

To Supply Leftover Food to Poor

College Name: TERF'S ACADEMY COLLEGE OF ARTS AND SCIENCE

College Code: bru5e

TEAMID: NM2025TMID27517

TEAM MEMBERS:

Team LeaderName: VEGASHINI S

Email: vegashini4@gmail.com

Team Member1: VIKASH D

Email: dvikash211@gmail.com

Team Member2: VISHAL V

Email: vvishal122005@gmail.com

Team Member3: YASWANTH M

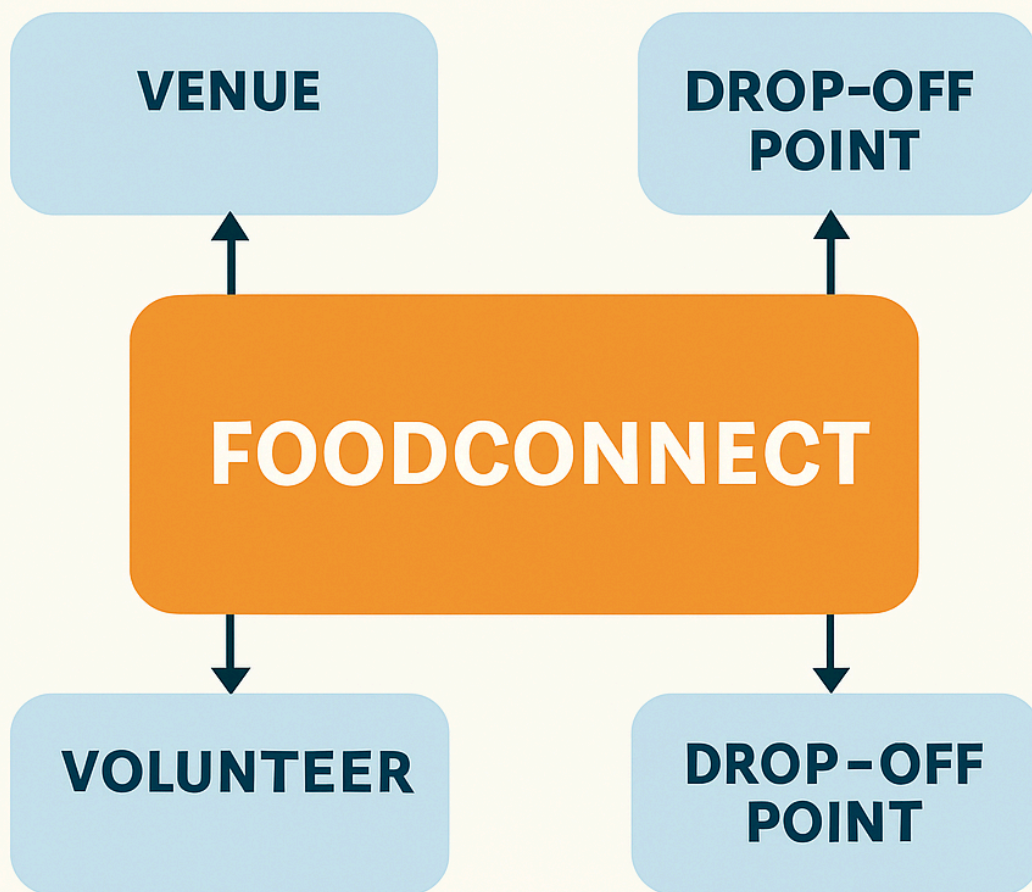
Email: yaswanth4327@gmail.com

1. INTRODUCTION

1.1 Project Overview

The **FoodConnect System** is a Salesforce-based application designed to streamline the process of **collecting and distributing leftover food**.

It handles venue management, volunteer coordination, drop-off points, and execution tracking with automation features such as flows, approval processes, triggers, and reports.



1.2 Purpose

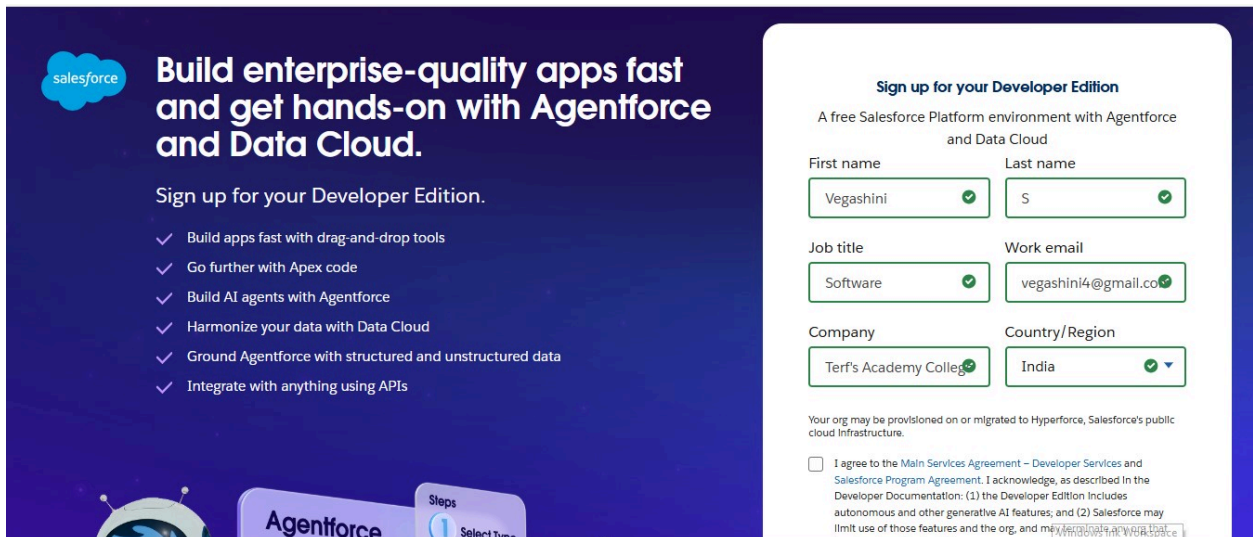
The main objective of this project is to **reduce food wastage** and ensure that surplus food is efficiently supplied to poor and needy people.

It connects venues like hotels or events with NGOs and volunteers. The system reduces manual intervention, improves coordination, and ensures food reaches people on time.

2. DEVELOPMENT PHASE

Creating Developer Account

By using this URL – <https://developer.salesforce.com/signup>



Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition
A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Vegashini ✓ Last name: S ✓

Job title: Software ✓ Work email: vegashini4@gmail.co ✓

Company: Terf's Academy Colleg ✓ Country/Region: India ✓

☐ I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may provision your org on Hyperforce, Salesforce's public cloud infrastructure.

Created Objects

The following custom objects were created to manage leftover food supply:

- **Venue** – Stores details of where food is available (Hotels, Restaurants, Events).
- **Drop-Off Point** – Temporary storage or NGO centers for collected food.
- **Volunteer** – People who help transport and distribute food.
- **Task** – Assigns food collection or delivery activities.
- **Execution Details** – Tracks task completion and delivery status.

Configured Fields and Relationships

- Lookup relationships between **Venue** → **Drop-Off Point**
- Master-Detail relationship between **Volunteer** → **Execution Details**
- Master-Detail relationship between **Task** → **Execution Details**
- Lookup from **Task** → **Venue** and **Task** → **Drop-Off Point**

Developed Lightning App with Tabs

A Lightning App named **FoodConnect** was created.
It includes tabs for:

- **Venue**

- Drop-Off Point
- Volunteer
- Task
- Execution Details
- Reports

Implemented Flow for Venue Creation

A **Flow** was developed to quickly capture venue details such as:

- Venue name
- Contact details
- Food category (Veg/Non-Veg/Snacks)
- Location

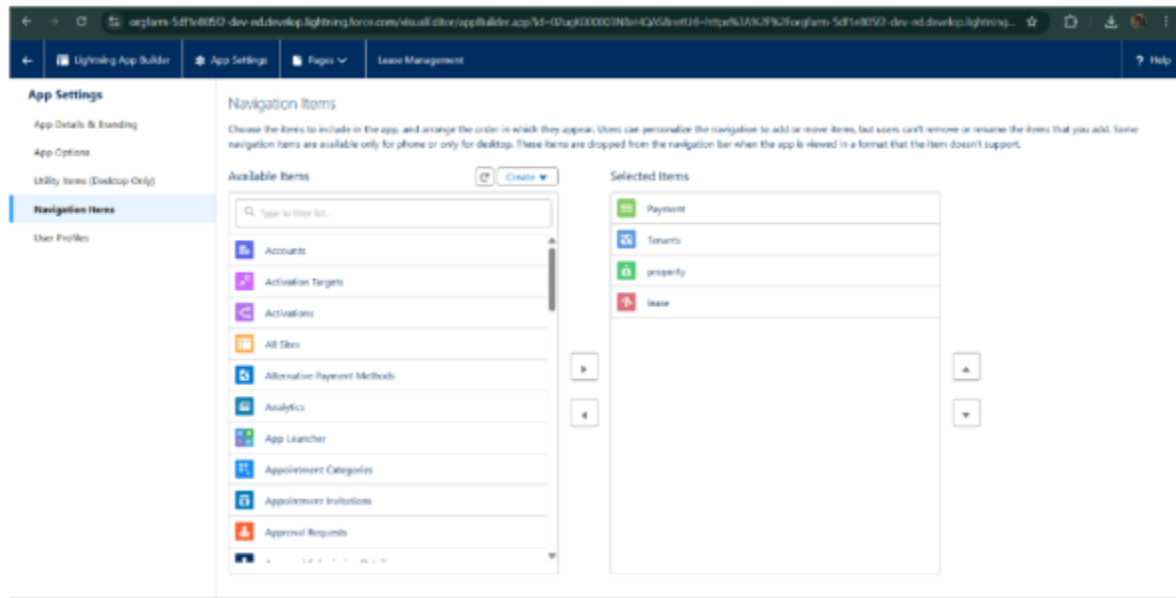
This reduces manual data entry and ensures accurate records.

Validation Rules

Validation rules were added to ensure:

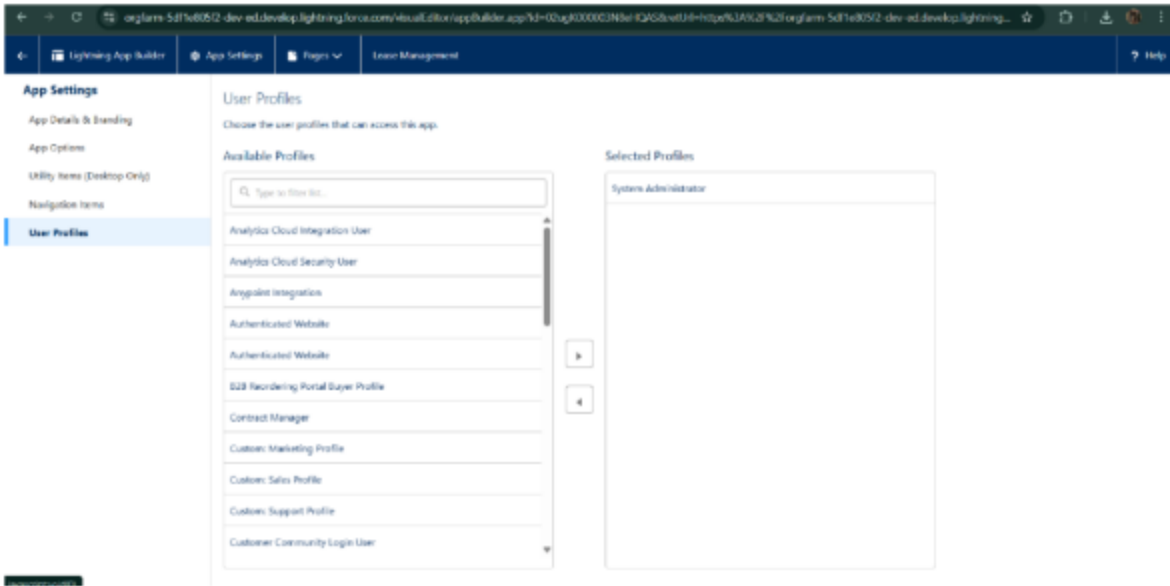
- Contact details (Phone, Email) are mandatory

- Food quantity must be greater than zero
- Volunteers must have age and contact number filled



Apex Trigger

Trigger on **Drop-Off Point** object to calculate distance automatically and assign it to the Distance field for sharing rules.



Scheduled Apex Class

A scheduled Apex class was implemented to send **daily notifications to volunteers and NGOs** about upcoming food collection tasks.

Email Templates

Custom email templates were created for:

- New food available notification
- Volunteer task assignment
- Task completion confirmation
- Feedback requests

Approval Process

Approval process for **food distribution tasks**:

- NGO Admin approves volunteer assignment
- Task is marked as active only after approval

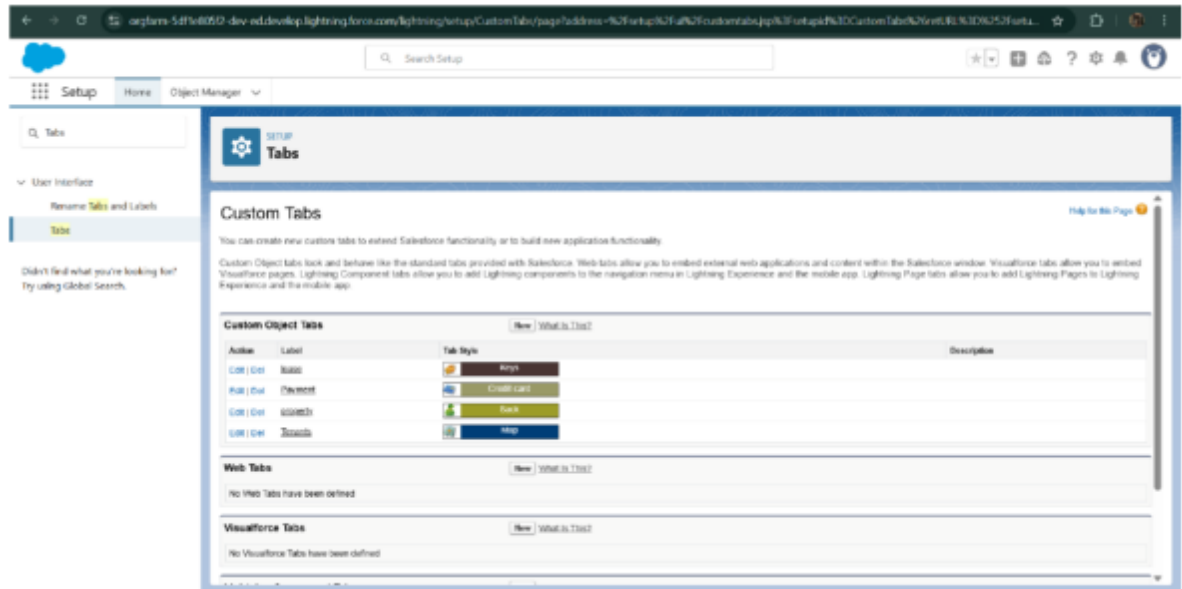
3. FUNCTIONAL AND PERFORMANCE TESTING

- Trigger validation by entering incorrect or duplicate values
- Validation Rule checking (e.g., empty contact details)
- Flow tested for venue creation
- Approval process tested through email alerts and status updates
- Reports verified for accuracy
- Dashboard tested for visualization

4. RESULTS

Output Screenshots

- Tabs for Venue, Drop-Off Point, Volunteer, Task, Execution Details



- Flow form for Venue creation
- Reports for Venue → Drop-Off → Volunteer mapping
- Dashboard showing food collection and distribution statistics
- Email notifications to volunteers and NGOs
- Trigger updates for distance calculation

5. ADVANTAGES & DISADVANTAGES

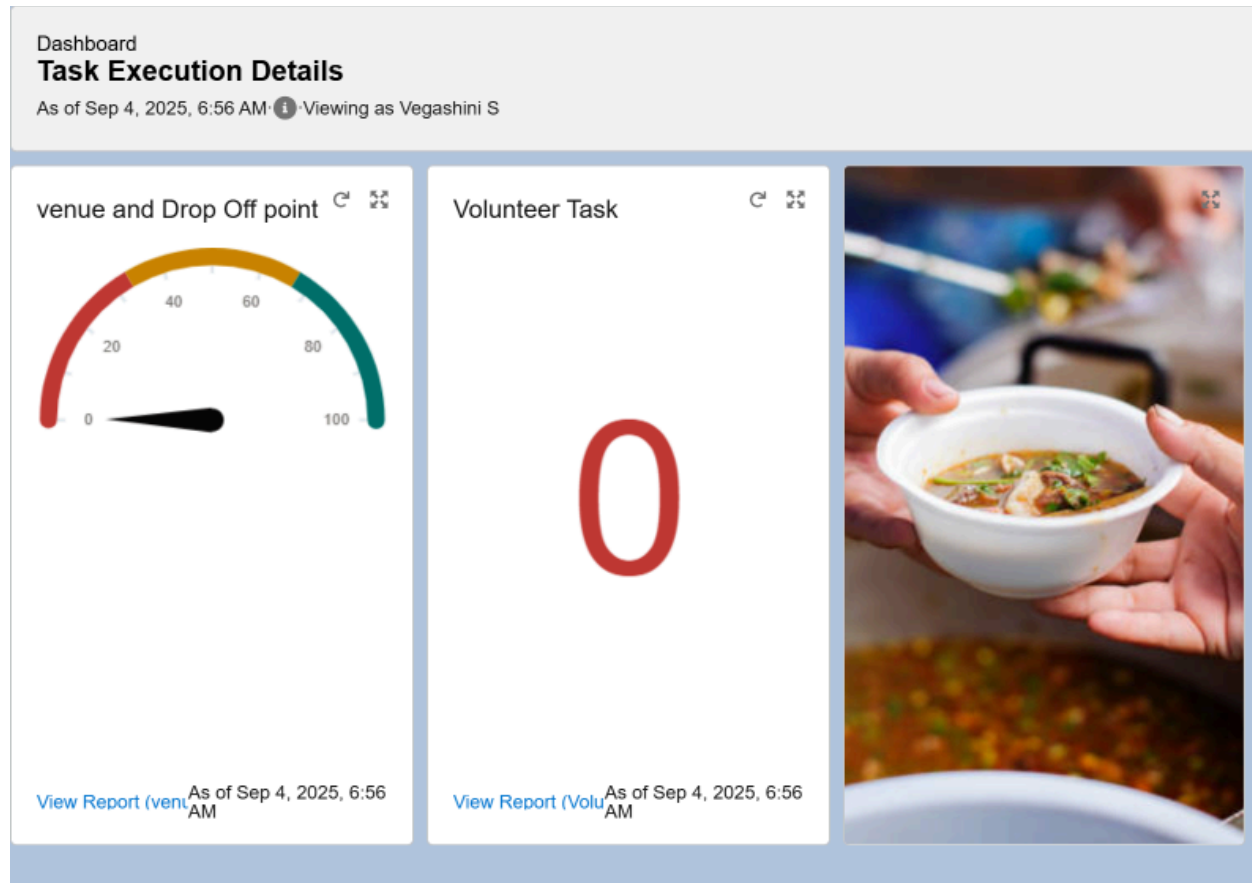
Advantages

- Reduces food wastage

- Ensures timely delivery to needy people
- Automates NGO and volunteer coordination
- Easy to track tasks and execution

Disadvantages

- Dependent on internet access
- Requires Salesforce knowledge for setup
- Volunteer availability may affect distribution



6. CONCLUSION

The **FoodConnect System** successfully streamlines leftover food management using Salesforce.

It reduces waste, improves coordination between NGOs, volunteers, and venues, and ensures food reaches the poor on time.

This project makes a strong **social impact** by using technology to solve a real-world problem.

7. APPENDIX

Sample Trigger (Drop-Off Point)

```
trigger DropOffTrigger on Drop_Off_Point__c (before insert) {
```

```

for(Drop_Off_Point__c d : Trigger.new){
    d.Distance__c = d.Distance_Calculation__c;
}
}

```

Sample Scheduled Class

```

global class DailyFoodReminder implements Schedulable {
    global void execute(SchedulableContext sc) {
        sendDailyReminders();
    }
    public static void sendDailyReminders() {
        List<Volunteer__c> vols = [SELECT Id, Email__c FROM
Volunteer__c];
        for(Volunteer__c v : vols){
            Messaging.SingleEmailMessage mail = new
Messaging.SingleEmailMessage();
            mail.setToAddresses(new String[]{v.Email__c});
            mail.setSubject('Food Collection Task Reminder');
            mail.setPlainTextBody('Please check your assigned tasks for
today.');
```

```

            Messaging.sendEmail(new
Messaging.SingleEmailMessage[]{mail});
        }
    }
}

```