**FOOD CONNECT**

**Meals That Matter: Feed The Hungry**

**Project Overview**:

**Food Connect** is a cloud-based CRM application that links food donors and restaurants with underserved and impoverished communities. The software offers a methodical and effective approach to overseeing the distribution and donation of excess food. The system enables contributors, including eateries, event planners, and private citizens, to register and record excess food products, which are subsequently collected by a network of registered volunteers. By carefully allocating these volunteers according to their availability and location, food is delivered to those in need in a timely and secure manner. Food Connect uses Salesforce tools like Process Builder, Flows, Apex, and Dashboards to automate important tasks like tracking donations, assigning volunteers, and confirming deliveries. All things considered, Food Connect reduces food waste and gives communities the tools they need to effectively combat hunger inside a scalable and intuitive CRM ecosystem.

**Objectives:**

The primary goal of **Food Connect** is to develop an easy-to-use and effective system that enables eateries and private citizens to donate excess food to the underprivileged and needy. The project's goal is to use Salesforce CRM to link food donors with local NGOs and volunteers who can promptly pick up and distribute the food. In addition to reducing food waste and helping to feed the hungry, this makes the process more efficient, well-organized, and beneficial to the community.

**Requirement Analysis & Planning:**

**Business Requirements:**

A simple method for handling food donors, such as eateries and people wishing to donate leftover food, must be included in the **Food Connect** app. It should facilitate the coordination of volunteers who swiftly pick up and distribute the meals. To ensure that everyone is aware of when food is donated, picked up, and delivered, the app must display real-time updates. Additionally, it should automatically send out communications to organizations, volunteers, and funders. Maintaining a registry of shelters and charity helps ensure that the food ends up in the appropriate hands. Lastly, the app ought to be secure and expand as more users sign up for the program.

**Project Scope:**

The goal of the Food Connect project is to create a CRM application that runs on Salesforce to handle the distribution of surplus food from businesses and people to underprivileged areas. Developing modules for beneficiary (NGO/shelter) tracking, volunteer management, and donor registration is part of the scope. It will allow for automated alerts to inform all parties and real-time tracking of food donation requests, pickups, and delivery. Information on volunteer activity and donation levels will be available through dashboards and reporting. The project includes user training, system design, development, testing, and deployment.

**Project Objectives:**

1. **User-Friendly Platform:**  
   Establish an easy-to-use system that effectively connects volunteers, nonprofit groups, and food contributions.
2. **Efficient Food Collection and Delivery:**  
   Make certain that food scraps from eateries and private households are promptly collected and distributed to those in need.
3. **Automated Communication:**  
   For seamless coordination, automate messages and notifications between volunteers, donors, and NGOs.
4. **Reduce Food Waste:**  
   Redirect excess food to the hungry and impoverished to reduce food waste.
5. **Data Reporting and Monitoring:**  
   Create reports to monitor contributions, volunteer work, and enhance the giving process.

**Data Model:**

**1. Donor (Account / Custom Object):**

Represents people or groups that donate extra or leftover food on a regular basis.   
**Relationship:** A single donor can generate numerous food donations.

**2. Food Donation (Custom Object)**

Records the pertinent information about every food donation, such as the kind of food, how much is being donated, when it expires, and its current state.   
**Relationship:** Every food donation has a donor, a designated volunteer, and a recipient or distribution center.

**3. Volunteer (Contact / Custom Object):**

Symbolizes someone who assists in gathering and delivering food from donors to recipients or facilities.   
**Relationship:** A single volunteer can manage numerous food donations.

**4. Beneficiary (Custom Object):**

Refers to the recipients of the given food, which may be families, individuals, or support groups.   
**Relationship:** More than one food donation can be given to the same beneficiary.

**5. Distribution Centre (Custom Object):**

A temporary storage point where donated food can be held before it is distributed to beneficiaries  
**Relationship:** One Distribution Centre can hold multiple Food Donations

**6. Delivery Record (Custom Object):**

Tracks the final delivery status, condition of food, and feedback from the recipient after a donation is delivered  
**Relationship:** Each Delivery Record is connected to one Food Donation and one Beneficiary.

**Security Model:**

**Object-Level Security:**

**1. Donor:**

Access is given to Admins and Donation Managers; Volunteers have read-only access

**2. Food Donation:**

Admins and Donation Managers have full access; Volunteers have read and edit access only for assigned records

**3. Volunteer:**

Only Admins can create or update volunteer profiles; Volunteers can view their own information

**4. Beneficiary:**

Accessible by Admins and Delivery Coordinators; Volunteers have read-only access if linked to a delivery

**5. Distribution Centre:**

Full access for Admins; read-only for Logistics Team; not visible to Volunteers

**6. Delivery Record:**

Admins and Delivery Coordinators can create and update; Volunteers can view records they are assigned to

**Role Hierarchy:**

**1. System Admin:**

Has full access to all records and configuration settings

**2. Donation Manager:**

Manages donor data and food donation records; can assign volunteers

**3. Volunteer:**

Can view assigned donations and delivery records; limited access to other modules

**4. Delivery Coordinator:**

Handles distribution and delivery tracking; has access to beneficiaries and delivery records

**5. Logistics Team:**

Manages distribution centres and inventory-related info; cannot access donor or volunteer data

**Record-Level Security:**

**1. Organization-Wide Defaults (OWD):**

1. Donor: Private
2. Food Donation: Controlled by Parent
3. Volunteer: Private
4. Beneficiary: Public Read Only
5. Delivery Record: Controlled by Parent
6. Distribution Centre: Public Read Only

**2. Sharing Rules:**

1. Auto-share Food Donations with assigned Volunteers
2. Share Delivery Records with related Delivery Coordinators
3. Manual sharing enabled for special cases

**3. Profiles and Permission Sets:**

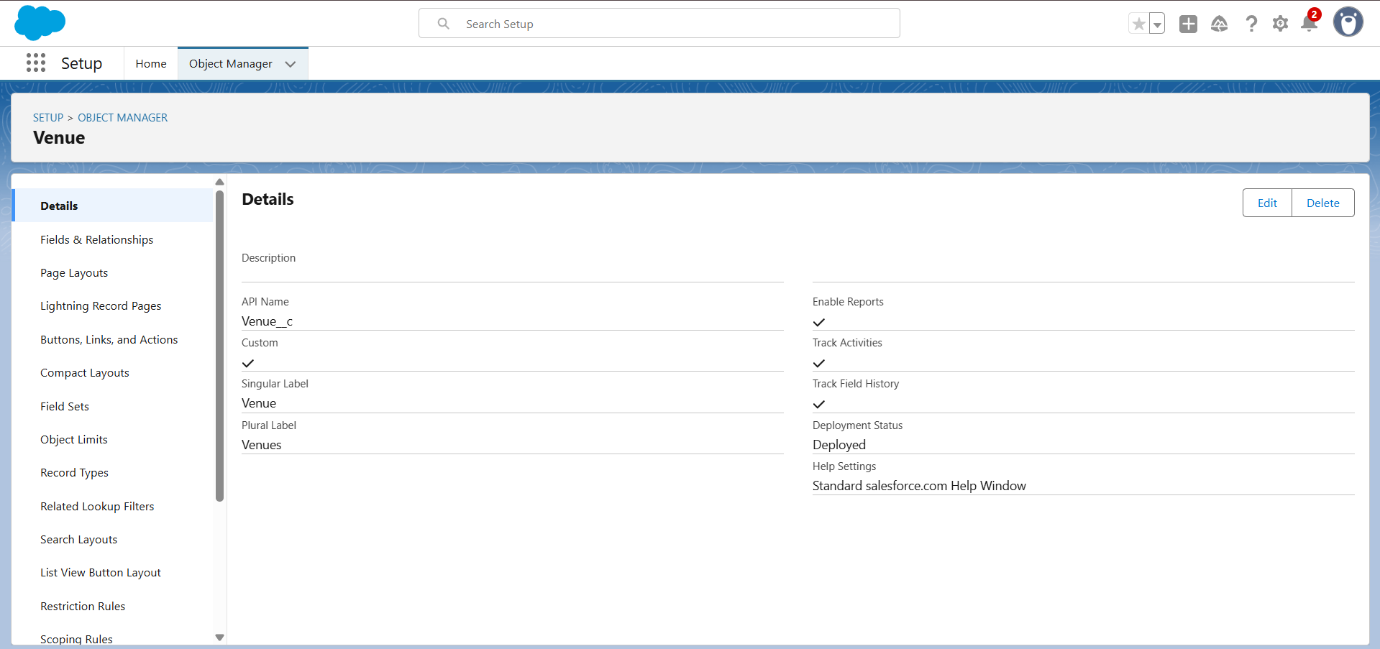
1. Profiles control baseline access per role
2. Permission Sets extend access for specific tasks (e.g., assign volunteers, export reports)

**Salesforce Development - Backend & Configurations:**

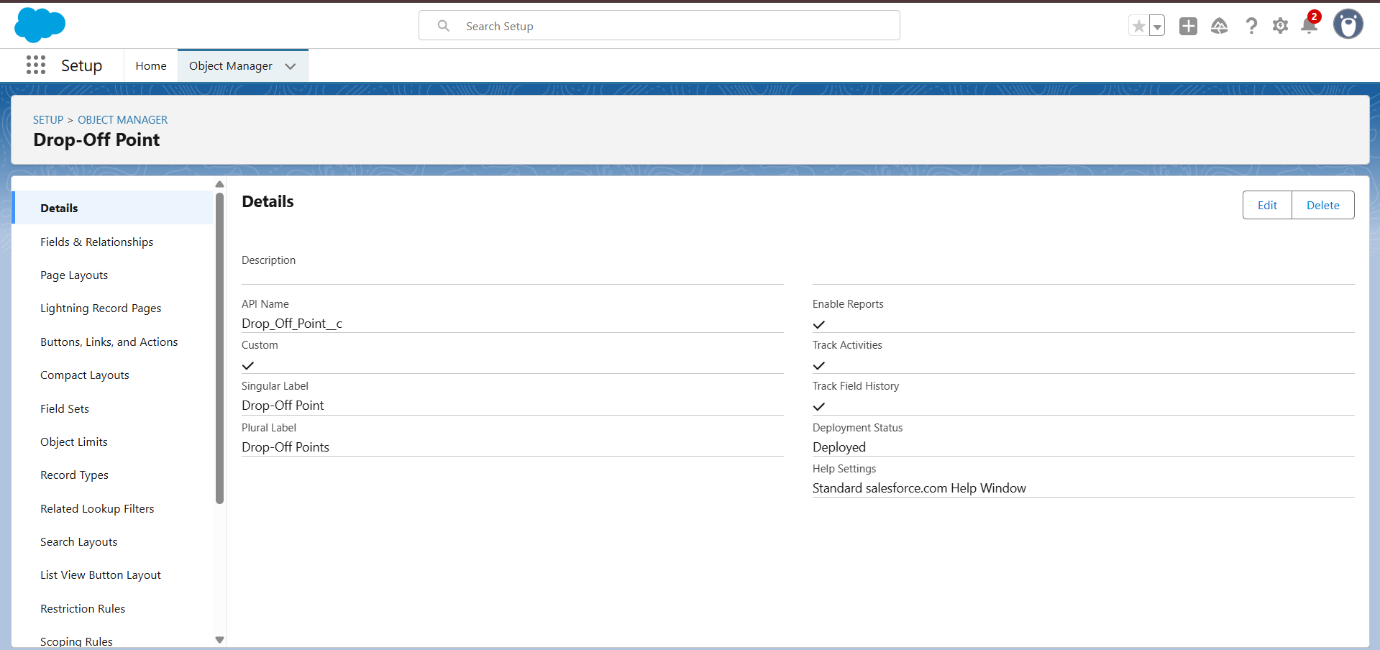
**Objects:**

1. **Venue:**

* Enter the label name >> Venue
* Plural label name >> Venues
* Enter Record Name Label and Format
* Record Name >> Venue Name
* Data Type >> Text

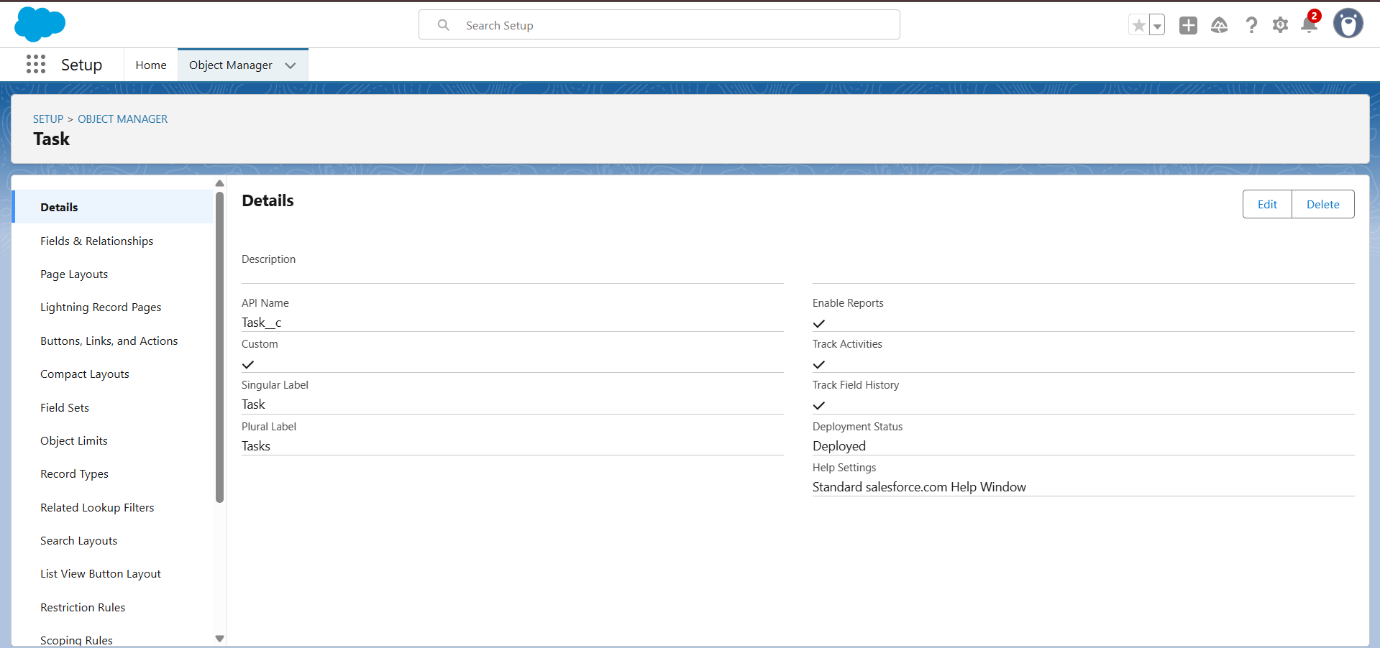
****

1. **Drop-Off Point:**

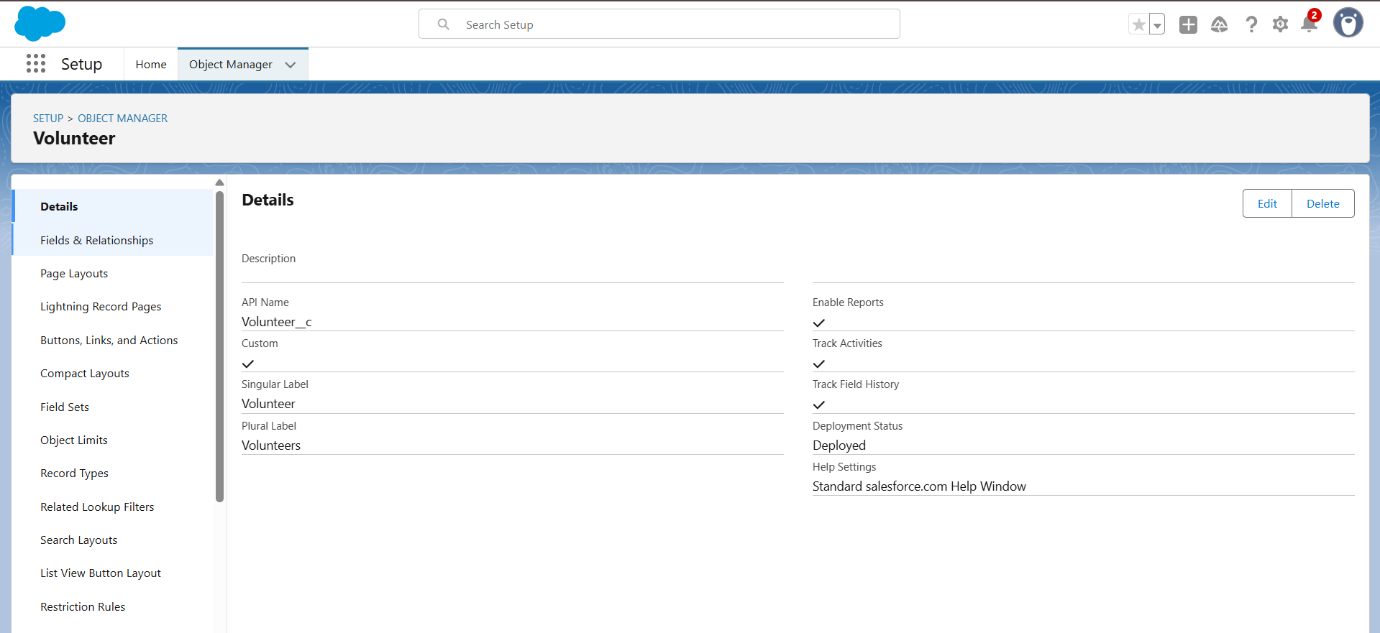
* Enter the label name >> Drop-Off Point
* Plural label name>> Drop-Off Points
* Enter Record Name Label and Format
* Record Name >> Drop-Off point Name
* Data Type >> Text

1. **Task:**

* Enter the label name>> Task
* Plural label name>> Tasks
* Enter Record Name Label and Format
* Record Name >> Task Name
* Data Type >> Text

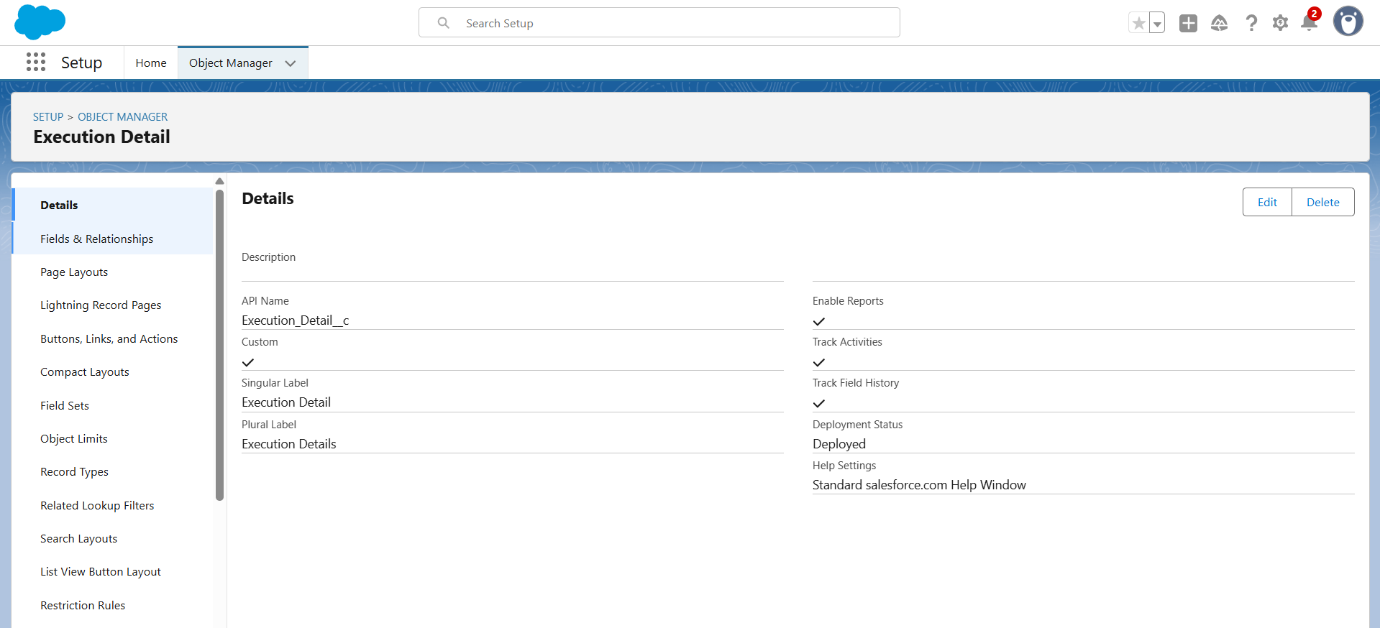
****

1. **Volunteer:**

* Enter the label name>> Volunteer
* Plural label name>> Volunteers
* Enter Record Name Label and Format
* Record Name >> Volunteer Name
* ****Data Type >> Text

1. **Execution Details:**

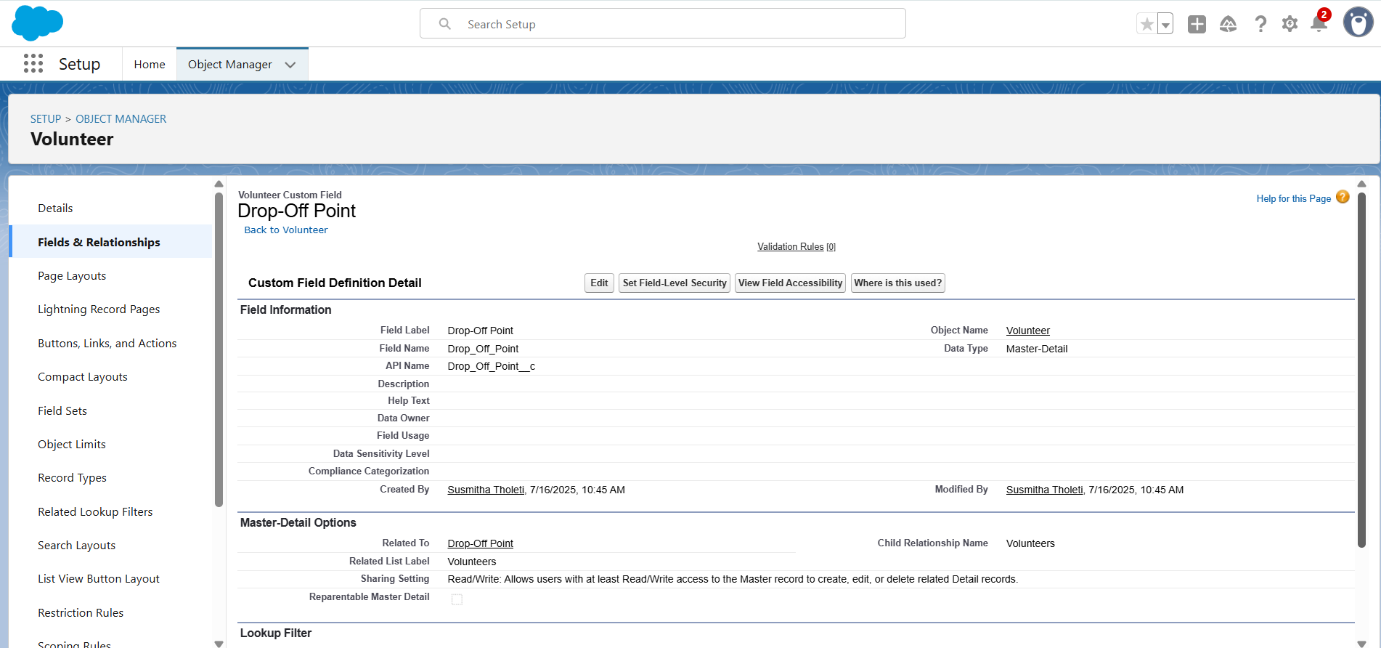
* Enter the label name >> Execution Detail
* Plural label name >> Execution Details
* Enter Record Name Label and Format
* Record Name >> Execution Detail Name
* Data Type >> Text

****

**Creation of Relationship fields in objects:**

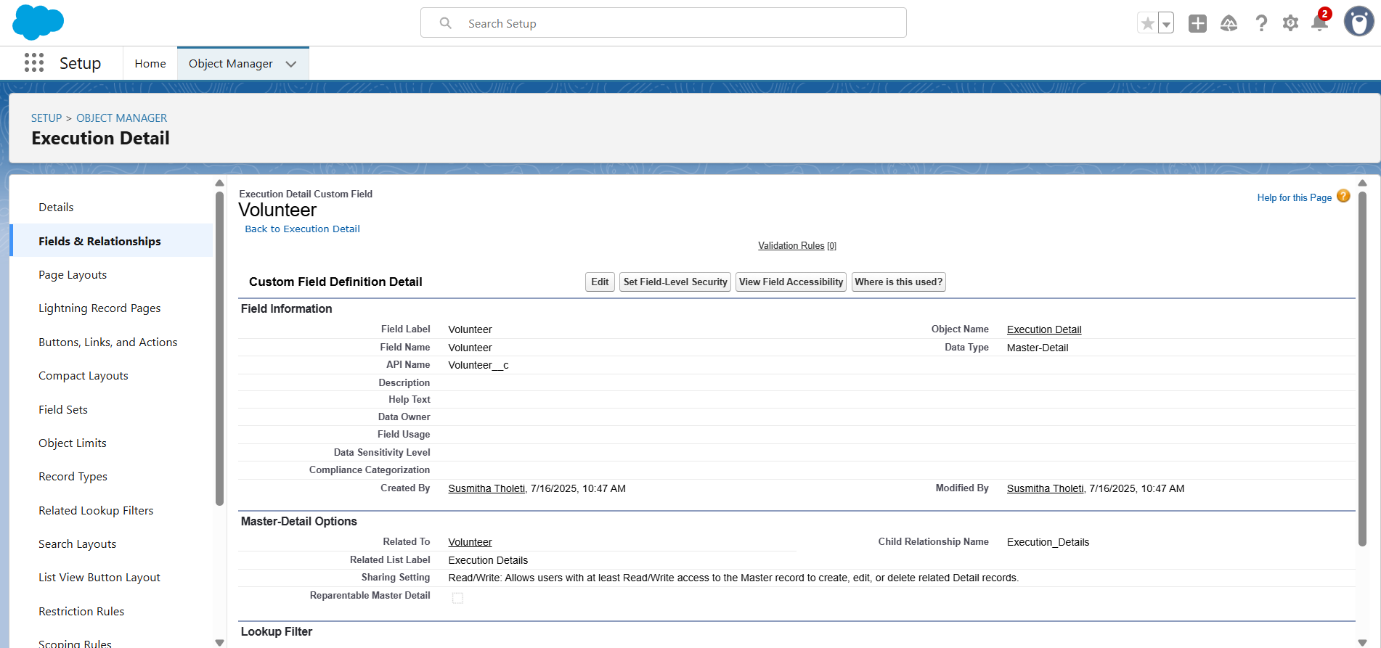
Creation of Lookup Relationship Field on Volunteer Object:

1. Select Master Detail relationship
2. Field Name: Drop\_Off\_point
3. Field label: Auto generated



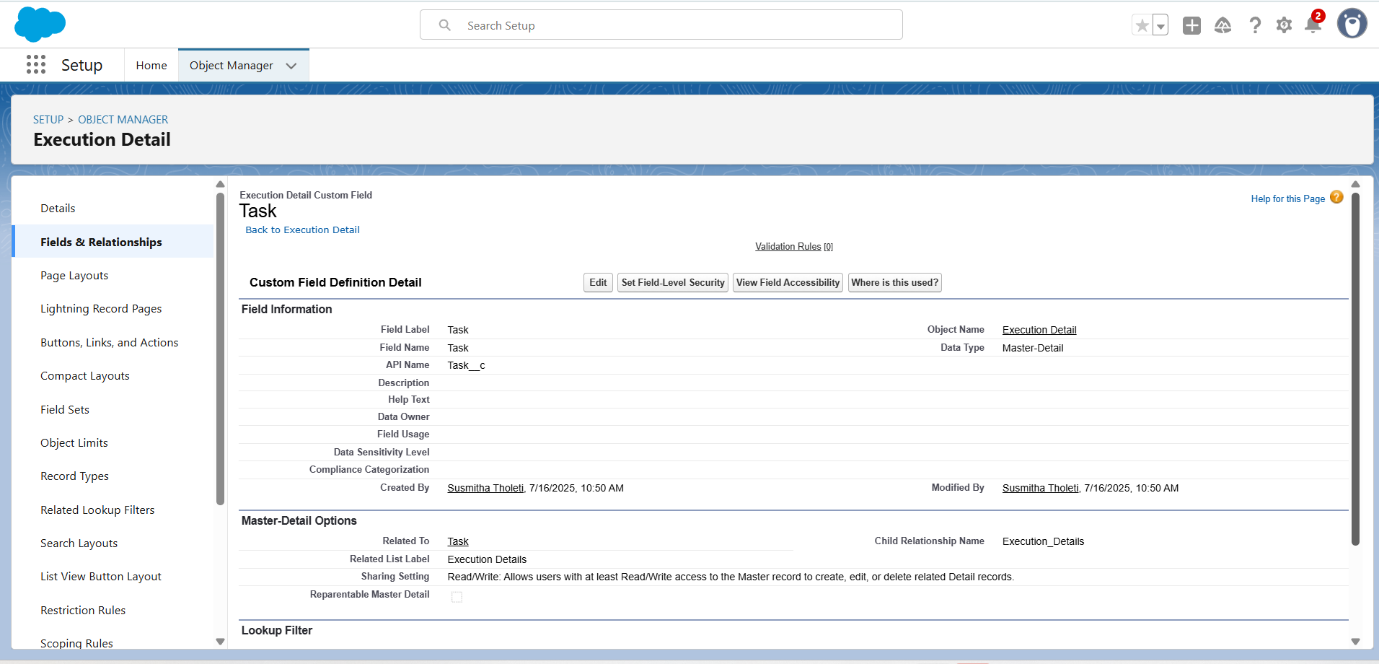
Creation of Master Detail Relationship Field on Execution Details Object:

1. Select Master Detail relationship
2. Field Name: Volunteer
3. Field label: Auto generated



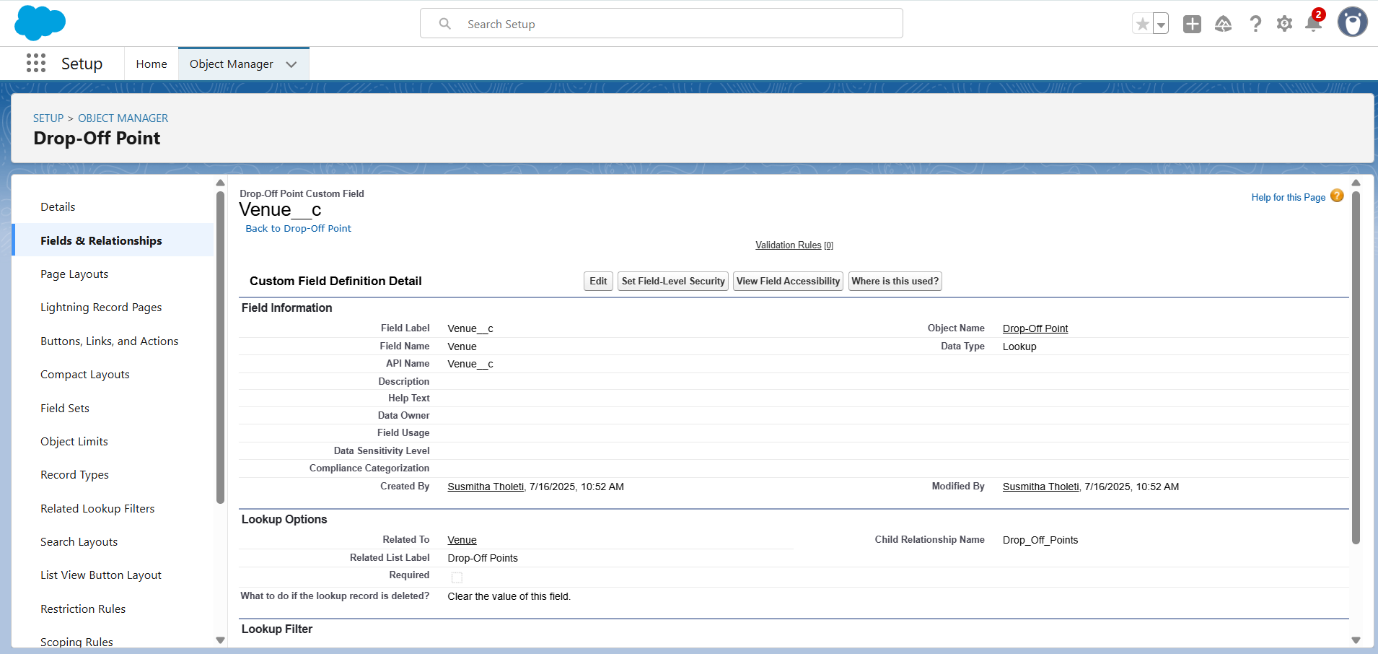
Creation of Master Detail Relationship Field on Execution Details Object:

1. Select Master Detail relationship
2. Field Name: Task
3. Field label: Auto generated



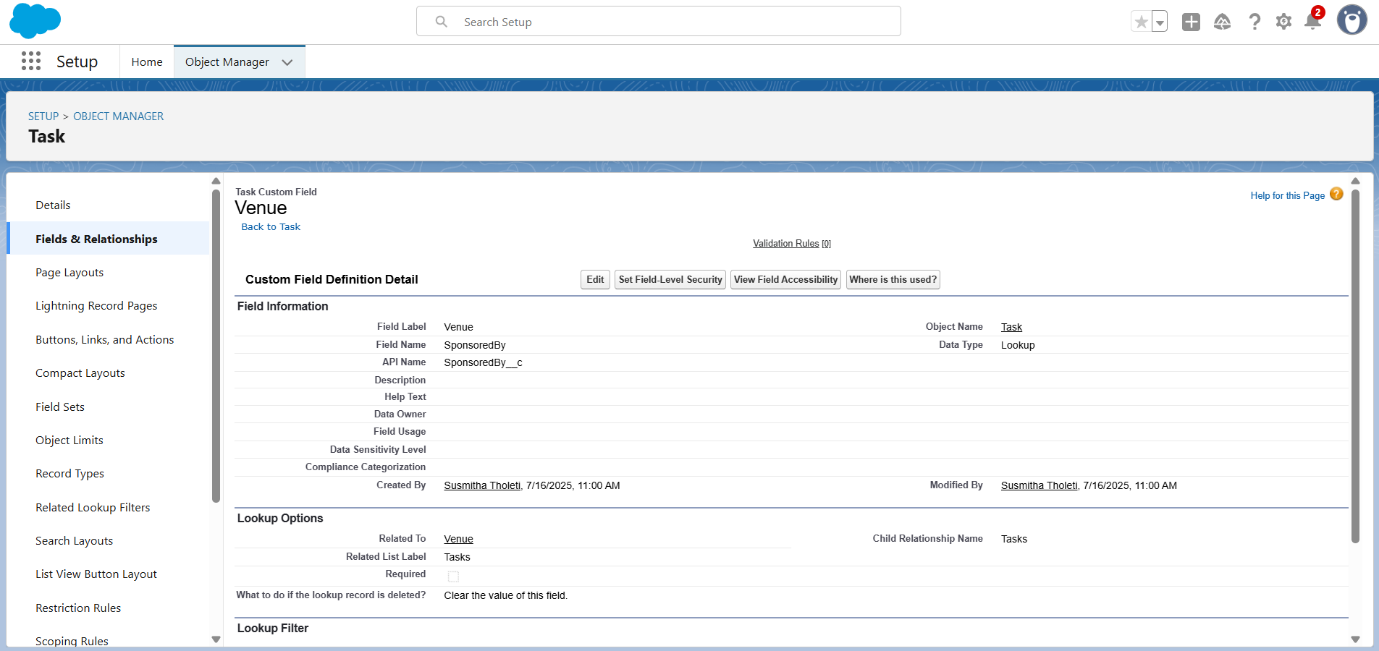
Creation of Lookup Relationship Field on Drop-Off Point Object:

1. Select Lookup relationship
2. Field Name: Venue
3. Field label: Venue\_\_c



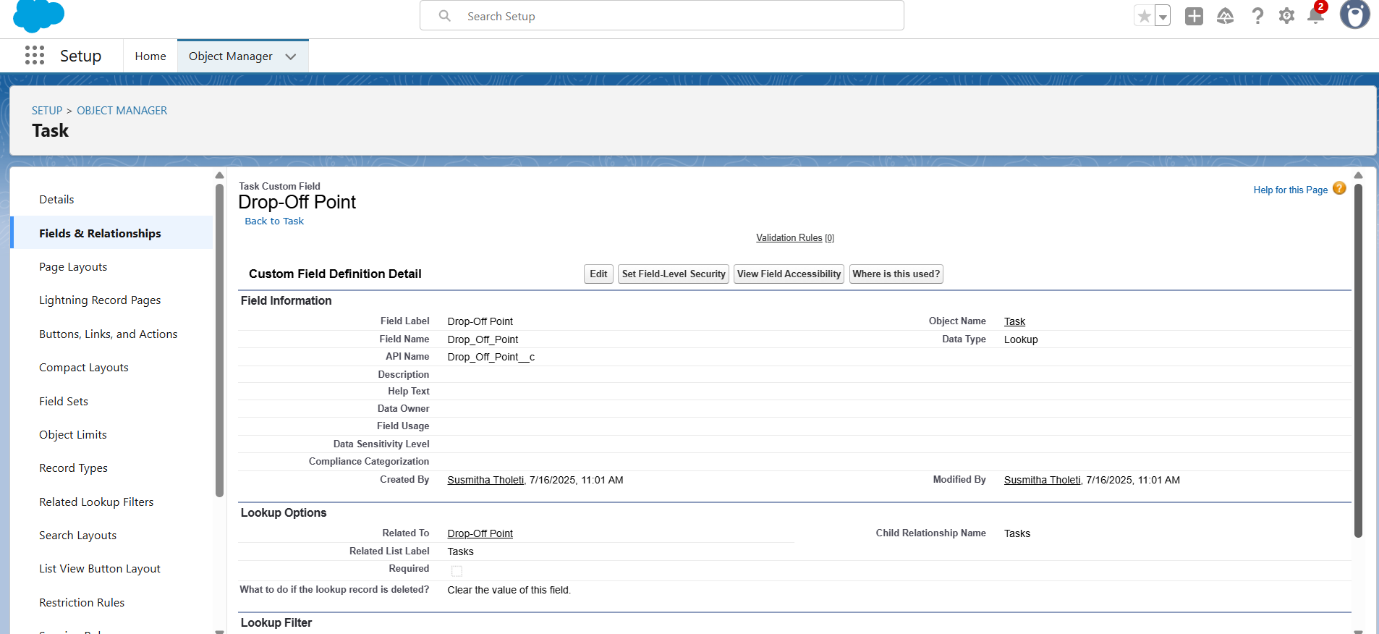
Creation of Lookup Relationship Field on Task Object:

1. Select Lookup relationship
2. Field Name: Sponsored By
3. Field label: Auto generated



Creation of Lookup Relationship Field on Task Object:

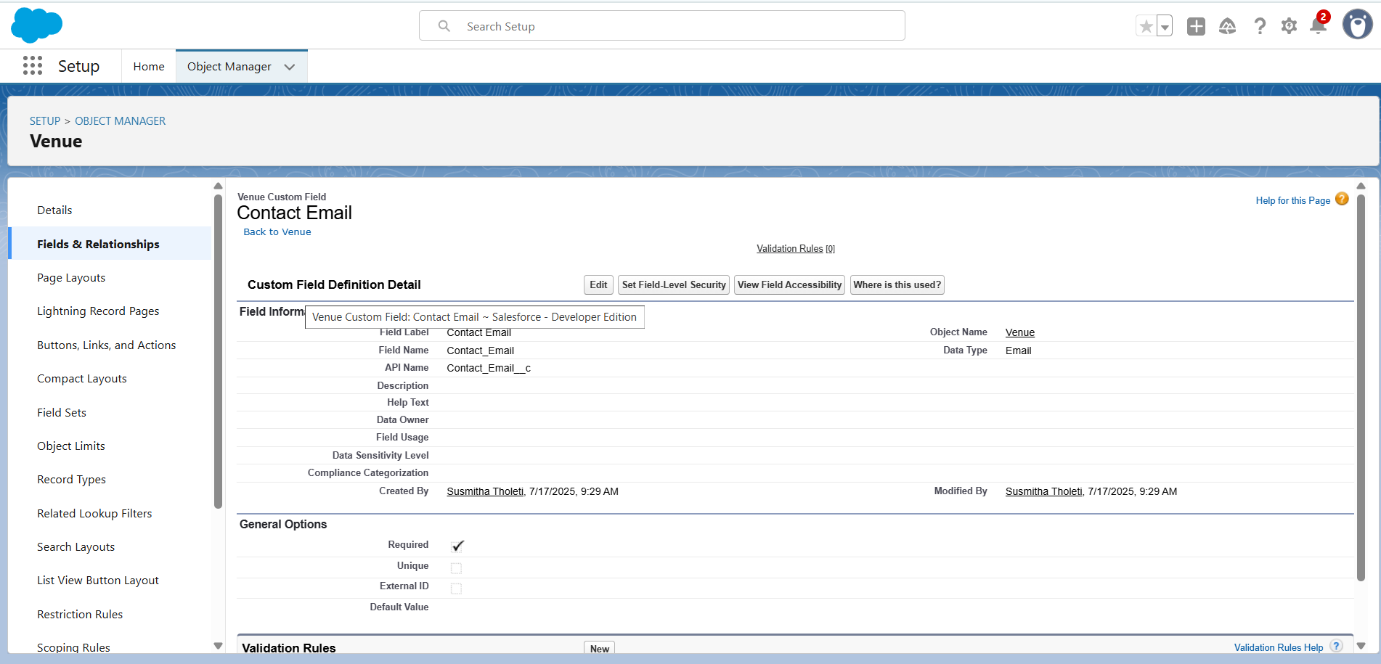
1. Select Lookup relationship
2. Field Name: Drop-Off point
3. Field label: Auto generated



**Creation of fields for the Venue object:**

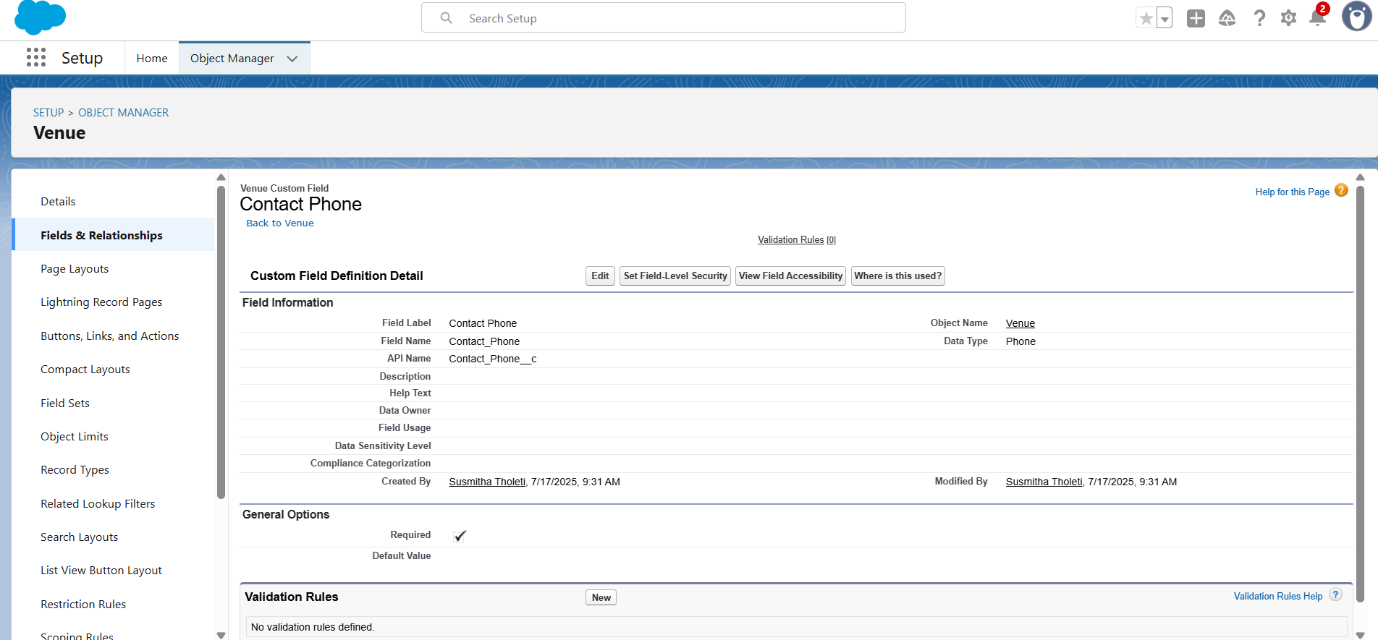
Email field:

1. Data Type: Email
2. Field Label: Contact Email
3. Field Name: Contact Email
4. Click on required check box



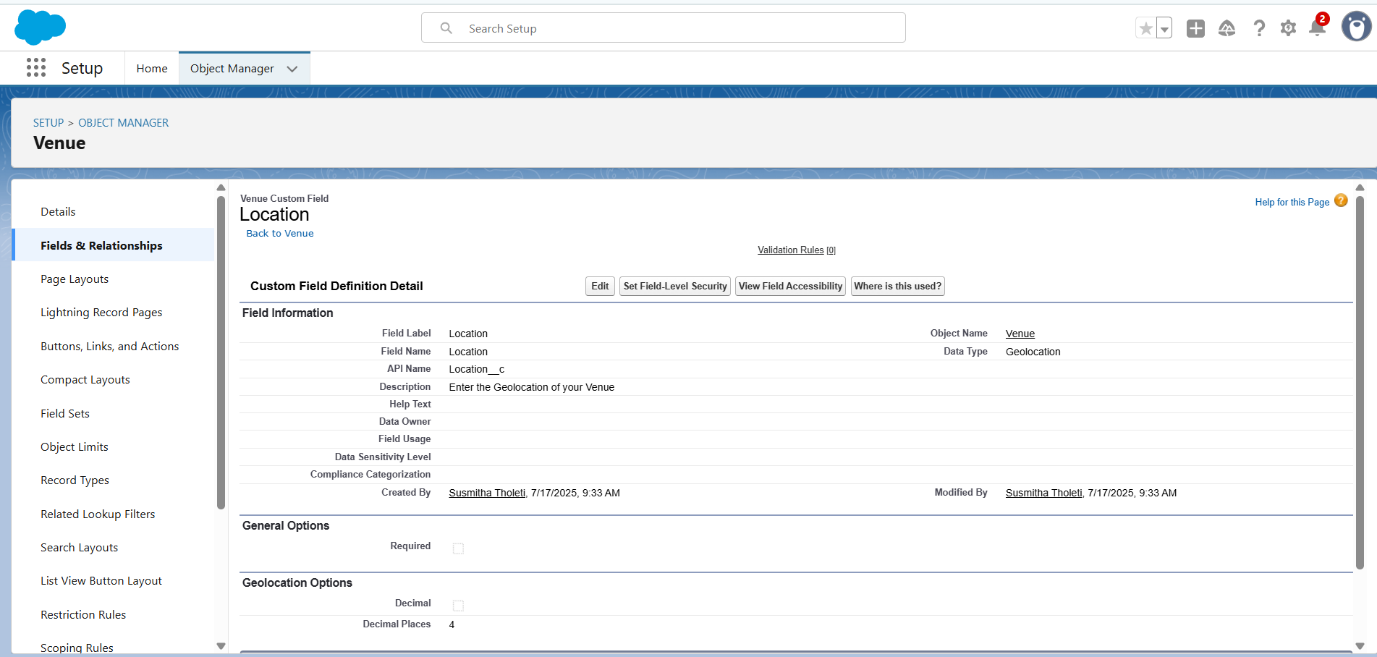
Phone field:

1. Data Type: Phone
2. Field Label: Contact Phone
3. Field Name: Contact Phone
4. Click on required check box



Location field:

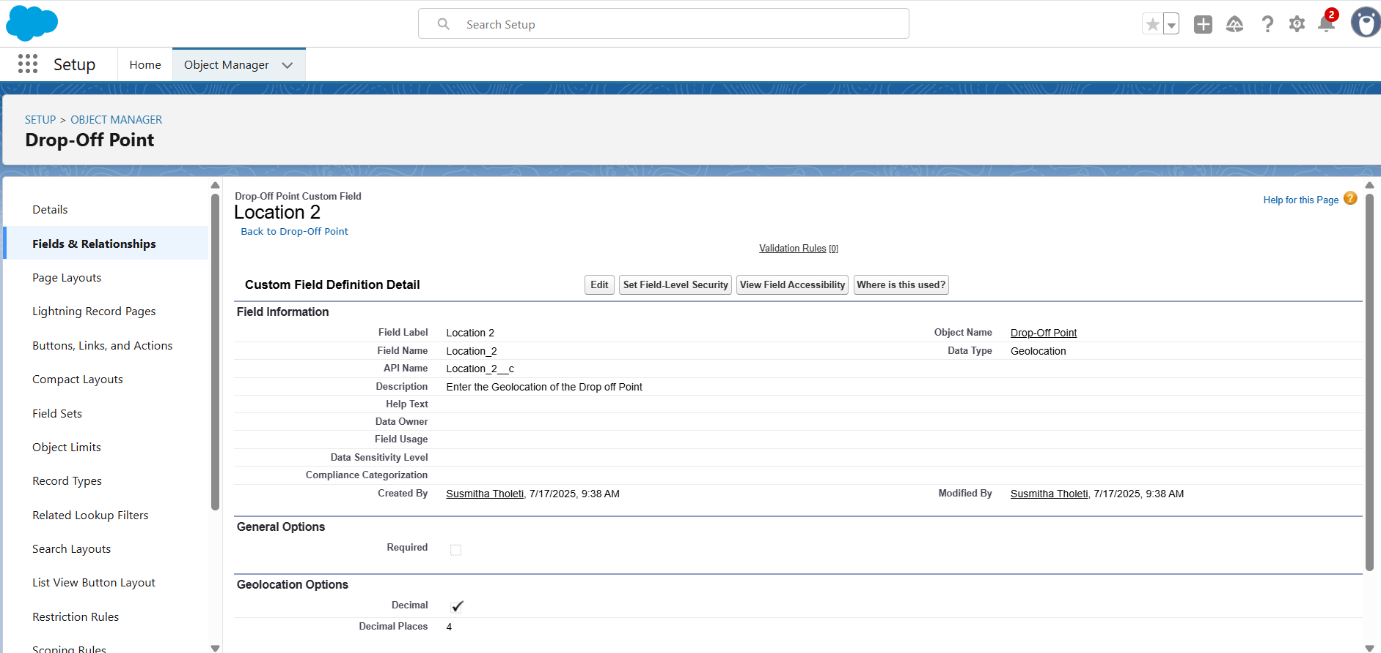
1. Field Label: Location
2. Decimal Places: 4
3. Field Name: Location
4. Description: Enter the Geolocation of your Venue



**Creation of fields for the Drop-Off point object:**

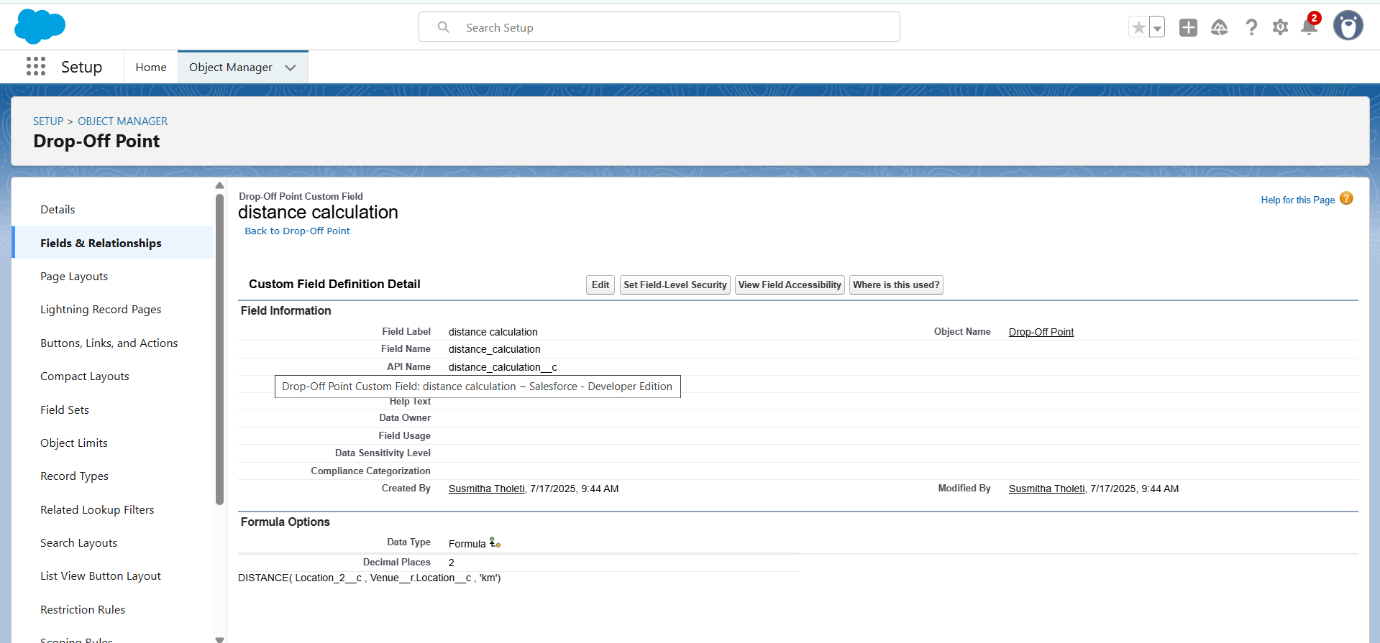
Location field:

1. Field Label: Location 2
2. Field Name: gets auto generated
3. Description: Enter the Geolocation of the Drop off Point
4. Geolocation Options: select Decimal
5. Decimal Places: 4



Distance calculation field:

1. Field Label: distance calculation
2. Field Name: distance\_calculation
3. Formula Return Type: Number
4. Formula Options: DISTANCE (Location\_2\_\_c ,  Venue\_\_r.Location\_\_c , 'km')

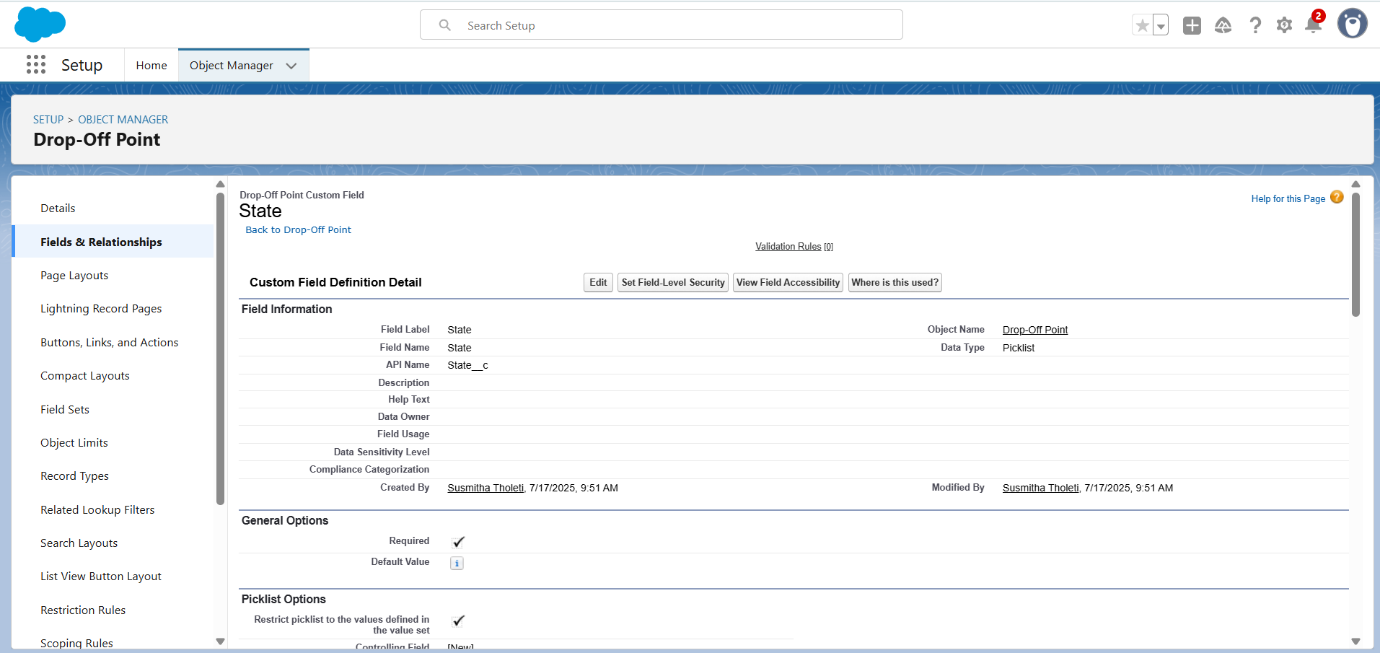


State field:

1. Field Label: State
2. Field Name: State
3. Enter values, with each value separated by a new line:

* Andhra Pradesh
* Arunachal Pradesh
* Assam
* Bihar
* Chhattisgarh
* Goa
* Gujarat
* Haryana
* Himachal Pradesh
* Jharkhand
* Karnataka
* Kerala
* Maharashtra
* Madhya Pradesh
* Manipur
* Meghalaya
* Mizoram
* Nagaland
* Odisha
* Punjab
* Rajasthan
* Sikkim
* Tamil Nadu
* Tripura
* Telangana
* Uttar Pradesh
* Uttarakhand
* West Bengal
* Andaman & Nicobar (UT)
* Chandigarh (UT)
* Dadra & Nagar Haveli and Daman & Diu (UT)
* Delhi [National Capital Territory (NCT)]
* Jammu & Kashmir (UT)
* Ladakh (UT)
* Lakshadweep (UT)
* Puducherry (UT)

1. Click on required check box



Distance field:

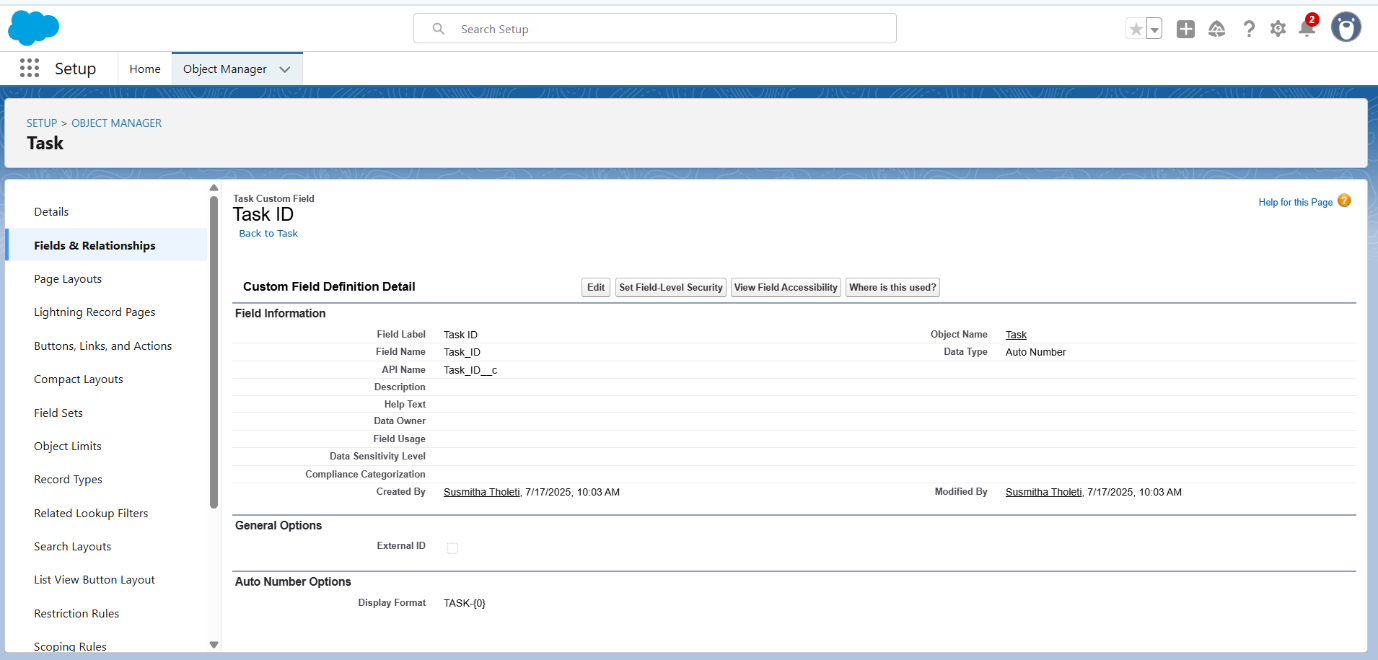
1. Field Label: Distance
2. Field Name: Distance
3. Length: 14
4. Decimal Places: 4
5. Click on required check box



**Creation of fields for the Task object:**

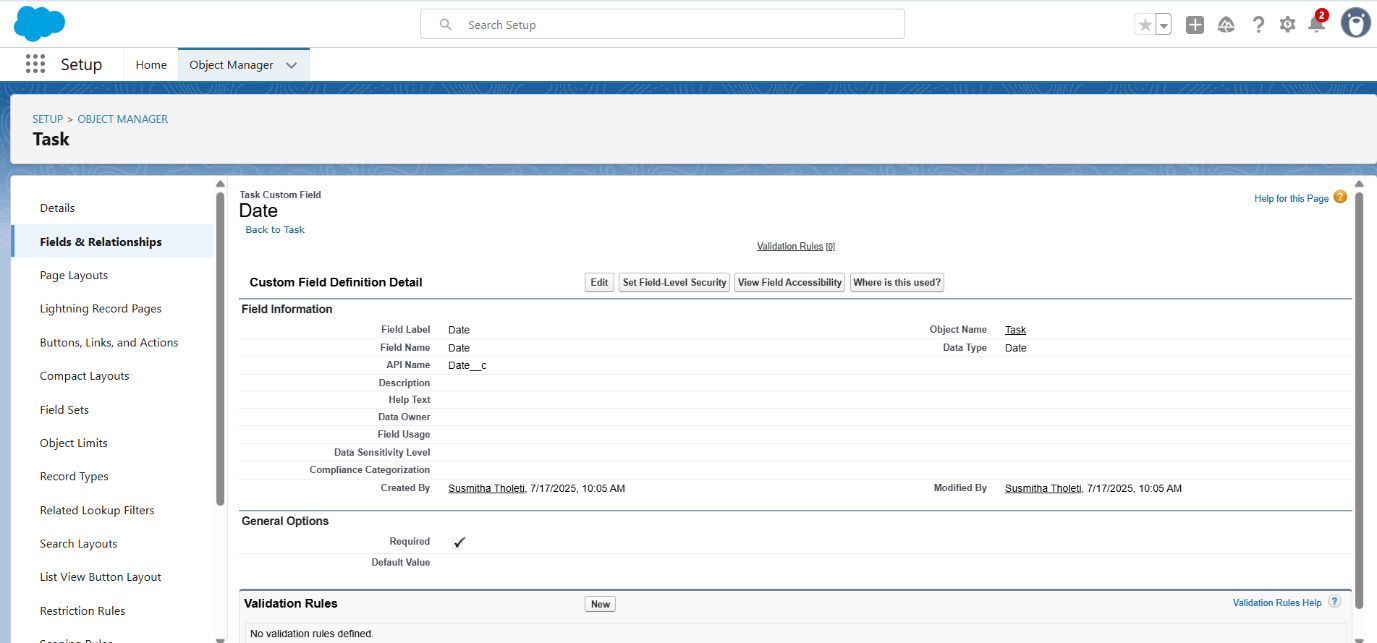
Task ID field:

1. Field Label: Task ID
2. Display Format: TASK-{0}
3. Starting Number: 1
4. Field Name: gets auto generated
5. Click on required check box



Date field:

* Field Label: Date
* Field Name: Date
* Click on required check box

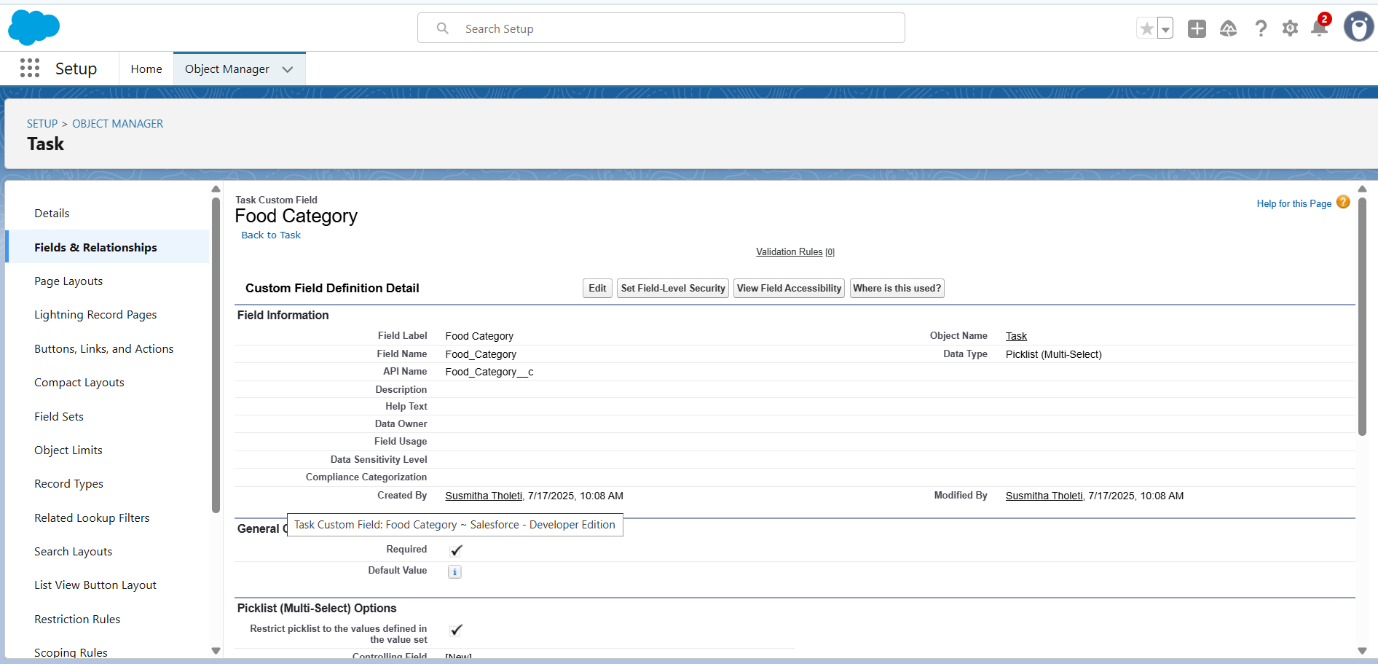


Food Category field:

1. Field Label: Food Category
2. Field Name: Food Category
3. Enter values, with each value separated by a new line:

* Veg
* Non-Veg
* Salad
* Snack

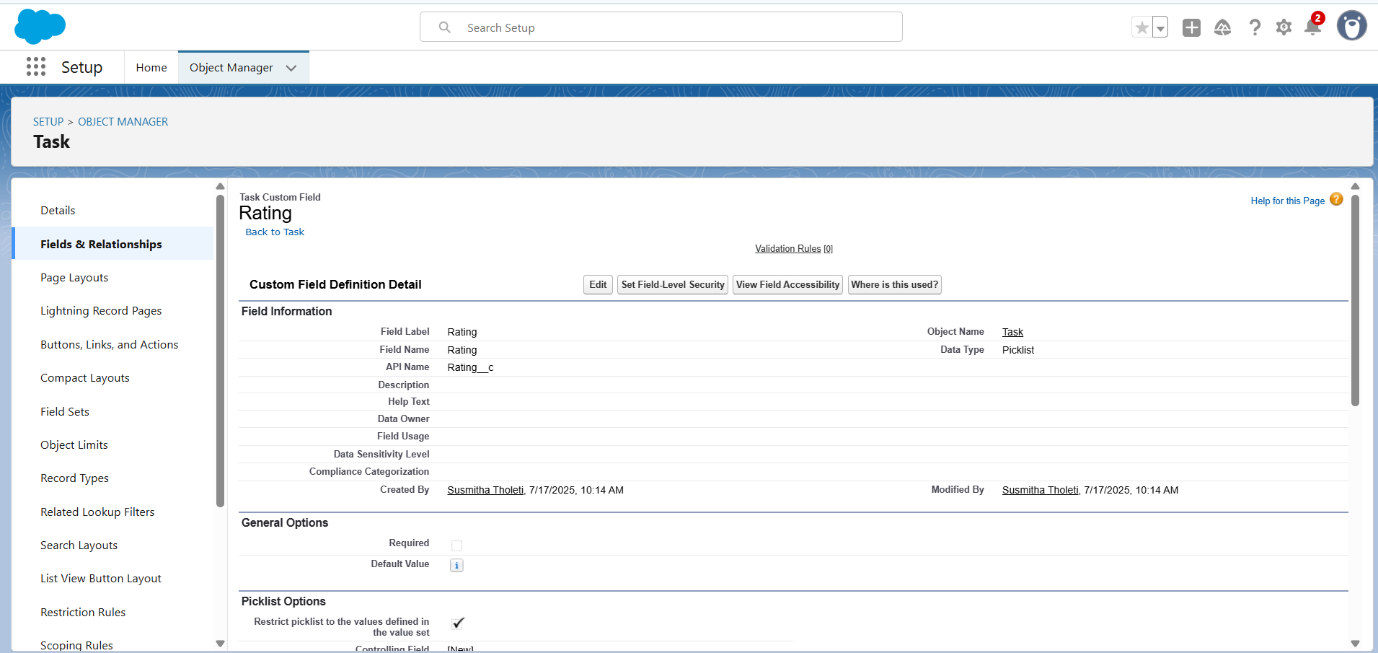
1. Click on required check box



Rating field:

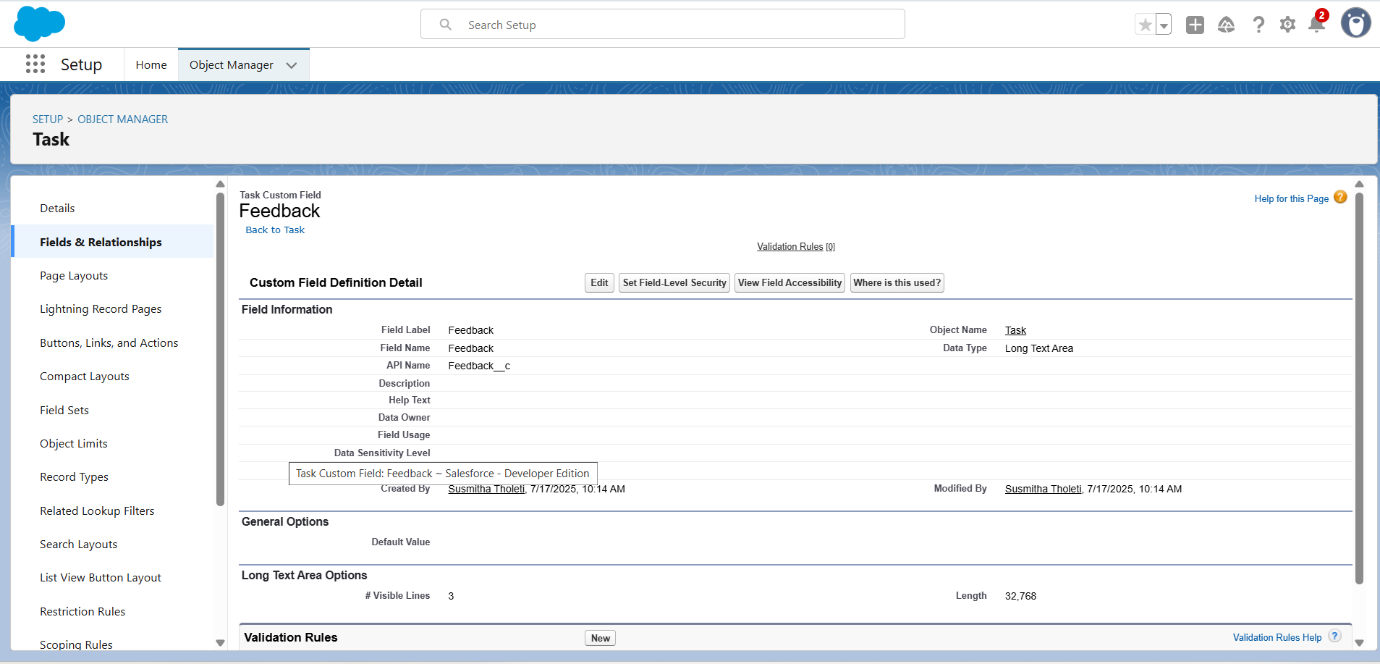
1. Field Label: Rating
2. Field Name: Rating
3. Enter values, with each value separated by a new line:

* 1
* 2
* 3
* 4
* 5



Feedback field:

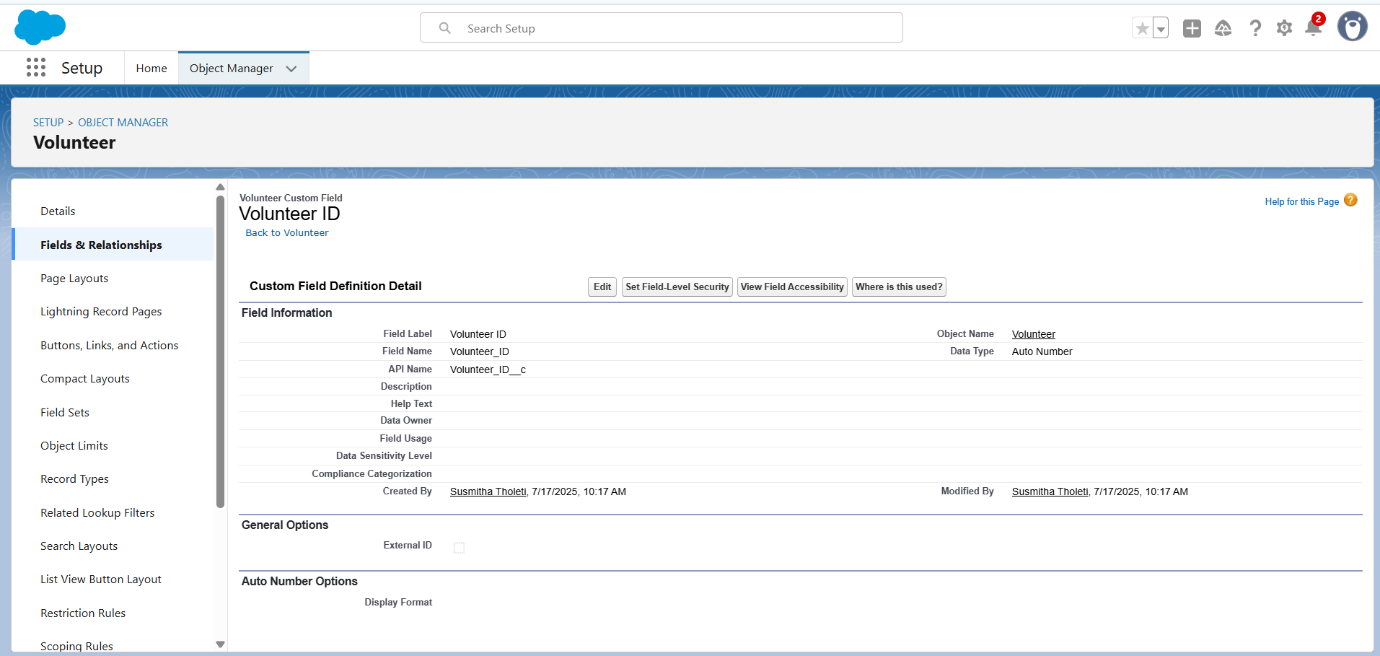
1. Data Type: long text area
2. Field Label: Feedback
3. Field Name: Feedback



**Creation of fields for the Volunteer object:**

Volunteer ID field:

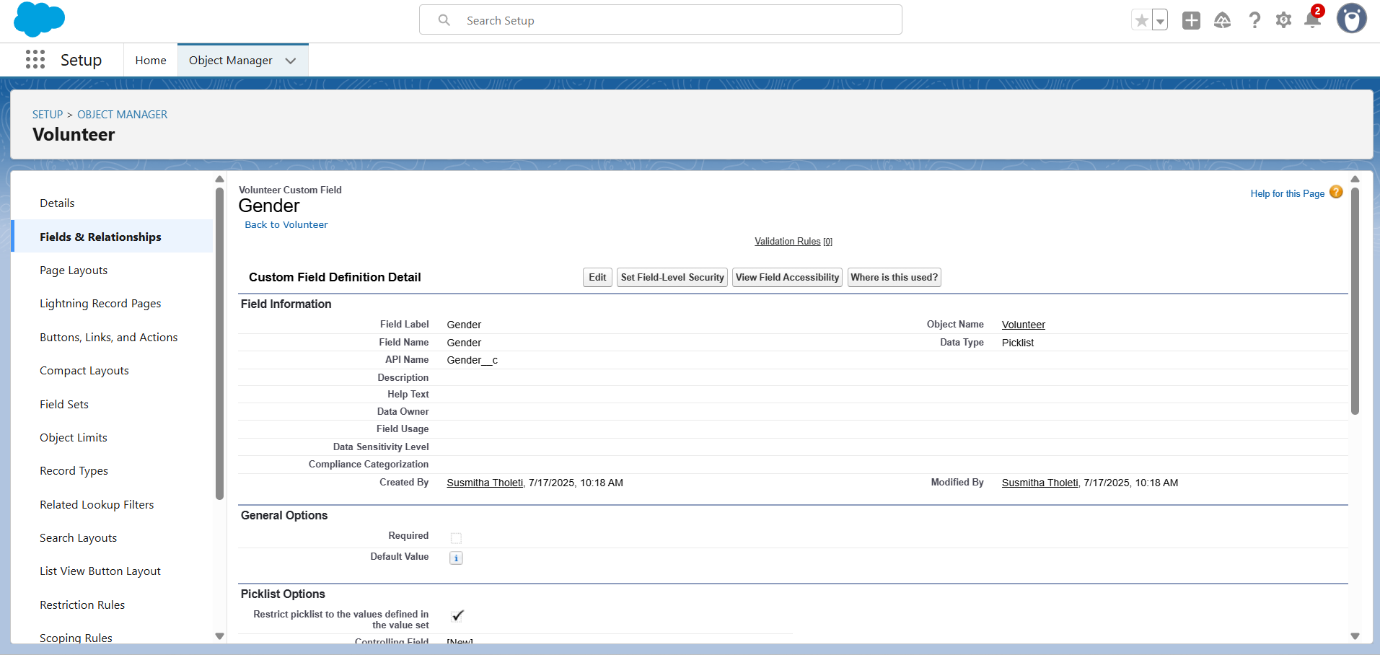
1. Field Label: Volunteer ID
2. Field Name: gets auto generated
3. Click on required check box



Gender field:

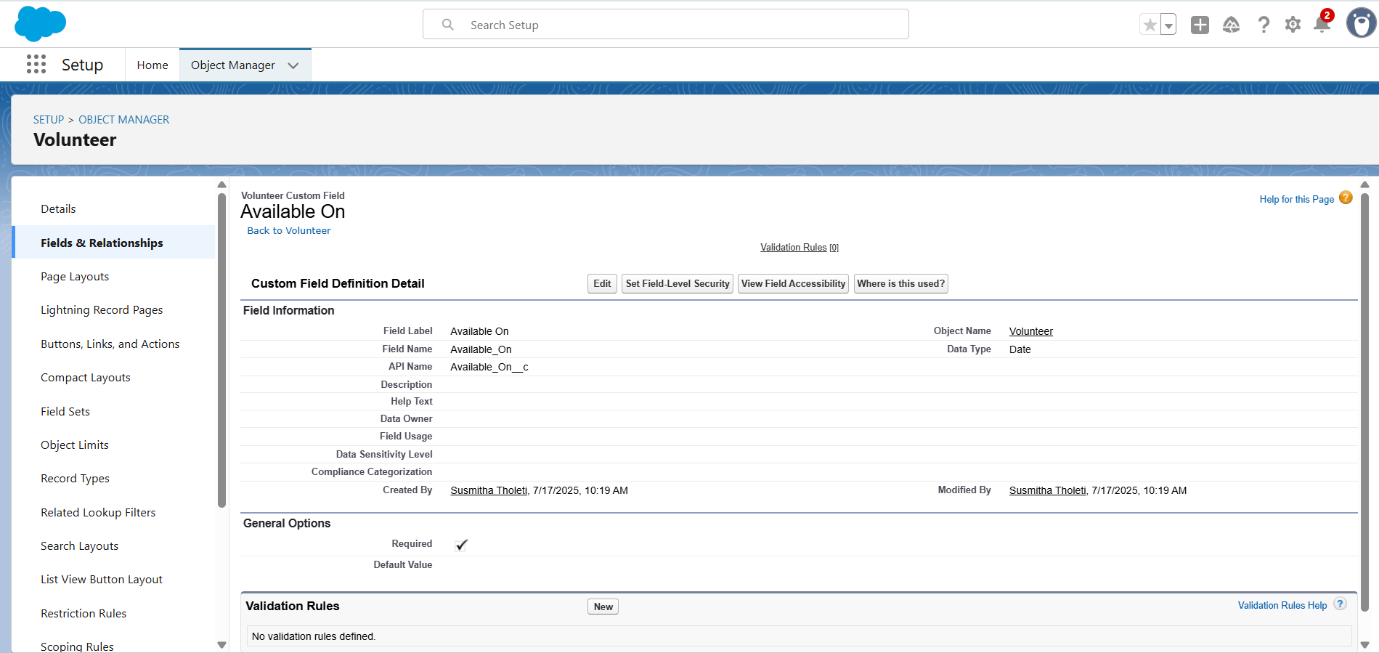
1. Field Label: Gender
2. Field Name: Gender
3. Enter values, with each value separated by a new line:

* Female
* Male



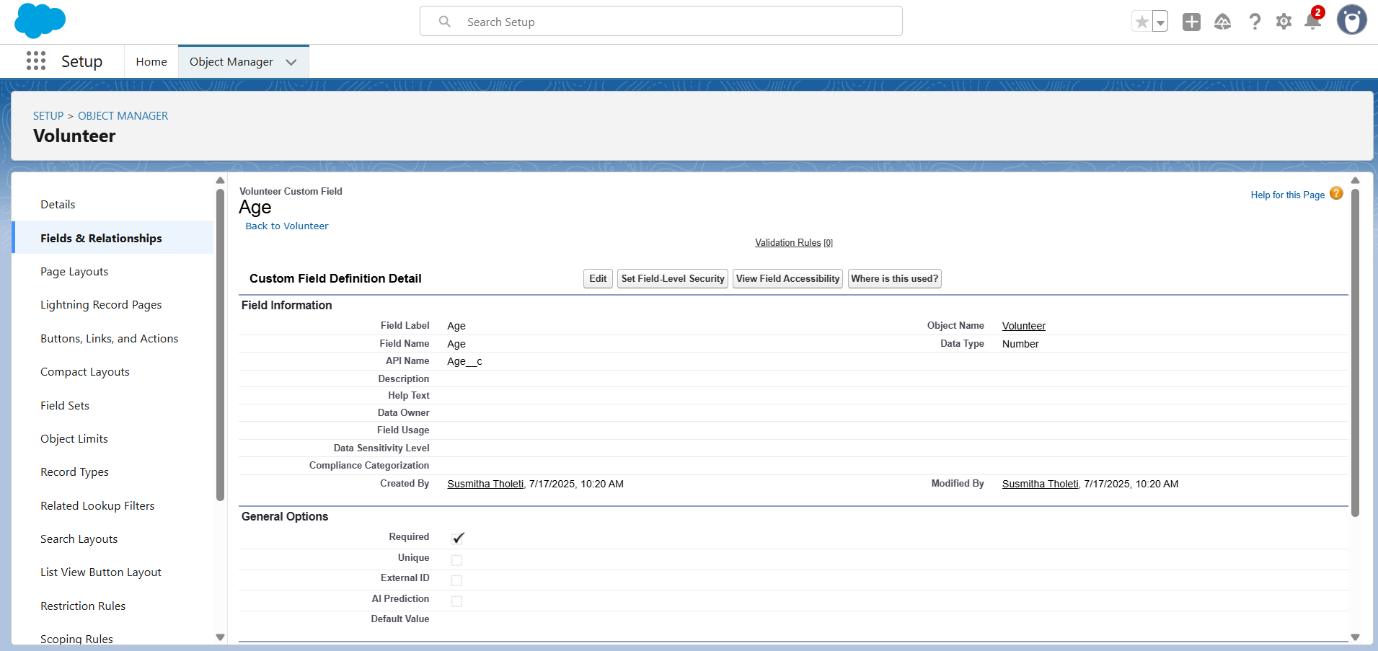
Available On field:

1. Field Label: Available On
2. Field Name: Available On
3. Click on required check box



Age field:

1. Field Label: Age
2. Field Name: Age
3. Click on required check box



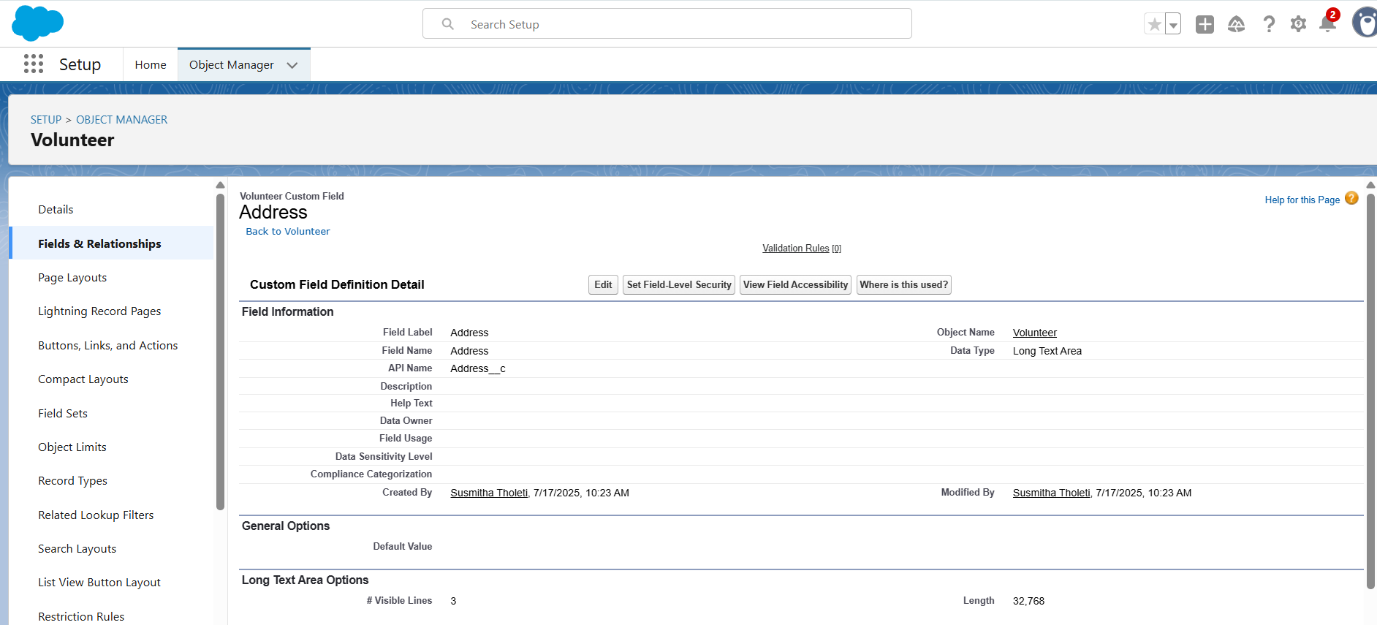
Contact label:

1. Field Label: Contact Number
2. Field Name: Contact\_Number
3. Click on required check box



Address field:

1. Data Type: Text Area (Long)
2. Field Label: Address
3. Field Name: Address



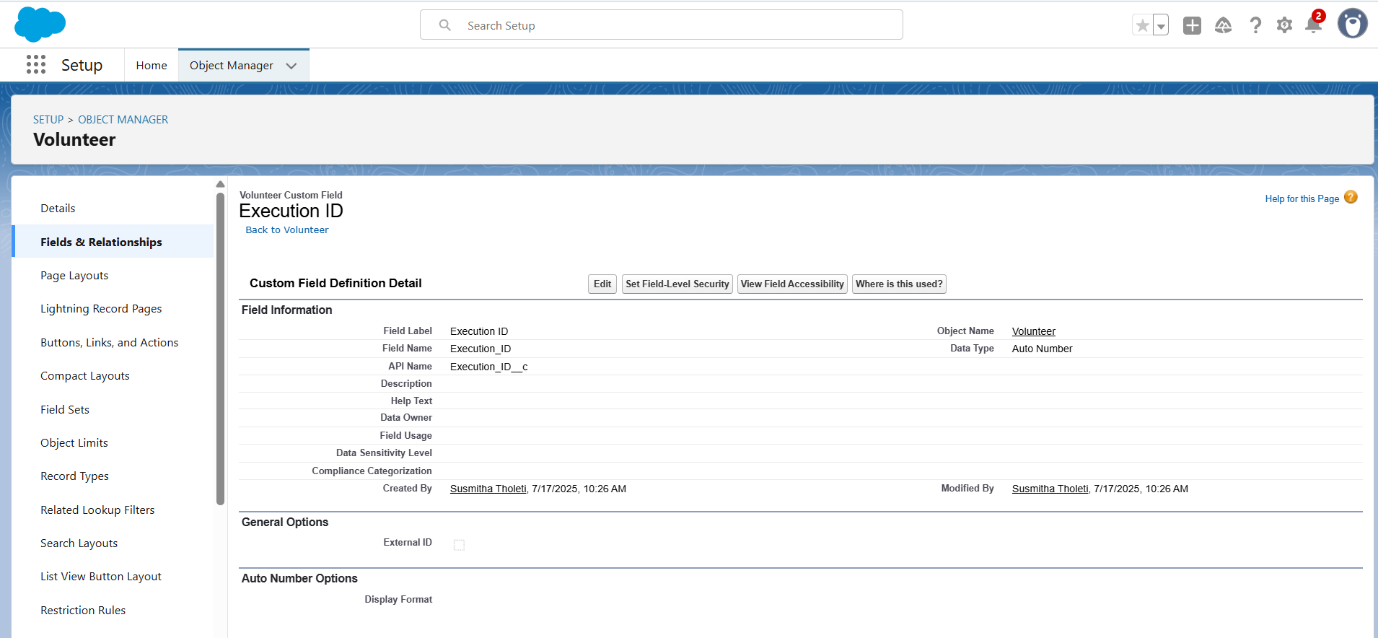
DOB field:

1. Data Type: Date
2. Field Label: Date of Birth
3. Field Name: Date\_of\_Birth



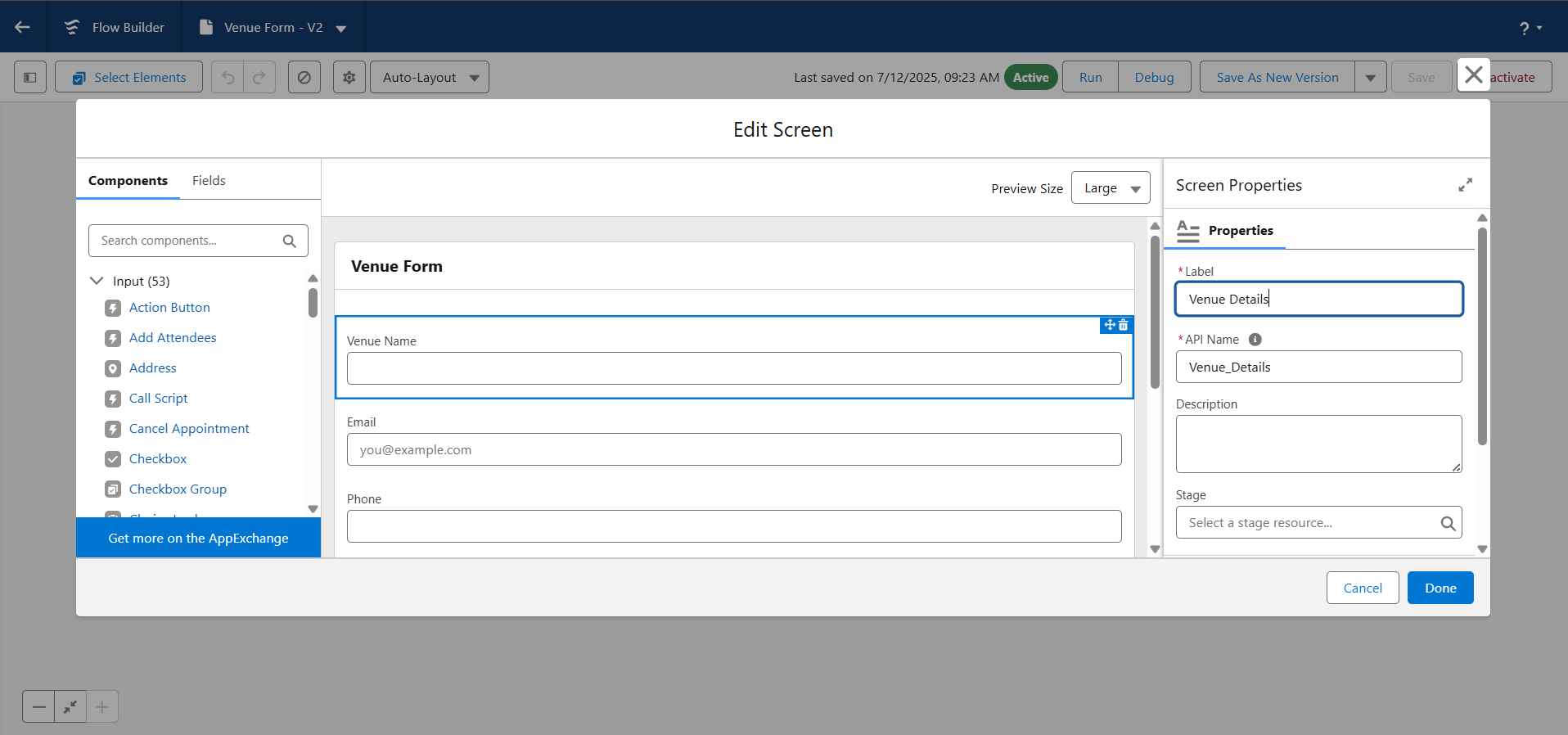
**Creation of fields for the Execution Details object:**

* Field Label: Execution ID
* Field Name: gets auto generated
* Click on required check box



**Create Flow to create a record in Venue object:**

1. Select the Screen flow. Click on create.
2. Under the Screen Properties:
   1. Label: Venue Details
   2. API Name: Venue\_Details
3. Now let’s add components in this flow. Click on Text Component and name it as:
   1. Label: Venue Name
   2. API Name: Venue\_Name
4. Click on Email Component and name it as:
   1. Label: Email
   2. API Name: Contact\_Email
5. Click on Phone Component and name it as:
   1. Label: Phone
   2. API Name: Contact\_Phone
6. Click on Text Component and name it as:
   1. Label: Venue Location
   2. API Name: Venue\_Location
7. Click on Number Component and name it as:
   1. Label: Latitude
   2. API Name: Latitude
8. Click on Number Component and name it as:
   1. Label: longitude
   2. API Name: longitude



1. Now label it as:

Label: Create Venue Record

API Name: Create\_Venue\_Record

How Many Records to Create: One

How to Set the Record Fields: Use separate resources, and literal values

Object: Venue

Set Field Values for the Venue: Click on ‘Add Field’ 5 times

Field: Value = Contact\_Email\_\_c: {!Contact\_Email.value}

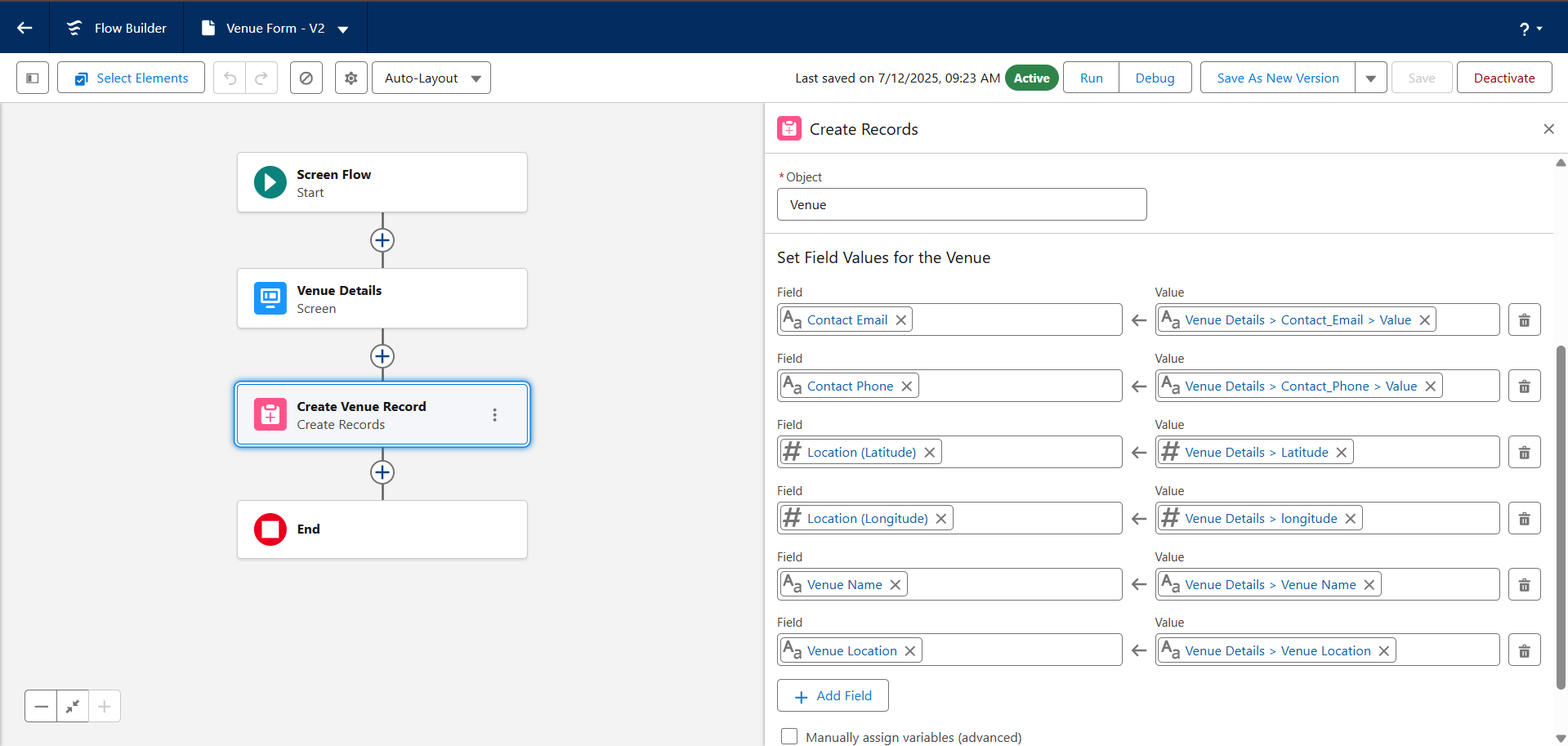
Field: Value = Contact\_Phone\_\_c: {!Contact\_Phone.value}

Field: Value = Name: {!Venue\_Name}

Field: Value = Venue\_Location\_\_c: {!location}

Field: Value = Location\_\_Latitude\_\_s: {!latitude}

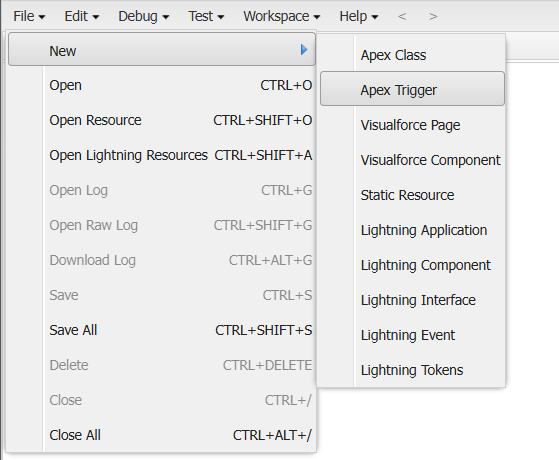
Field: Value = Location\_\_Longitude\_\_s: {!longitude}

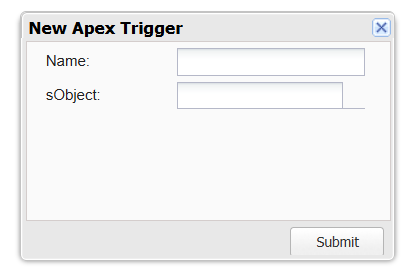


1. Click on Save as:
2. Flow Label: Venue Form
3. Flow API Name: Venue\_Form

**Create a Trigger:**

1. Click on developer console and you will be navigated to a new console window.
2. Click on the File menu in the toolbar, and click on new >> Trigger.
3. Enter the trigger name and the object to be triggered.



 Enter Name: DropOffTrigger

sObject: Drop-Off Point

Click on Submit.

**Trigger Code:**

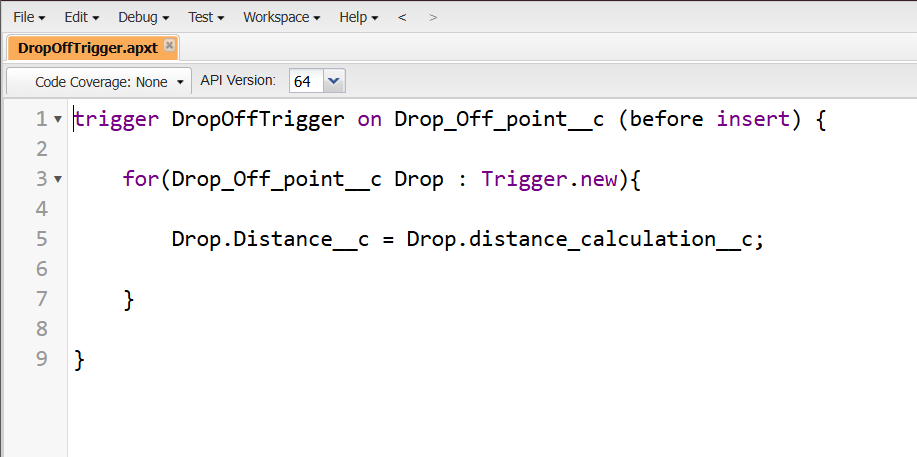
trigger DropOffTrigger on Drop\_Off\_point\_\_c (before insert) {

    for(Drop\_Off\_point\_\_c Drop : Trigger.new){

        Drop.Distance\_\_c = Drop.distance\_calculation\_\_c;

    }

}



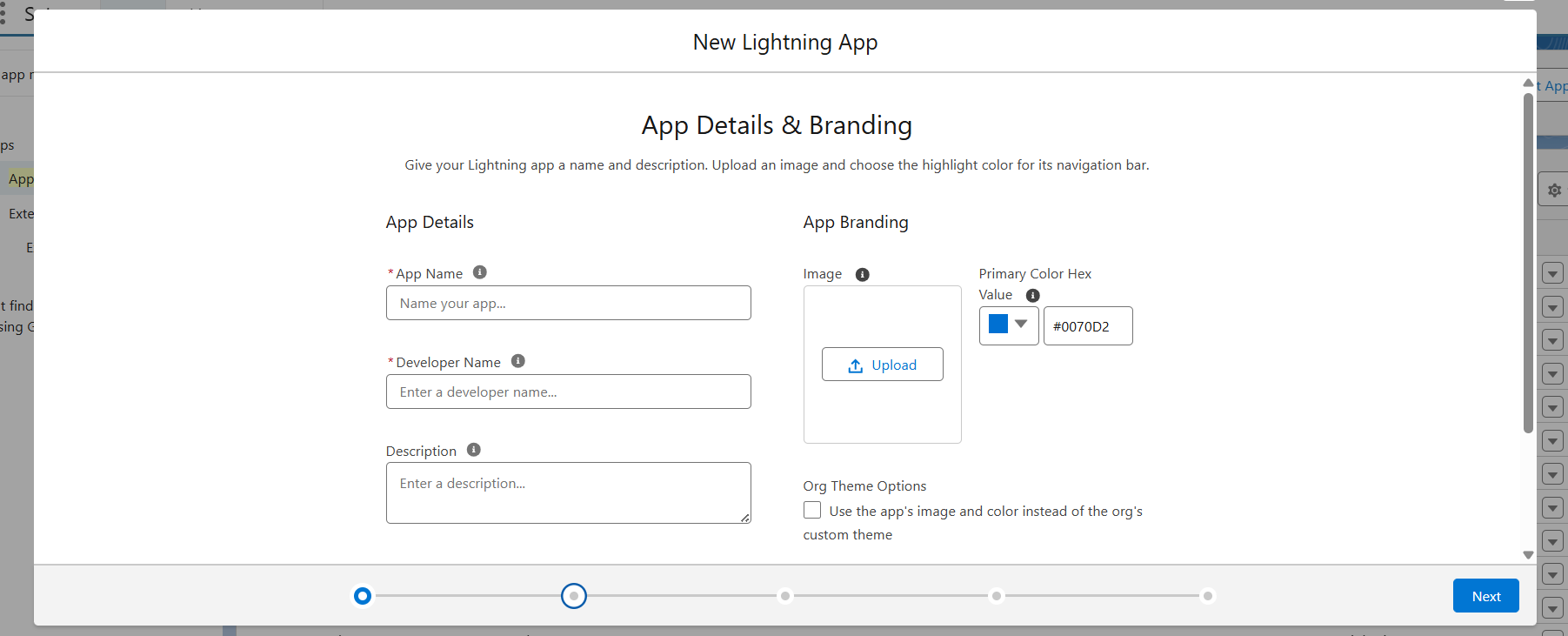
Phase 3:UI/UX Development and customization:

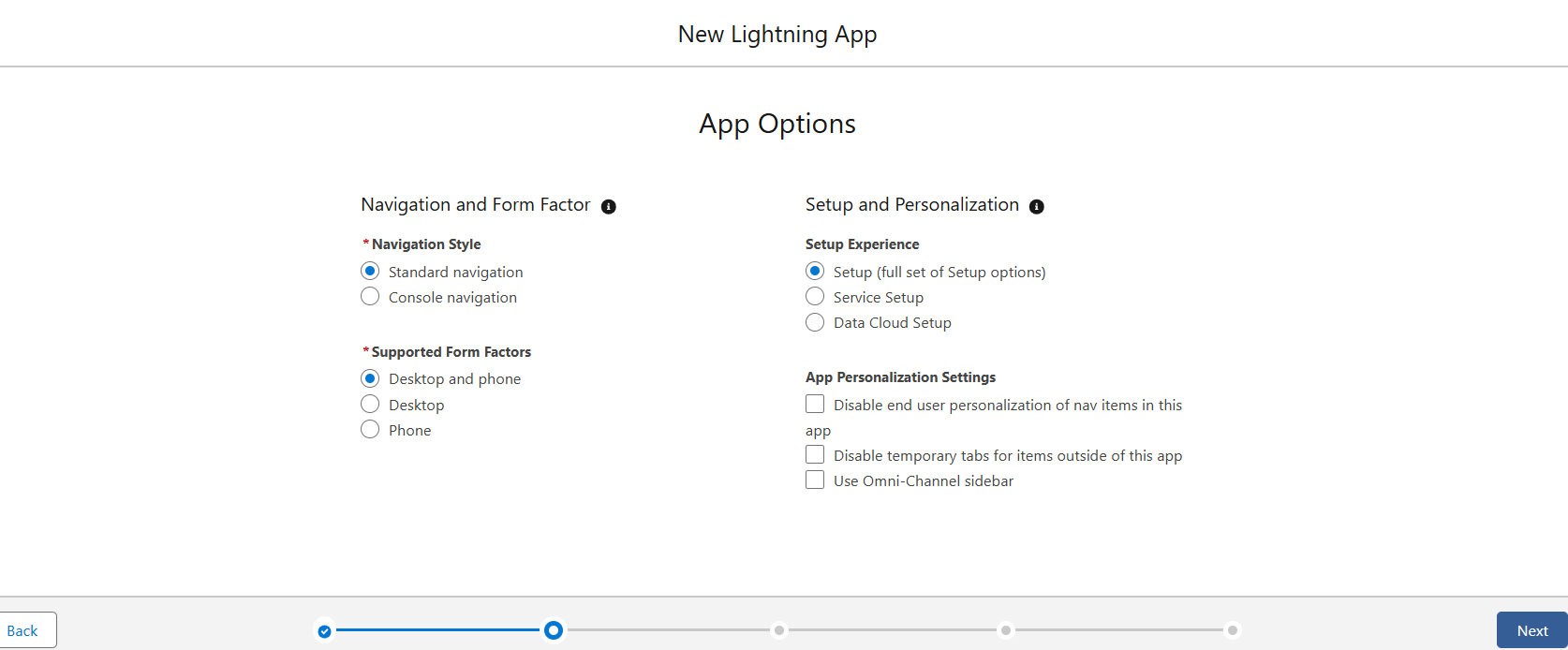
To create a lightning app page:

Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

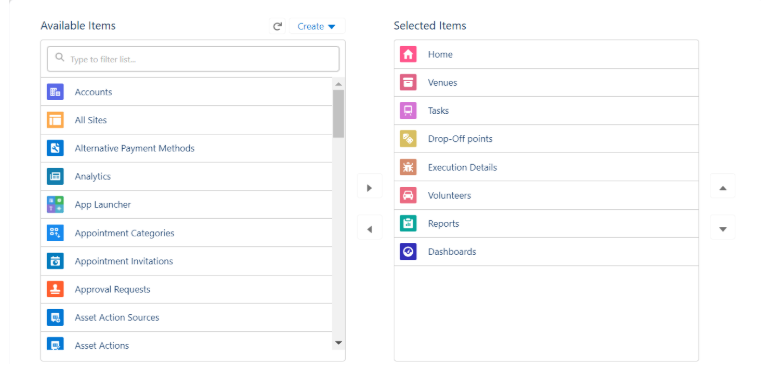
Fill the app name in app details and branding as follow  
App Name : FoodConnect  
Developer Name : This will auto populated  
Image : optional (if you want to give any image you can otherwise not mandatory) Primary color hex value : keep this default.

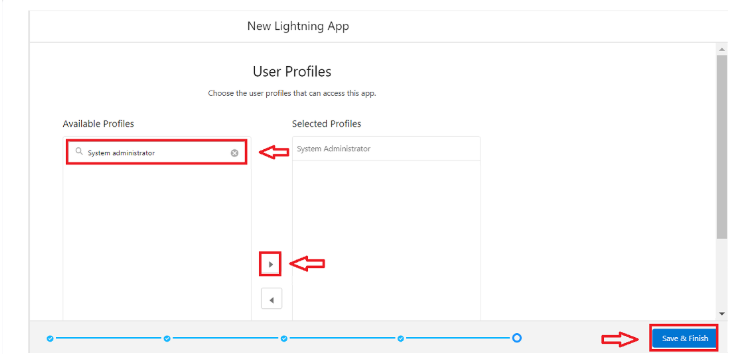
.Then click Next  >> (App option page)Set Navigation Style as Standard Navigation >> Next.



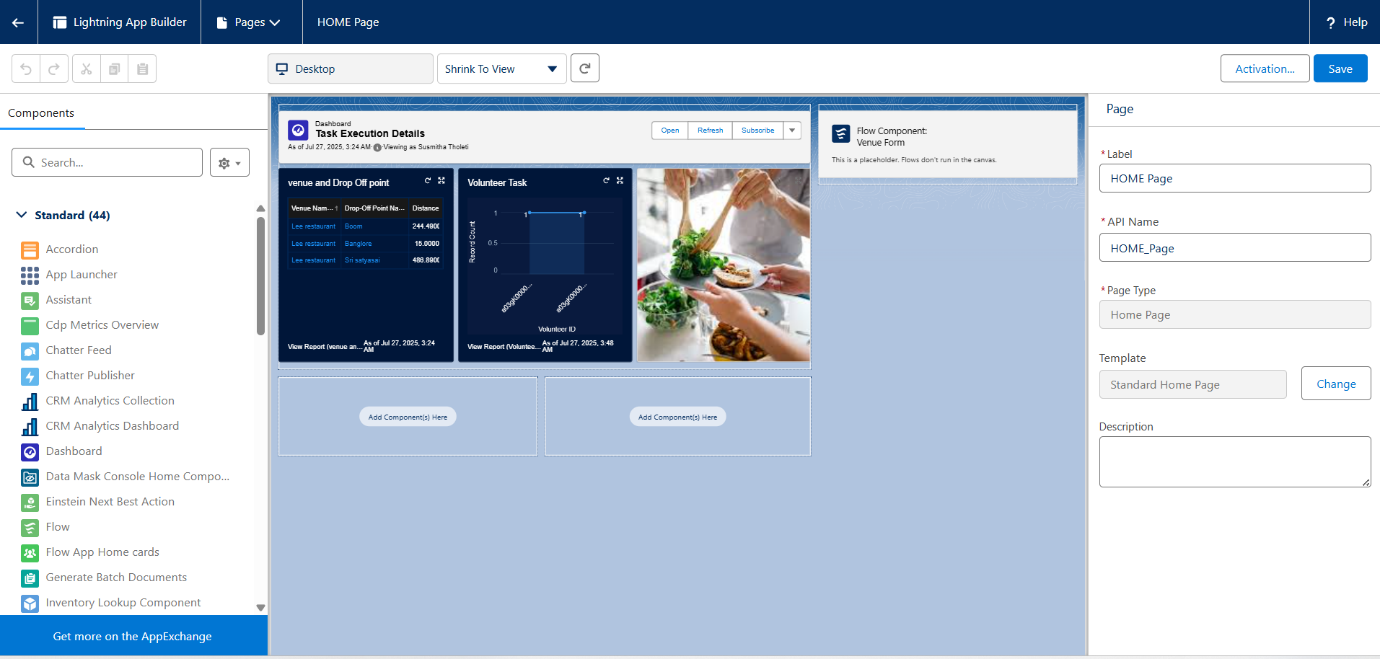


Search for the item in the (Home, Venue, Drop-Off Point, Task, Volunteer, Execution Details, Reports) from the search bar and move it using the arrow button >> Next >> Next.





Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



User Management:

1. Go to setup page >> type users in Quick Find bar >> click on users>> New user.
2. In General Information give details as: (Note: create users as per your wish NGO’s)

First Name: Iksha Foundation

Last Name: Iksha\_Foundation

Alias: iiksh

Email: Give Your Email

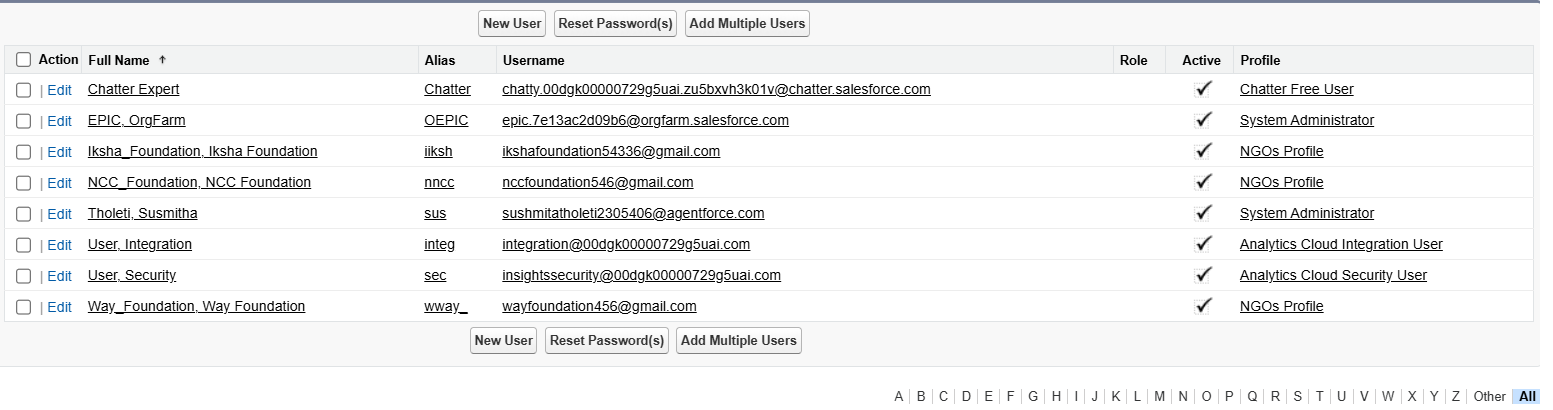
Username: [ikshafoundation@sb.com](mailto:ikshafoundation@sb.com) (give the username different)

Nickname: Auto Populated

User License: Salesforce Platform

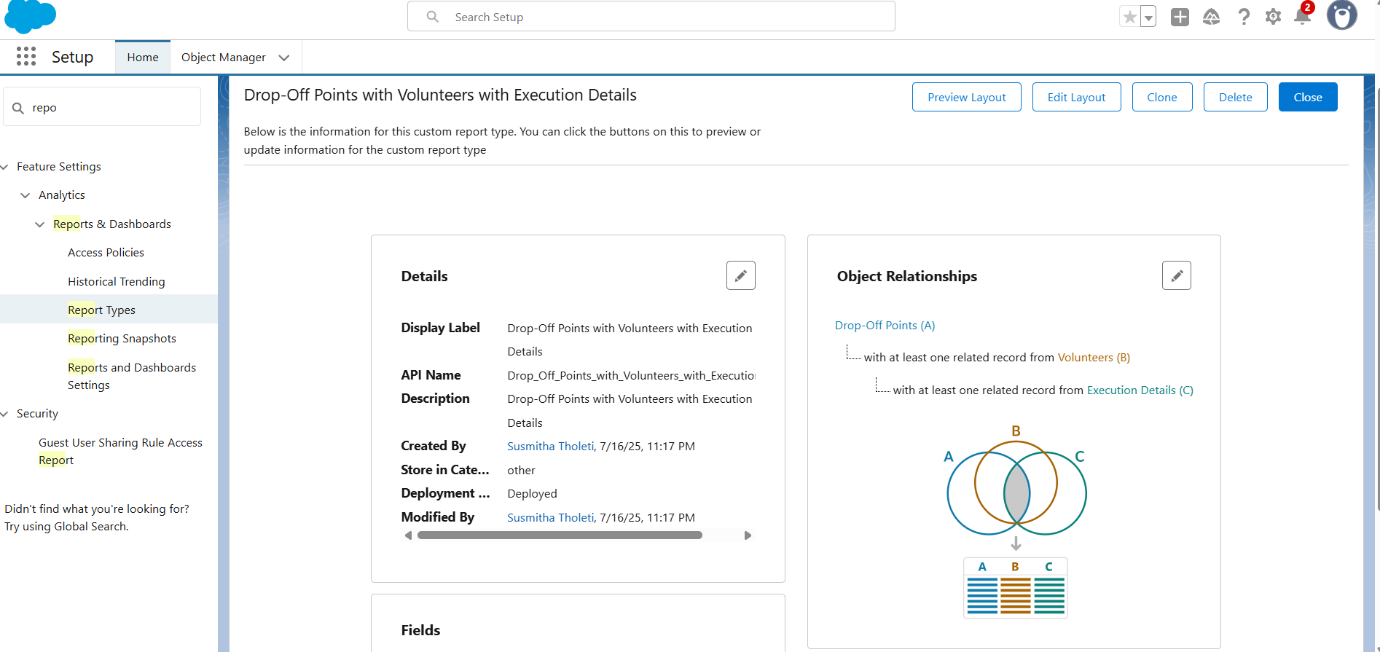
Profile: NGOs Profile

Active: Check



Reports:

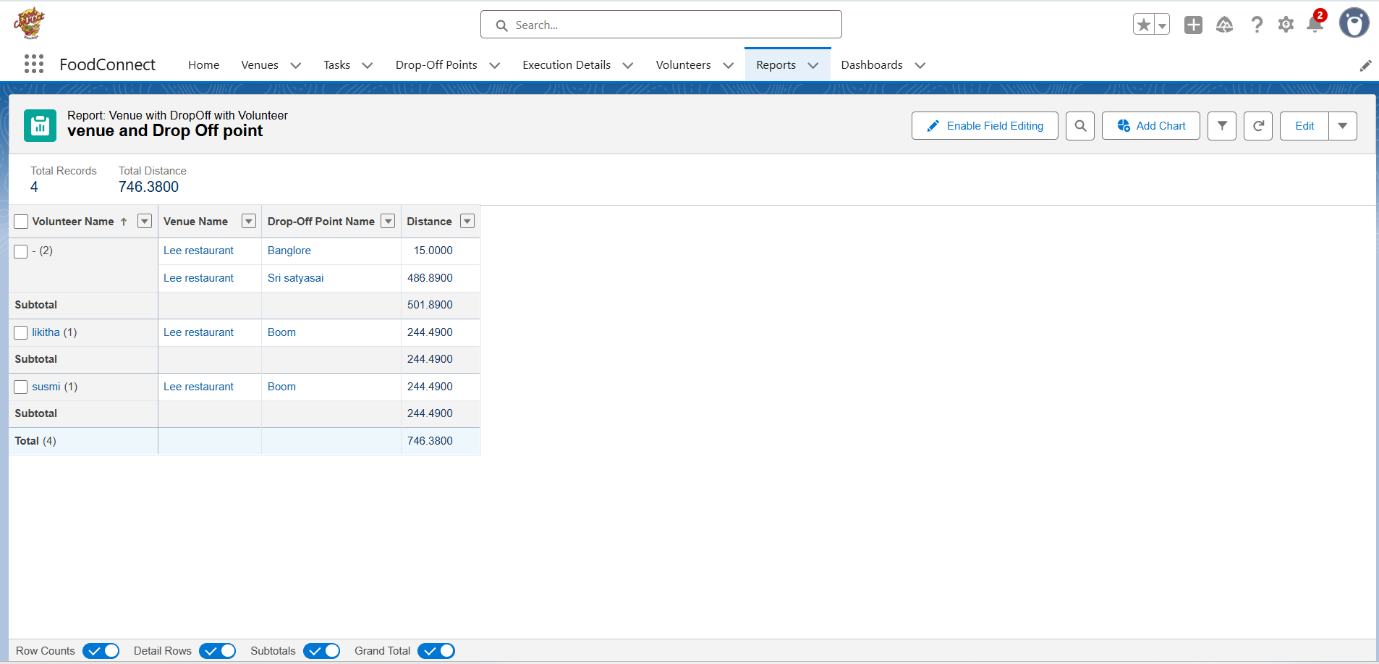
1. Go to setup page  >>  type Report Types in Quick Find bar  >> click on Report Types  >>  click on Continue   >>  Click on New Custom Report Type.
2. In Define the Custom Report Type:
   1. Primary Object: Select Venues
   2. Report Type Label: Venue with DropOff with Volunteer
   3. Report Type Name: Venue\_with\_DropOff\_with\_Volunteer
   4. Description: Venue with DropOff with Volunteer
   5. Store in Category: Select Other Reports
   6. Deployment Status: Deployed
3. Click on Next
4. Near Click to relate another Object Select Drop-Off Points.
5. And also select "A" records may or may not have related "B" records.
6. Now again Near Click to relate another Object Select Volunteers.
7. Now click on Save.



Creation of report:

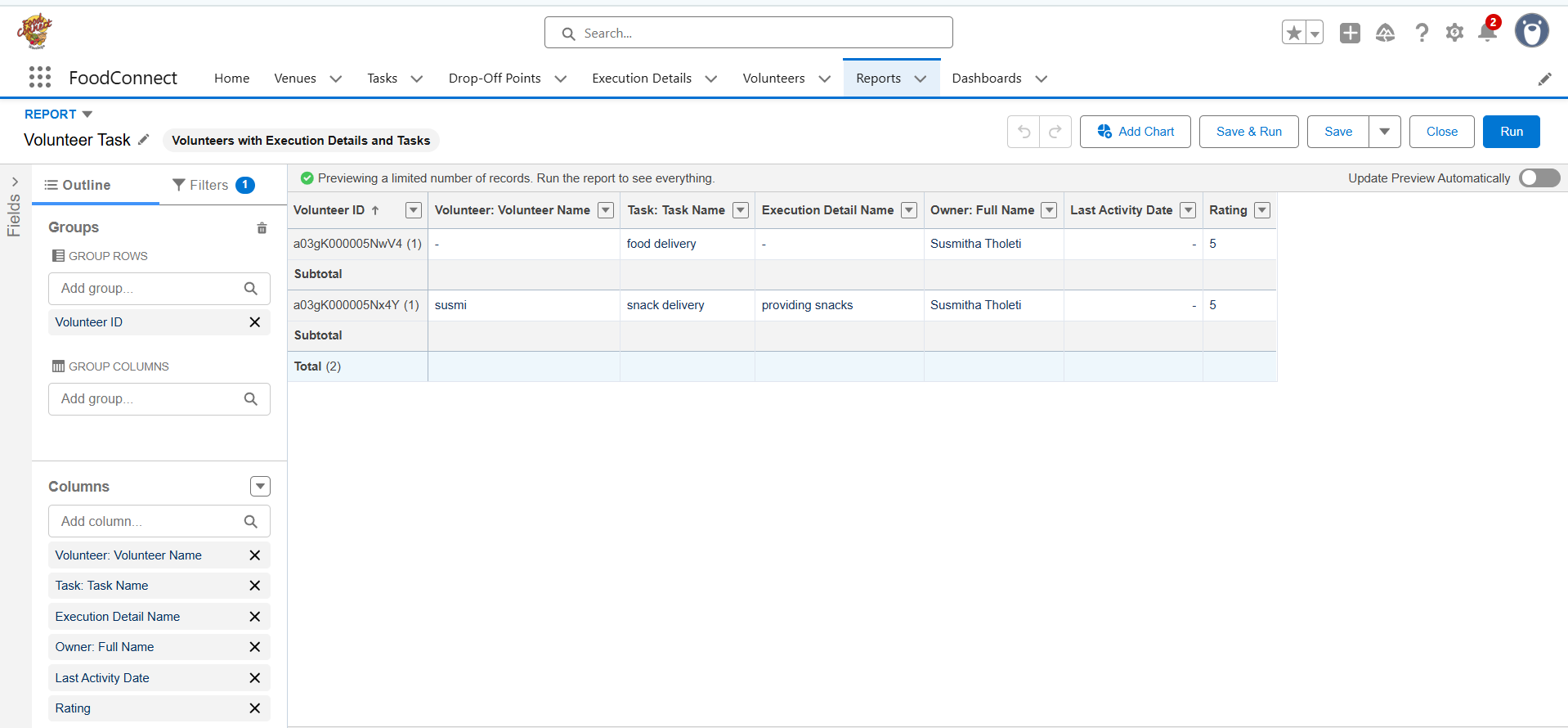
Creation of Report on Venue with DropOff with Volunteer:

1. Go to the app(FoodConnect)  >>  click on the reports tab
2. Click on New Folder.
3. Folder Label: Custom Reports
4. Folder Unique Name: CustomReports
5. Open Custom Reports and click on New Report
6. Select Report Type: Venue with DropOff with Volunteer
7. Then click on Start Report.
8. In GROUP ROWS: Add Volunteer Name
9. In Columns: Add Venue Name, Drop-Off point Name, Distance.
10. Now click on Save & Run.
11. Give Label as:
12. Report Name: venue and Drop Off point
13. Report Unique Name: Auto Populated
14. Click on Select Folder and select Custom Report, then click on Save.



**Creation of Report on Volunteers with Execution Details and Tasks:**

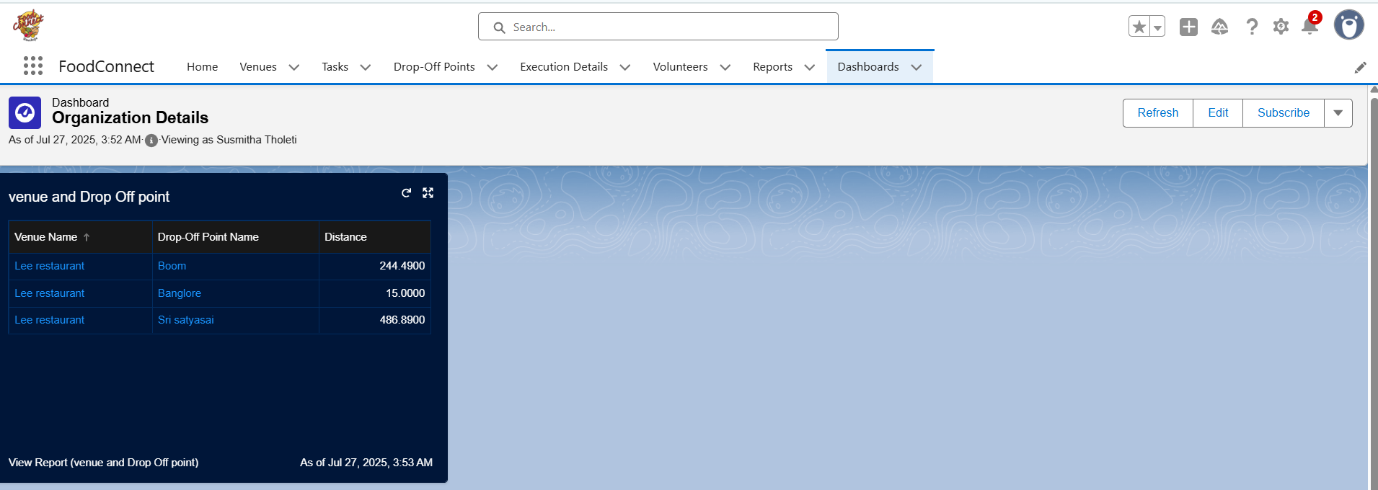
1. Go to the app(FoodConnect)  >>  click on the reports tab
2. Click on Custom Reports Folder and click on New Report
3. Select Report Type: Volunteers with Execution Details and Tasks.
4. Then click on Start Report.
5. In GROUP ROWS: Volunteer ID
6. In Columns: Add Volunteer: Volunteer Name, Task: Task Name, Execution Detail: Execution Detail Name, Volunteer: Owner Name, Task: Date, Task: Rating.
7. Now click on Save & Run.
8. Give Label as:
9. Report Name: Volunteer Task
10. Report Unique Name: Auto Populated
11. Click on Select Folder and select Custom Report, then click on Save.



Dashboards:

Adding venue and Drop Off point Report to the Dashboard:

1. Go to the app(FoodConnect)  >>  click on the Dashboards tab.
2. Click on New Folder.
3. Folder Label : Custom Dashboards
4. Folder Unique Name : Auto Populated
5. Open Custom Dashboards and click on New Dashboards
6. Name : Organization Details
7. Click on Widget and select Chart or Table
8. In Select Report : Select venue and Drop Off point Report.
9. Then click on select
10. In Add Component:
11. Display As : Select Lightning Table

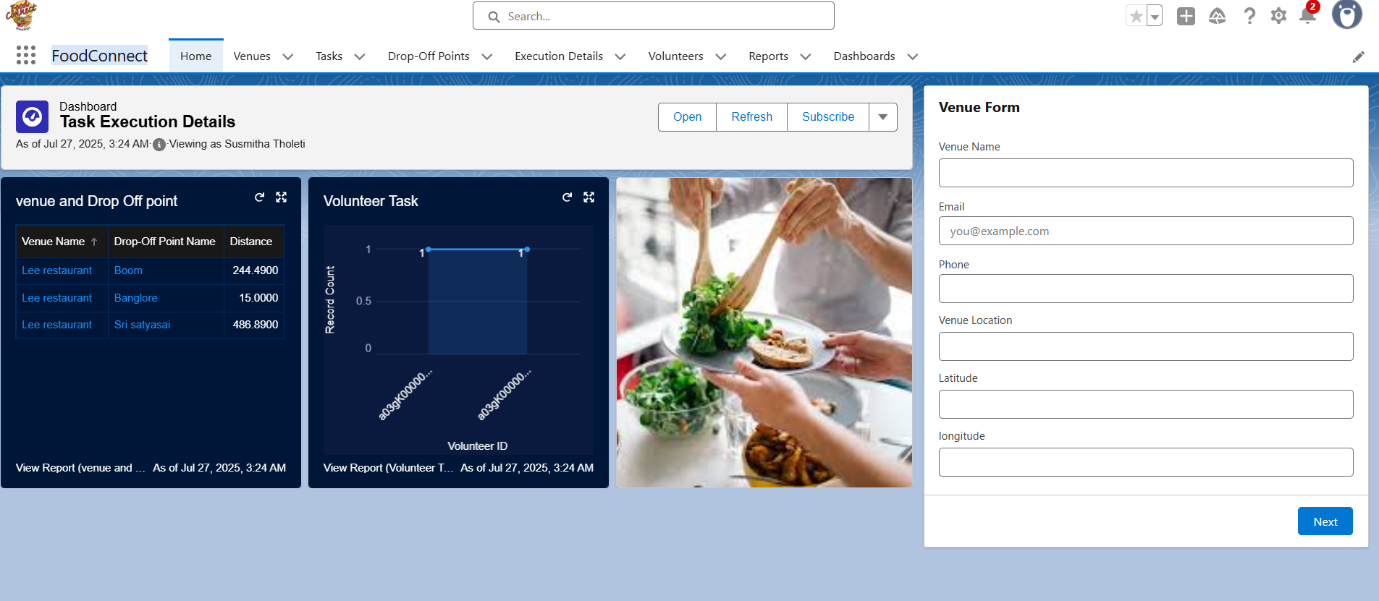


Adding Volunteer Task Report to the Dashboard:

1. Click on Widget and select Chart or Table
2. In Select Report : Select Volunteer Task Report.
3. Then click on select
4. In Add Component:
5. Display As : Select Line Chart



1. Now click on save.



Profiles, Role, Role Hierarchy, Permission Sets, Sharing Rules:

In a CRM system, users are individuals who access the platform, each assigned a profile that defines what they can see and do—such as object-level permissions, field visibility, and app access. Roles establish a user's position within the role hierarchy, which determines data access—higher roles inherit access from lower ones, enabling visibility across teams. Sharing rules extend access beyond role-based settings, allowing broader collaboration by automatically sharing records with groups or roles based on criteria. Permission sets provide extra privileges layered onto profiles for flexibility, letting admins tailor access without altering base settings. Together, these elements form a secure and scalable access model that balances control and collaboration across an organization.

Profiles:

1. Go to setup page >> type Profiles in Quick Find bar >> click on Profiles >> click on ‘S’
2. Click on Clone beside Standard Platform User.
3. Under Clone Profile:

Profile Name: NGOs Profile

1. Then click on Save.

Creation of sharing rules:

1. Go to setup >> type Sharing Settings in quick find box >>  Click on the Sharing Settings.
2. Scroll down and find Drop-Off point Sharing Rules.
3. Click on new near Drop-Off point Sharing Rules and Name it as:

Label: Rule 1

Rule Name: Rule\_1

1. Select your rule type: Select Based on criteria.
2. Select which records to be shared:

Field: Operator: Value = Distance: less than: 15

1. Select the users to share with: Near Share With

Public Groups: Iksha

1. Click on Save.
2. Click on new near Drop-Off point Sharing Rules and Name it as:

Label: Rule 2

Rule Name: Rule\_2

1. Select your rule type: Select Based on criteria.
2. Select which records to be shared:

Field: Operator: Value = Distance: greater than: 15

Field: Operator: Value = Distance: less or equal: 30

1. Select the users to share with: Near Share With

Public Groups: NCC

1. Click on Save.
2. Click on new near Drop-Off point Sharing Rules and Name it as:

Label: Rule 3

Rule Name: Rule\_3

1. Select your rule type: Select Based on criteria.
2. Select which records to be shared:

Field: Operator: Value = Distance: greater than: 30

Field: Operator: Value = Distance: less or equal: 50

1. Select the users to share with: Near Share With

Public Groups: Way

1. Click on Save.

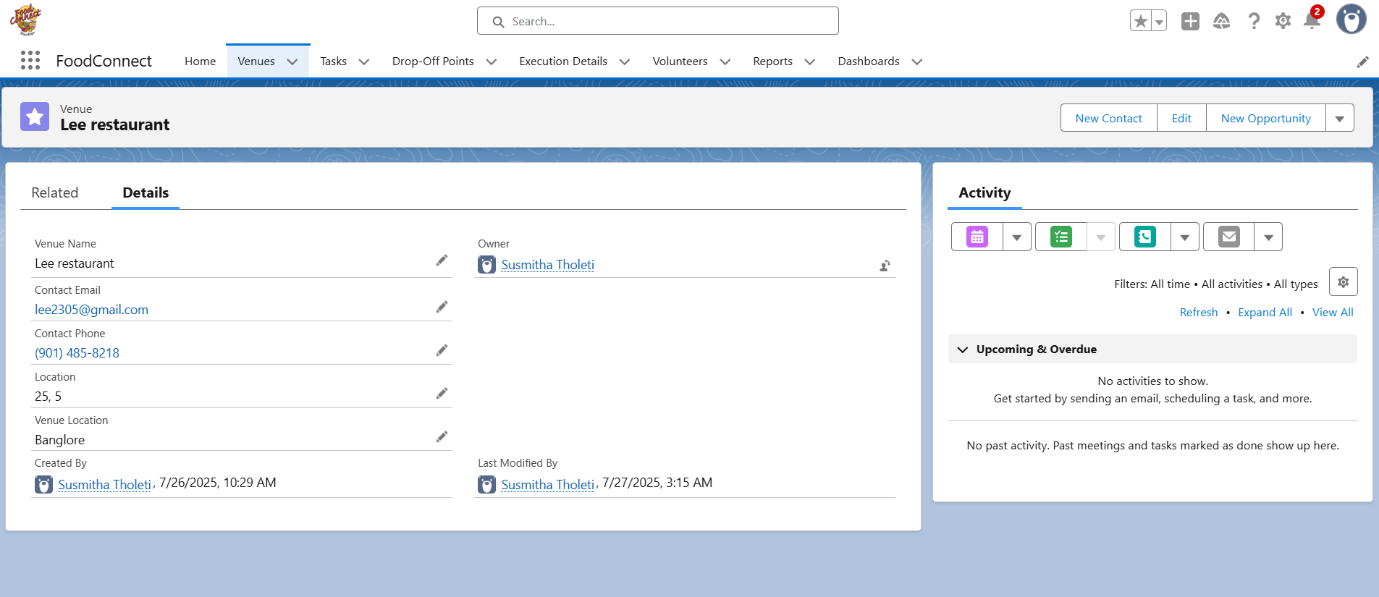
**Phase 5: Deployment, Documentation & Maintenance**

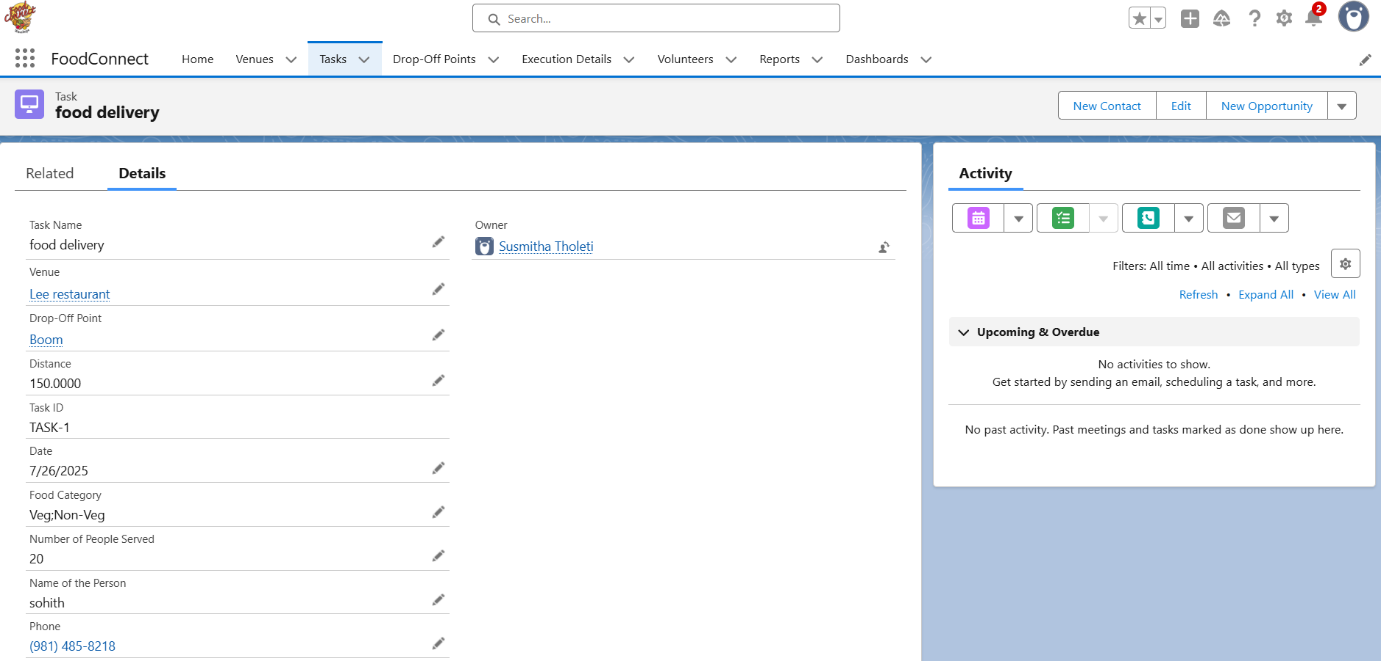
**Deployment Strategy:**

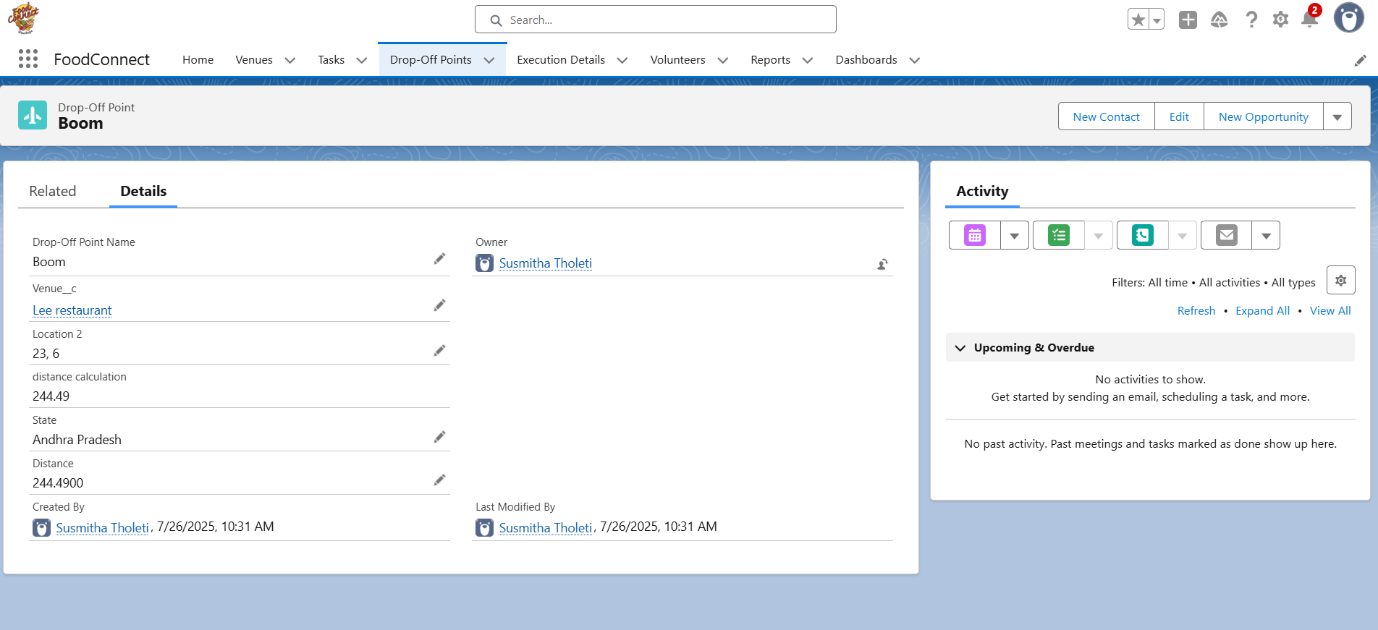
Change Sets were used to deploy the **FOODCONNECT CRM** from the sandbox environment to production. This allowed for the safe and systematic migration of metadata components, such as custom objects, fields, validation rules, flows, and Apex code. By ensuring that all dependencies were appropriately packed and tested prior to final deployment, this approach reduced the possibility of live environment disturbances.

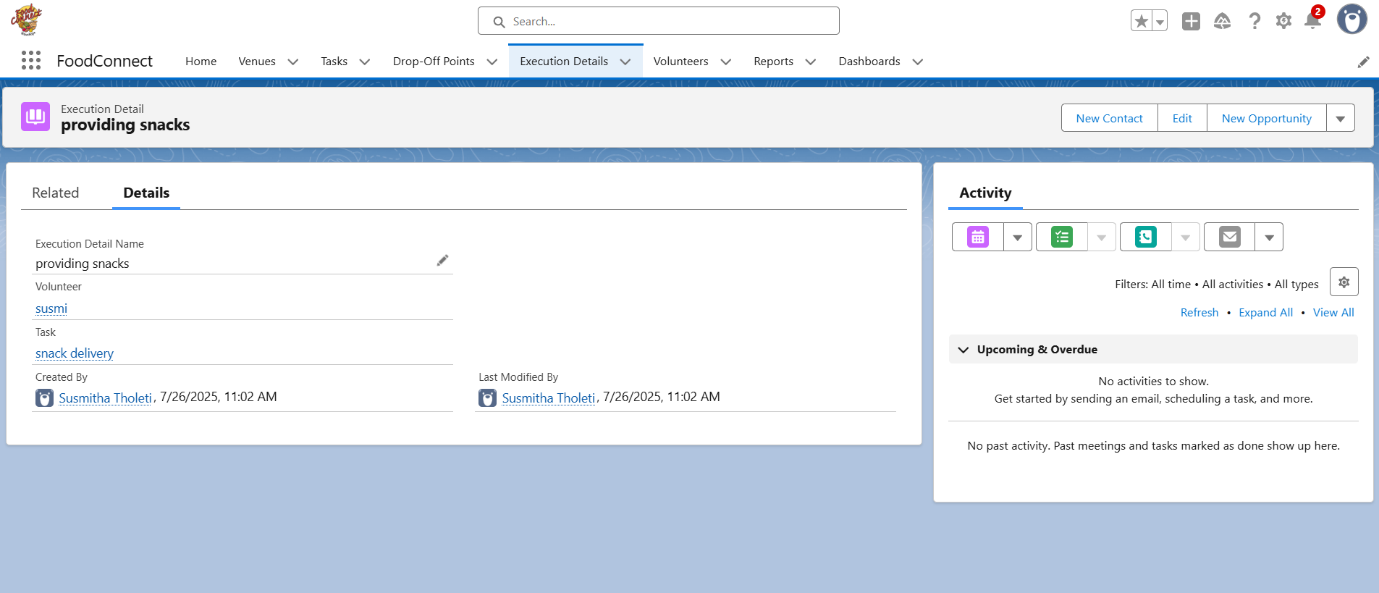
**System Maintenance and Monitoring:**

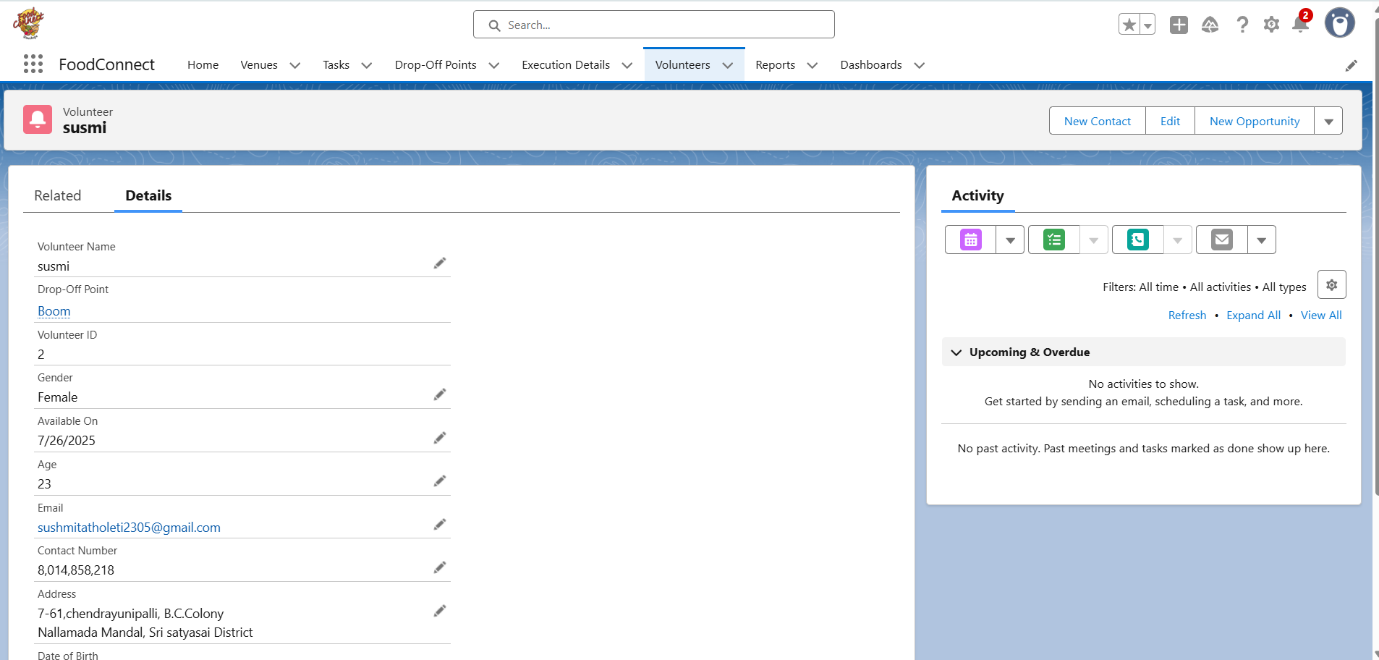
The CRM is made to make regular maintenance simple. Daily operations are monitored with the aid of scheduled reports and dashboards, which highlight metrics such as completed distributions and pending pickups. Duplicates and inconsistencies are found through routine data quality assessments. Periodically reviewing automation processes and making sure that validation rules and flows stay in line with changing business needs are the responsibilities of administrators. Before going live, any necessary improvements or new features will be safely created and tested in sandbox settings.

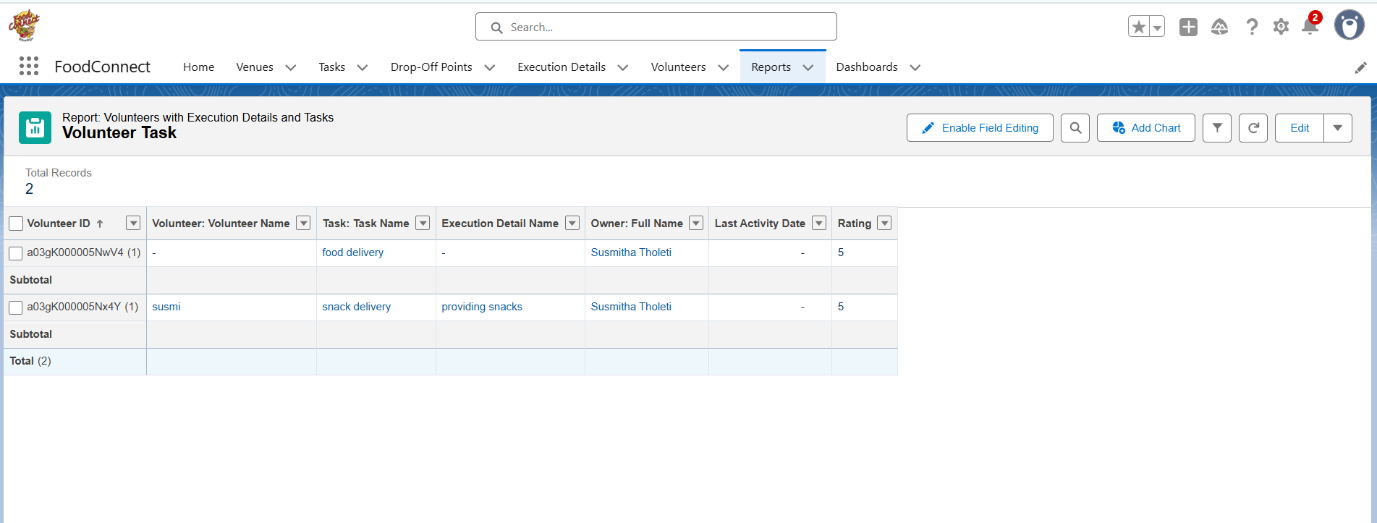


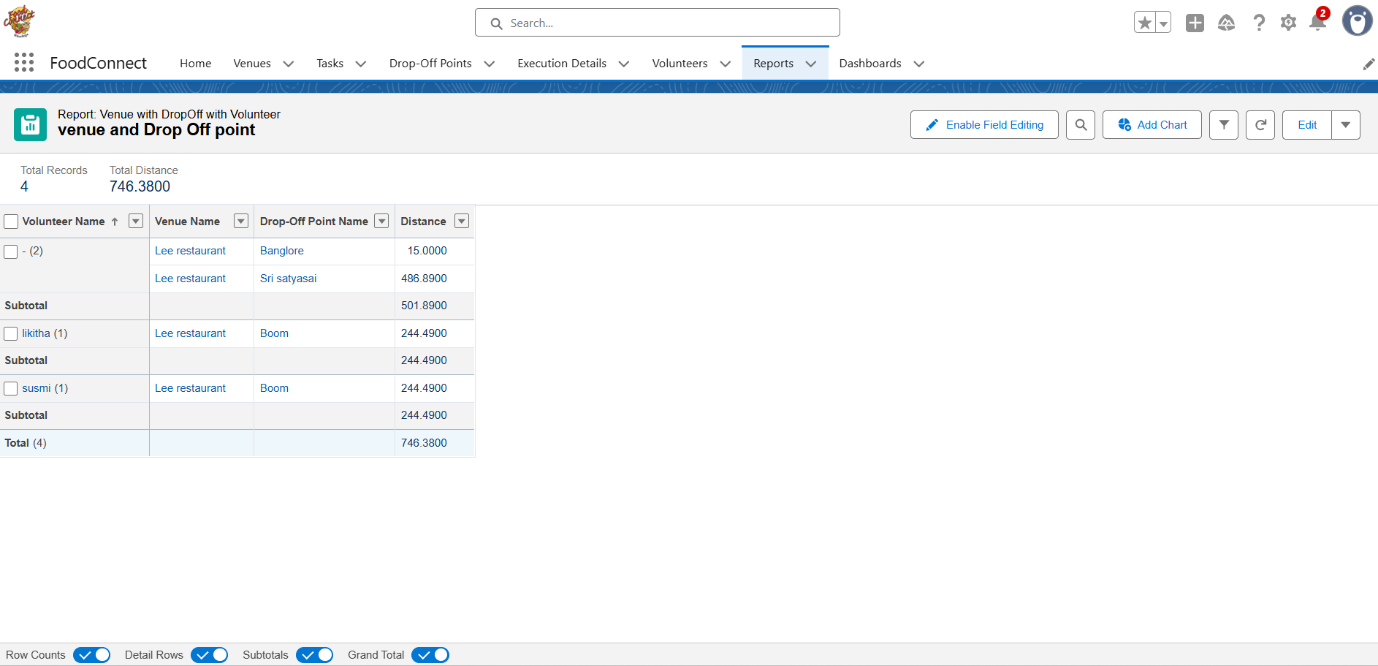


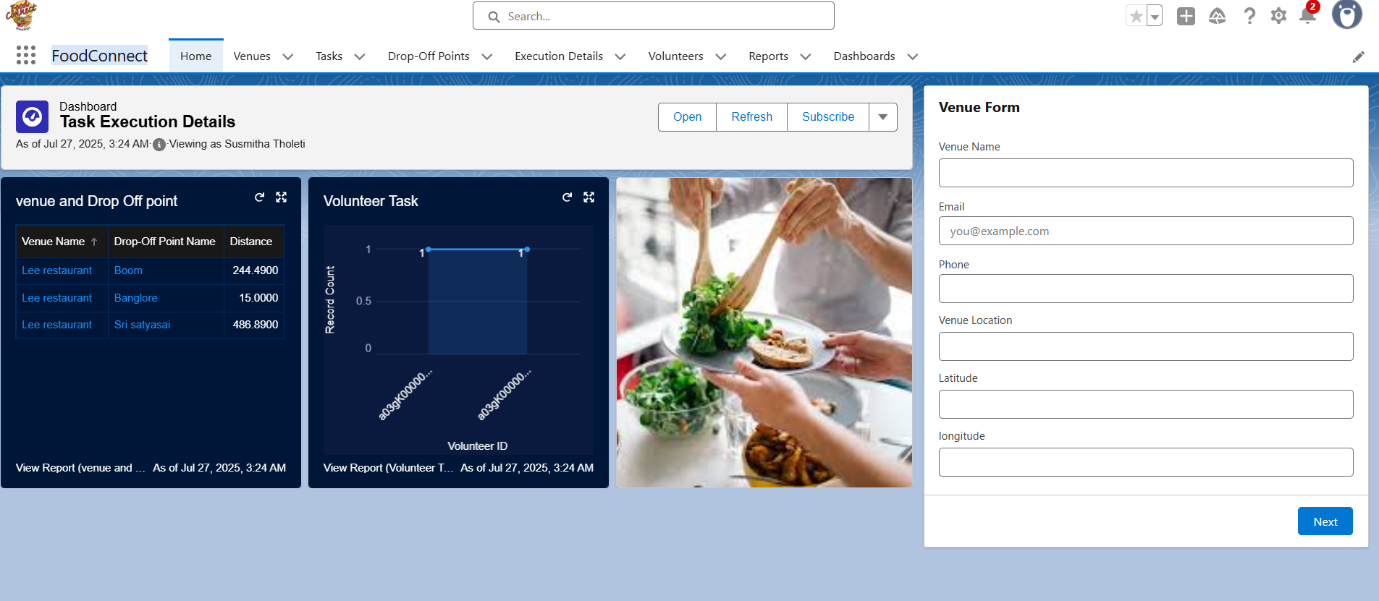












**Troubleshooting Approach:**

Common problems including unsuccessful pickups, delayed notifications, and inconsistent data are addressed through a systematic troubleshooting procedure to guarantee seamless operations and reduce interruptions. In order to swiftly locate and fix the underlying causes of failures in Apex triggers and automation flows, technical teams rely on debug logs. By describing object relationships, business rules, and known error messages, thorough CRM documentation aids in this process. This proactive and methodical approach supports the organization's goal of reducing waste and combating

hunger by preserving system reliability and guaranteeing that leftover food is delivered to those in need on time.

**Conclusion:**

Through the use of Salesforce CRM, the Food Connect project develops a dependable, expandable platform that links surplus food suppliers with underserved populations. By automating crucial procedures such as food distribution, scheduling, and collection, the system minimizes food waste, cuts down on manual labor, and guarantees that gifts go to the poor quickly and effectively. Real-time dashboards, thorough data recording, and automatic donor communication further enhance transparency and operational oversight. As the network grows, Food Connect's growth-oriented design offers a versatile platform for bringing on more volunteers, NGOs, and donors. With AI-powered demand forecasting and chatbot-enabled donor support, among other innovative possibilities, the platform has a lot of promise to advance its goal of eradicating hunger and helping marginalized populations.