To Supply Leftover Food to Poor

(A Salesforce-Based Food Distribution Management System)

Introduction:

Food wastage is a major issue in urban areas where large quantities of edible food are discarded daily from events, restaurants, and institutions. Simultaneously, many underprivileged people struggle for one meal a day.

This project, "To Supply Leftover Food to Poor," aims to create a Salesforce-based system that streamlines collection, tracking, and distribution of surplus food to needy individuals, ensuring transparency, efficiency, and accountability.

The project uses Salesforce's cloud platform to manage venues, volunteers, drop-off points, and task execution details efficiently.

Requirements Phase:

1.Functional Requirements

- Manage venues where leftover food is available.
- · Track drop-off points for food delivery.
- Assign volunteers for pickup and delivery tasks.
- Record execution details for each delivery.
- Generate reports and dashboards for monitoring.

2. Non-Functional Requirements

- Accessible via Salesforce platform (web-based).
- Easy to use with minimal training.
- Scalable for multiple NGOs and organizations.
- Secure data access using sharing rules.

3. Software Requirements

- Salesforce Developer Edition Account
- Web Browser (Chrome/Edge)
- Stable Internet Connection

Design Phase:

1.System Architecture

The architecture is based on Salesforce CRM's multi-object model integrating custom objects and relationships:

- Venue
- · Drop-Off Point
- Task
- Volunteer
- Execution Details

Each object interacts through lookup and master-detail relationships to form a complete operational workflow.

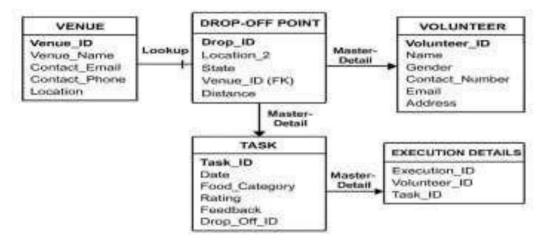
2. ER Diagram (Conceptual Design)

Entities:

- Venue (Venue ID, Venue Name, Contact Email, Contact Phone, Location)
- Drop-Off Point (Drop ID, Location 2, State, Venue ID, Distance)
- Volunteer (Volunteer ID, Name, Gender, Contact Number, Email, Address)
- Task (Task ID, Date, Food Category, Rating, Feedback, Drop Off ID, Venue ID)
- Execution Details (Execution ID, Volunteer ID, Task ID)

Relationships:

- Venue → Drop-Off Point (Lookup)
- Drop-Off Point → Volunteer (Master-Detail)
- Volunteer → Execution Details (Master-Detail)
- Task → Execution Details (Master-Detail)



Development Phase:

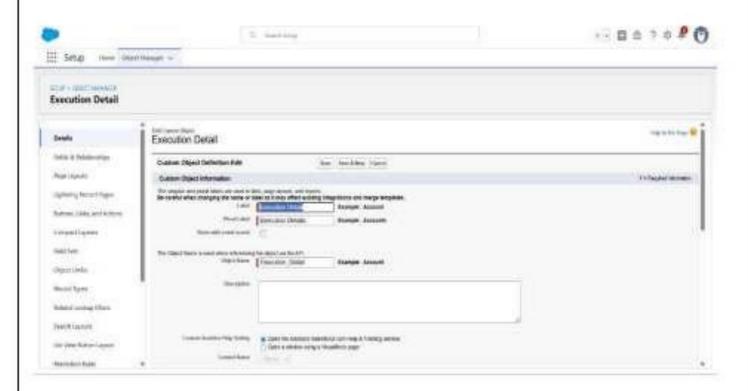
Salesforce Developer Account Creation

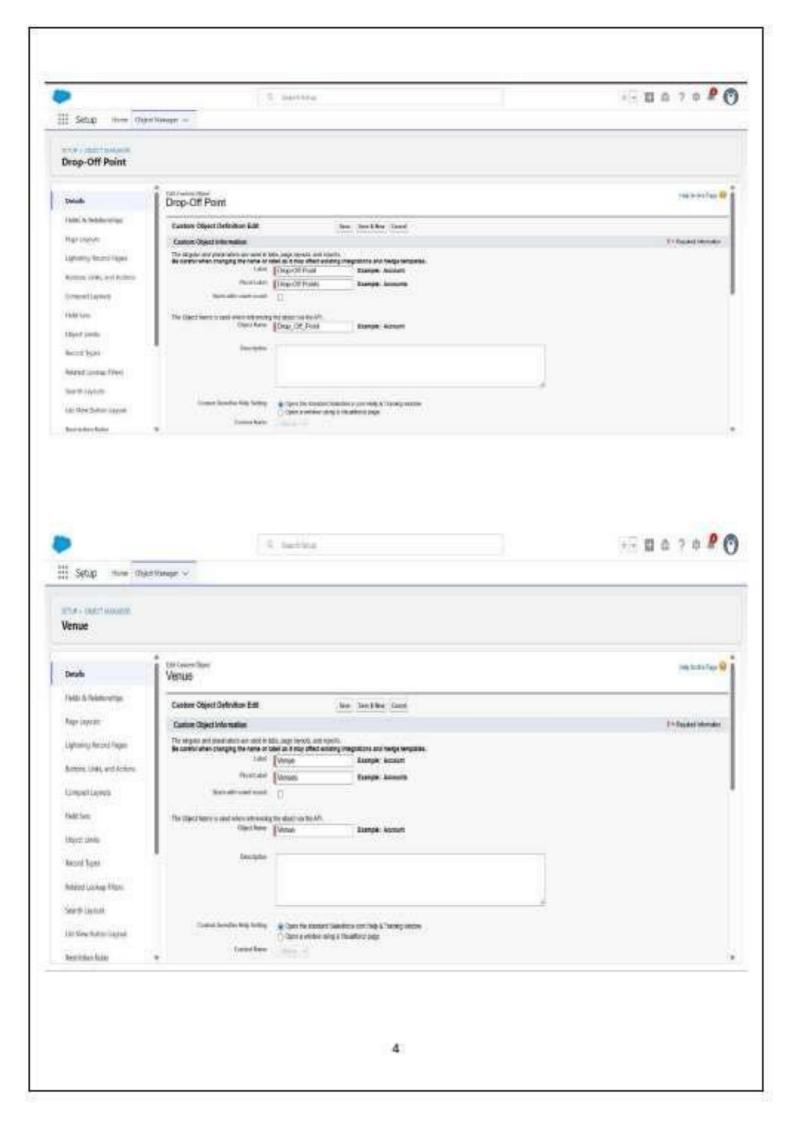
- Go to https://developer.salesforce.com/signup.
- Fill in your details: name, email, role (Developer), company name, country (India), and postal code.
- Username format: username@organization.com
- Verify account through email and set a password.

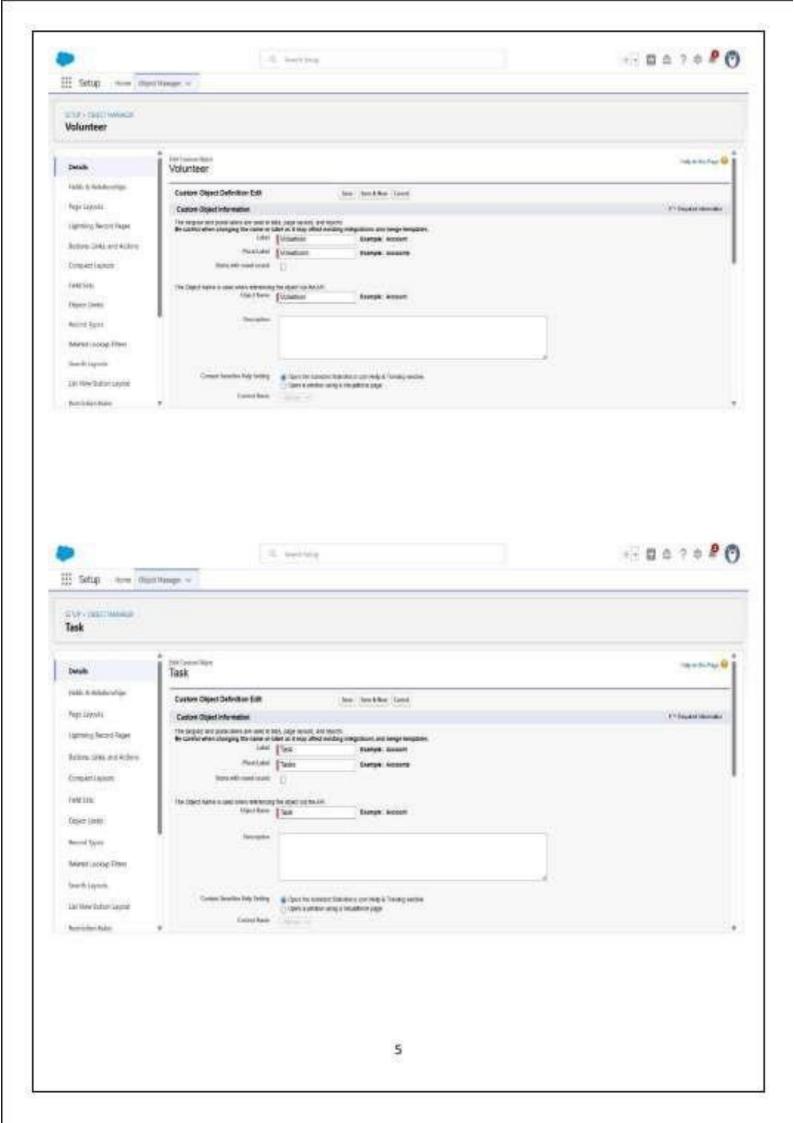
Object Creation

- Venue to manage food source details
- Drop-Off Point to track distribution points
- Task to record deliveries and feedback
- · Volunteer to manage individuals assisting in operations
- Execution Details to log volunteer-task execution records

Each object includes key fields such as Email, Phone, Geolocation, Text Area, Picklist, and Lookup relationships.







Relationships

| Object | Relationship Type | Related object |
|----------------------|-------------------|----------------|
| Volunteer | Master-Detail | Drop-Off Point |
| Execution Details | Master-Detail | Volunteer |
| Execution Details | Master-Detail | Task |
| Drop-Off Point | Lookup | Venue |
| Task | Lookup | Venue |
| Task | Lookup | Drop-Off Point |

Lightning App Creation

App Name: Food Connect

Navigation Items: Home, Venue, Drop-Off Point, Task, Volunteer, Execution Details,

Reports.

Profiles: System Administrator and NGO users.

Automation

Flow Creation

Flow Name: Venue Form

Type: Screen Flow

Purpose: To create Venue records from UI form input automatically.

Fields include Venue Name, Contact Email, Contact Phone, Latitude, and Longitude.



Apex Trigger

```
Trigger Name: DropOffTrigger
Object: Drop-Off Point

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

Purpose: To calculate and assign the "Distance" field automatically for sharing rule processing.

Sharing Rules

Rules created based on Distance criteria:

- Rule 1: Distance < 15 → Shared with Iksha Group
- Rule 2: 15 ≤ Distance ≤ 30 → Shared with NSS Group
- Rule 3: 30 ≤ Distance ≤ 50 → Shared with Street Cause Group

Reports and Dashboards

Reports:

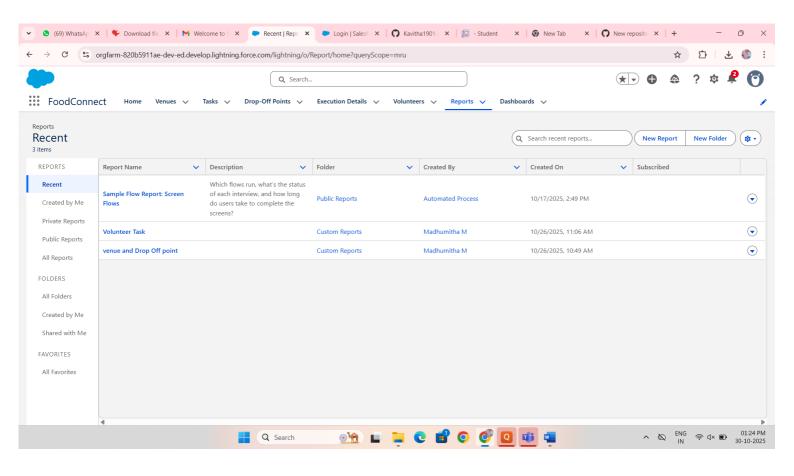
- 1. Venue with Drop-Off with Volunteer
 - Displays Venue, Drop-Off Points, and Volunteer Names.
- Volunteer Task
 - Shows Volunteers, Task Details, Ratings, and Execution Data.

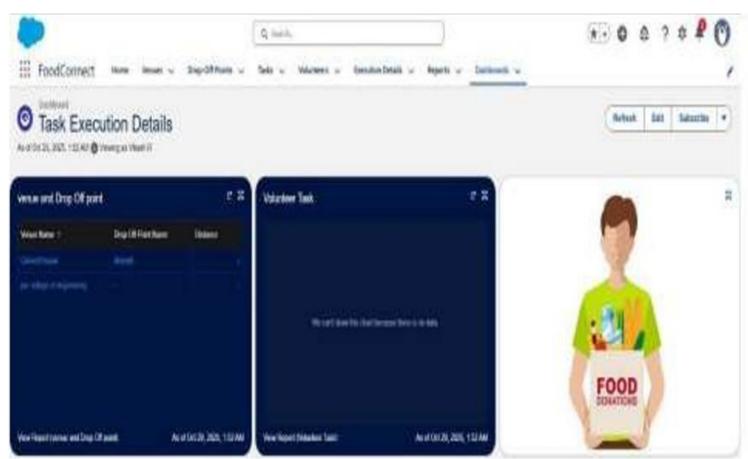
Dashboard:

Name: Organization Details

Components:

- Lightning Table for Venue & Drop-Off Report
- Line Chart for Volunteer Task Report
- Optional image logo





Testing Phase:

Test Cases

| Test Case | Description | Expected Output | Expected Output |
|--------------|--------------------------------|--------------------------------|-----------------|
| TC01 | Create new Venue | Record saved successfully | Pass |
| TC02 | Volunteer assigned to Drop-Off | Linked correctly | Pass |
| TC03 | Flow form submission | Venue record auto- created | Pass |
| TC04 | Trigger calculation | Distance updated automatically | Pass |
| TC05 | Dashboard view | Displays data correctly | Pass |

Validation Rules

Validation rules ensure correct data entry

(e.g., mandatory fields, valid emails, and phone numbers).

Example:

ISBLANK(Contact Email c)

→ Displays error if email is missing.

Deployment Phase:

The project was deployed in a Salesforce Developer Org. Steps for replication:

- Create Salesforce Developer Org.
- Recreate custom objects and fields as listed.
- Configure relationships and flows.
- · Add users, public groups, and sharing rules.
- Build reports and dashboards.

Once verified, assign app to NGO user profiles and test record creation via Home Page flow.

Limitations:

- System relies on manual record input; no IoT-based automation yet.
- Distance calculation accuracy depends on geolocation precision.
- Limited offline accessibility.
- Requires internet connectivity for all operations.

Future Enhancements:

- Integrate mobile app with real-time location tracking.
- Enable automatic food pickup scheduling through Apex scheduler.
- Add WhatsApp/SMS notification integration.
- Build community portal for donors and NGOs.

Conclusion:

This project demonstrates how Salesforce CRM can be effectively utilized for social welfare and resource optimization.

The system provides a structured way to connect food donors, volunteers, and distribution points through cloud-based automation, ensuring accountability, efficiency, and transparency. It can serve as a scalable foundation for NGOs and civic bodies working towards reducing hunger and food waste.

References:

- https://developer.salesforce.com/
- https://trailhead.salesforce.com/
- Salesforce Documentation & Flow Builder Guide