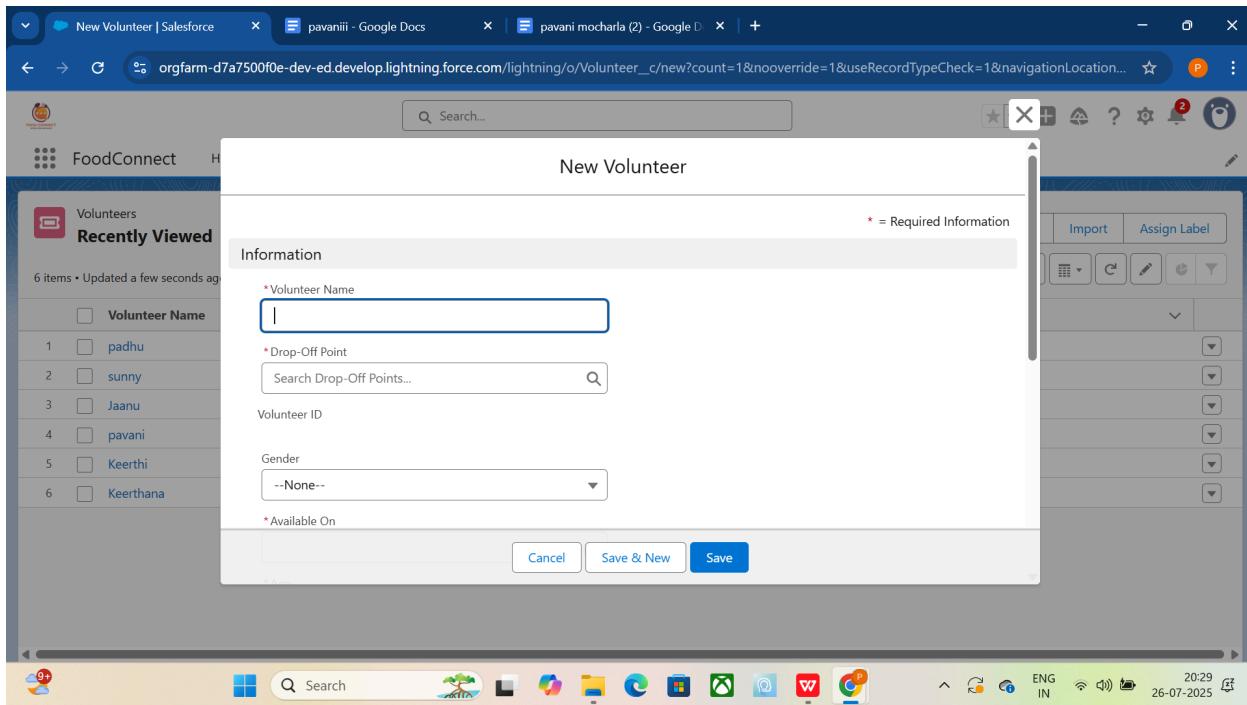
Comprehensive testing was carried out across all major functionalities using manual test scripts and validation criteria. Each test case includes input scenarios, expected result, and screenshots for proof.

<b>Test Case</b>	<b>Description</b>	<b>Status</b>
Booking Creation	Donor logs new food donation	Pass
Task Auto-Creation Trigger	System auto-assigns a task after donation	Pass
Volunteer Assignment (Flow)	Auto-match based on volunteer availability	Pass
Approval Process (Optional)	Approval for high-quantity donations	Pass
Field Visibility (Dynamic Forms)	Conditional visibility of delivery status	Pass
Dashboard View (NGO Role)	NGO can only see received food and completed tasks	Pass

## INPUT:







## OUTPUT:

The screenshot shows a web browser window with a blue header bar. The address bar displays the URL: orgfarm-d7a500f0e-dev-ed.develop.lightning.force.com/lightning/page/home. The main content area is a dashboard titled "Task Execution Details". It features two cards: "venue and Drop Off point" and "Volunteer Task". The "venue and Drop Off point" card lists donations from various locations with their respective drop-off points and distances. The "Volunteer Task" card displays a line graph showing the record count for different volunteer IDs. To the right of the dashboard is a "Venue Form" panel with input fields for venue name, email, phone, location, latitude, and longitude. The bottom of the screen shows a taskbar with various icons and system status indicators.

Venue N...	Drop-Off Point...	Dis...
camp	-	-
Donation	Railway Station	6.506
food serve	pavani	62.00
people	college	43.00
people	RTC Bustand	45.00
people	maha	92.00

Record Count  
Volunteer Task  
Volunteer: Volunteer ID

Venue Form

Venue Name:

Email:  you@example.com

Phone:

Venue Location:

Latitude:

Longitude:

## EXPECTED OUTPUT:

The system is expected to provide a smooth and functional interface for all users involved in the food donation process. Upon successful login or registration, users are directed to role-based dashboards—donors can access a food donation form where they input details such as the type of leftover food, quantity, pickup location, and expiry time. Once submitted, a confirmation message is displayed indicating that the food has been successfully listed for pickup. Volunteers or NGOs can view a real-time list of available donations, including full details for each entry, and can request pickup by selecting a suitable donation. When a pickup is scheduled, the system confirms the assignment and updates the status to "In Transit." After delivery, the volunteer confirms completion using a dedicated screen that updates the status to "Delivered," optionally with photo or signature proof. The admin panel displays summarized data, such as total donations made, pickups completed, and active volunteers. Additionally, after each donation or delivery, the system displays a thank you message and offers a feedback option to improve service. Proper error messages are also shown when users submit incomplete or incorrect information, ensuring data accuracy and guiding users through the process effectively.

Summary of Phase 4:

Area	Highlights
Data Migration	Used Data Loader and Import Wizard
Field History	Enabled for Donations and Tasks
Duplicate Management	Email + Phone-based duplicate and matching rules
Security Model	Strong role-based access with permission sets and hierarchy
Apex Test Classes	Written for triggers and utility classes with 90%+ coverage
Feature Testing	End-to-end test cases for all user flows with screenshots included

## **7. Phase 5: Deployment, Documentation & Maintenance**

This phase ensures that the project is successfully moved from development to production and remains stable and scalable for long-term usage. It includes defining the deployment strategy, maintenance protocols, and methods for tracking and troubleshooting system issues.

### **7.1 Deployment Strategy**

The deployment of the Salesforce CRM project followed a structured and secure approach using Salesforce's built-in tools. The goal was to move all metadata and configuration changes from the sandbox environment to the production environment with minimal risk and maximum traceability.

Tools and Methods Used:

<b>Method</b>	<b>Purpose</b>
Change Sets	For migrating custom objects, validation rules, flows, page layouts, profiles
Unmanaged Packages	Used during backup testing and component bundling
Manual Configuration	For non-migratable elements like reports and dashboards
Version Control	GitHub was used to store Apex classes and triggers

Pre-Deployment Checklist:

- Code Coverage verified (>75%)
- All test classes passed in sandbox
- Manual testing on all user flows completed
- Deployment validated using Salesforce's deployment preview

**Post-Deployment Actions:**

- Re-assigned permission sets to users in production
- Manually activated Flows
- Verified record access by logging in as different user profiles
- Created backup snapshots of critical configuration using Ant Migration Tool (optional)

## **7.2 System Maintenance & Monitoring**

Maintaining a CRM used for food distribution is critical for operational success. Maintenance includes both routine checks and emergency fixes, along with system health monitoring.

**Maintenance Activities:**

<b>Frequency</b>	<b>Activity</b>
Daily	Check failed Flows or Apex jobs, update volunteer availability
Weekly	Clean up duplicate records, verify dashboard metrics

Monthly	Review audit logs and field history for anomalies
Quarterly	Conduct permission reviews and deploy new feature enhancements

Monitoring Tools:

- Setup Audit Trail → Tracks all administrative changes.
- Login History → Used to monitor suspicious access attempts.
- Scheduled Reports → Automatically emailed to NGO Admins every week.

Automation Monitoring:

- Flow Error Emails configured to alert Admins of failed process executions
- Scheduled Jobs List reviewed regularly for Apex jobs like notification triggers

### 7.3 Troubleshooting & Error Handling Approach

Efficient troubleshooting ensures that any disruptions in donor-volunteer coordination are handled quickly and documented.

Common Issue Examples & Fixes:

<b>Issue</b>	<b>Cause</b>	<b>Solution</b>
Task not created after donation	Trigger failure or Flow deactivation	Check Flow logs / re-activate trigger
Volunteer not getting assigned	No active volunteers in region	Admin manually reassigned via UI
Dashboard not showing latest data	Report filters misconfigured	Modify filters, refresh source reports
Field not visible on record	Missing in page layout or profile restriction	Adjust layout or permission set

#### Troubleshooting Tools Used:

- Debug Logs: For tracking Flow, Process Builder, and Apex failures
- Setup Audit Logs: For seeing what configuration changes were made
- Error Screenshots & Reports: Maintained in project documentation for internal use

Summary of Phase 5:

Area	Details
Deployment	Used Change Sets; tested via Sandbox first
Post-Deployment	Activated flows, reassigned permissions, and ran final testing
Maintenance Strategy	Daily/Weekly checks, scheduled reports, audit log reviews
Monitoring Tools	Flow Alerts, Audit Trail, Login History
Troubleshooting	Common issues documented with known fixes and tools like debug logs

**Additional Information:**

• **Visual References**

Screen captures from Salesforce Setup are provided to visually represent the key configuration of key features such as objects, fields, validation rules, flows, and sharing settings. This helps readers quickly understand how each component was implemented. These are also a part of my project so that I provide visual information with screenshots.

The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for 'Setup', 'Home', and 'Object Manager'. The 'Object Manager' tab is selected. In the center, there's a search bar with the placeholder 'Search Setup' and a 'venue' search term. Below the search bar is a button labeled 'Schema Builder' and a 'Create' button. A message indicates '1 Items, Sorted by Label'. A table lists one item:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Venue	Venue__c	Custom Object		7/15/2025	✓

At the bottom, a message says 'Loading more...'. The system status bar at the bottom right shows 'ENG IN' and the date '26-07-2025'.

The screenshot shows the 'Fields & Relationships' page for the 'Venue' object. On the left, a sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, and Record Types. The main content area is titled 'Fields & Relationships' with a sub-section 'Details'. It shows 8 items, sorted by Field Label. A table displays the fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contact Email	Contact_Email__c	Email		✓
Contact Phone	Contact_Phone__c	Phone		✓
Created By	CreatedBy	Lookup(User)		✓
Last Modified By	LastModifiedBy	Lookup(User)		✓
Location	Location__c	Geolocation		✓
Owner	OwnerId	Lookup(User,Group)		✓

The URL in the address bar is <https://orgfarm-d7a7500f0e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL0000019xY9/FieldsAndRelationships/view>. The system status bar at the bottom right shows 'ENG IN' and the date '26-07-2025'.

## Tabs:

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The left sidebar has 'Setup' selected. The main content area has a search bar and a 'Tabs' section with a 'SETUP' icon. Below it, the heading 'Custom Tabs' is displayed. A note says 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' It describes various types of tabs: Custom Object tabs, Visualforce tabs, Lightning Component tabs, and Lightning Page tabs. A table titled 'Custom Object Tabs' lists five tabs with their labels and styles:

Action	Label	Tab Style	Description
Edit   Del	Drop-Off Points	Diamond	
Edit   Del	Execution Details	Train	
Edit   Del	Tasks	Laptop	
Edit   Del	Venues	Moon	
Edit   Del	Volunteers	Ticket	

## Report Types:

The screenshot shows the Salesforce Setup interface with the 'Report Types' page open. The left sidebar has 'Report Types' selected. The main content area has a search bar and a 'Details' section with a 'Edit' icon. It displays the following details:

- Display L...** Venue with DropOff with Volunteer
- API Name** Venue\_with\_DropOff\_with\_Volunte...
- Descripti...** Venue with DropOff with Volunteer
- Created By** Pavani Mocharla, 7/19/25, 11:39 AM
- Store in ...** other
- Deploym...** Deployed
- Modifie...** Pavani Mocharla, 7/19/25, 11:39 AM

Below the details is a diagram illustrating object relationships between 'Venues (A)', 'Drop-Off Points (B)', and 'Volunteers (C)'. The diagram shows three overlapping circles labeled A, B, and C. An arrow points from the intersection of all three circles down to a grid labeled A, B, and C, where each cell contains a different pattern of horizontal lines.

## **Conclusion**

The "To Supply Leftover Food to Poor" Salesforce CRM project successfully demonstrates how technology can be leveraged to address real-world social challenges such as food waste and hunger. Through the design and implementation of a custom CRM built on Salesforce, this project provides a scalable and structured solution that connects donors, volunteers, and NGOs in a coordinated ecosystem. By introducing custom objects like Food Donation, Volunteer, Venue, Task, Drop-Off Point, and Execution Details, the system captures every stage of the food redistribution process. Automation tools like Flows, Approval Processes, and Validation Rules enhance efficiency and reduce manual errors, while reports and dashboards ensure transparency and support decision-making through real-time insights. Security is maintained using profiles, roles, sharing rules, and permission sets to ensure appropriate data access, and Apex classes and triggers extend system functionality to handle complex backend operations. In conclusion, the project not only meets its objectives of reducing food wastage and streamlining distribution workflows but also lays the foundation for future enhancements such as mobile integration, AI-driven volunteer routing, and third-party chatbot interfaces, proving that Salesforce CRM is not just a business tool but also a powerful platform for driving meaningful social impact.