**To Supply Leftover Food to the Poor**

Salesforce is a **cloud-based Customer Relationship Management (CRM) platform** that helps businesses manage customer data, sales, marketing, and support operations efficiently.

It offers integrated tools like Sales Cloud, Service Cloud, and Marketing Cloud, plus AI-powered features (like Agentforce) to automate tasks, personalize customer interactions, and improve team productivity.

Salesforce can be used at the student level to **enhance learning, career readiness, and academic support**:

* 🎓 **Student Success Platforms**: Tools like Education Cloud help track academic progress, schedule advising, and manage personalized care plans.
* 🤖 **AI-Powered Support**: Students can access automated help for FAQs, appointment scheduling, and academic planning via Agentforce.
* 📊 **Career & Engagement Tools**: Institutions use Salesforce to manage internships, tutoring, and wellbeing programs, improving student outcomes.

Salesforce offers **15 types of cloud services**, each tailored to specific business functions or industries. Here are the **6 major types** most commonly used:

* 💼 **Sales Cloud**: Manages leads, opportunities, and customer relationships to boost sales.
* 📣 **Marketing Cloud**: Automates and personalizes campaigns across email, social media, and mobile.
* 🛒 **Commerce Cloud**: Supports B2C and B2B e-commerce with personalized shopping experiences.
* 🛠️ **Service Cloud**: Enhances customer support with case management, chatbots, and omnichannel tools.
* 🌐 **Experience Cloud**: Builds branded portals and communities for customers, partners, and employees.
* 📊 **Analytics Cloud**: Provides advanced data visualization and AI-driven insights.

Additional specialized clouds include **Education Cloud**, **Health Cloud**, **Financial Services Cloud**, **Manufacturing Cloud**, and more.

In Salesforce, the **Administration** and **Developer** roles serve distinct but complementary functions:

🧭 **Administration**

* Uses **point-and-click tools** (no coding) to configure the platform.
* Manages **users, security, data**, and **automation** via tools like Flow and Process Builder.
* Builds **reports, dashboards**, and custom objects to support business processes.
* Ideal for roles like CRM Manager, Business Analyst, or Functional Consultant.

💻 **Developer**

* Uses **code** (Apex, Visualforce, Lightning Web Components) to extend platform capabilities.
* Builds **custom apps**, integrations (via APIs), and advanced automation.
* Handles **deployment**, performance optimization, and CI/CD pipelines.
* Suited for technical roles like Solution Architect or Technical Consultant.

Salesforce is built on a modular architecture with key components that support customization, automation, and scalability:

🔧 **Core Components**

* **Objects**: Standard (e.g., Account, Contact) and Custom objects to store data.
* **Fields**: Define data types and attributes within objects.
* **Records**: Individual entries of data stored in objects.
* **Apps**: Collections of tabs and functionalities tailored to business needs.
* **Tabs**: Navigation elements for accessing objects and apps.

⚙️ **Functional Components**

* **Automation Tools**: Workflow Rules, Process Builder, Flow, and Apex Triggers.
* **Security Model**: Profiles, Permission Sets, Roles, and Sharing Rules.
* **Reports & Dashboards**: Visualize and analyze data in real time.
* **Lightning Components**: UI building blocks for custom interfaces.
* **APIs**: REST and SOAP APIs for integration with external systems.
* These components work together to deliver a flexible CRM platform.

**Salesforce components and how they map to Admin vs Developer roles:**

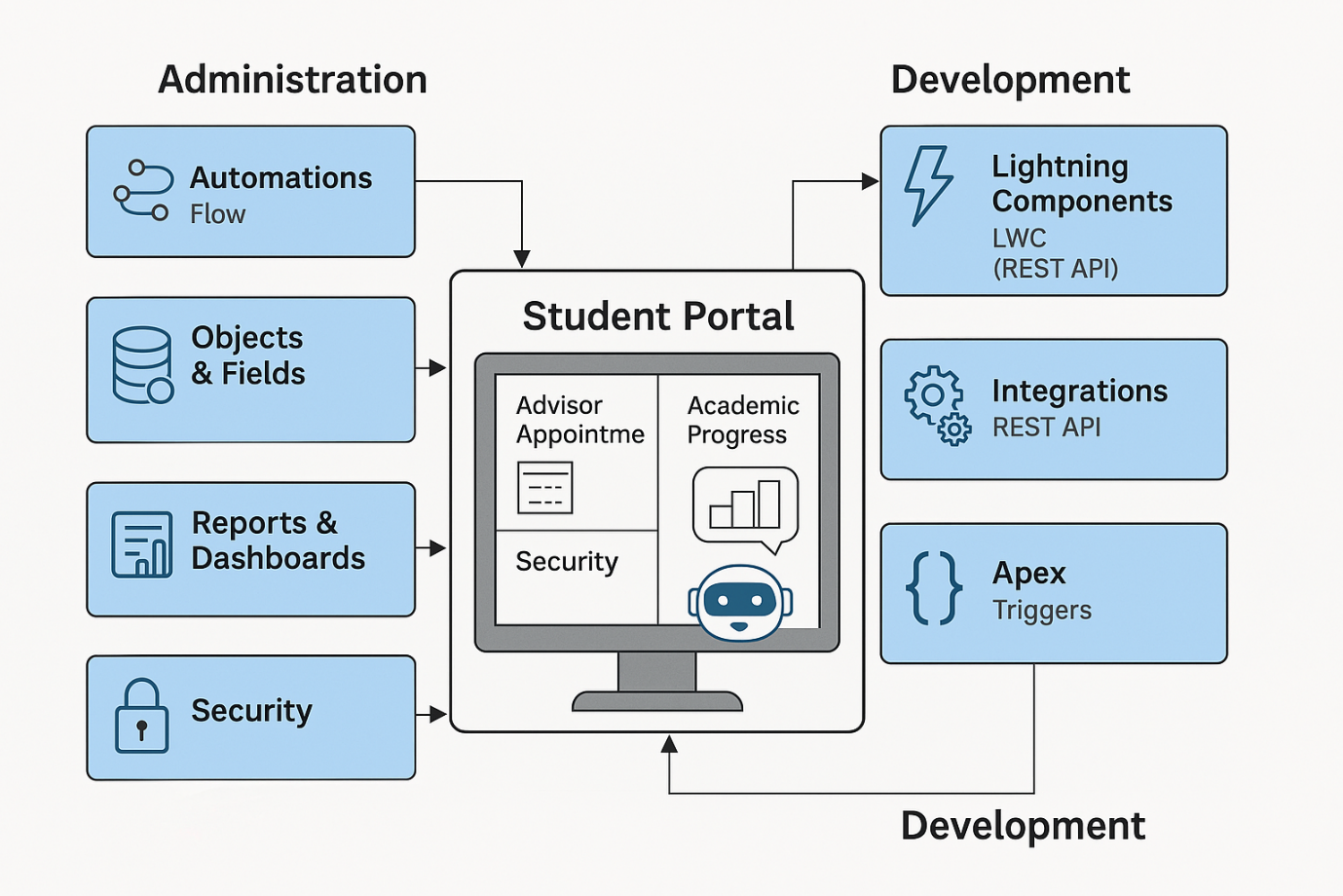
**🧩 Salesforce Architecture Overview**

+-----------------------------+  
|        User Interface       |  
|  (Lightning Experience,     |  
|   Mobile App, Communities)  |  
+-----------------------------+  
|     Application Layer       |  
|  (Standard & Custom Objects |  
|   Workflows, Reports, etc.) |  
+-----------------------------+  
|     Business Logic Layer    |  
|  (Apex Code, Triggers,      |  
|   Validation Rules, Flows)  |  
+-----------------------------+  
|     Data Layer              |  
|  (Database, SOQL, SOSL)     |  
+-----------------------------+  
|     Integration Layer       |  
|  (APIs, External Services)  |  
+-----------------------------+

**🔄 Admin vs Developer Role Mapping**

|  |  |  |
| --- | --- | --- |
| **Component** | **Admin Role 🧭** | **Developer Role 💻** |
| **Objects & Fields** | Create & manage via UI | Extend with metadata API |
| **Automation** | Use Flow, Process Builder | Write Apex Triggers & Classes |
| **Security** | Configure Profiles, Roles, Sharing | Enforce via Apex & Custom Auth |
| **Reports/Dashboards** | Build & customize | Embed in custom apps |
| **UI Customization** | Page layouts, Lightning App Builder | Lightning Web Components (LWC) |
| **Integrations** | Use External Services, Named Credentials | REST/SOAP APIs, Middleware (e.g. MuleSoft) |
| **Deployment** | Change Sets | CI/CD with Git, Salesforce CLI |

The picture provided below shows how Salesforce components come together in a practical use case like a student portal or energy system dashboard. You’ll see how data flows between admin settings, user interfaces, and backend automations.



📘 **Objective**

To build a Salesforce-based system that efficiently tracks and manages the distribution of leftover food to impoverished communities, minimizing waste and ensuring timely delivery.

🛠️ **Project Overview**

This project utilizes Salesforce to create a centralized database where donors can log surplus food availability and volunteers can track supply pickups and deliveries. It streamlines communication between stakeholders, automates record-keeping, and supports data-driven decisions for resource allocation.

📊 **Key Features**

* **Food Donation Log**: Donors enter food details, quantity, and expiration times.
* **Distribution Tracker**: Records delivery locations and recipient details.
* **Volunteer Management**: Assigns and tracks volunteer schedules and activities.
* **Analytics Dashboard**: Provides real-time insights into food supply, demand hotspots, and delivery success rates.

🌱 **Impact**

* Reduces food waste from restaurants, events, and households.
* Supports hunger relief by enabling efficient and traceable donations.
* Encourages community engagement through volunteer involvement.

**📍Milestone 1**

🏗️ **Salesforce Org Creation**

To begin development of the “To Supply Leftover Food to the Poor” project, a new Salesforce Developer Org is established using the provided sign-up link. This org serves as a blank canvas, requiring manual creation of all relevant data structures, objects, and configurations tailored to the project’s goals.

📌 **Purpose**

The org acts as the foundation for the project, enabling configuration of data models, user interfaces, automation flows, and security settings tailored to the food distribution use case.

🧩 **Key Development Tasks**

* **Custom Data Models**: Define custom objects such as Donors, Food Items, Distribution Logs, and Volunteers.
* **Field Definitions**: Create relevant fields for tracking quantities, expiration, location, and contact details.
* **Automation Setup**: Build Flows, Validation Rules, and Approval Processes to streamline operations.
* **Security & Access**: Configure Profiles, Roles, and Permission Sets for secure stakeholder interaction.
* **UI Configuration**: Use Lightning App Builder to design user-friendly interfaces for donors and volunteers.

**📍Milestone 2**

**🔓 Account Activation**

After creating the Salesforce Developer Org, the account activation is completed via the confirmation link sent to the registered email address. This process verifies ownership and grants full access to the development environment.

**✅ Activation Steps**

* Open the email from Salesforce and click the activation link.
* Set up a secure password and security question.
* Log into the Salesforce Developer Org to begin configuring custom components and development settings.

**📍Milestone 3**

**🧱 Core Object Types**

Salesforce objects are like database tables that store data. They come in several types, each serving a unique purpose:

|  |  |
| --- | --- |
| **Type** | **Description** |
| **Standard Objects** | Predefined by Salesforce (e.g., Account, Contact, Lead, Opportunity) |
| **Custom Objects** | Created by users to store data specific to their organization |
| **External Objects** | Represent data stored outside Salesforce, accessed via APIs |
| **Big Objects** | Handle massive volumes of data for archiving or analytics |
| **Platform Events** | Enable event-driven architecture for real-time communication |
| **Change Data Capture** | Track changes to records for syncing with external systems |

**🔗 Relationship Types**

* **Lookup Relationship**: Loose link between objects
* **Master-Detail Relationship**: Strong dependency; child record tied to parent
* **Many-to-Many Relationship**: Managed via junction objects

Would you like a diagram showing how these object types interact in a real-world app like a student portal or energy dashboard? I can sketch that next.

🧱 **Custom Object Creation**

To support the operational workflows of the “To Supply Leftover Food to Poor” initiative, the following custom objects are created within the Salesforce Org. Each object captures key data entities involved in food logistics and volunteer coordination.

📘 **Defined Custom Objects**

|  |  |
| --- | --- |
| **Object Name** | **Purpose** |
| **Venue** | Stores details of locations where food is collected or events are held |
| **Drop Off Point** | Tracks designated food delivery locations including contact and address |
| **Task** | Manages individual assignments linked to food pickup, delivery, or setup |
| **Volunteer** | Holds profiles, roles, and availability details of volunteer personnel |
| **Execution Details** | Logs overall activity data—timestamps, route info, completion status |

**📍Milestone 4**

🗂️ **Custom Tab Setup**

To streamline navigation and improve user experience, custom tabs are created for each key object in the “To Supply Leftover Food to Poor” project: **Venue**, **Drop Off Point**, **Task**, **Volunteer**, and **Execution Details**.

📌 **Tab Configuration Steps**

* Go to **Setup** → **Tabs** → **Custom Object Tabs.**
* Choose the corresponding custom object.
* Assign tab label, icon, and user visibility settings.
* Add tabs to relevant apps via the **App Manager.**

🎯 **Purpose of Creating Tabs**

* **Direct Access**: Allows users to open and view records for each object without searching or navigating manually.
* **Improved Usability**: Enhances user interface clarity, especially for volunteers and coordinators.
* **Modular Design**: Keeps functionalities well-organized across different roles and workflows.

**📍Milestone 5**

⚡ **Lightning App Creation – *“To Supply Leftover Food to Poor”***

To unify navigation across custom objects and streamline user access, a dedicated Lightning App is created with the project title. This app organizes tabs, layouts, and permissions tailored to donor coordination, volunteer management, and food logistics.

🧭 **Key Setup Steps**

* Go to **Setup** → **App Manager** → **New Lightning App**
* Enter the App Name: *“To Supply Leftover Food to Poor”.*
* Choose a custom icon and primary color for branding.
* Add navigation tabs: Venue, Drop Off Point, Task, Volunteer, Execution Details.
* Assign user profiles (e.g., Admin, Volunteer Coordinator) for app visibility.
* Save and activate the app for users to start working within a unified interface.

🎯 **Purpose**

* Provides a centralized workspace for managing all aspects of the food distribution workflow.
* Simplifies task access, improves productivity, and enhances user experience.

**📍Milestone 6**

In Salesforce, **relationships define how objects are connected** so data can be linked and accessed efficiently. There are six main types:

* **Lookup Relationship**: Loosely connects two objects; child records remain even if the parent is deleted.
* **Master-Detail Relationship**: Strongly binds objects; deleting the parent also deletes child records. Enables roll-up summaries and shared security.
* **Many-to-Many Relationship**: Achieved using a junction object with two master-detail fields to link records from both sides.
* **Self-Relationship**: An object relates to itself, useful for hierarchies like campaign structures.
* **External Relationships**: Connect Salesforce objects to external data using indirect or external lookups.
* **Hierarchical Relationship**: Specific to the User object, used to define reporting structures like manager relationships.

Let me know if you'd like a quick comparison table or examples for each!

**🔗 Relationship Field Configuration**

To establish data connectivity between objects in the *“To Supply Leftover Food to Poor”* project, various **Master-Detail** and **Lookup Relationship** fields are configured. These enable structured record linkage and dependency across workflows.

**📘 Configured Relationships**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Relationship Type** | **Related To** | **Field Name** | **Purpose** |
| Volunteer | Master-Detail | Drop-Off Point | Drop\_Off\_point | Links volunteers with assigned drop-off sites |
| Execution Details | Master-Detail | Volunteer | Volunteer | Connects execution logs with responsible volunteer |
| Execution Details | Master-Detail | Task | Task | Ties execution actions with specific tasks |
| Drop-Off Point | Lookup | Venue | Venue\_\_c | Identifies delivery location’s physical venue |
| Task | Lookup | Venue | Sponsored By | Indicates which venue sponsors the task |
| Task | Lookup | Drop-Off Point | Drop-Off point | Specifies location for task execution |

**⚙️ Field Creation Process (General Steps)**

1. Navigate to **Setup** → **Object Manager** → Select Object
2. Open **Fields & Relationships** → Click **New**
3. Choose **Master-Detail** or **Lookup Relationship** as required
4. Select related object → Name field → Configure settings → Save

**🔗 Relationship Field Descriptions**

* **Volunteer → Drop-Off Point (Master-Detail)**  
  Assigns each volunteer to a specific drop-off location for operational tracking.
* **Execution Details → Volunteer (Master-Detail)**  
  Links execution logs to the responsible volunteer for accountability.
* **Execution Details → Task (Master-Detail)**  
  Connects execution records to related tasks for proper workflow management.
* **Drop-Off Point → Venue (Lookup)**  
  References the venue where food is received or distributed.
* **Task → Venue (Lookup, Sponsored By)**  
  Identifies the supporting venue responsible for the task.
* **Task → Drop-Off Point (Lookup)**  
  Specifies the location where the task is executed or delivered.

Here’s a more detailed breakdown of each field in your Venue object, tailored for clarity and purpose-based documentation:

**🏛️ Venue Object Field Configuration**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose & Behavior** |
| **Contact Email** | Email | Stores the venue coordinator’s email address. Used for automated communication, alerts, and scheduling confirmations. |
| **Contact Phone** | Phone | Captures the venue coordinator’s mobile or landline number. Enables quick manual outreach or SMS notifications. |
| **Location** | Geolocation | Stores latitude and longitude values. Useful for mapping, routing, and proximity-based filtering. |
| **Venue Location** | Long Text Area | Provides a detailed description of the venue’s address, landmarks, and special instructions. Supports multiline inputs. |

This setup ensures easy venue management, especially for logistics, contact coordination, and geo-services integration.

**🆔 Execution ID – Field Configuration**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose** |
| **Execution ID** | Auto Number | Automatically assigns a unique ID to each record upon creation. Useful for unambiguous reference, report filtering, audit trails, and linking with related records. |

**⚙️ Additional Setup Notes**

* **Format**: You can customize the prefix (e.g., EXC-{0000}) to reflect object identity.
* **Uniqueness**: This ID is system-generated and guaranteed to be unique—ideal for tracking operational executions or process iterations.
* **Visibility**: Often set as read-only to preserve data integrity and avoid manual edits.

**🙋 Volunteer Object – Field Configuration Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose & Usage** |
| **Volunteer ID** | Auto Number | System-generated unique ID for each volunteer. Ideal for tracking, referencing, and linking records reliably. |
| **Gender** | Picklist | Captures gender identity. Useful for demographic analysis, personalized communication, and event diversity planning. |
| **Available On** | Date | Indicates the volunteer’s next availability. Supports task scheduling, shift assignments, and reminder automation. |
| **Age** | Number | Stores age value. Helps enforce age restrictions and plan suitable volunteer activities. |
| **Email** | Email | Contact email for formal communication, notifications, or volunteer confirmations. |
| **Contact Number** | Number | Phone or mobile number for quick communication, urgent updates, and SMS alerts. |
| **Address** | Long Text Area | Accommodates detailed residential or preferred contact address, including street name, city, zip, and notes. |
| **Date of Birth** | Date | Tracks birthdate for eligibility verification, age calculations, and personalized interactions. |

This structure promotes streamlined volunteer onboarding, efficient coordination, and enhanced communication workflows.

**📦 Drop-Off Point Object – Field Configuration Details**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose & Usage** |
| **Location 2** | Geolocation | Captures the latitude and longitude of each drop-off point. Useful for mapping, route optimization, and proximity analysis. Can be paired with custom logic or third-party mapping tools. |
| **Distance Calculation** | Formula (Number) | Calculates the distance (in km) between this drop-off point and its related Venue. Typically uses the Haversine formula with lat-long values from both records. Enables efficient logistics planning. |
| **State** | Picklist | Provides a standardized list of Indian states and Union Territories. Supports regional filtering, administrative reporting, and location-based dashboards. |

🛠️ If you’re applying this in Salesforce, ensure:

* **Distance Calculation** references both geolocation fields via formula or Apex if precision is key.
* **State Picklist** uses a restricted value set for consistency across records.

**📋 Task Object – Field Configuration Details**

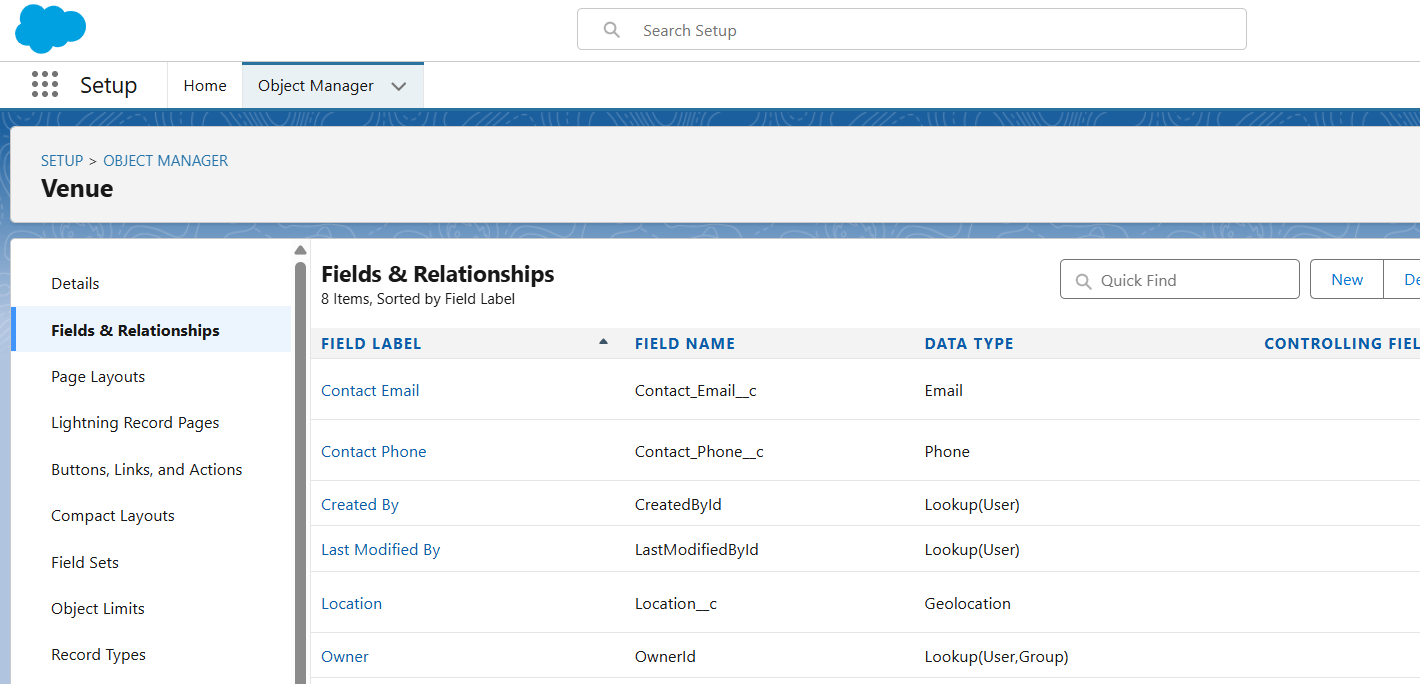
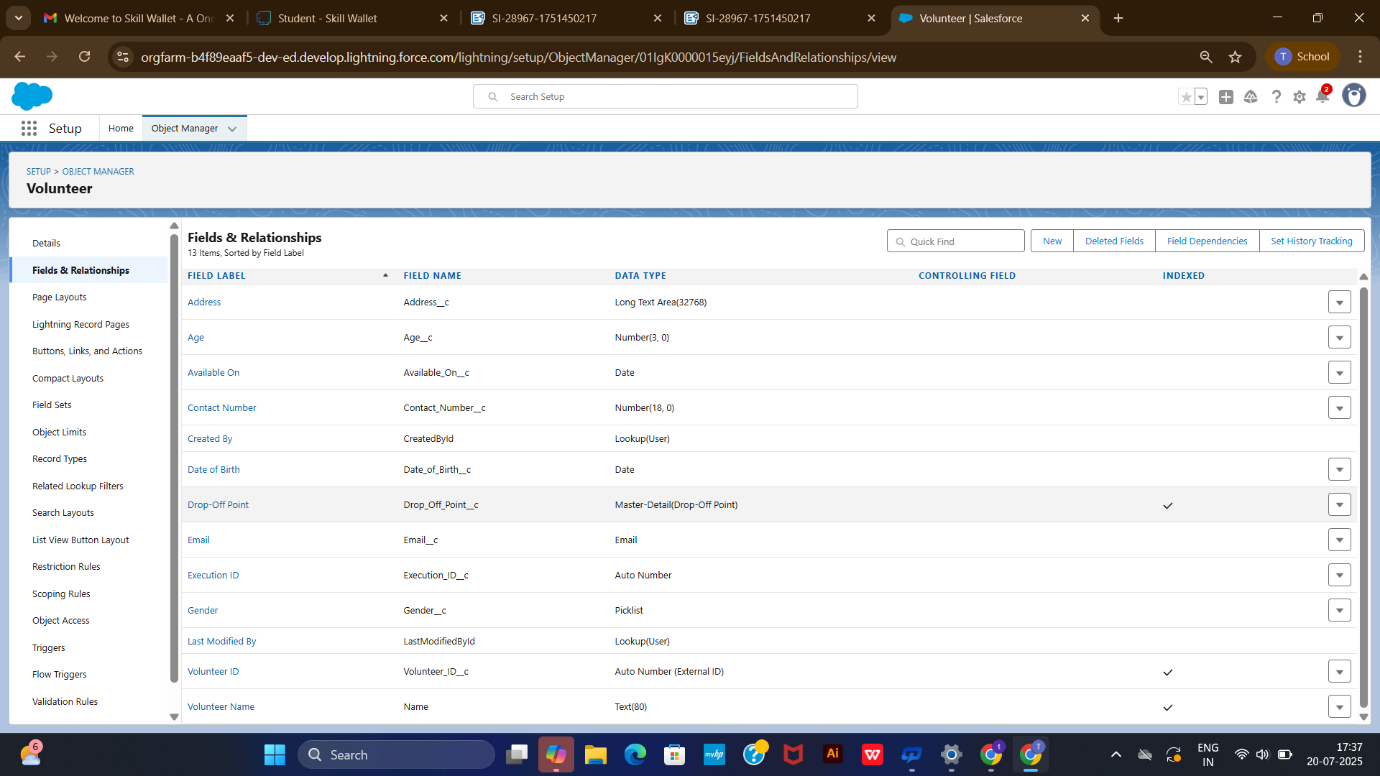
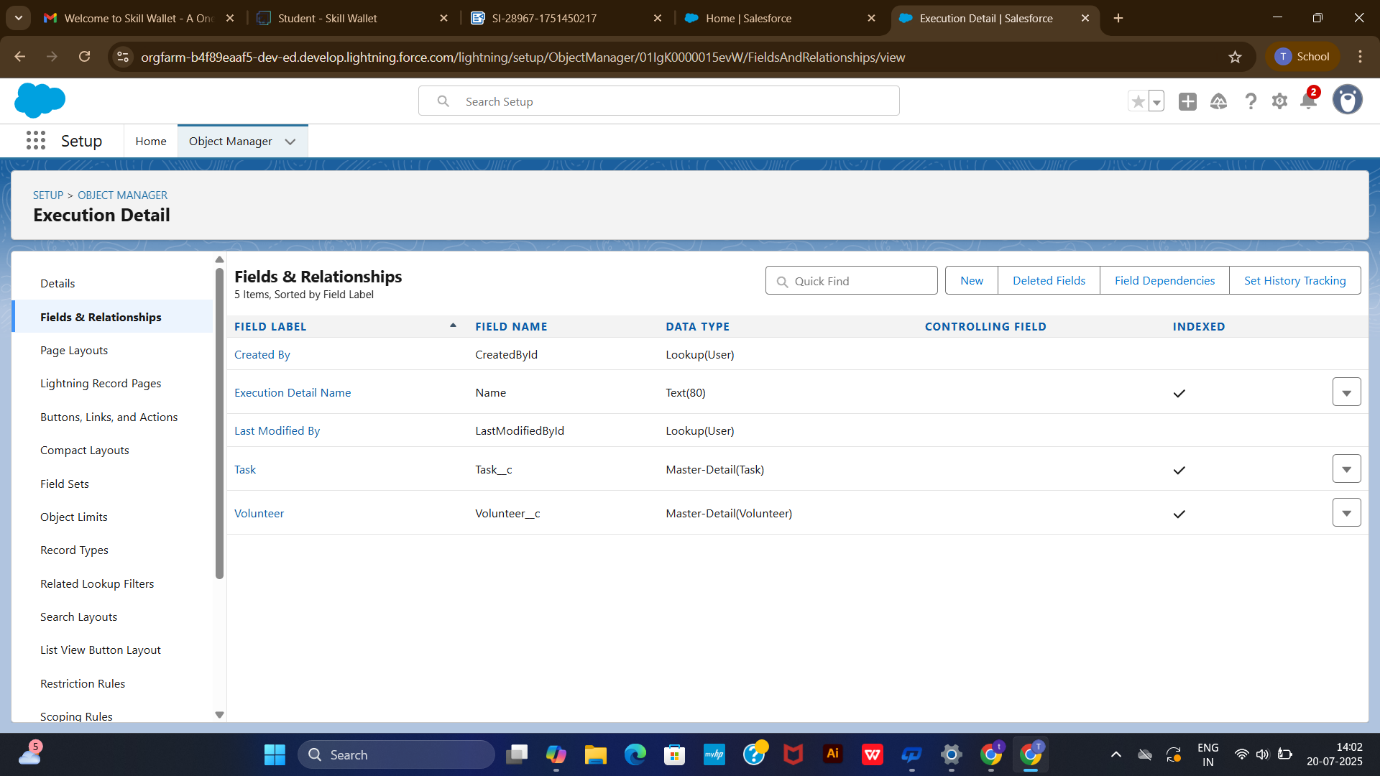
|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Purpose & Behavior** |
| **Task ID** | Auto Number | Auto-generates a unique task identifier, e.g. TASK-0001. Enhances record traceability, auditability, and consistency. |
| **Date** | Date | Stores the scheduled or execution date. Crucial for reports, history tracking, and reminders. |
| **Food Category** | Picklist (Multi-Select) | Enables selection of multiple food types like Veg, Non-Veg, Salad, Snack. Assists in dietary planning and inventory coordination. |
| **Number of People Served** | Number | Captures impact metrics by logging recipients per task. Useful for performance dashboards and service analysis. |
| **Name of the Person** | Text | Stores the organizer or point-of-contact name. Supports personalization and team-based filtering. |
| **Phone** | Phone | Logs mobile or landline number for communication or last-minute changes. |
| **Rating** | Picklist | Allows selection from predefined ratings (1 to 5). Useful for evaluating task effectiveness or volunteer coordination. |
| **Feedback** | Long Text Area | Captures open-ended comments, issues, or suggestions. Valuable for service improvement and record enrichment. |
| **Distance** | Number (Precision 4) | Records task-related travel distance—e.g., delivery or volunteer travel. Enables logistics optimization and expense tracking. |

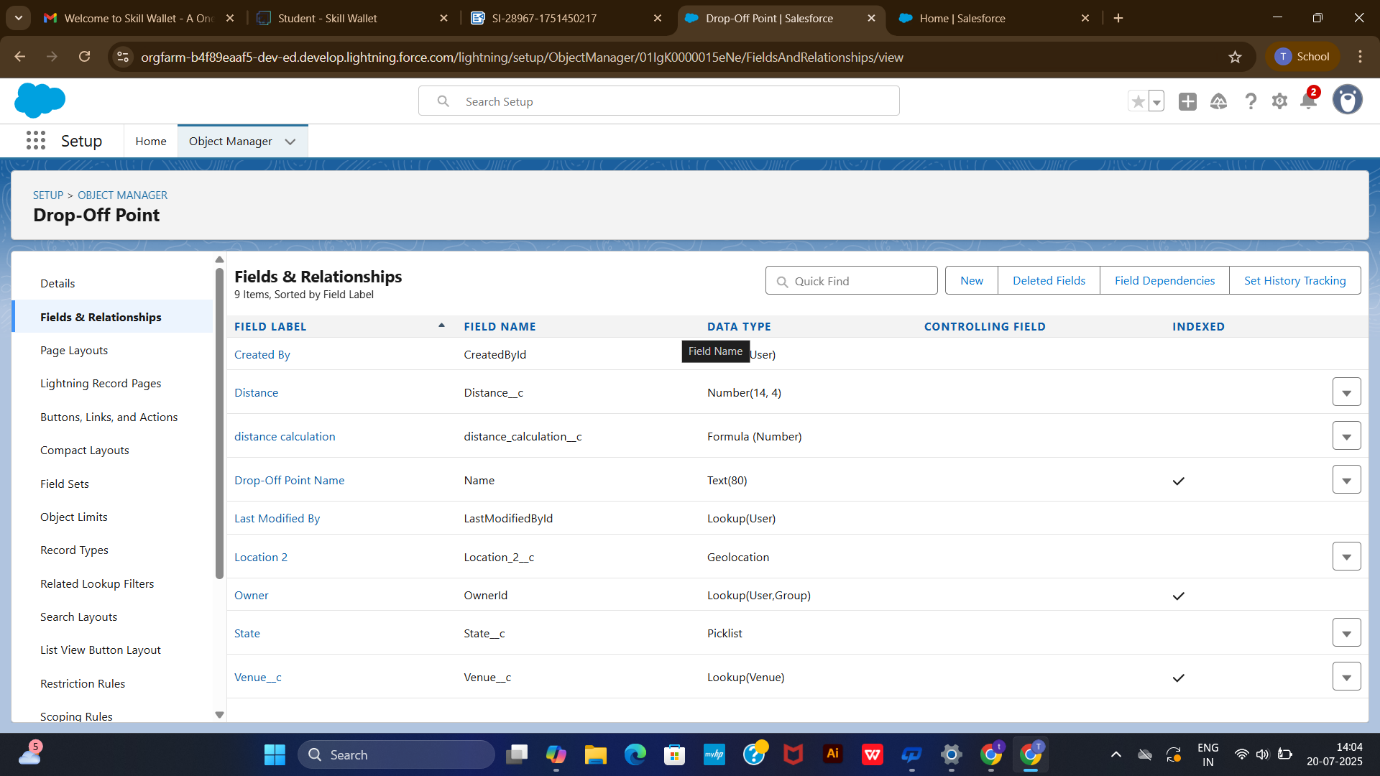
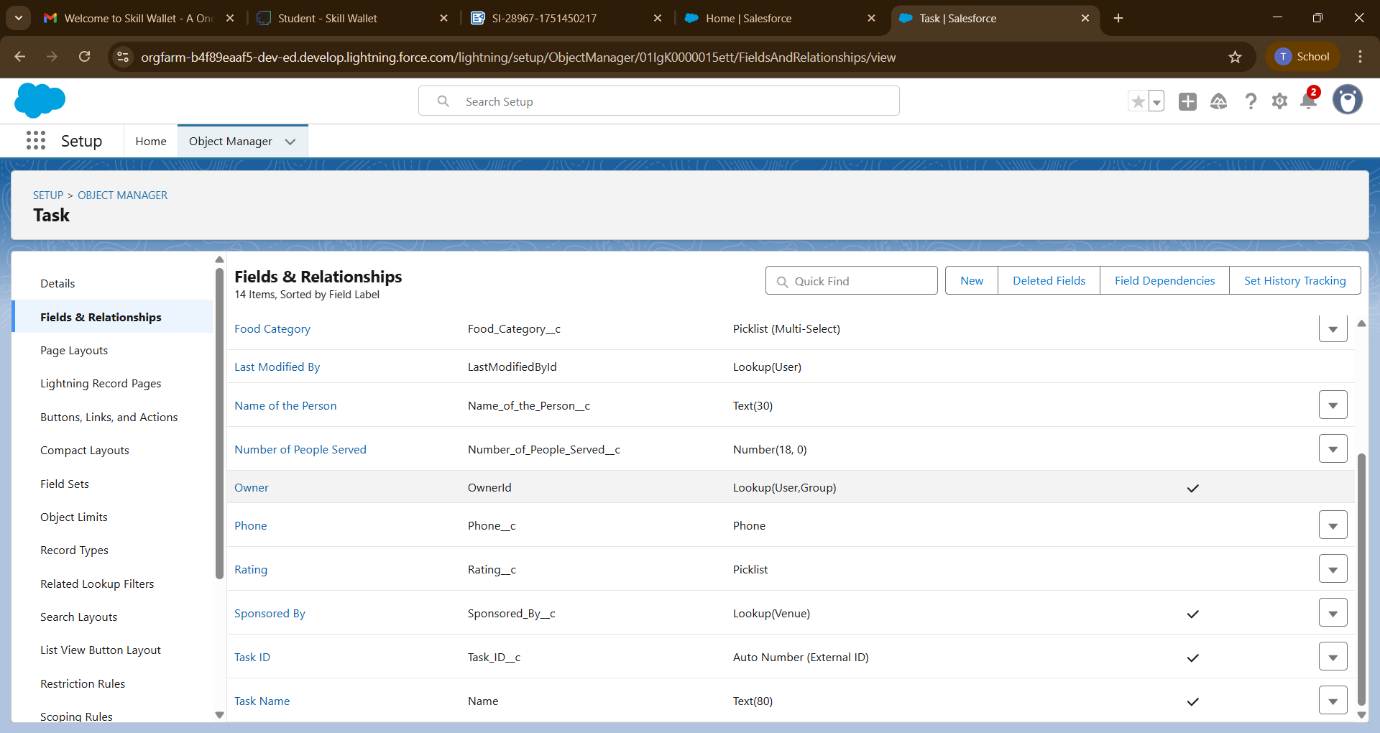
🧩 This structure facilitates seamless planning, execution insights, and impact measurement across food distribution efforts.

**✅ Summary of Field Configurations**

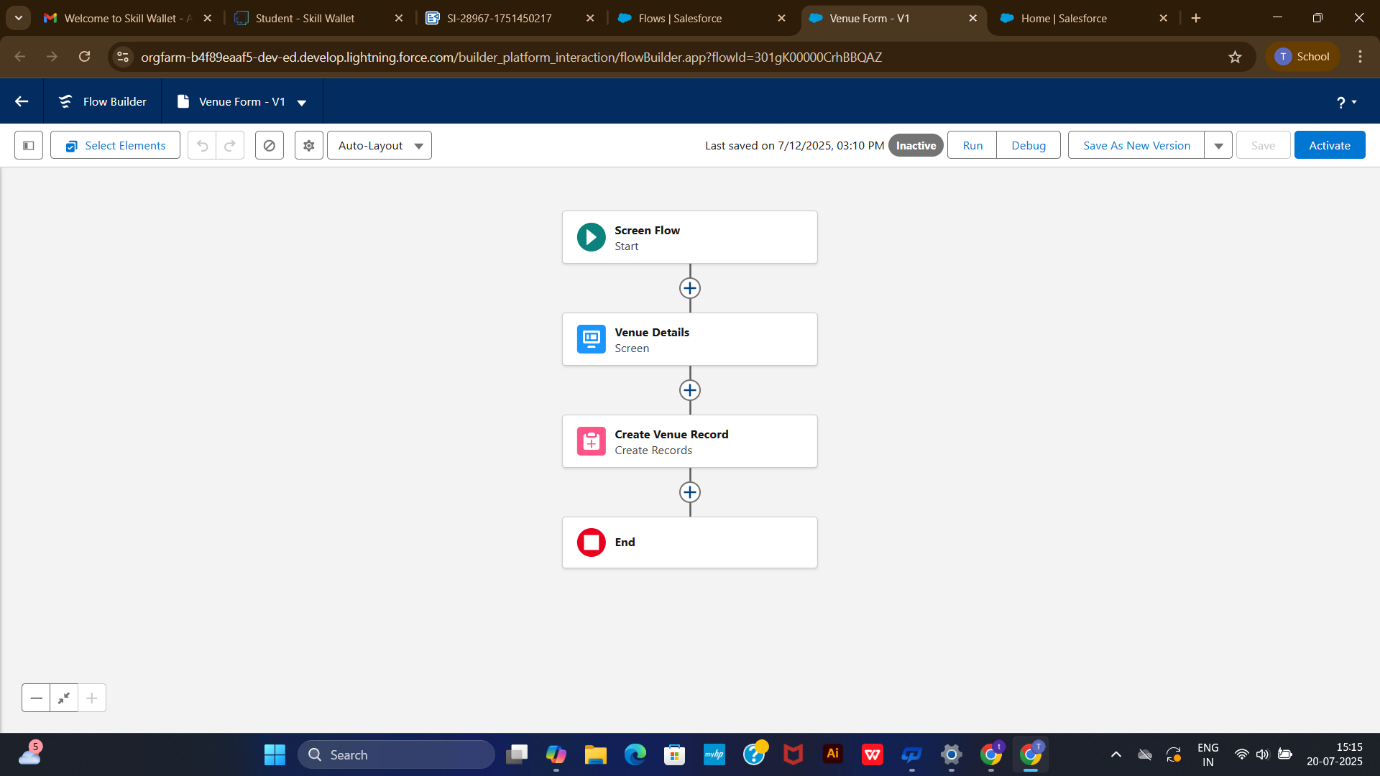
|  |  |
| --- | --- |
| **Object** | **Focus** |
| **Venue** | Contact info, geolocation, and descriptive address for event coordination. |
| **Execution Details** | Auto-generated ID for unique tracking of every execution instance. |
| **Volunteer** | Key personal and logistical details for optimized volunteer management. |
| **Drop-Off Point** | Location mapping, state identification, and distance calculation for logistical planning. |
| **Task** | End-to-end task metadata—type of food, people served, contact details, impact tracking. |

📌 With these fields in place, your Salesforce system is well-equipped for smooth data capture, tracking, and reporting. Each object contributes to a cohesive data model that supports food distribution logistics and volunteer coordination.





**📍Milestone 7**



In Salesforce, a **Flow** is a powerful automation tool that lets you build logic using clicks—not code. It collects data, applies conditions, updates records, sends emails, and more—all within a visual interface called **Flow Builder**.

**🔄 Types of Flows**

* **Record-Triggered Flow**: Runs when a record is created, updated, or deleted.
* **Screen Flow**: Displays UI screens to users for input or guidance.
* **Scheduled Flow**: Executes at defined intervals (e.g., daily or weekly).
* **Autolaunched Flow**: Runs in the background, triggered by Apex, another flow, or API.
* **Platform Event Flow**: Responds to incoming platform event messages.

Flows are ideal for automating business processes like approvals, data updates, and task assignments. Want help designing one for your food distribution project?

🔄 **Flow Creation – *Venue Record Entry***

A **Screen Flow** named **Venue Form** is designed to capture venue details and automatically create a record in the **Venue** object.

🧾 **Description**

* Presents a form with fields for venue name, contact email & phone, location, latitude, and longitude.
* Maps user inputs to corresponding fields in the **Venue** object via a **Create Record** element.
* Uses separate resources and literal values for data binding and record generation.

🎯 **Purpose**

To streamline and standardize venue data entry by providing an interactive UI for users or volunteers—ensuring accurate and complete record creation with geolocation support.

**📍Milestone 8**

In Salesforce, a **trigger** is a piece of Apex code that runs **before or after** specific database events—like insert, update, delete, or undelete—on a record.

**⚙️ Trigger Basics**

* **Purpose**: Automate custom logic when data changes occur.
* **Types**:
  + **Before Trigger**: Used to validate or modify data *before* it's saved.
  + **After Trigger**: Used to perform actions *after* data is saved (e.g., update related records).

**🧠 Example Use Cases**

* Automatically create a related record when a new one is inserted.
* Prevent deletion of critical records.
* Sync field values between parent and child objects.

Would you like a sample trigger for one of your custom objects like Task or Volunteer?

✨ **Trigger Summary: DropOffTrigger**

This Apex trigger runs **before insert** on the **Drop\_Off\_point\_\_c** object. It sets the value of the Distance\_\_c field equal to the pre-calculated value in the distance\_calculation\_\_c field for each new record.

🎯 **Purpose**

To ensure the Distance\_\_c field automatically reflects the computed distance upon record creation, enabling consistent usage in sharing rules and data visibility logic.

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;

}

}

‍**📍Milestone 9**

**💼 Purpose of Creating Profiles**

* **Access Control**: Profiles assign permissions for viewing, editing, creating, or deleting specific records or objects.
* **User Segmentation**: Different user roles (admin, volunteer, coordinator) get tailored access suited to their responsibilities.
* **Security & Efficiency**: Limits unnecessary data exposure and helps users focus only on tools relevant to their tasks.

**🛠️ Key Configurations in a Profile**

* **Object Permissions**: Define access to standard/custom objects (e.g. Contacts, Reports).
* **Field-Level Security**: Control visibility/edit rights for individual fields.
* **Tab Settings**: Decide which tabs are visible or hidden.
* **Page Layouts**: Set default layouts for users in that profile.
* **App Access**: Choose which apps are available.

Profiles are essential for ensuring data integrity and role-based system usage.

👥 **NGOs Profile – Description & Purpose**

🧾**Description**  
The **NGOs Profile** is a custom Salesforce user profile created by cloning the Standard Platform User. It defines access settings and permissions tailored for personnel affiliated with non-governmental organizations participating in the food distribution initiative.

🎯**Purpose**  
To grant NGO users secure, role-appropriate access to relevant objects, tabs, and apps—such as Venue, Drop-Off Point, Task, and Volunteer—enabling them to record, view, and track supply efforts without unnecessary administrative privileges.

**👥 NGOs Profile – Detailed Overview**

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| **Base Profile** | Cloned from **Standard Platform User**, which offers access to custom apps but restricts setup access. |
| **Customization** | Tailored to limit access only to relevant objects: **Venue**, **Drop-Off Point**, **Volunteer**, and **Task**. Ensures operational focus while preventing unauthorized configuration changes. |
| **Object Permissions** | - **Read/Create/Edit** on operational objects - **No Delete or Modify All** to protect data integrity - **Restricted Access** to standard CRM objects like Leads, Opportunities, etc. |
| **Tab Settings** | Tabs for custom objects are set to *Default On* or *Available*, ensuring visibility without cluttering the interface. |
| **App Access** | Access to a custom Lightning App designed for food distribution tracking. No access to sensitive or administrative apps. |
| **Field-Level Security** | Sensitive fields (e.g. personal contact numbers, geolocation) may have *Read-only* or *Hidden* settings depending on user role. |
| **Use Case Alignment** | Empowers NGO volunteers or coordinators to log and track contributions, outreach, and distribution without access to system setup or unrelated data. |

This profile ensures streamlined, secure access that aligns with their operational responsibilities while preserving data governance.

**📍Milestone 10**

**👤 Why Create User Accounts?**

* **Access Control**: Assign roles, profiles, and permissions so users can only access data relevant to their responsibilities (e.g. NGO personnel managing drop-off points and tasks).
* **Collaboration & Tracking**: Each user can log activities, update records, and participate in flows—allowing accurate tracking of contributions and accountability.
* **Auditability**: Salesforce logs user actions by account, helping with compliance, debugging, and historical analysis.
* **Personalized Experience**: UI settings, dashboards, and reports can be tailored per user based on their needs and role.

In your food distribution setup, it ensures NGO partners work efficiently without risking system-wide changes.

**👥 User Accounts**  
Multiple users are created in the Salesforce Org to represent participating NGOs, each configured with the **Salesforce Platform** user license and assigned the **NGOs Profile**. This setup enables controlled, role-specific access to project features like venue registration, food task monitoring, and execution tracking.

🎯**Purpose**

* Facilitate collaboration with verified NGO partners.
* Grant access to relevant tabs, records, and flows without admin privileges.
* Support food supply chain tasks, record updates, and progress tracking by external contributors.

**👥 NGO User Accounts – Detailed Setup**

|  |  |
| --- | --- |
| **Component** | **Configuration & Role** |
| **User Type** | Standard **Salesforce Platform License** — grants access to custom apps and objects without full CRM features. |
| **Profile Assigned** | **NGOs Profile** — tailored to restrict setup privileges and expose only relevant tabs like Venue, Drop-Off Point, Task, and Volunteer. |
| **Access Scope** | Users can *create*, *view*, and *update* records across operational objects but cannot delete or modify system-level data. |
| **Permissions Inherited** | From the profile—tab visibility, object-level security, flow execution, and field-level access. |
| **Role Assignment (Optional)** | Can be linked to **NGO Contributor Role** or similar hierarchy for sharing rules, visibility settings, and reporting. |
| **Collaboration Use Case** | Enables verified NGO staff to log activity, update service records, and monitor progress—without administrative access or risk to system integrity. |

🔐 This setup ensures secure collaboration with external partners, keeping your data organized, roles well-separated, and operations scalable.

**📍Milestone 11**

**🔒Public Groups**

**🔒 Benefits of Public Groups**

* **Centralized Sharing Rules**: Instead of setting access per user, you can grant permissions to an entire group—saving time and reducing errors.
* **Role-Based Collaboration**: Combine users (e.g. NGO staff + Admin) to work together on records like Tasks, Venues, or Volunteers.
* **Hierarchy Support**: When "Grant Access Using Hierarchies" is enabled, managers automatically inherit visibility over group-shared data.
* **Streamlined Security Management**: Groups let you control access across objects without custom permission sets or manual updates for each user.

🚀 In your project, public groups ensure NGO partners can collaborate securely and efficiently, with scalable access tailored to each organization's role.

👥 **Public Groups – Description & Purpose**

🧾**Description**  
Three public groups are created in Salesforce, each representing a participating NGO. These groups include their respective NGO user and the System Administrator. For example, the **Iksha** group includes the *Iksha Foundation* user and the admin.

🎯**Purpose**

* ✅ **Simplified Sharing Rules**: Public groups enable record access based on group membership rather than individual users.
* 🔐 **Access Control**: You can easily set sharing rules to allow group members to view, edit, or manage records across objects like Venue, Drop-Off Point, or Tasks.
* 🏢 **Hierarchical Permission Management**: With “Grant Access Using Hierarchies” enabled, managers or higher roles can inherit visibility across group data.

**👥 NGO-Based Public Groups – Configuration & Functionality**

|  |  |
| --- | --- |
| **Element** | **Details** |
| **Group Composition** | Each group includes one NGO user (e.g. *Iksha Foundation*) and the **System Administrator**. This pairing allows both operational and oversight access. |
| **Purpose of Creation** | Designed for centralized **record sharing** across critical objects like **Venue**, **Drop-Off Point**, **Volunteer**, and **Task**—without manually configuring individual access. |
| **Usage in Sharing Rules** | Sharing rules are defined at the group level, granting **Read/Edit/Create** access where needed. This simplifies security management and supports scalable access control. |
| **Hierarchical Settings** | With **Grant Access Using Hierarchies** enabled, senior roles (like NGO project leads or admins) automatically inherit visibility to lower-level records, promoting supervisory transparency. |
| **Object-Level Application** | Applied to records created by NGO users, ensuring they and their group members can collaborate without violating broader org security boundaries. |

🔧 This setup empowers efficient data sharing and collaborative record access while maintaining strict governance and clarity over who can do what.

**📍Milestone 12**

In Salesforce, **report types define the data structure** available for building reports. They determine which objects and fields can be used, and how those objects are related.

There are two main categories:

* **Standard Report Types**: Automatically created for standard and custom objects with direct relationships.
* **Custom Report Types**: Manually defined to include specific objects, relationships, and fields—ideal when standard types don’t meet your needs.

Each report type serves as a template for generating reports. Once selected, it can't be changed later.

📊 **Custom Report Type – *Venue with DropOff with Volunteer***

🧾**Description**  
This custom report type enables multi-object reporting by joining **Venues**, **Drop-Off Points**, and **Volunteers**. It is labeled *Venue with DropOff with Volunteer*, categorized under **Other Reports**, and marked as **Deployed** for active use.

**🔗Object Relationships**

* **Primary Object**: *Venues*
* Related to: *Drop-Off Points* (via "A may or may not have B")
* Further Related to: *Volunteers* (same flexible relationship)

🎯**Purpose**

* 🔎 Allows reporting on venue-related logistics and volunteer assignments across linked drop-off points
* 📋 Supports operational insights by displaying relationships between physical locations and people involved
* 📈 Facilitates dashboards, audits, and performance tracking for food supply and distribution activities.

📄 **Purpose of Creating Reports**

* **Track Progress**: Monitor key metrics like task execution, volunteer hours, or resource allocation in real-time.
* **Identify Patterns**: Spot trends, inefficiencies, and bottlenecks through visual summaries.
* **Make Informed Decisions**: Support strategic planning with accurate data insights.
* **Enhance Accountability**: Share outcomes with stakeholders and measure impact clearly.
* **Customize Views**: Tailor reports for different teams—logistics, admin, outreach—based on what matters most to them.

Reports are the backbone of performance tracking and data-driven management in any Salesforce-driven workflow.

**📍Milestone 13**

**🔍 What Is a Report in Salesforce?**

A **report** is a dynamic, customizable view of Salesforce data—structured in rows and columns, often grouped or summarized. It pulls data from objects like Opportunities, Cases, Leads, etc., based on filters and logic you define.

**🎯 Why We Create Reports**

* **Performance Tracking**
  + Monitor KPIs like sales revenue, case resolution time, or lead conversion rates.
  + Example: A report showing monthly revenue by sales rep helps assess individual performance.
* **Trend Analysis**
  + Identify patterns over time—like seasonal dips in sales or recurring support issues.
  + Example: A matrix report showing product sales by region and quarter.
* **Operational Oversight**
  + Ensure teams are meeting SLAs, handling workloads efficiently, and following processes.
  + Example: A case report grouped by priority and owner to track support backlog.
* **Decision Support**
  + Provide leadership with real-time insights for strategic planning.
  + Example: A dashboard combining opportunity pipeline and forecast reports.
* **Data Sharing & Collaboration**
  + Share filtered views with stakeholders via dashboards or scheduled emails.
  + Example: Weekly report emailed to managers showing open opportunities by stage.

**📊 Common Report Formats**

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | **Use Case** | **Grouping** | **Charts Supported** |
| Tabular | Simple lists (e.g., all open leads) | ❌ | ❌ |
| Summary | Grouped data (e.g., cases by priority) | ✅ Rows | ✅ |
| Matrix | Multi-dimensional (e.g., sales by region/time) | ✅ Rows & Columns | ✅ |
| Joined | Compare across objects (e.g., Accounts + Cases) | ✅ Blocks | ✅ |

📊 **Report 1: *Venue and Drop Off Point***

🧾**Description**  
A report built using the custom report type *Venue with DropOff with Volunteer* that visualizes relationships between venues, drop-off locations, and assigned volunteers.

**🔗Configuration**

* **Grouped by**: Volunteer Name
* **Columns**: Venue Name, Drop-Off Point Name, Distance
* **Saved as**: *Venue and Drop Off point* in the *Custom Reports* folder

🎯**Purpose**

* 🧭 Provides clarity on which volunteer is linked to which drop-off site and venue
* 🚚 Helps monitor food distribution distance and logistics
* 📍 Supports planning and optimization of delivery points

**📋 Report 2: *Volunteer Task***

🧾**Description**  
Uses the report type *Volunteers with Execution Details and Tasks* to track volunteer engagement and task execution details.

**🔗Configuration**

* **Grouped by**: Volunteer ID
* **Columns**: Volunteer Name, Task Name, Execution Detail Name, Owner Name, Task Date, Task Rating
* **Saved as**: *Volunteer Task* in the *Custom Reports* folder

🎯**Purpose**

* 👥 Tracks individual volunteer contributions across assigned tasks
* ✅ Enables monitoring of task performance and feedback through ratings
* 📅 Provides insight into execution timelines and responsible users

**📍Milestone 13**

📊 **Dashboard Creation**

Dashboard creation involves assembling various report components—like tables, charts, and images—into a unified, interactive layout. Each dashboard helps track specific operational insights such as venue logistics, volunteer activities, or task execution performance.

🛠️ **Key Steps in the Process**

* **Organize**: Create a dedicated folder (e.g. *Custom Dashboards*) to manage your dashboards.
* **Select Reports**: Choose relevant data sources (like *Venue and Drop-Off Point Report* or *Volunteer Task Report*).
* **Design Components**: Use widgets like Lightning Tables or Line Charts to represent data meaningfully.
* **Customize**: Apply themes (Dark mode), add branding images, and label components for clarity.
* **Save & Deploy**: Finalize the dashboard and place it in the selected folder for access and collaboration.

Creating a **dashboard in Salesforce** is essential for turning raw data into actionable insights. It provides a **visual summary of key metrics** from reports, helping users monitor performance, spot trends, and make informed decisions in real time.

Here’s why dashboards are necessary:

* 📊 **Real-Time Visibility**: Dashboards update dynamically, showing the latest data without manual refreshes.
* 🎯 **Goal Tracking**: Use gauges and charts to track progress against targets like sales quotas or case resolution SLAs.
* 👥 **Team Oversight**: Managers can monitor team performance across roles, regions, or time periods.
* 📈 **Decision Support**: Visualizations help stakeholders quickly interpret complex data and respond strategically.
* 🔄 **Collaboration & Transparency**: Dashboards can be shared across departments to align efforts and foster a data-driven culture.

🎯 **Purpose**

Dashboards serve as quick-reference tools for tracking metrics, identifying trends, improving logistics, and presenting organizational impact. They enhance transparency, performance review, and coordination across teams.

🗂 **Adding Venue and Drop-Off Point Report**

* **Purpose**: To visualize logistical touchpoints such as food collection and delivery sites for better coordination and oversight.
* **Steps Summary**:
  + Create a folder called **Custom Dashboards** to organize custom visualizations.
  + Inside it, build a dashboard named **Organization Details**.
  + Add a **Lightning Table** using the **venue and Drop Off point Report** to present detailed tabular data.
  + Optional: Apply a **Dark Theme** for better contrast and readability.

**📈 Adding Volunteer Task Report**

* **Purpose**: To monitor volunteer activities over time and identify performance trends or gaps.
* **Steps Summary**:
  + Add a new widget and choose **Line Chart** for temporal analysis of task assignments.
  + Use the **Volunteer Task Report** as the data source.
  + Optional: Apply **Dark Theme** for visual consistency.

**🖼️ Adding a Picture to the Dashboard (Optional)**

* **Purpose**: To personalize or visually enhance the dashboard—useful for branding, instructions, or emphasis.
* **Steps Summary**:
  + Download a relevant image to your system.
  + Use the Image widget to upload it.
  + Save it under the name **Task Execution Details** in the **Custom Dashboards** folder.

📌 **Overall Purpose of the Dashboard**  
This setup provides a streamlined, visual representation of organizational logistics, volunteer efforts, and custom branding for internal review or presentation. It helps stakeholders make informed decisions, spot inefficiencies, and demonstrate operational impact clearly.

**📍Milestone 14**

In Salesforce, **sharing rules** are used to automatically grant access to records for users who wouldn’t otherwise have it, based on your **organization-wide defaults (OWD)**.

**🔐 What Are Sharing Rules?**

Sharing rules are **exceptions to OWD settings**, allowing broader access to records for users in specific roles, groups, or territories. They apply to both standard and custom objects.

Types include:

* **Owner-Based**: Share records owned by specific users or roles.
* **Criteria-Based**: Share records that meet defined field conditions.
* **Guest User Rules**: Grant access to unauthenticated users (e.g., website visitors).

**🎯 Purpose of Implementing Sharing Rules**

✅ **Extend Access Beyond Role Hierarchy**  
Useful when users outside the hierarchy need visibility into records.

📈 **Enable Collaboration Across Teams**  
Share leads, cases, or opportunities between departments for joint efforts.

🔍 **Control Data Visibility Securely**  
Maintain privacy while allowing access only where needed.

🛠️ **Automate Record Sharing**  
Avoid manual sharing by setting rules that apply dynamically.

📊 **Support Business Processes**  
Ensure users have access to data relevant to their function—e.g., marketing seeing high-priority leads.

Salesforce offers several types of **sharing rules** to extend record access beyond the default settings.

**🔑 Types of Sharing Rules in Salesforce**

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Use Case Example** |
| **Owner-Based** | Shares records based on who owns them | Share leads owned by Sales team with Marketing team |
| **Criteria-Based** | Shares records that meet specific field conditions | Share cases where Priority = High with Support team |
| **Guest User Sharing** | Grants access to unauthenticated users (e.g., website visitors) | Share public knowledge articles on Experience Cloud |
| **Territory-Based** | Shares records based on user territories (Enterprise Territory Management) | Share accounts within a geographic region |

Each rule includes:

* **Which records to share** (owner or criteria)
* **Who to share with** (roles, groups, territories)
* **Access level** (Read-Only or Read/Write)

**🔐 Sharing Rules**

Sharing Rules in Salesforce define how records are shared with specific users or groups based on criteria or ownership. They help extend access to users who don’t normally have visibility based on role hierarchy or object-level permissions.

🎯 **Purpose**

* Enhance collaboration by granting visibility to records that meet specific criteria (e.g., location, status, distance).
* **Types**:
  + **Owner-based**: Shares records owned by certain users.
  + **Criteria-based**: Shares records that meet field conditions.

**🚚 Explanation of Drop-Off Point Sharing Rules**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rule** | **Criteria** | **Shared With (Group)** | **Purpose** |
| Rule 1 | Distance < 15 | Iksha | Gives nearest volunteers visibility into nearby drop-off points for quicker access. |
| Rule 2 | Distance > 15 AND Distance ≤ 30 | NSS | Enables mid-range volunteers to plan tasks efficiently for moderately distant points. |
| Rule 3 | Distance > 30 AND Distance ≤ 50 | Street Cause | Shares far-distance drop-off locations for long-range task assignments or monitoring. |

* All rules use **criteria-based sharing** to dynamically allocate access based on distance fields.
* Each rule specifies a **Public Group** to streamline visibility among relevant teams.
* “Near Share With” is likely a label in the UI or configuration tag denoting the access pathway.

**🎯 Overall Purpose**

These sharing rules optimize **geographic task distribution**, ensuring volunteers from different groups can access relevant drop-off data based on location proximity. It promotes efficient routing, clearer responsibilities, and better utilization of resources across the organization.

**📍Milestone 14**

In Salesforce, the **Home Page** is the first screen users see after logging in. It’s designed to provide **quick access to key tools, data, and actions** tailored to each user’s role.

**🎯 Purpose of Creating a Home Page**

* **Centralized Workspace**  
  Displays recent records, tasks, events, and updates—reducing navigation time.
* **Role-Based Insights**  
  Custom components show relevant metrics, dashboards, or list views based on user profiles.
* **Improved Productivity**  
  Users can act on reminders, view performance summaries, and access frequently used features instantly.
* **Personalized Experience**  
  Admins can design different home pages for different apps or profiles using Lightning App Builder.

To create a **custom Home Page in Salesforce Lightning**, follow these steps using the **Lightning App Builder**:

**⚙️ Steps to Create a Home Page**

1. **Go to Setup**
   * Click the gear icon → Select **Setup**.
2. **Open Lightning App Builder**
   * In the Quick Find box, search for **Lightning App Builder** → Click it.
3. **Create New Page**
   * Click **New** → Choose **Home Page** → Click **Next**.
4. **Name & Template**
   * Enter a label (e.g., “Food Distribution Home”) → Select a layout template → Click **Done**.
5. **Add Components**
   * Drag standard components like:
     + **Assistant** (updates on Leads/Opportunities)
     + **Performance** (sales metrics)
     + **Today’s Tasks/Events**
     + **Key Deals**, **Reports**, or **Dashboards**
6. **Set Component Visibility (Optional)**
   * Configure filters to show components based on user profile or role.
7. **Save & Activate**
   * Click **Save** → Then **Activate**.
   * Choose activation scope:
     + **Org Default**
     + **App Default**
     + **App + Profile Assignment**

**🖼️ Creation of home Page**

The described Home Page setup using Lightning App Builder is designed to provide a customized, intuitive landing experience for users of the **FoodConnect App** in Salesforce.

**🧩 Description:**

* A **Standard Home Page** layout integrating two primary components:
  + A **Flow component** on the right panel to guide users through the “Venue Flow” process interactively.
  + A **Dashboard component** in the top section for real-time visual insights and metrics related to FoodConnect operations.
* Set as **App Default** for FoodConnect, ensuring all users land on this page when accessing the app.

**🎯 Purpose:**

* **Streamline User Experience**: Centralizes key actions (via Flow) and important data (via Dashboard) in one place.
* **Boost Productivity**: Ensures users can initiate processes and review performance immediately upon logging in.
* **Tailored Access**: Activation for the FoodConnect App aligns the page to its users' specific needs, improving app relevance and usability.

🧭 **Objective:**

* Offer a simplified, actionable entry point into the FoodConnect App.
* Enable users to initiate workflows and monitor performance from a single screen.
* Improve efficiency, clarity, and user engagement by combining process automation with data visualization.

Salesforce is a **comprehensive cloud-based CRM platform** that empowers businesses to manage customer relationships, automate workflows, and drive growth across sales, service, marketing, and analytics.

**🧭 Final Takeaways on Salesforce**

* **Unified Customer View**: Centralizes data across departments, enabling personalized engagement and smarter decision-making.
* **Scalable & Customizable**: Adapts to businesses of all sizes with flexible architecture, metadata-driven development, and robust APIs.
* **Automation & AI**: Streamlines operations using tools like Flow, Process Builder, and Einstein AI for predictive insights.
* **Ecosystem & Integration**: Offers thousands of apps via AppExchange and seamless integration with external systems.
* **Security & Compliance**: Built on a secure, multitenant infrastructure with global data residency support via Hyperforce.

Salesforce continues to evolve with innovations like generative AI, real-time data processing, and industry-specific solutions—making it a strategic platform for digital transformation.

Would you like a visual summary or a tailored conclusion for your food distribution project?

**Conclusion**

The project utilized the Salesforce platform to create a clear and efficient system for surplus food management. By organizing volunteer efforts and ensuring prompt food distribution, it successfully reduced food insecurity and optimized resource use.Building on this system, the project enhanced communication among stakeholders and fostered a reliable network for food redistribution. Its impact extended beyond logistics—promoting community engagement, reducing waste, and supporting sustainable development goals.