SALESFORCE PROJECT IMPLEMENTATION PHASES

Phase 1: Problem Understanding & Industry Analysis

Problem: During natural disasters (floods, earthquakes, cyclones), victims often struggle to request essential resources efficiently, and NGOs/government agencies have difficulty tracking requests and assigning volunteers/resources.

Industry Analysis:

- Sector: Disaster Management / Humanitarian Aid
- Current Challenges:
  - Delayed response due to manual request tracking
  - Lack of real-time visibility of volunteer availability
  - Inefficient allocation of resources
- Salesforce Role: Centralized system to manage requests, volunteers, and resources with automation and dashboards.

1. Requirement Gathering

- Victims should be able to submit requests for essential resources (food, water, medicine, shelter).
- Volunteers should be able to view and accept/complete assignments.
- Coordinators should have a dashboard to track requests, volunteers, and resources.
- The system should automatically assign volunteers/resources based on availability and priority.
- Notifications/alerts for pending or overdue requests.
- Reports and analytics for completed vs pending requests, resource utilization, and volunteer workload.

2. Stakeholder Analysis

Stakeholder	Role	Expectations
Victims (Affected People)	Submit resource requests	Easy-to-use portal, real-time tracking
Volunteers	Fulfill requests	Clear assignments, update status easily
Coordinators/NGOs	Manage relief operations	Central dashboard, automated assignment, analytics
Government Agencies	Oversight/monitoring	Reports, transparency, faster response

3. Business Process Mapping

Process Flow:

1 Victim submits a request.

2 System validates request and auto-assigns volunteer/resource.

3 Volunteer receives task notification and updates status.

4 Coordinator tracks progress and reallocates if needed.

5 Reports and dashboards provide insights into overall relief efforts.

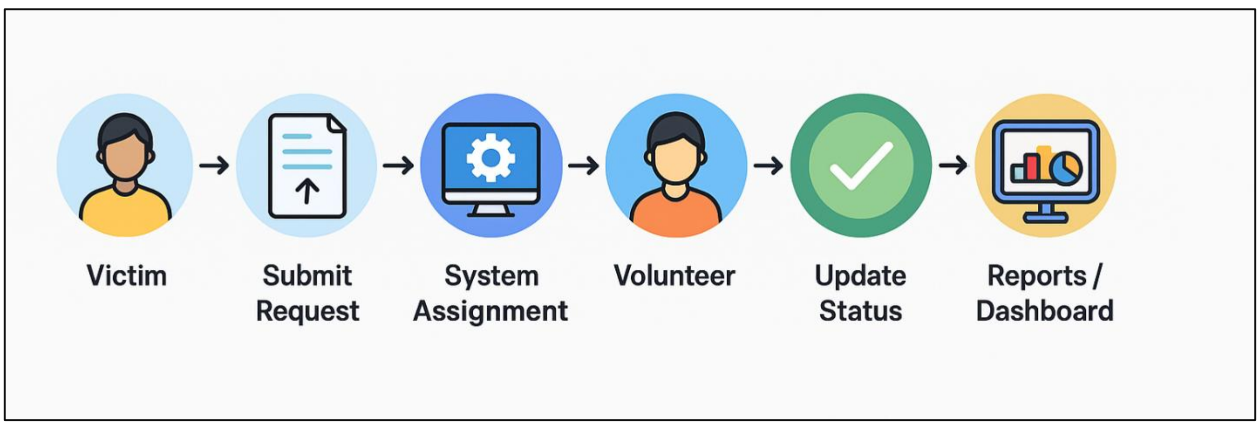


Figure 1. Process Flow Diagram

4. Industry-specific Use Case Analysis

- Use Case: Flood Relief in a rural district
- Victims request food, water, and shelter.
- Volunteers from NGOs and community groups are available for deployment.
- Coordinators need real-time tracking of requests and volunteer workload.
- Outcome: Faster allocation of relief resources, reduced duplication of effort, and improved victim support.

5. AppExchange Exploration

- Volunteer Management Apps (for NGOs) → provide partial solutions but are generic.
- Nonprofit Success Pack (NPSP) → widely used in nonprofit operations but not disaster-specific.
- Case Management Apps → good for tracking issues but lack volunteer/resource assignment logic.
- **Differentiation:** Our solution focuses specifically on disaster relief scenarios, integrates volunteer and resource assignment, and includes LWC dashboards for real-time tracking — all built in a free Salesforce Developer Org.

Phase 2: Org Setup & Configuration – ReliefConnect *(Disaster Relief & Resource Management System)*

0. Preliminaries

1. Sign up / log into **Salesforce Developer Edition** (https://developer.salesforce.com).

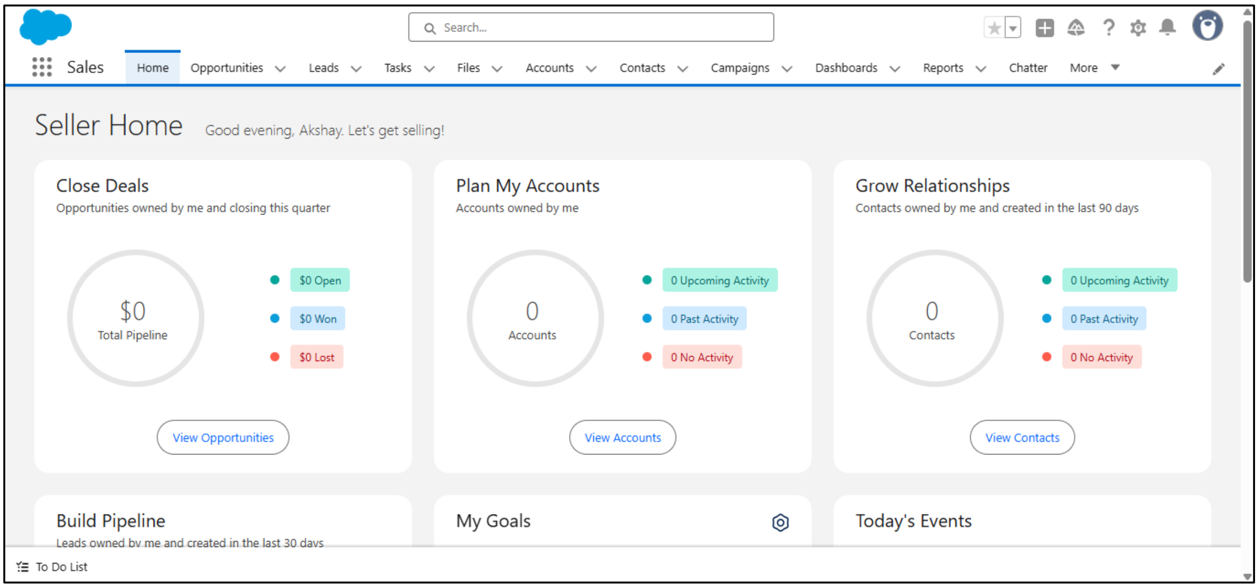


Figure 1 Developer Org home

1. Company Profile (Company Information)

1. Setup → Quick Find: Company Information → Open.
2. Edit → set Organization Name = **ReliefConnect**
   - Set Default Time Zone = Asia/Kolkata, Default Currency = INR. → Save.

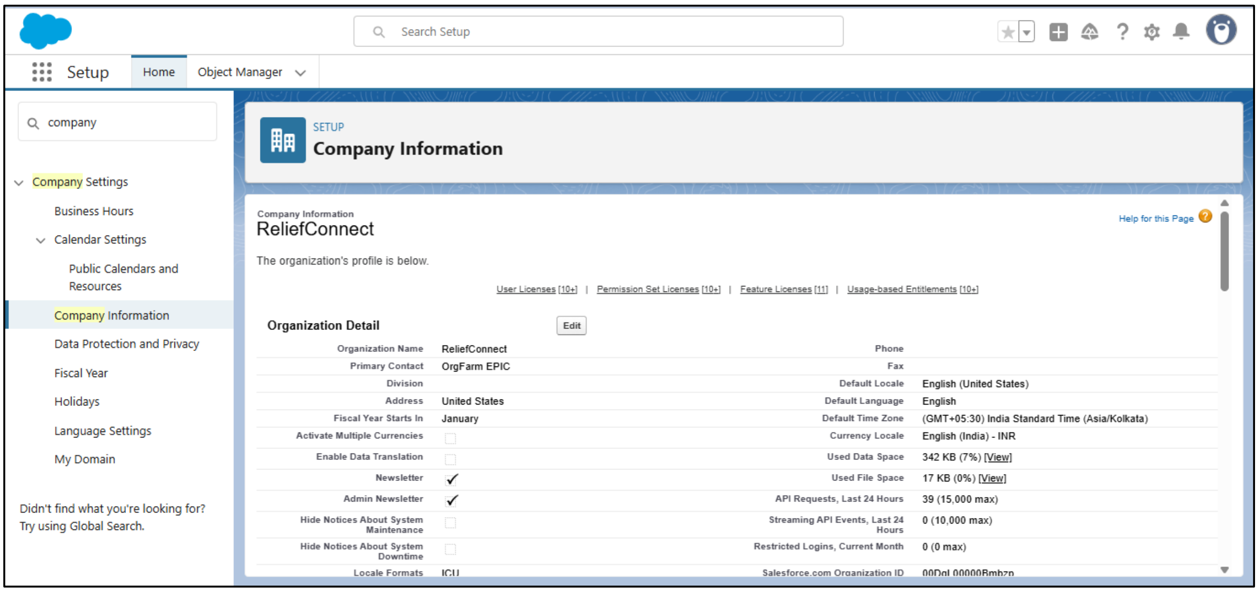


Figure 2 Company info with org name, timezone, currency.

2. Business Hours & Holidays

1. Setup → Quick Find: **Business Hours** → New Business Hours.
    - Label: Emergency Operations (24x7) → Set Mon–Sun 12:00 AM–12:00 AM → Save.

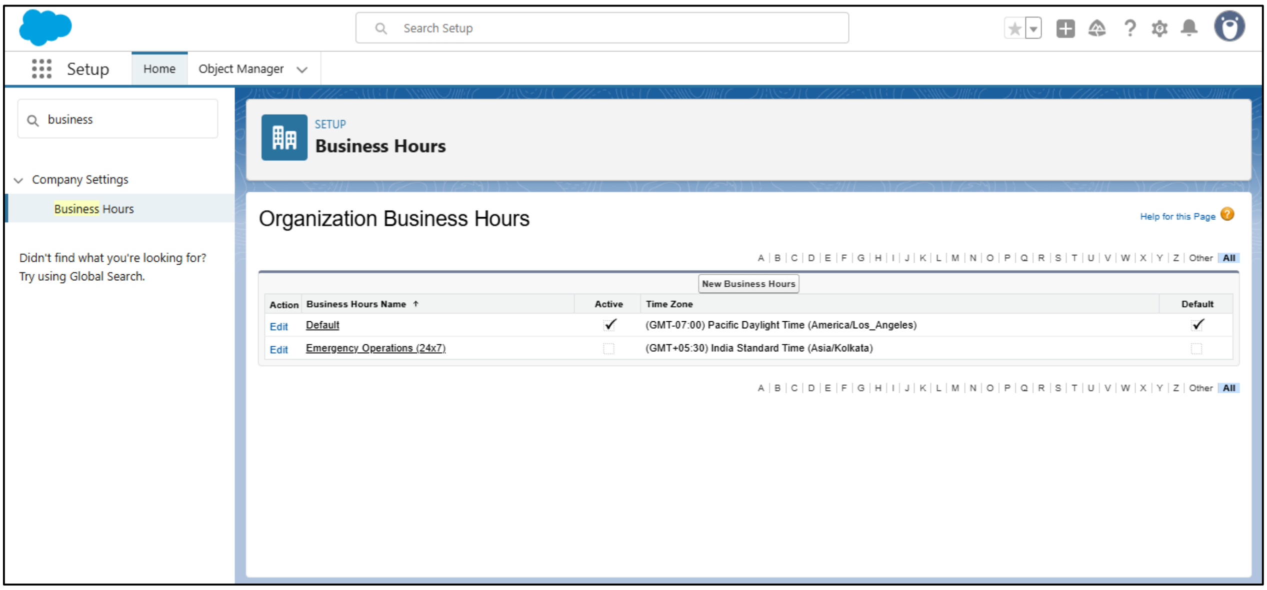


Figure 3 Business Hours

Setup → Quick Find: **Holidays** → New Holiday.

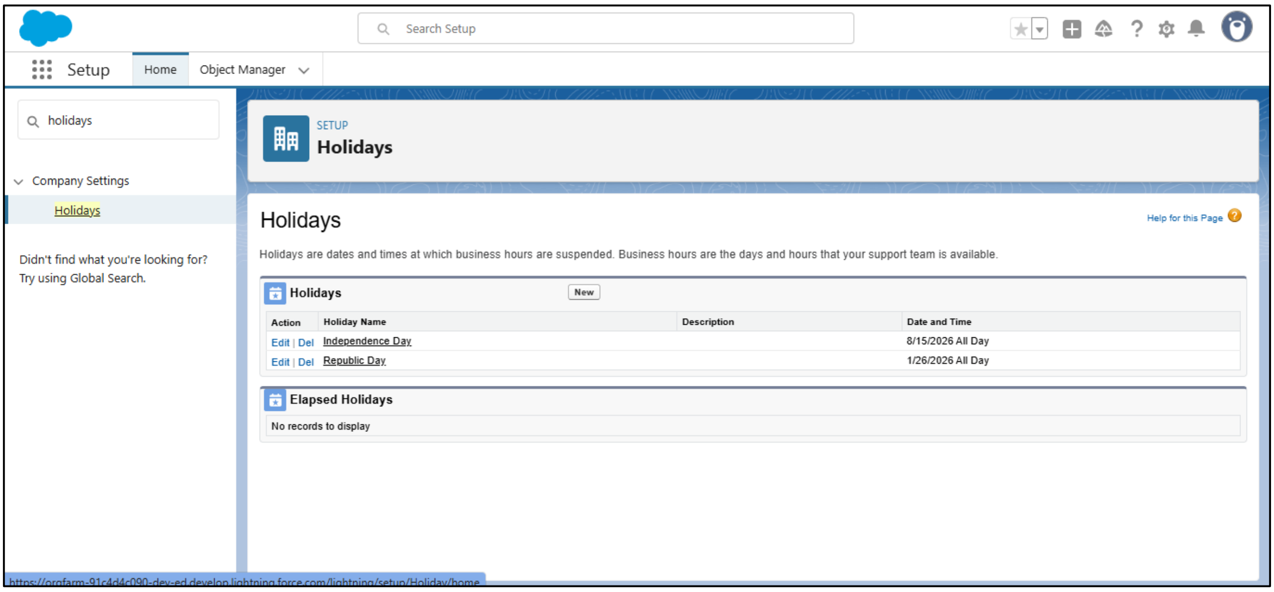


Figure 4 Holidays

3. Fiscal Year

1. Setup → Quick Find: **Fiscal Year** → Verified default fiscal year (Standard). set **April-March**.

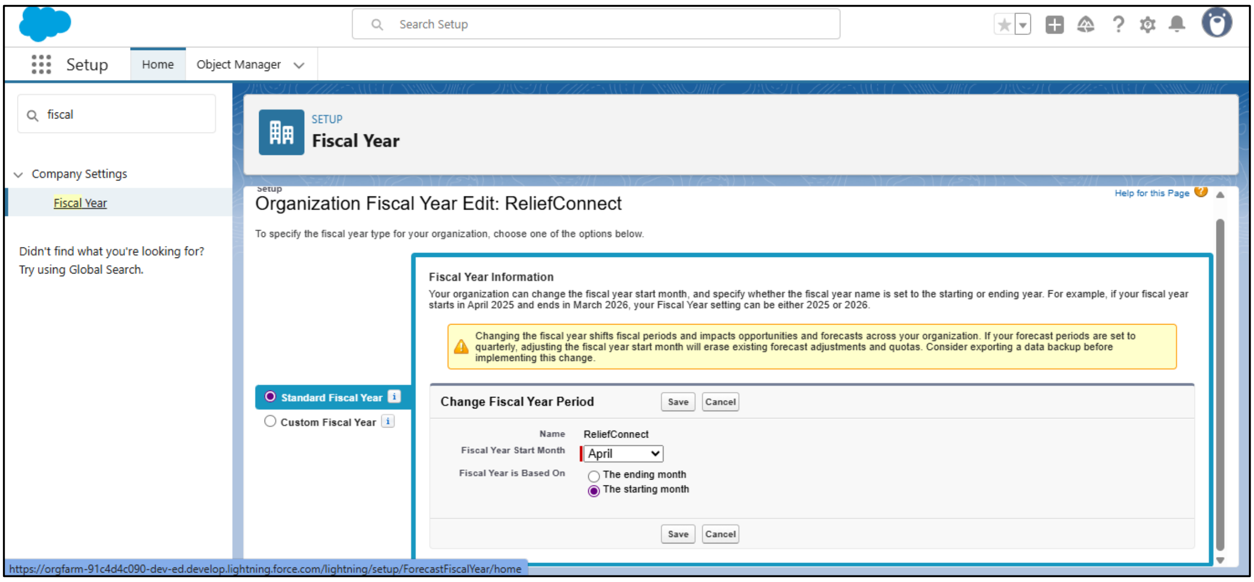


Figure 5 Fiscal Year

4. Create Sample Users & Assign Licenses

1. Setup → Quick Find: **Users** → Users → New User. Create three users:
   - Relief Head — relief.head@demo.com — Profile: System Administrator
   - Regional Coordinator — coordinator@demo.com — Profile: Standard User
   - Volunteer — volunteers@demo.com — Profile: Standard User
2. Fill First/Last name, Username (unique), Email, License = Salesforce or Platform if available. Save

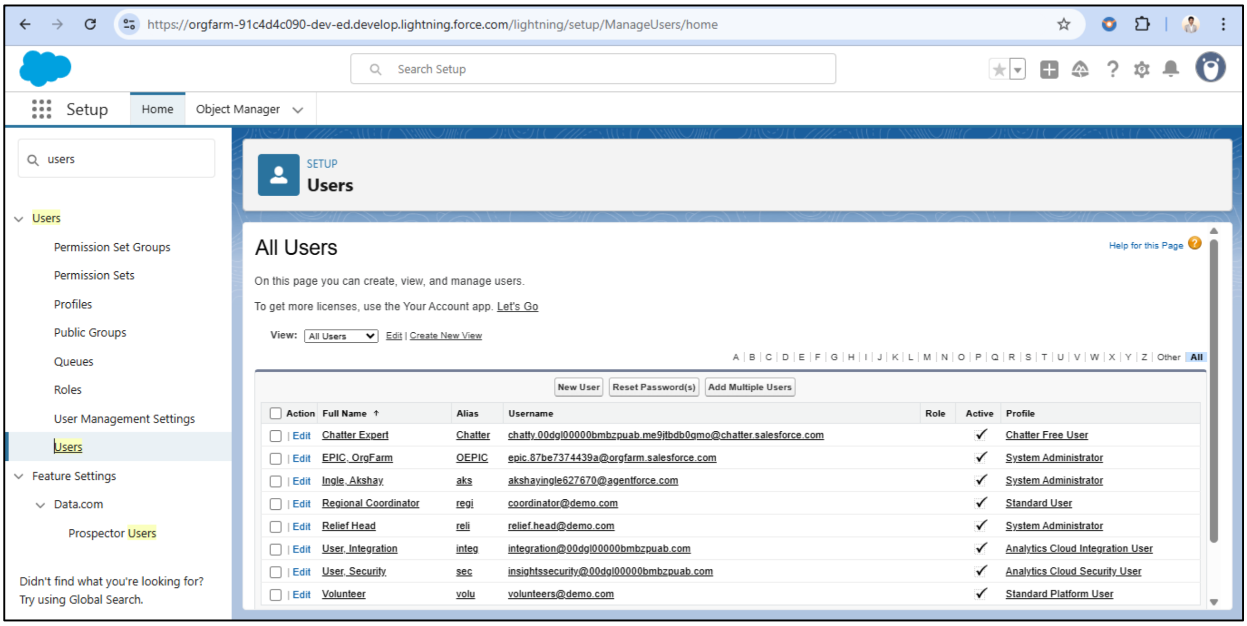


Figure 6 Sample Users

5. Role Hierarchy

1. Setup → Quick Find: **Roles** → Click **Set Up Roles** → Expand and **Add Role**:
   - Top Role: Relief Head
   - Child under Relief Head: Regional Coordinator
   - Child under Regional Coordinator: Volunteer
2. Save.

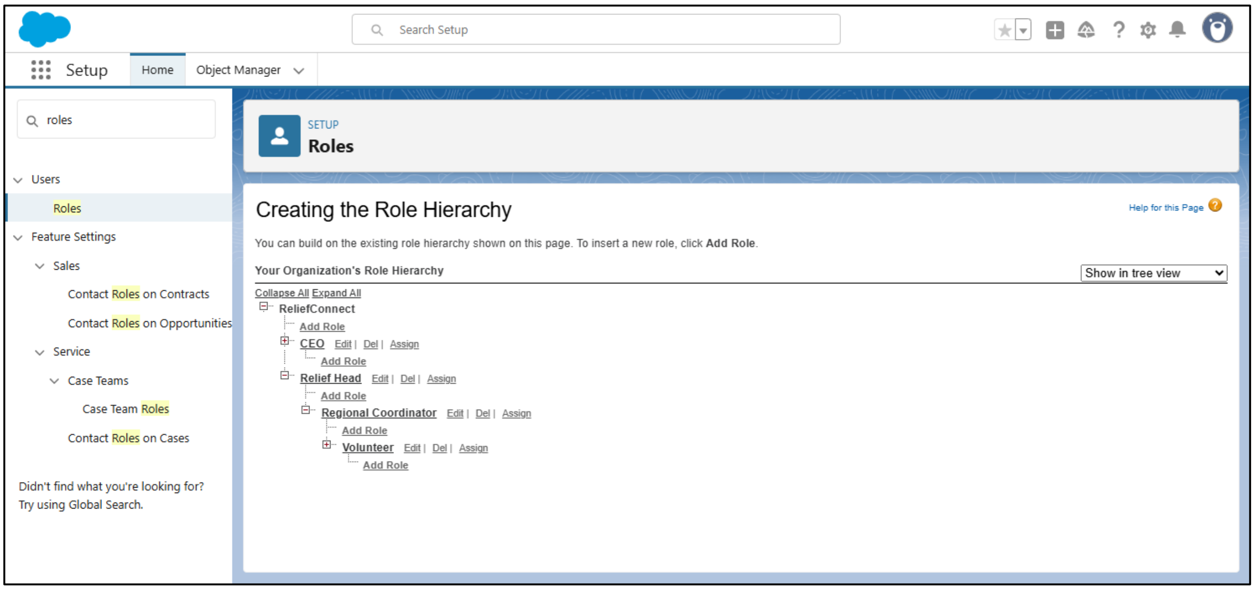


Figure 7 Role hierarchy tree (Relief Head > Coordinator > Volunteer).

6. Profiles (clone & edit)

1. Setup → Quick Find: **Profiles** → Open Standard User → Click **Clone** → Name: Coordinator Profile.
2. Clone another profile → Volunteer Profile

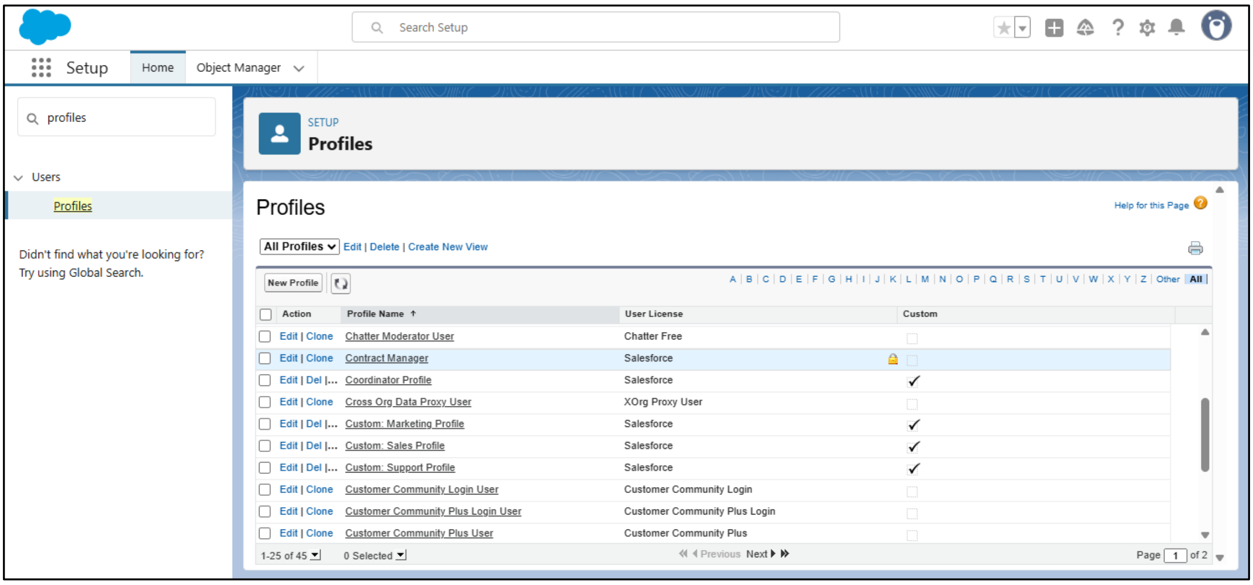


Figure 8 Coordinator Profile object permissions

7. Organization-Wide Defaults (OWD)

1. Setup → Quick Find: **Sharing Settings** → Edit.
2. Set defaults:
   - **Requests** → Private
   - **Resources** → Public Read Only
   - **Volunteer** → Private
3. Save.

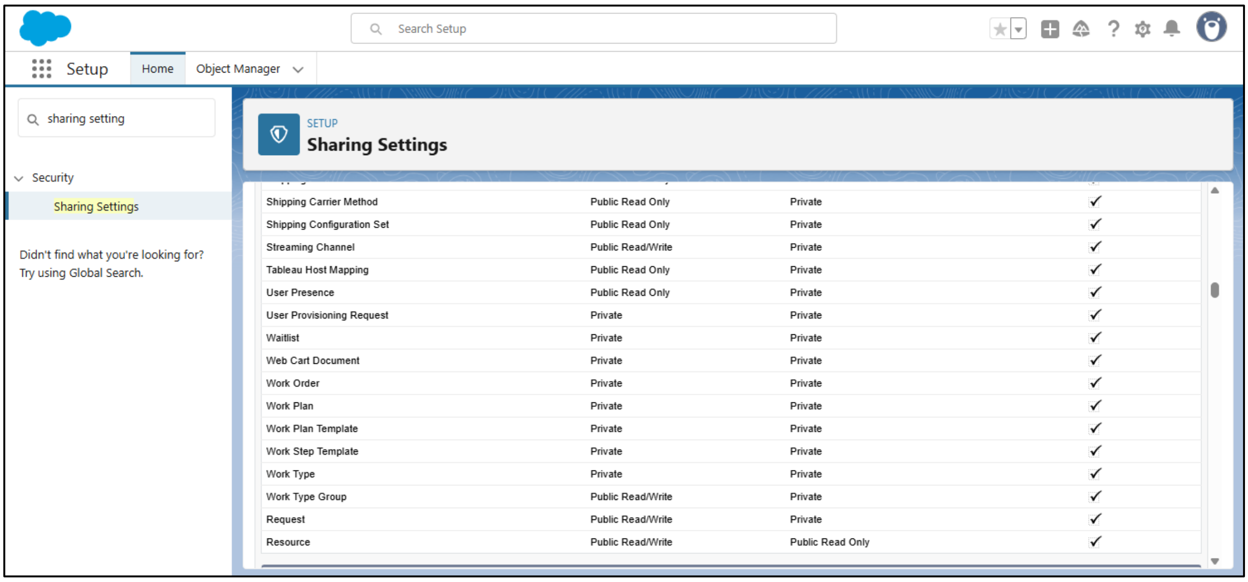


Figure 9 OWD table showing Requests private, Resources public read only

8. Sharing Rules

1. Setup → Quick Find: **Sharing Settings** → Scroll to object Requests → Click **New** under Sharing Rules.
2. Create a rule:

- Rule Type: **Owner based**
- Owned By: Regional Coordinator (or Owner = Volunteer) → Share with: Role: Regional Coordinator or a Public Group.
- Access Level: Read/Write. → Save.

9. Login Access Policies, IP Ranges & Login Hours

1. Setup → Quick Find: **Login Access Policies** → Enable **Administrators Can Log in as Any User**

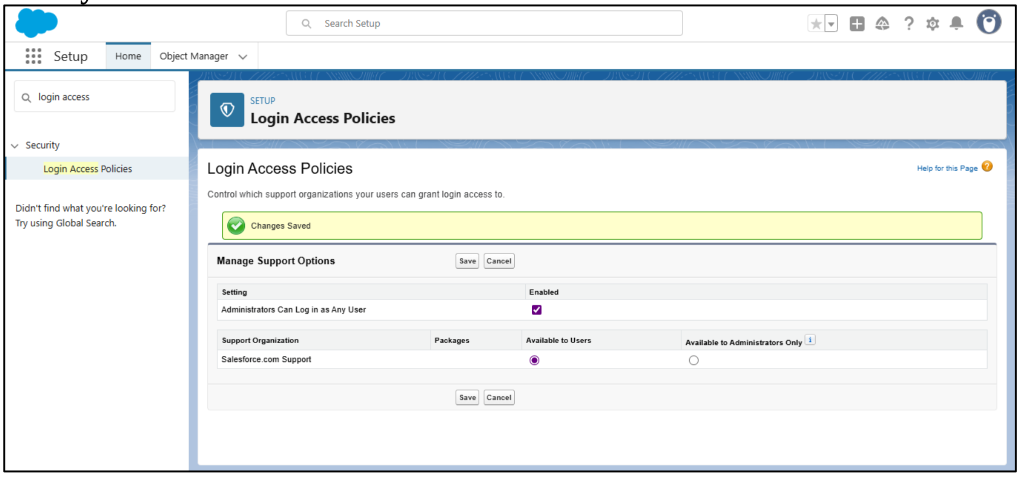


Figure 10 Login Access Policies

2. Setup → Profiles → open Volunteer Profile → Login Hours → Edit → Set 08:00–22:00 for allowed days → Save.

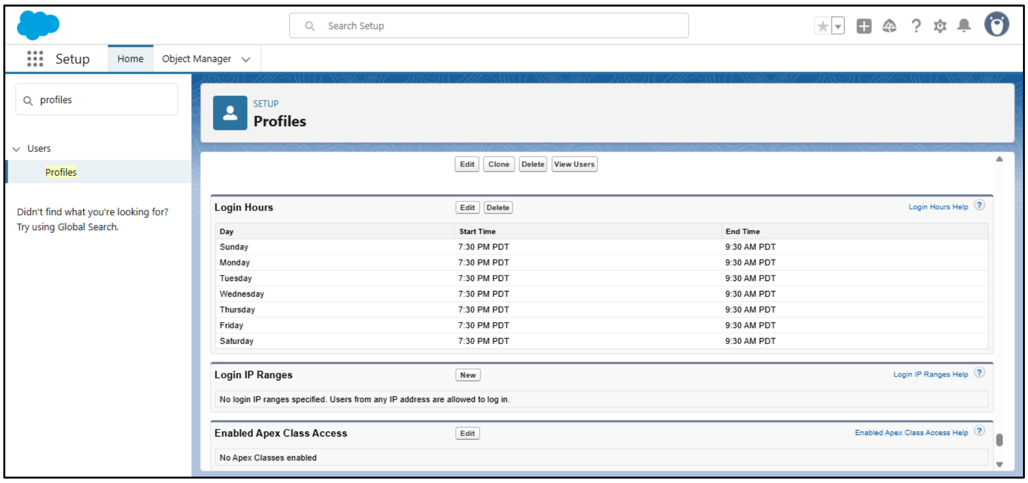


Figure 11 Login Hours

3. Setup → Profiles → open Coordinator Profile → Login IP Ranges → Add IP ranges to restrict logins → Save.

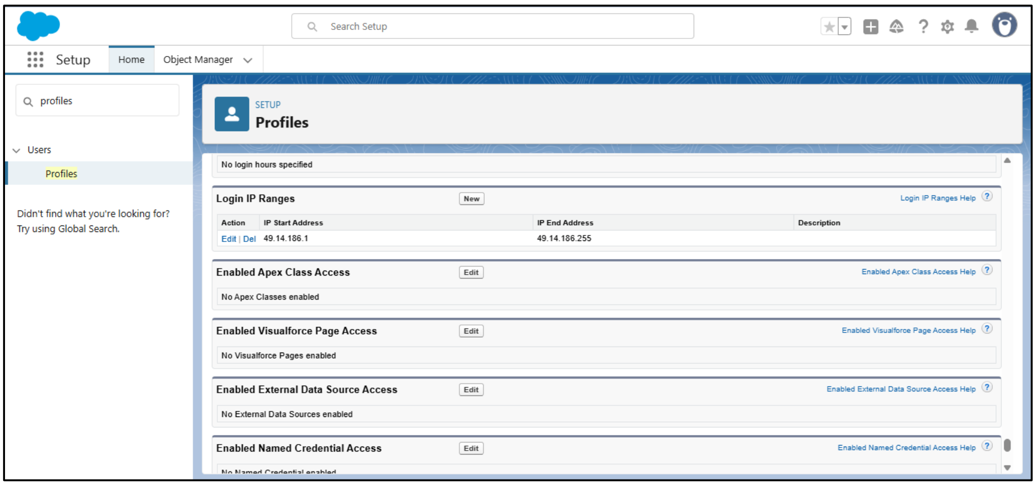


Figure 12 IP Assignment

10. Create Lightning App (ReliefConnect)

1. Setup → Quick Find: **App Manager** → New Lightning App.
   - App Name: ReliefConnect → Branding icon added → Next.
   - Add Nav Items: Requests, Volunteers, Resources, Locations, Reports, Dashboards → Save & Finish.

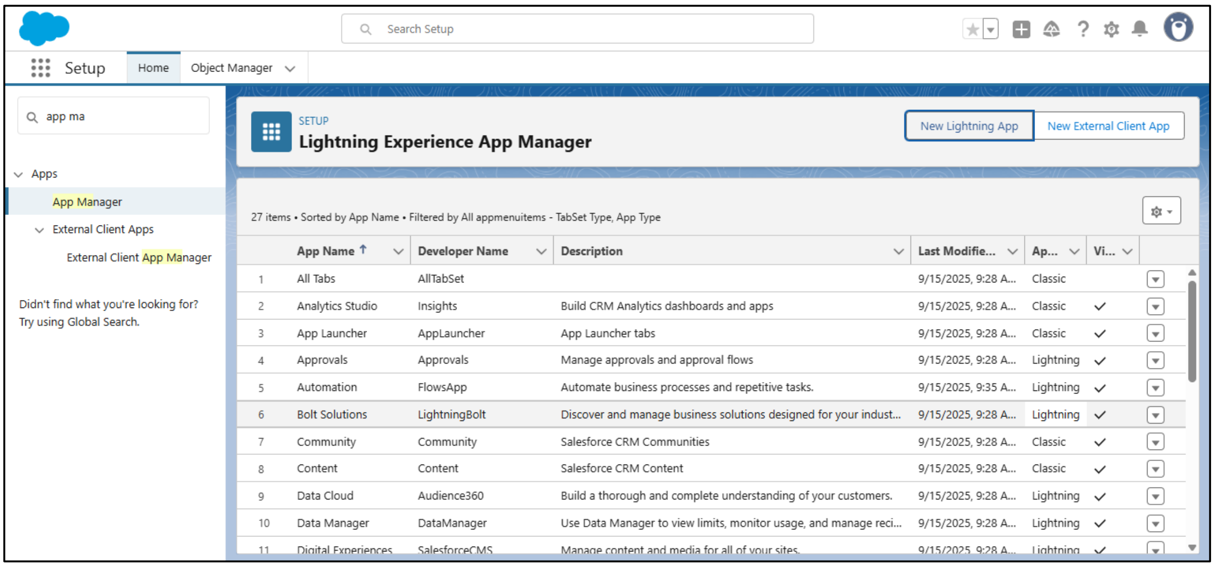


Figure 13 App Manager

App Launcher → search for ReliefConnect → click →ReliefConnect App is opened

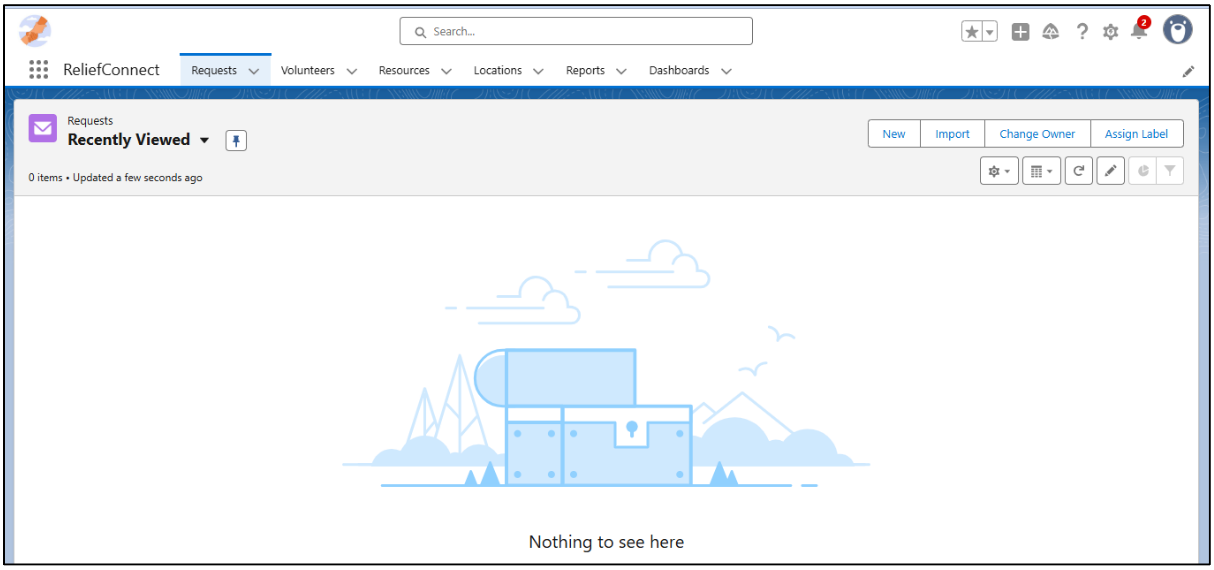
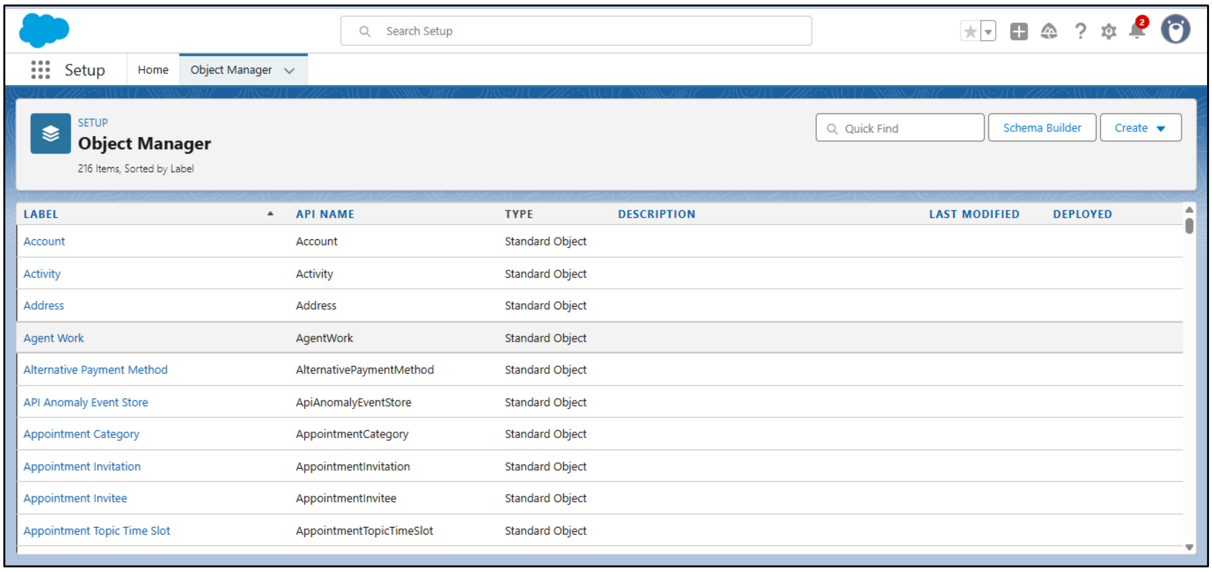


Figure 14 ReliefConnect App

Phase 3: Data Modeling & Relationships

Step 1 — Standard & Custom Objects

1. Go to Setup → Object Manager → Create → Custom Object.
2. Create these objects:
    - Request → for disaster relief requests.
    - Volunteer → for volunteers.
    - Resource → for supplies/resources.
    - Location → for geographical areas.
    - Assignment → junction object between Request, Volunteer, Resource.
3. The **Standard Objects** I have used: User, Contact, Report, etc.

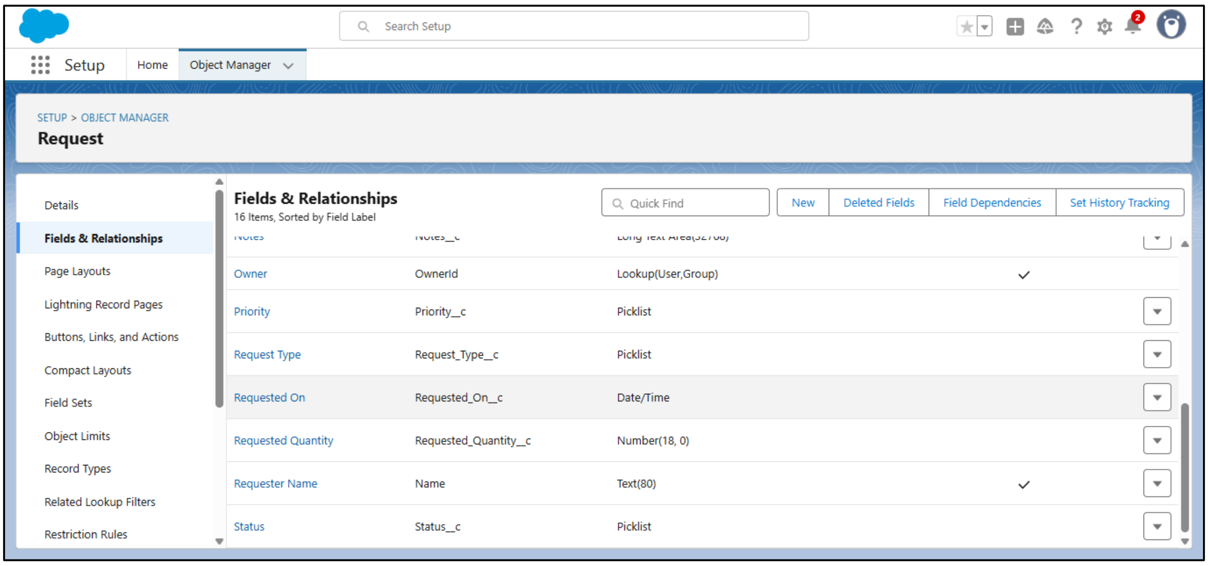


Step 2 — Fields

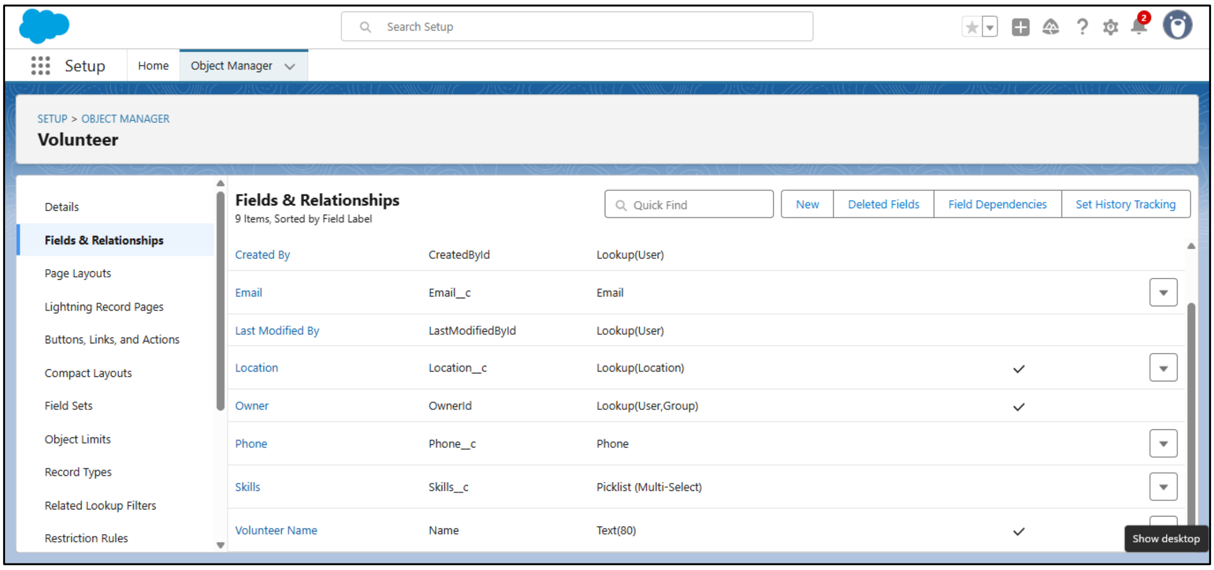
For each object, added fields:

- **Request**
    - Requester Name (Text)
    - Phone (Phone)
    - Email (Email)
    - Request Type (Picklist: Food, Water, Medicine, Shelter, Other)
    - Priority (Picklist: High, Medium, Low)
    - Status (Picklist: Submitted, Assigned, In Progress, Fulfilled, Closed)
    - Location (Lookup → Location)
    - Assigned Volunteer (Lookup → Volunteer)

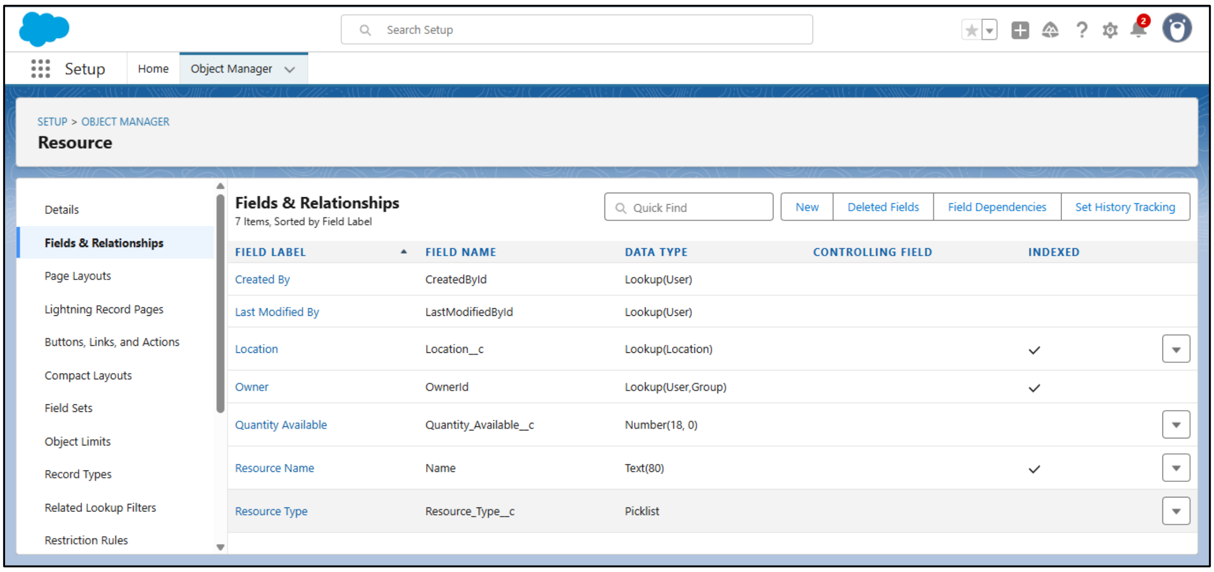
- Assigned Resource (Lookup → Resource)
- Requested On (Date/Time)
- Notes (Long Text Area)



- **Volunteer**
  - Phone (Phone)
  - Email (Email)
  - Skills (Multi-picklist: Medical, Logistics, Transport, Food Distribution)
  - Availability (Picklist: Available, Busy, Inactive)
  - Location (Lookup → Location)

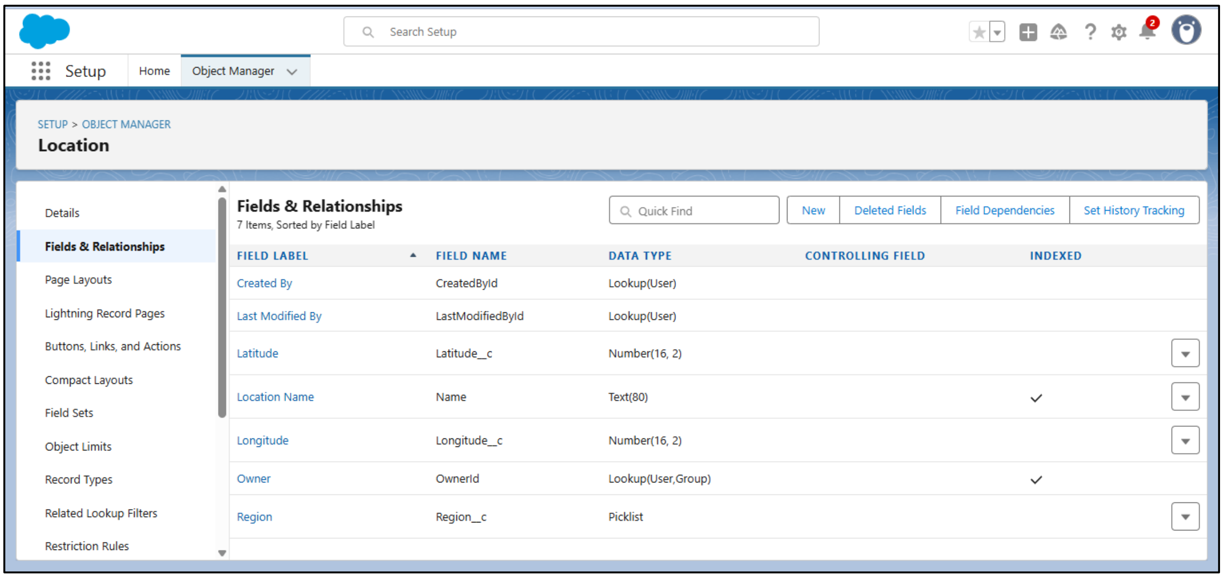


- **Resource**
    - Resource Type (Picklist: Medicine, Food, Shelter Material, Clothing)
    - Quantity Available (Number)
    - Location (Lookup → Location)

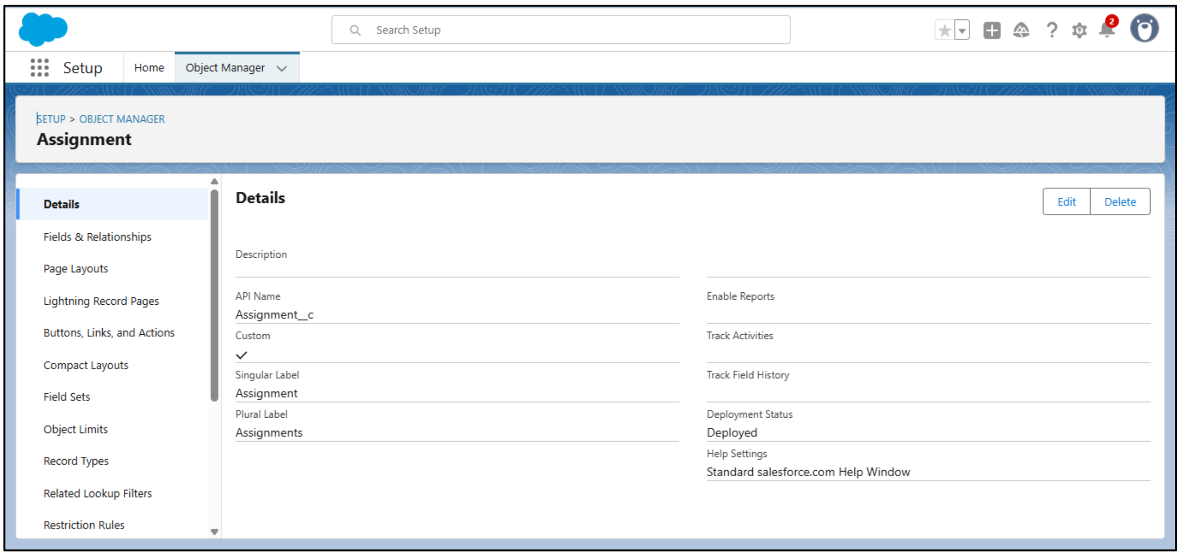


Location

- Region (Picklist: North, South, East, West)
- Latitude (Number, 2 decimal places)
- Longitude (Number, 2 decimal places)

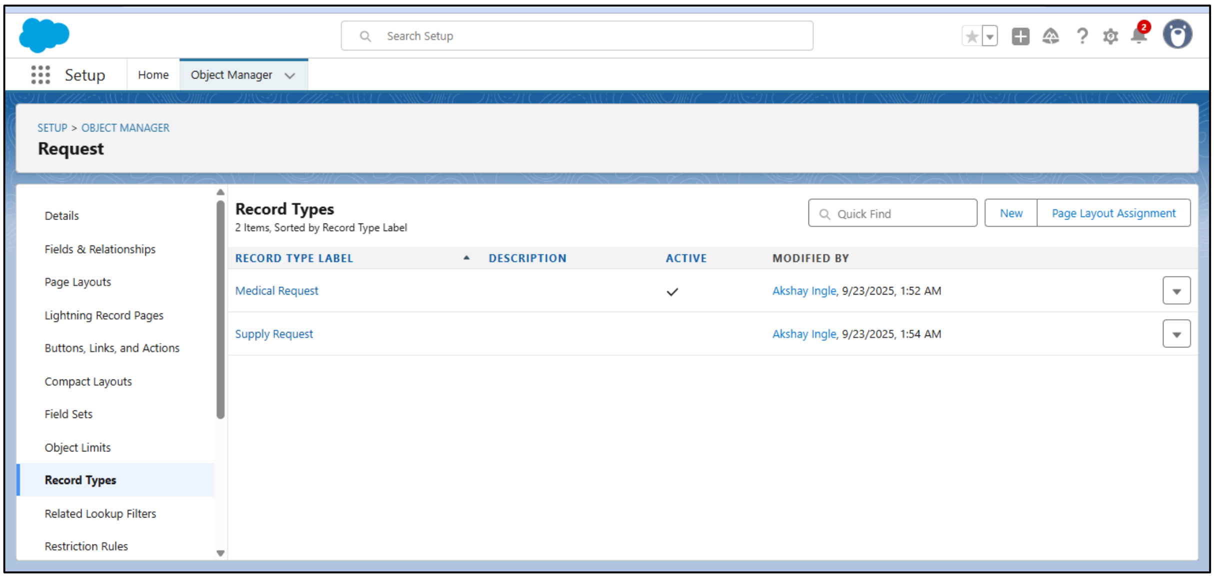


- **Assignment** (junction object)
    - Request (Master-Detail → Request)
    - Volunteer (Lookup → Volunteer)
    - Resource (Lookup → Resource)
    - Status (Picklist: Assigned, In Progress, Completed)
    - Assigned On (Date/Time)



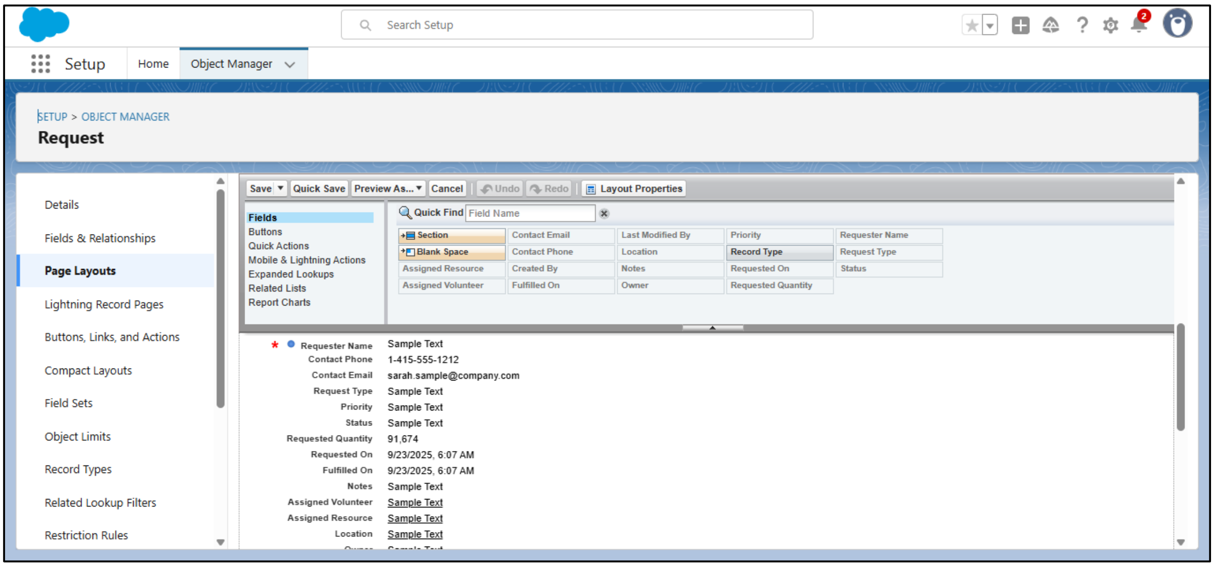
Step 3 — Record Types

- Go to **Object Manager → Request → Record Types → New**.
- Create record types for **Medical Request** and **Supply Request**.
- Assign different **picklist values** and **page layouts**.



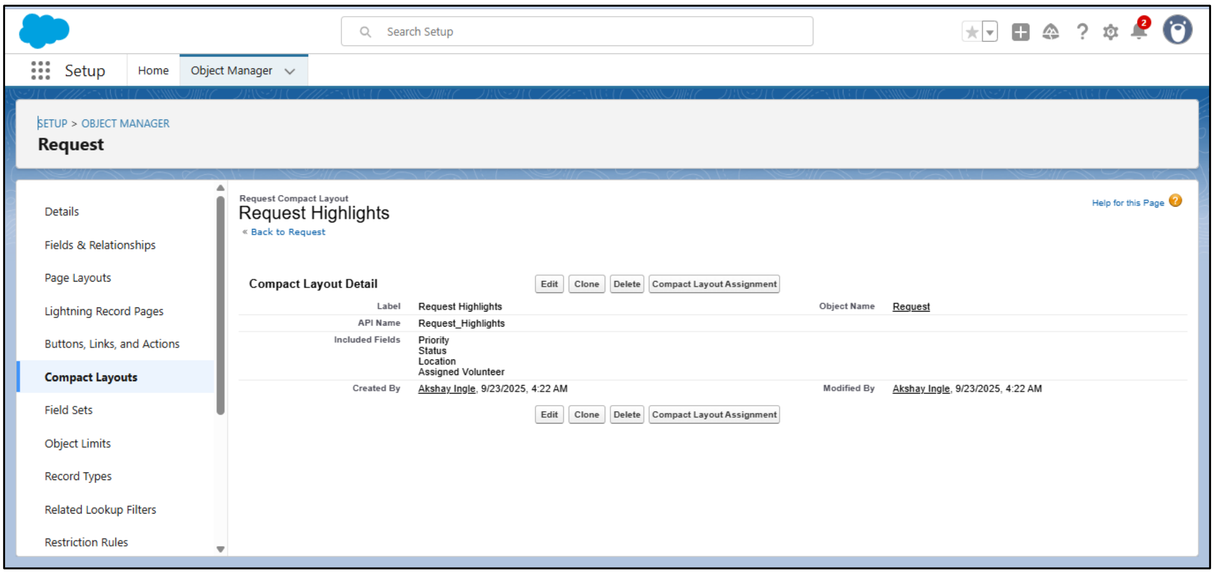
Step 4 — Page Layouts

- Edit **Request Page Layout**: show Requester details, Priority, Status, Assigned Volunteer.
- Edit **Volunteer Page Layout**: show Skills, Availability, Location.
- Different layouts for **Coordinator vs Volunteer** (assign via Profiles).



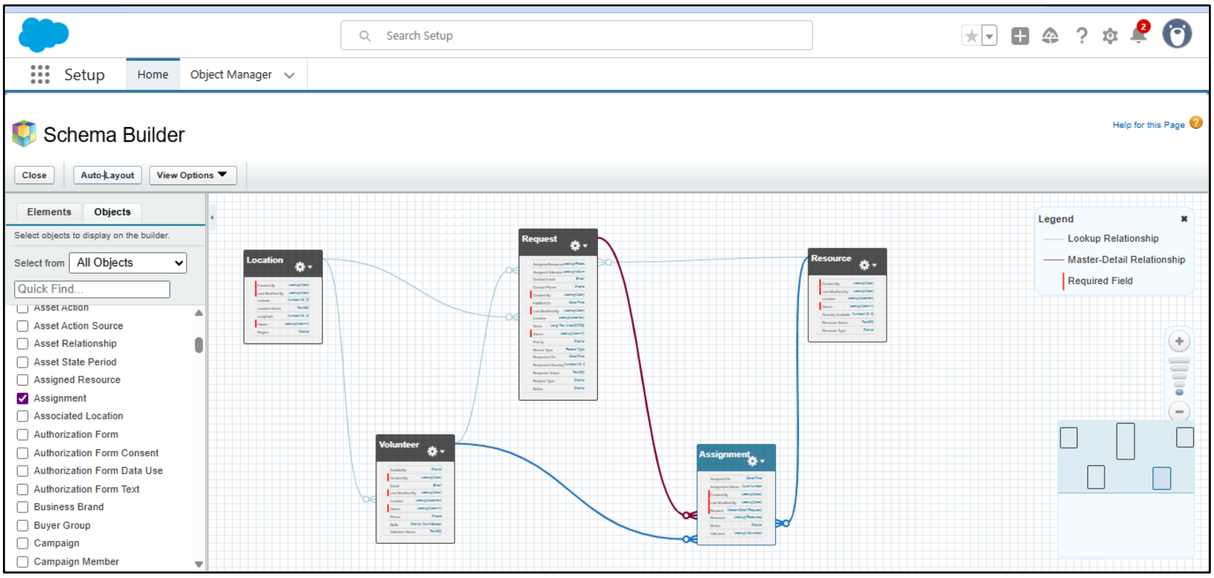
Step 5 — Compact Layouts

- Go to **Object Manager → Request → Compact Layouts → New**.
- Fields to show: Priority, Status, Location, Assigned Volunteer.
- Set as **Primary Compact Layout**.



Step 6 — Schema Builder

- Setup → Quick Find → **Schema Builder**.
- Drag objects: Request, Volunteer, Resource, Location, Assignment.
- Relationships will appear visually (lookups & master-detail).



Step 7 — Lookup vs Master-Detail vs Hierarchical Relationships

• Master-**Detail (Request – Assignment)**

- Reason: An Assignment cannot exist without a Request. If the Request is deleted, all related Assignments should also be deleted.
- Example: If Request #101 (Food Needed in Nagpur) is removed, its related volunteer/resource assignments are also removed.

• Lookup **(Assignment – Volunteer, Assignment – Resource)**

- Reason: A Volunteer or Resource can exist independently, even if they're not currently assigned.
- Example: Volunteer "John Doe" may be free (no assignments) but still exists in the system.

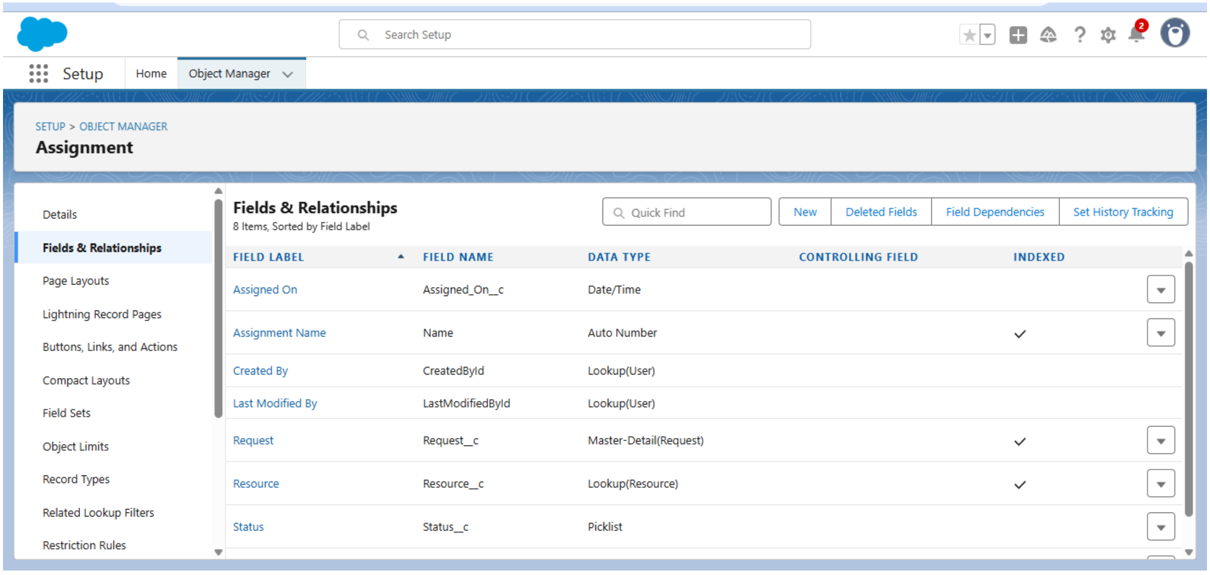
• Lookup **(Request – Location, Volunteer – Location, Resource – Location)**

- Reason: Requests, Volunteers, and Resources are linked to a region (Location), but Locations can exist without them.

- Hierarchical **(Not used here)**

Step 8 — Junction Objects

- Assignment is the **junction object**.
- Links many Volunteers & Resources to many Requests.
- Demonstrate a record where 1 Request is assigned to multiple Volunteers.



Phase 4: Automation & Logic

1 Validation Rules

Purpose: Stop invalid data (prevent save if condition fails).

Example rule: A request cannot be marked **Fulfilled** unless an assigned volunteer exists.

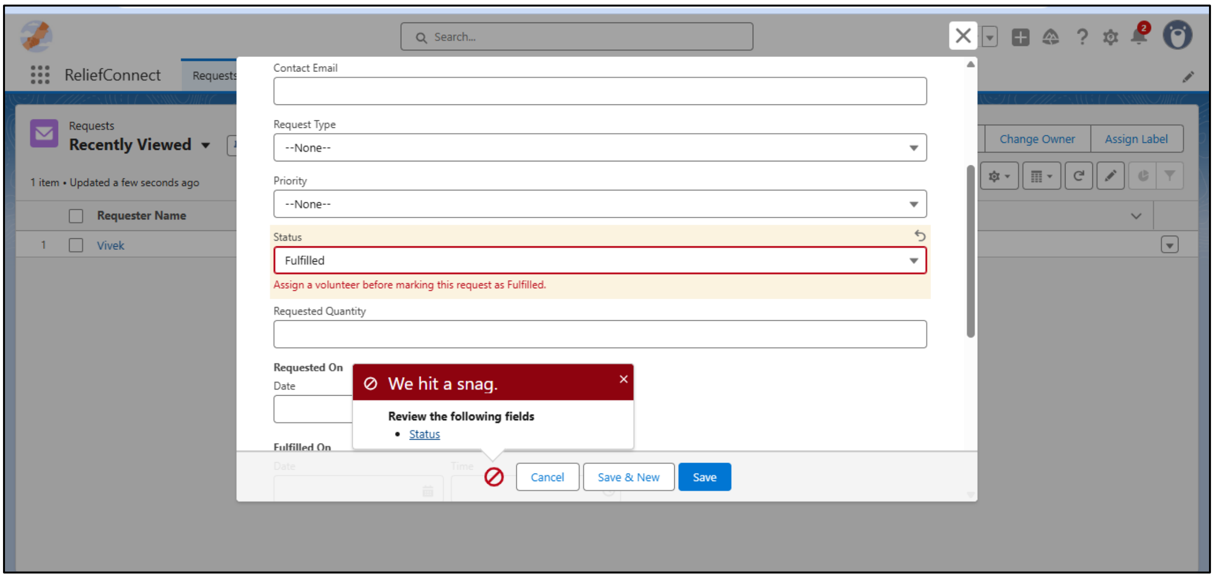
Steps

1. Setup → Quick Find → **Object Manager** → Request → **Validation Rules** → **New**.
2. Fill:
   - **Rule Name:** VR_Request_Fulfilled_Assigned
   - **Error Condition Formula:**

     ```
     AND(
       ISPICKVAL(Status__c, "Fulfilled"),
       ISBLANK(Assigned_Volunteer__c)
     )
     ```
   - **Error Message:** Assign a volunteer before marking this request as Fulfilled.
   - **Error Location:** Field → Status.
3. Save & Activate.

Test

- Open a Request → set Status = Fulfilled without Assigned Volunteer → Save → you should see the error.
- Assign a Volunteer → set Status = Fulfilled → Save succeeds.



2 **Email Alerts** (used by Workflow/Flow)

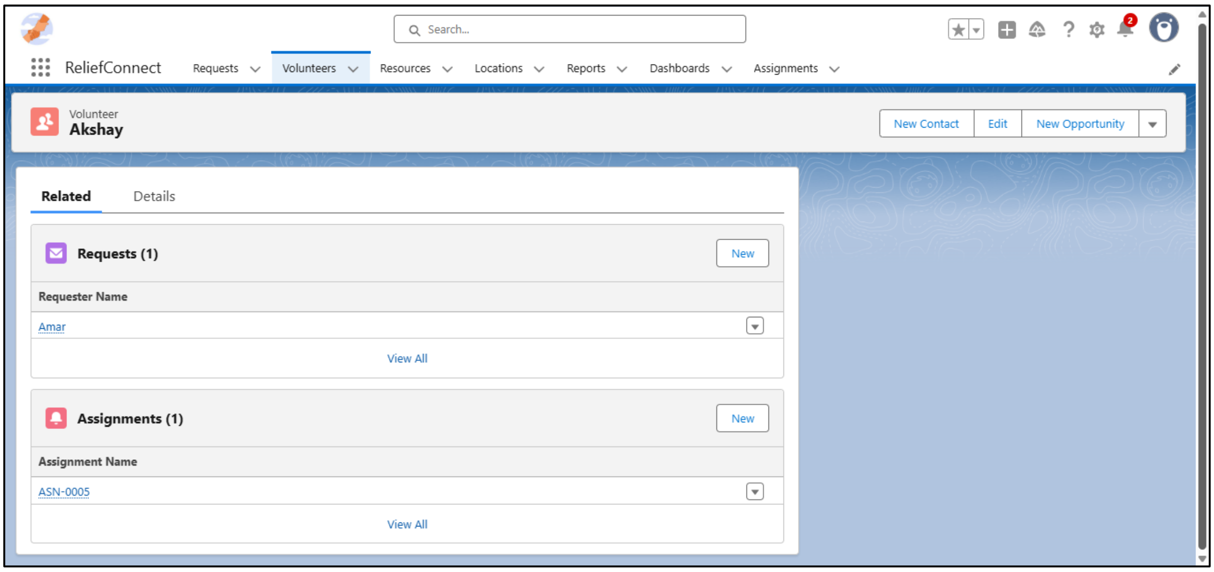
Purpose: send templated emails to users on events.

Steps

1. Create an Email Template:
    - Setup → Quick Find → **Email Templates** → **New Email Template (Lightning)**.
    - Name: ET_Request_Assigned
    - Subject: You have been assigned to Request {!Request__c.Name}
    - Body: include merge fields like {!Request__c.Requester_Name__c}, {!Request__c.Location__c}.
    - Save.
2. Create Email Alert (used by Workflow/Flow):
    - Setup → Quick Find → **Email Alerts** → **New Email Alert**.
    - Label: EA_Request_Assigned
    - Object: Assignment (or Request if you prefer)
    - Email Template: ET_Request_Assigned
    - Recipient Type: User/Email Field → Volunteer__c Save.

Test

- Create an Assignment (or update Request to Assigned) that triggers the Email Alert.
- Check the recipient inbox (or ActivityHistory Email record) in Salesforce for the sent message.



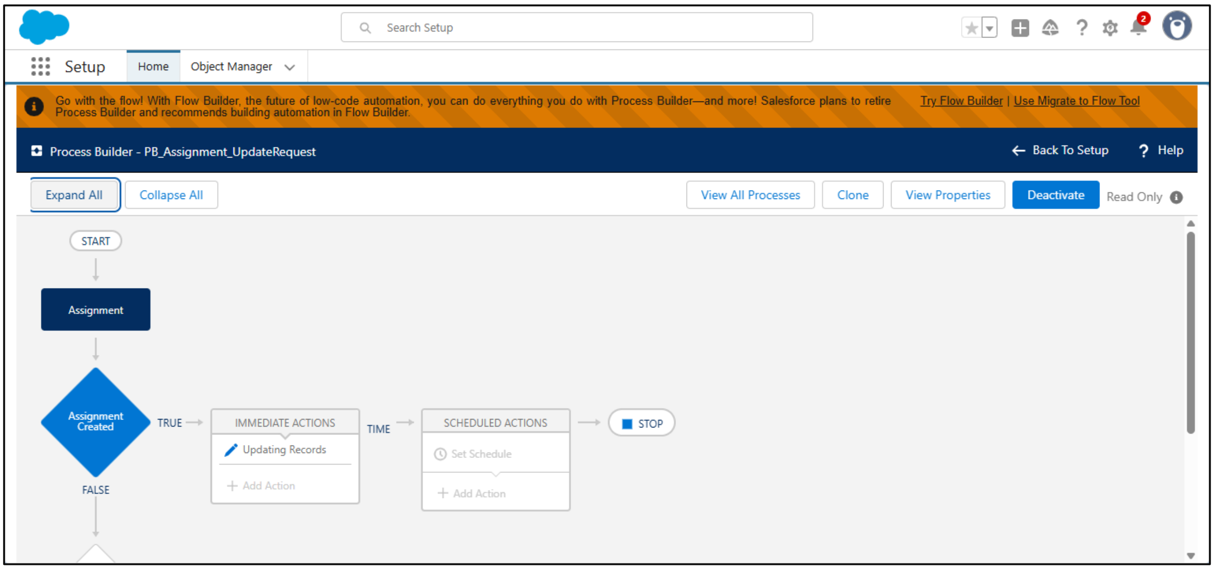
3 Process Builder

Purpose: if/then branching on object events (use Flow instead, but here's how).
Example: When Assignment created → update Request Status to In Progress.
Steps

1. Setup → Quick Find → **Process Builder** → New.
   - Name: PB_Assignment_UpdateRequest.
2. Add Object: Assignment → start the process when a record is created.
3. Add Criteria Node:
   - Criteria Name: Always (or Assignment Created) → no conditions or Status = Assigned.
4. Immediate Actions:
   - Action Type: **Update Records** → select the associated Request record ([Assignment].Request__c) → set Status__c = In Progress.
5. Activate.

Test

- Create an Assignment linked to a Request → the Request's status should be updated.



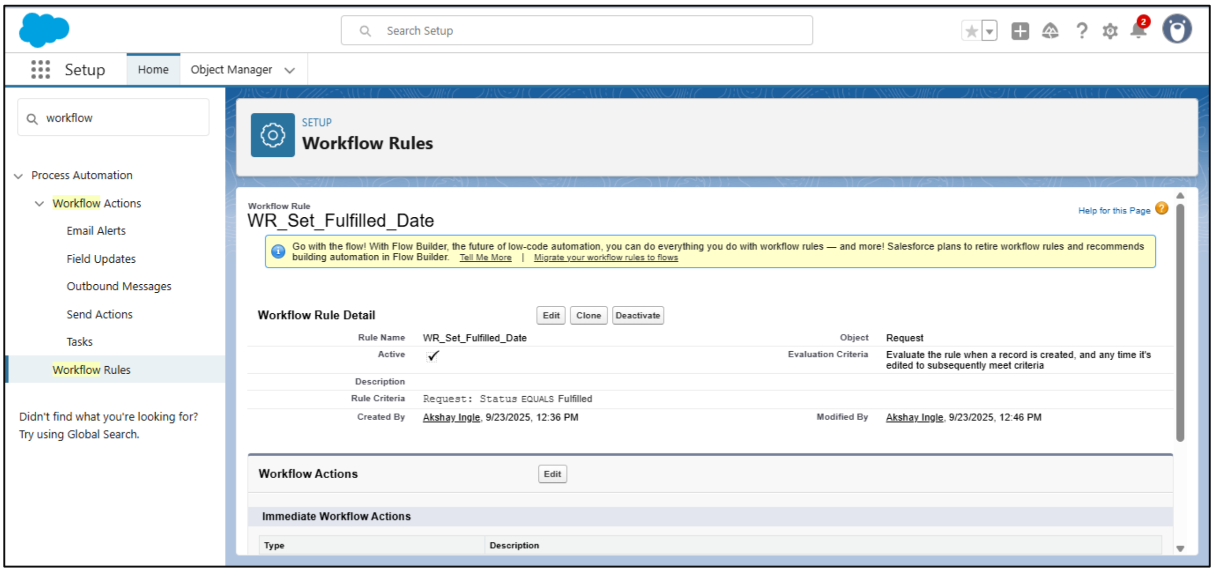
4 Workflow Rules

Steps:

1. Setup → **Workflow Rules** → New Rule.
2. Object: **Request__c**.
3. Rule Name: WR_Set_Fulfilled_Date.
4. Rule Criteria: Status__c Equals Fulfilled.
5. Action: **New Field Update**.
    - Name: FU_Set_Fulfilled_Date.
    - Field to Update: Fulfilled_On__c.
    - Formula: NOW ()
6. Save → Activate Workflow.

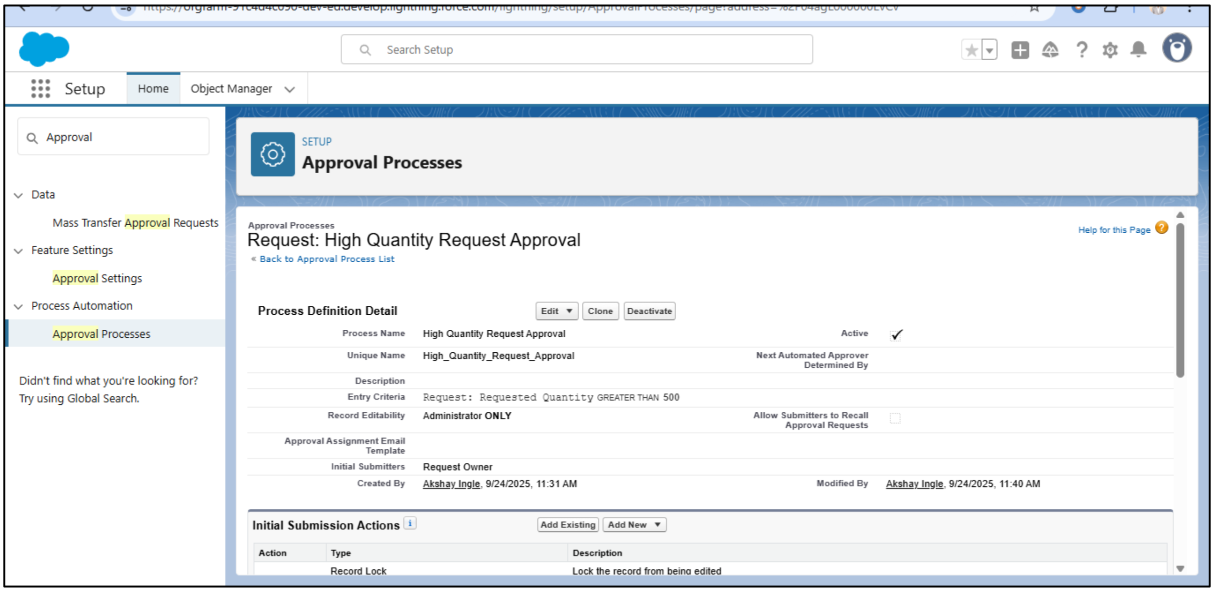
Test:

- Open a Request record.
- Change Status = Fulfilled → Save.
- Check that Fulfilled_On__c is now set.



5 Approval Process: High Quantity Request

- Navigate: Setup → Quick Find → Approval Processes
- Object: Request__c
- Create Process: Jump Start Wizard
- Basic Info: Name = High Quantity Request Approval
- Entry Criteria: Requested_Quantity__c > 500
- Assign Approver: Automatically → Relief Head
- Initial Submission: Status__c = Pending Approval
- Final Approval: Status__c = Approved
- Final Rejection: Status__c = Rejected
- Activate: Save & Activate



Notify Volunteer Flow

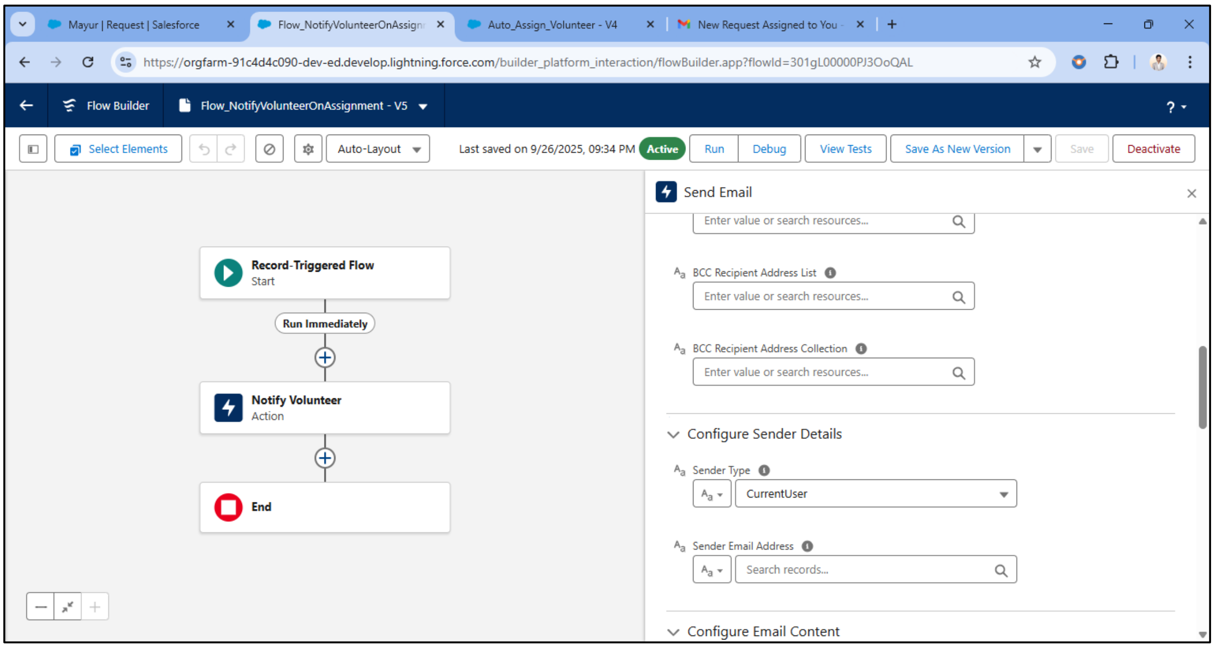
Step 1: Create Flow

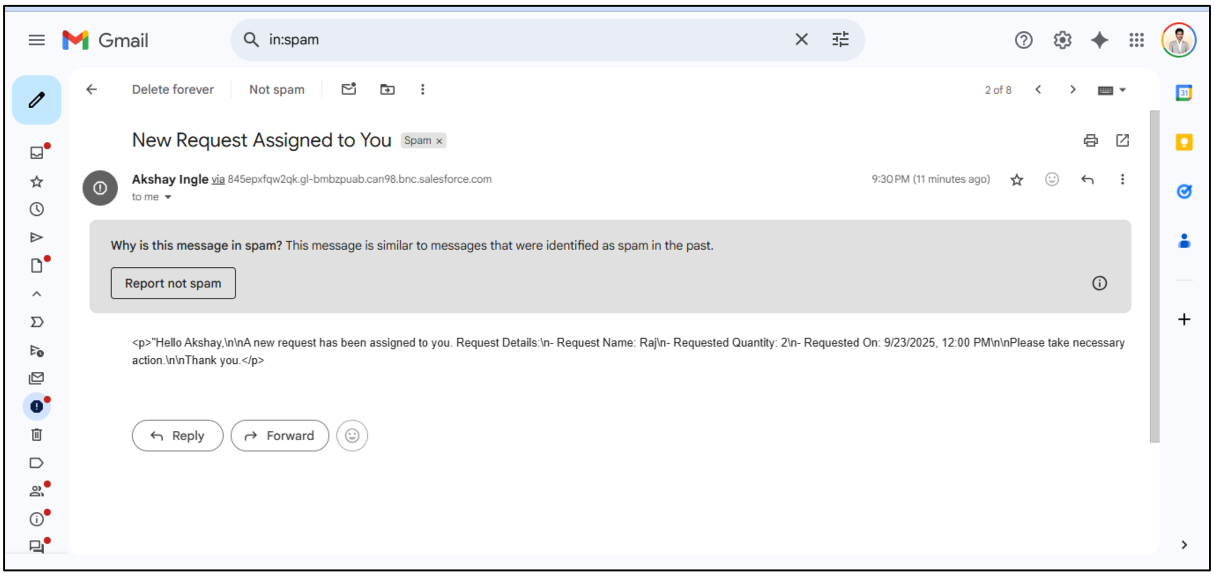
- **Type:** Record-Triggered Flow
- **Object:** Request__c
- **Trigger:** On record update
- **Condition:** Assigned_Volunteer__c Is Null = False
- **Optimize:** Actions and Related Records

Step 2: Add Email Action

- **Action:** Email Alert (or Send Email)
- **Template:** "New Request Assigned to You"
- **Body:** "Hello {!Assigned_Volunteer__c.Name}, you are assigned to Request {!Request__c.Name}."
- **Recipient:** {!$Record.Assigned_Volunteer__r.Email__c}

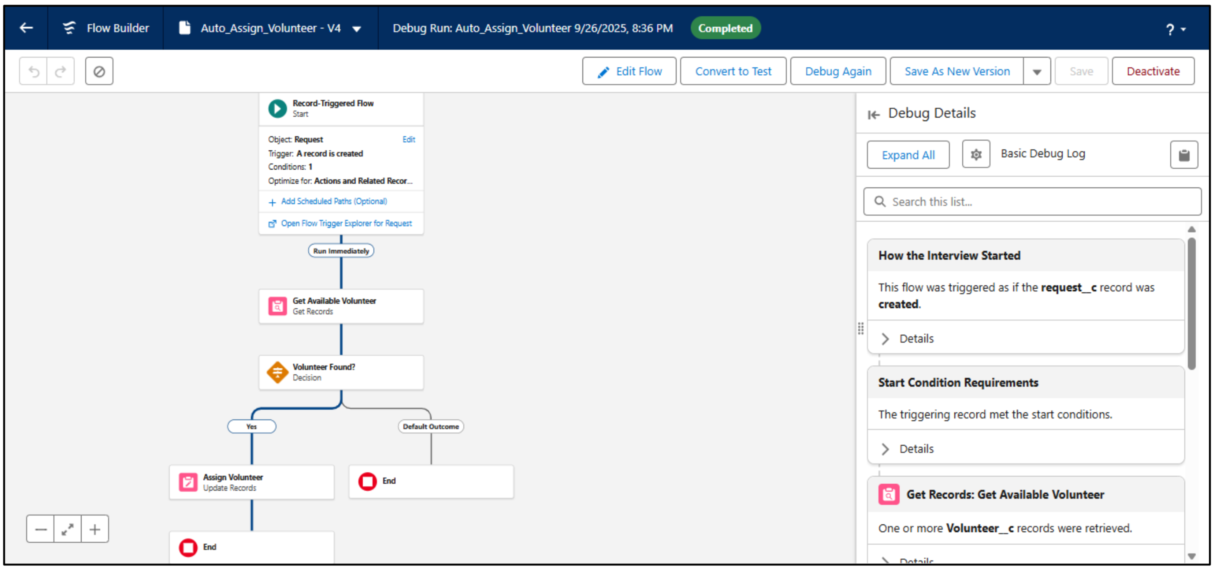
Step 3: Save & Activate





Auto Assign Volunteer Flow

- Flow Trigger
  - Flow Name: Flow_AutoAssignVolunteer
  - Object: Request__c
  - Trigger: When a record is created or updated
- Get Records Element
  - Element Label: Get Available Volunteer
  - Object: Volunteer__c
  - Filter Criteria: Availability__c Equals Available
- Decision Element
  - Condition: {!Get_Available_Volunteer.Id} Is Null = False
  - Paths:
  - Yes: Update Records
  - Default: Do nothing
- Update Records Element
  - Record to Update: Record that triggered the flow
  - Field to Update: Assigned_Volunteer__c
  - Value: {!Get_Available_Volunteer.Id}

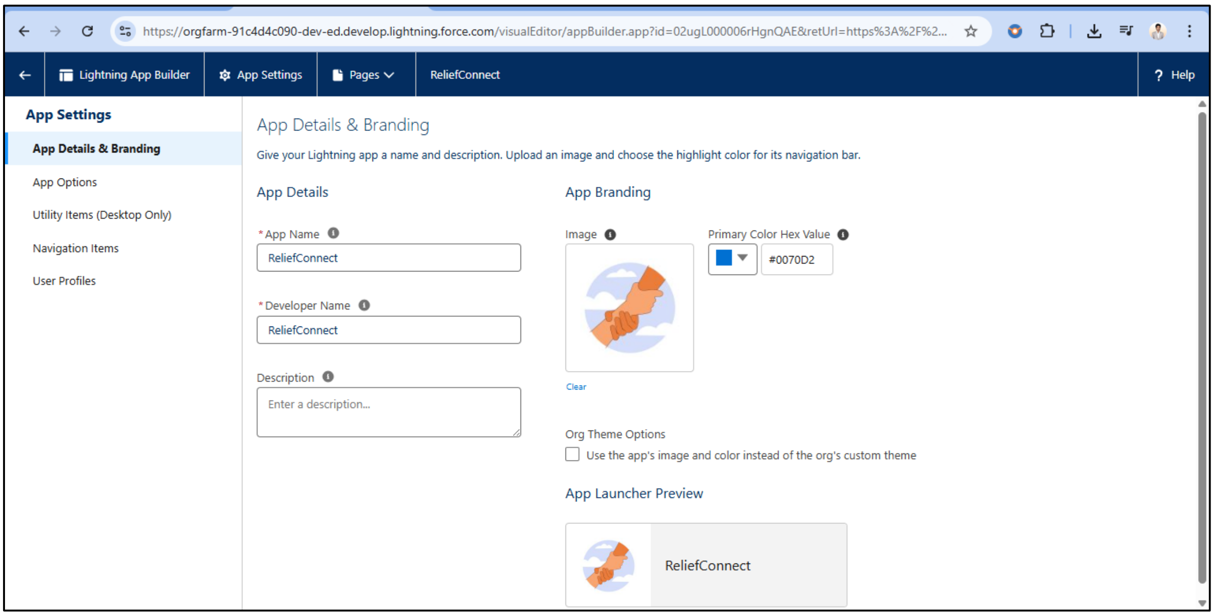


Phase 6: User Interface Development

In this phase, the focus was on designing the **Lightning Experience user interface** to provide users with an intuitive and accessible way to interact with the Disaster Response System.

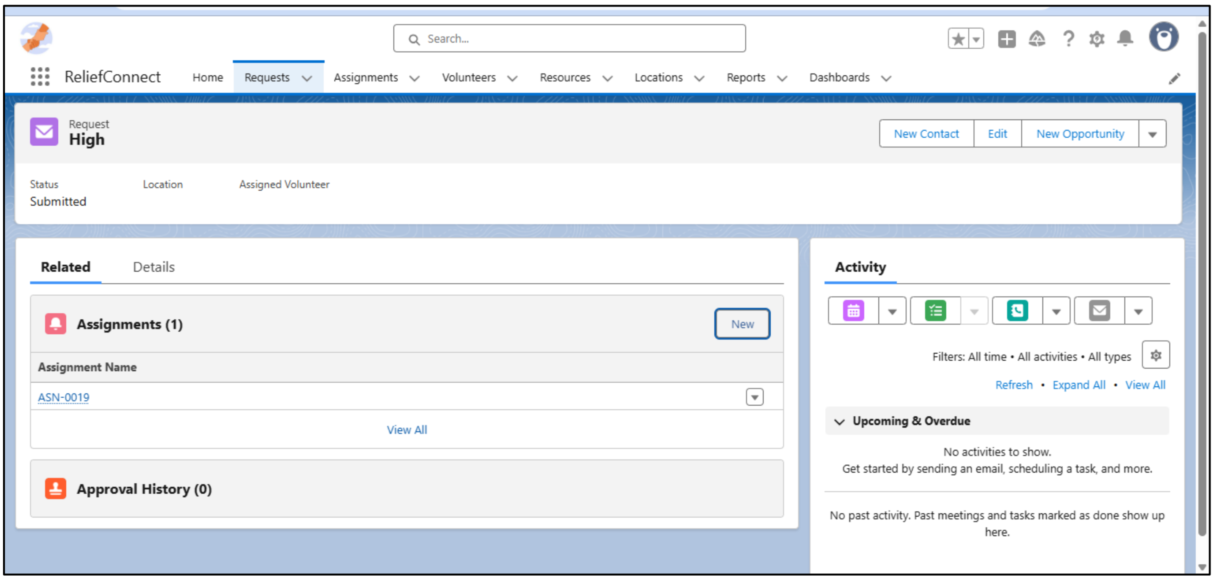
Lightning App Builder

- Customized the **Home Page** using Lightning App Builder.
- Added regions and placed standard components (e.g., Rich Text, Recent Items, Reports/Charts).
- Ensured quick navigation to **Requests**, **Volunteers**, and **Reports**.



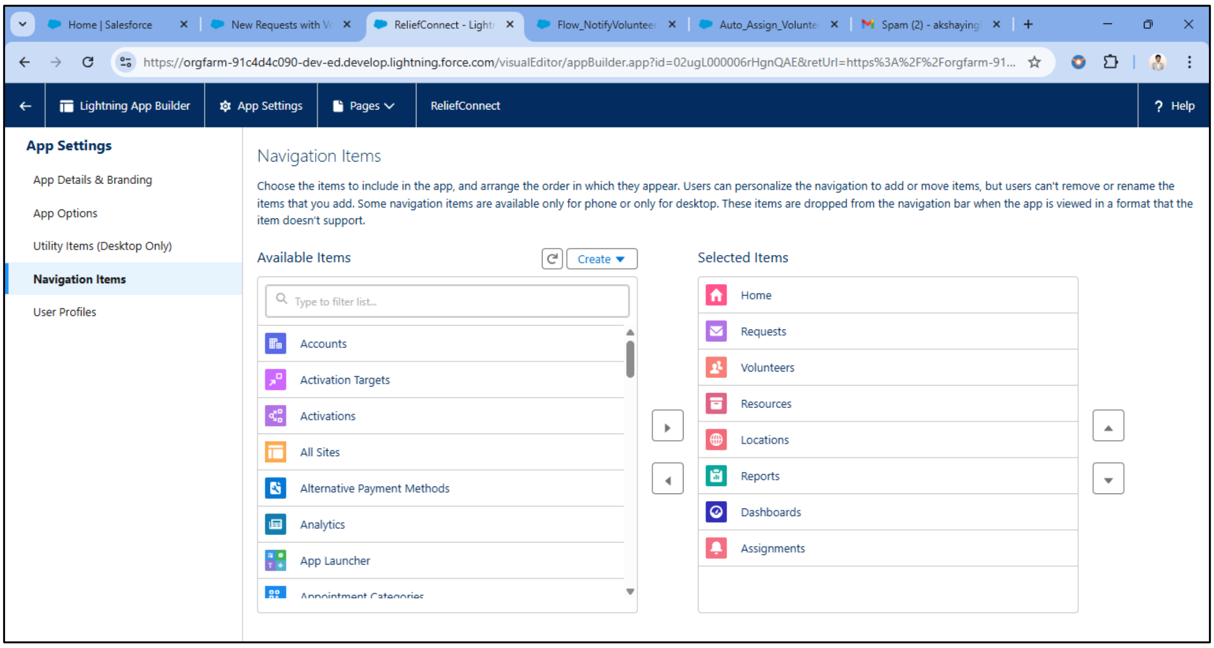
Record Pages

- Customized the **Request__c record page** to display request details, assigned volunteer, and status in a user-friendly format.



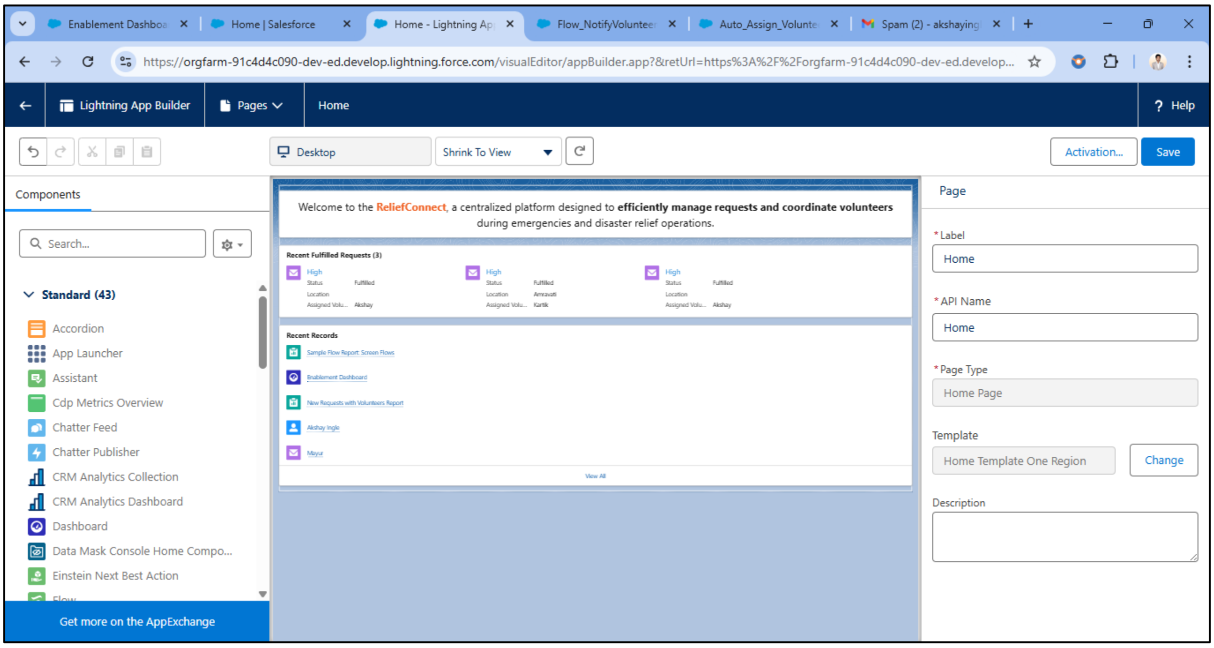
Tabs & Navigation

- Created tabs for **Requests** and **Volunteers** so users can access these objects directly from the Lightning App.



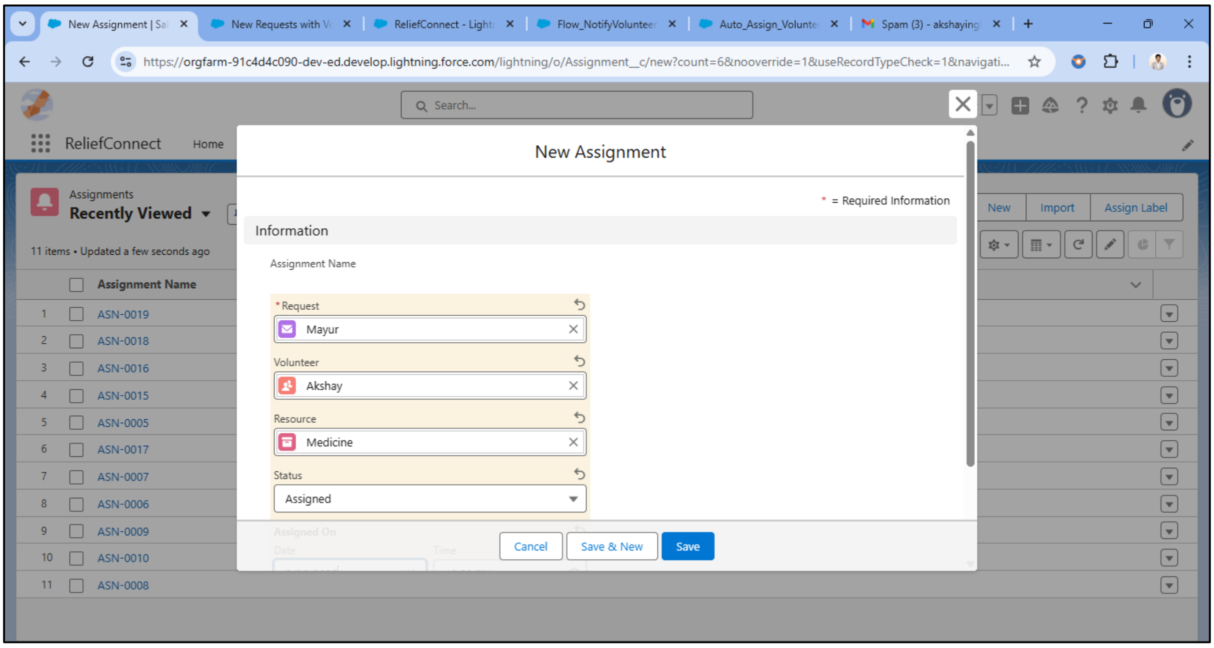
Home Page Layouts

- Added a **Rich Text component** introducing the project, key features, and usage instructions.

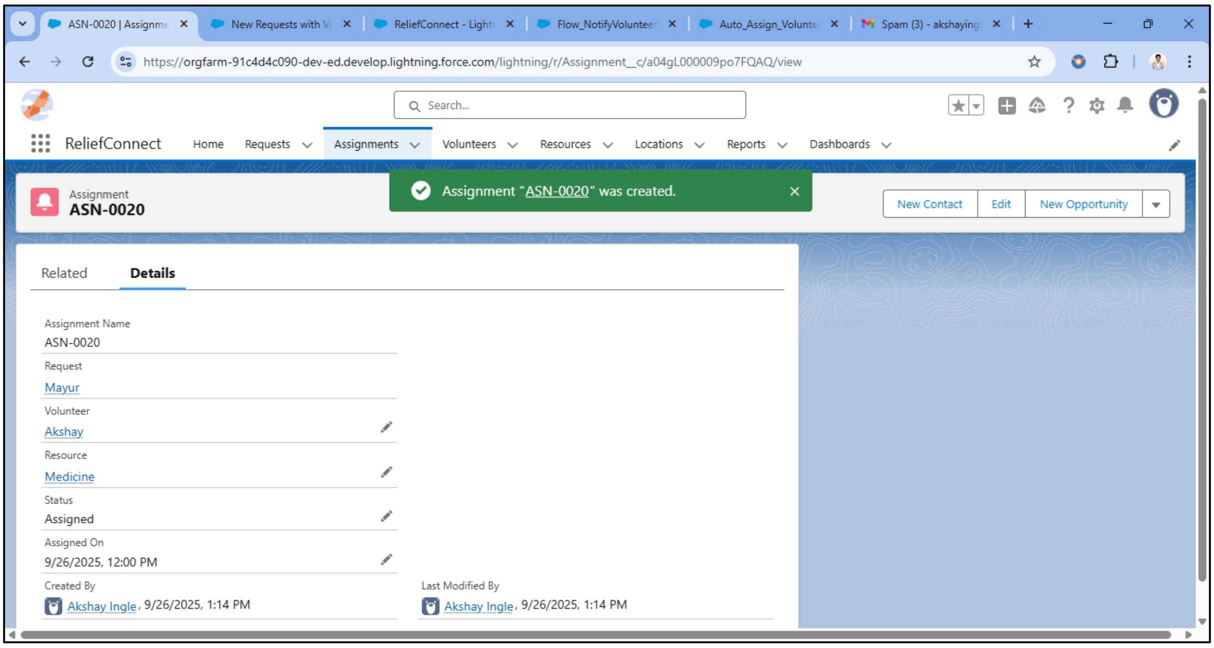


To simplify the tracking of which volunteer is assigned to which request, an **Assignments tab** was added in the Lightning App

The Assignments tab acts as a central place where users can view and manage the relationship between **Requests** and **Volunteers**.



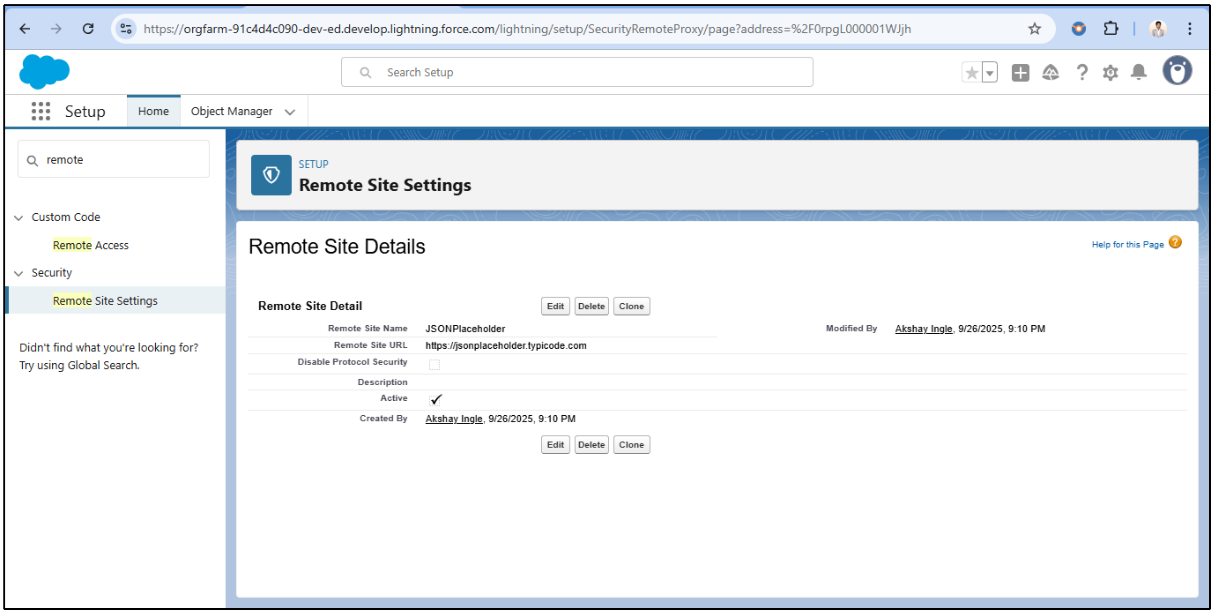
- Displays all request–volunteer assignments in a list view.
  - Request Name
  - Volunteer Name
  - Assignment Status (Active / Completed)
  - Requested Quantity
  - Fulfilled On Date



Phase 7: Integration & External Access

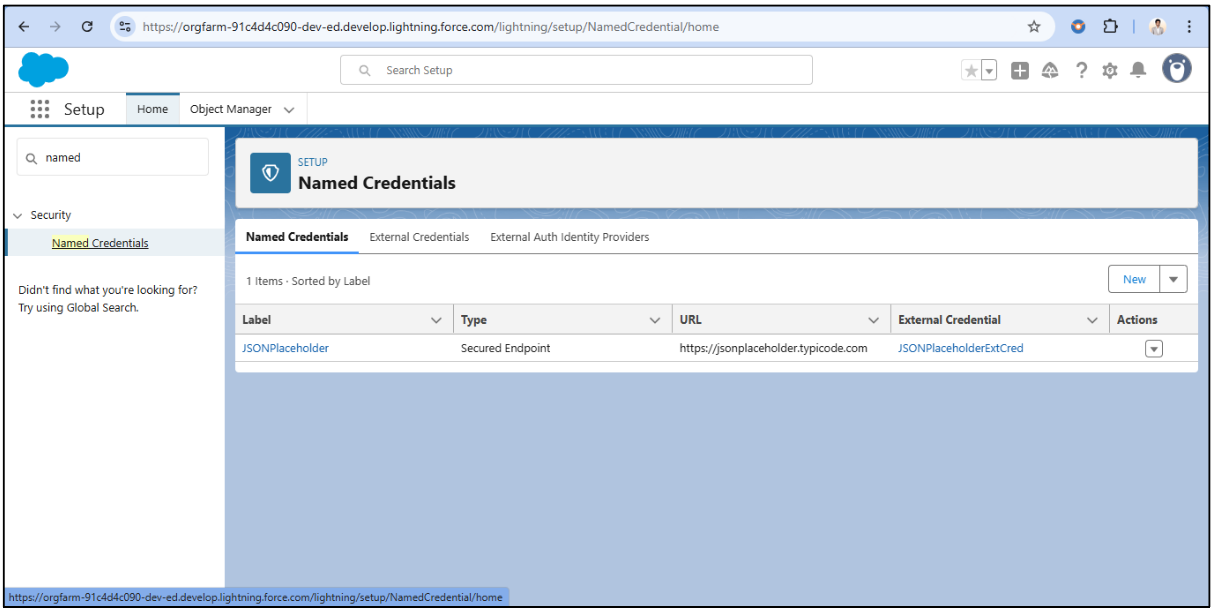
A Remote Site Setting (required for Apex callouts)

1. Setup → Quick Find → Remote Site Settings → New Remote Site.
2. Label: JSONPlaceholder (example). Remote Site URL: https://jsonplaceholder.typicode.com → Save.
3. Why: allows Salesforce to call that external endpoint (demo/test).



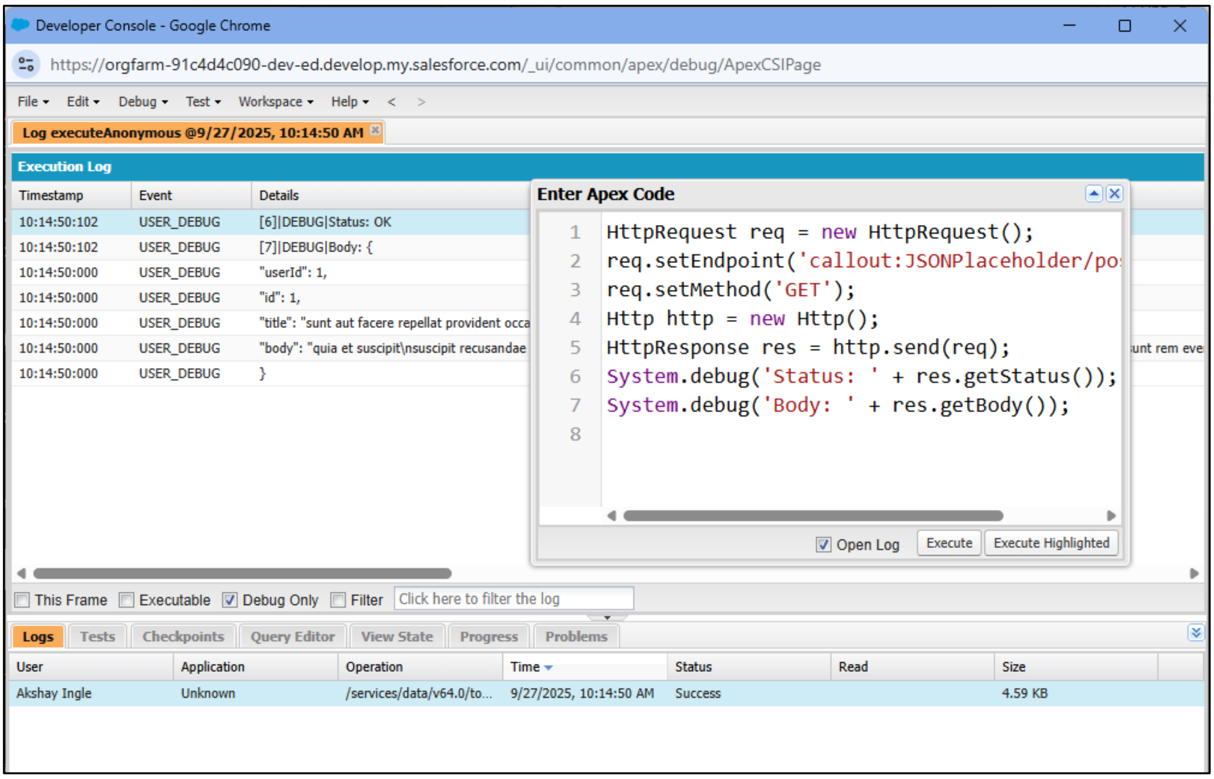
B Named Credential

1. Setup → Quick Find → **Named Credentials** → **New Named Credential**.
2. Label: JSONPlaceholder_NC. URL: https://jsonplaceholder.typicode.com. Identity Type: **Anonymous** (for demo) or **Named Principal** if you have credentials. Authentication: **No Authentication** (for demo). Save.



C Test an Apex Callout using Named Credential

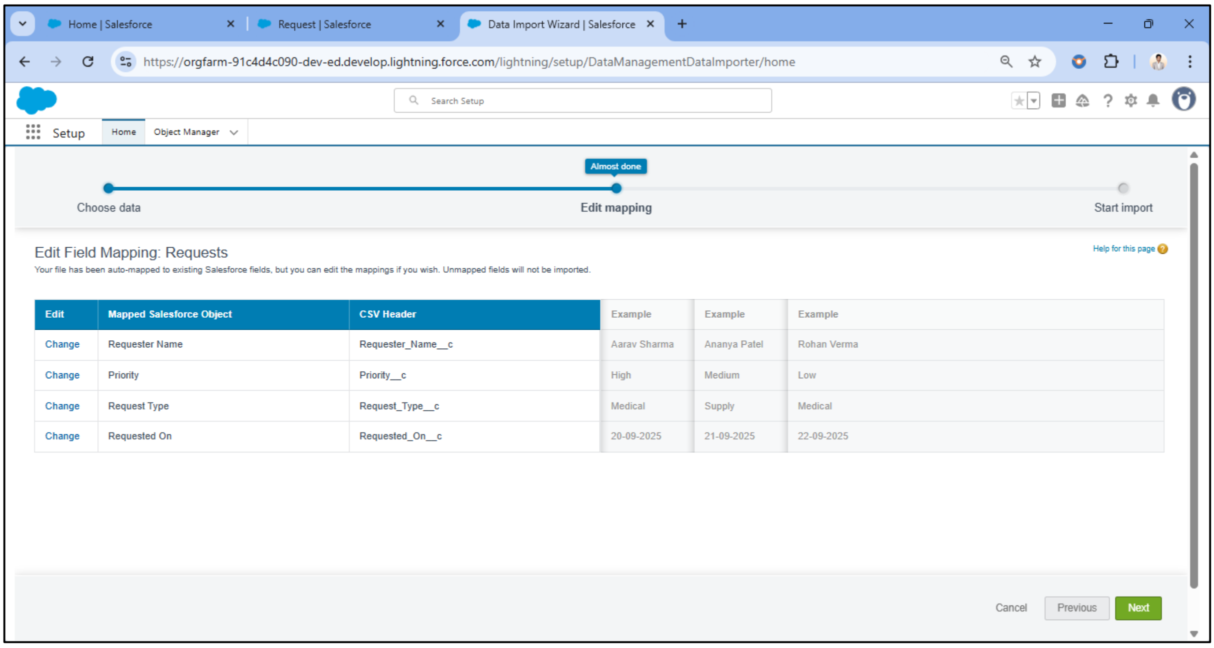
- Open Developer Console (avatar → Developer Console).
- Debug → Open Execute Anonymous Window.
- Check Open Log then Execute.



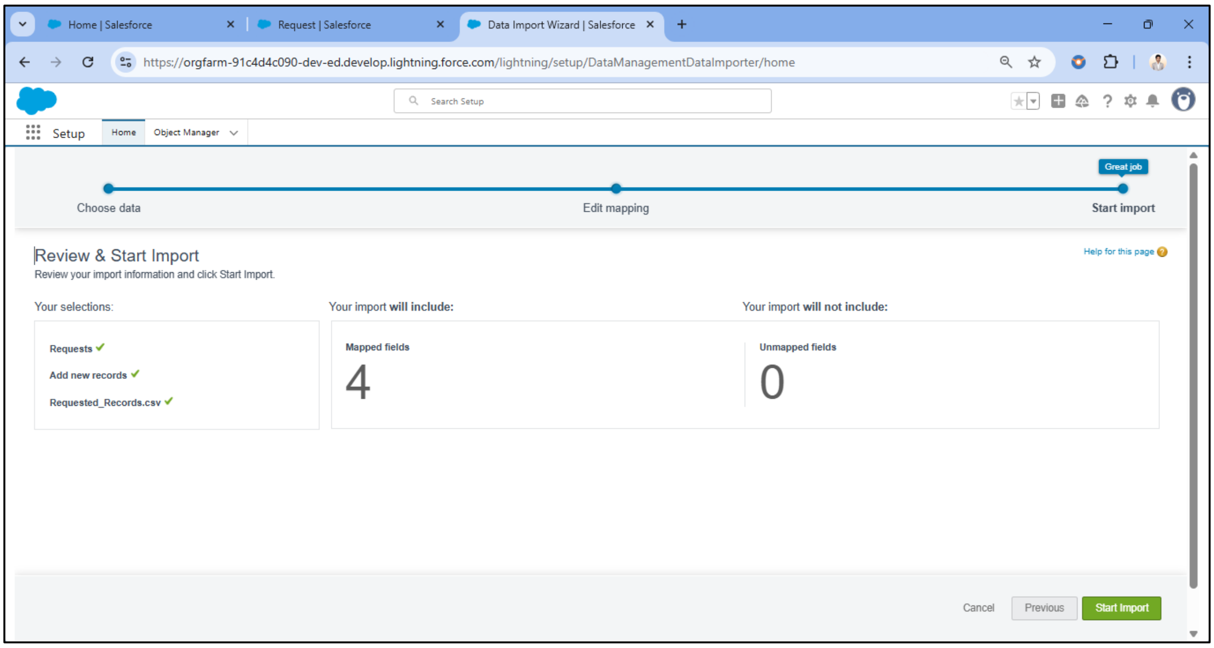
Phase 8 — Data Management & Deployment

A — Data Import Wizard

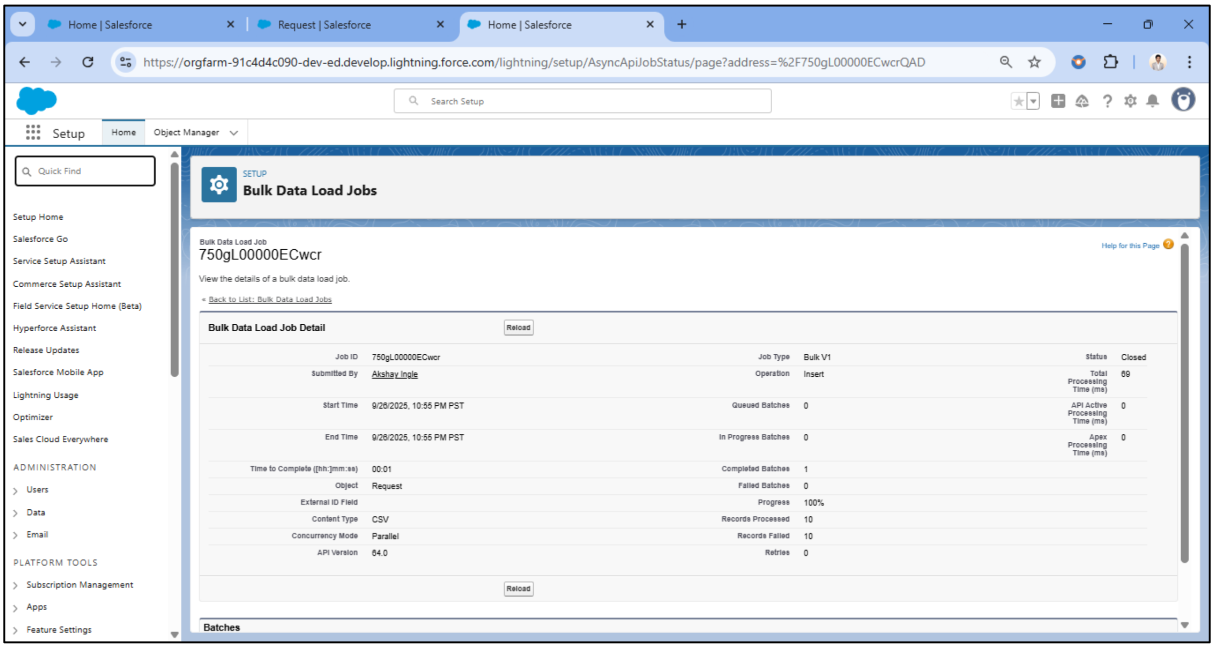
1. Setup → Quick Find → **Data Import Wizard** → Launch Wizard.
2. Select **Custom Objects** → Request (or Volunteers/Resources).
3. Upload your CSV file (prepare columns matching API names: e.g., Requester_Name__c, Priority__c, Request_Type__c, Requested_On__c).



Map fields (Data Import Wizard shows mapping) → Start Import

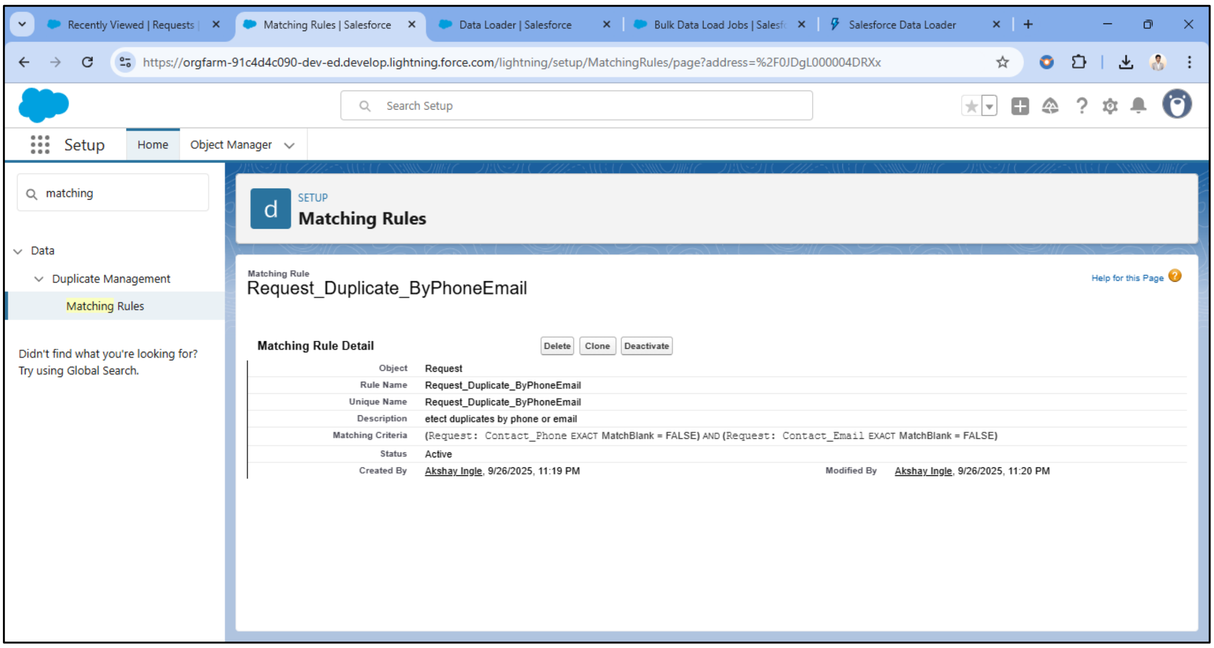


B- Data Loader



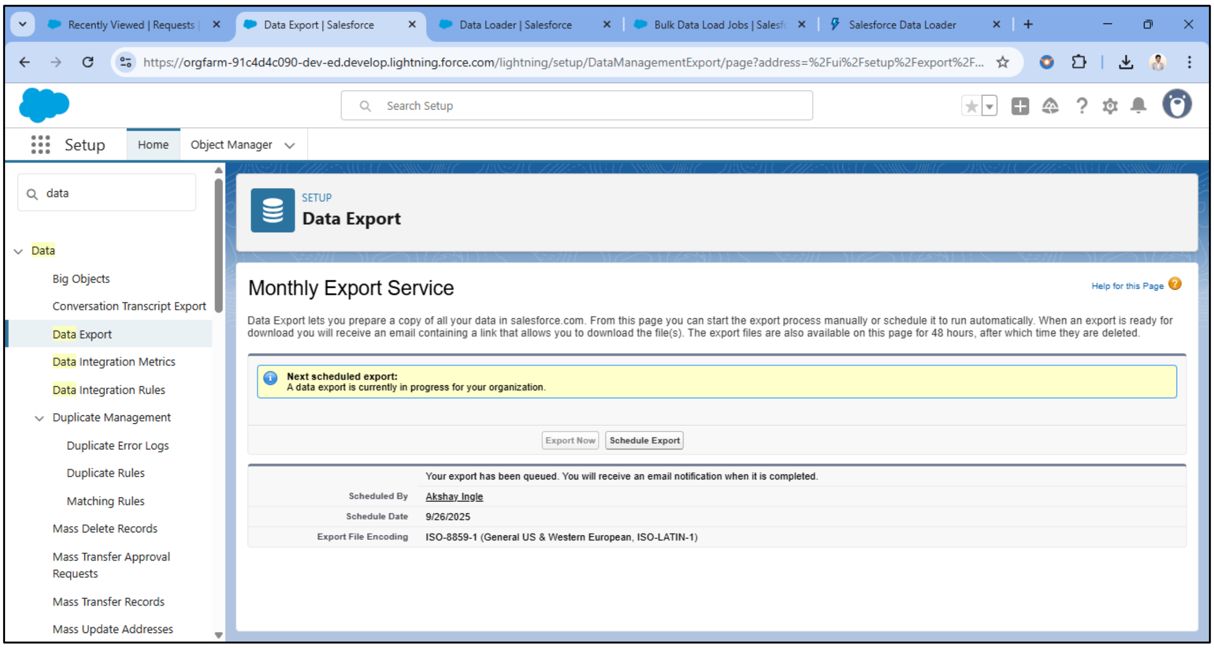
C — Duplicate Rules & Matching Rules

1. Setup → Quick Find → **Matching Rules** → **New** (create matching rule for Request by phone/email).
2. Setup → Quick Find → **Duplicate Rules** → **New** → choose object Request__c → link to matching Rule → Action: Block or Alert (Alert recommended for demo). Activate.
3. Test: try inserting duplicate Request with same phone → system shows duplicate alert.



D — Data Export & Backup

1. Setup → Quick Find → **Data Export** → **Export Now** or Schedule Export → choose objects (Requests, Volunteers, Resources) → Start Export.

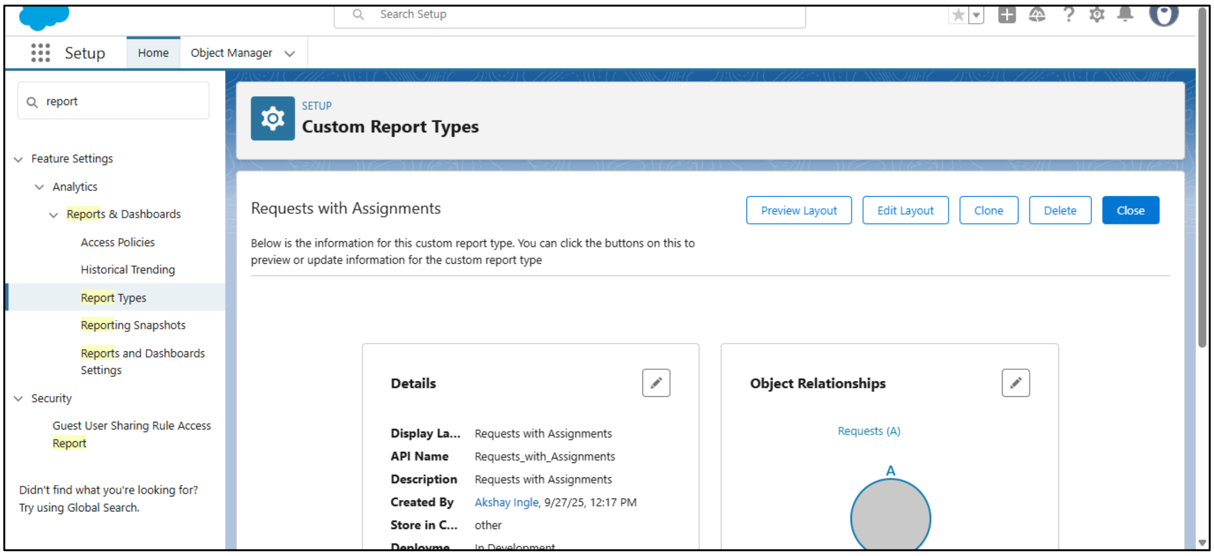


Phase 9 — Reporting, Dashboards & Security Review

Relevant: Reports types, Dashboards (including dynamic), Sharing & Field-level security, Login IP ranges, Audit Trail.

A — Custom Report Type (for Requests + Assignments)

1. Setup → Quick Find → **Report Types** → New Custom Report Type.
   - Primary Object: Request__c.
   - Give it a name: Requests with Assignments.
   - Click Save.



B — Create Key Reports (Tabular & Summary)

1. Summary Report

1. App Launcher → Reports → New Report → select Requests with Assignments.
2. Group Rows by **Location__c**.
3. Columns: Requester Name, Priority, Status, Assigned Volunteer.
4. Filter: Status ≠ Fulfilled.
5. Save as **Open Requests by Location**.

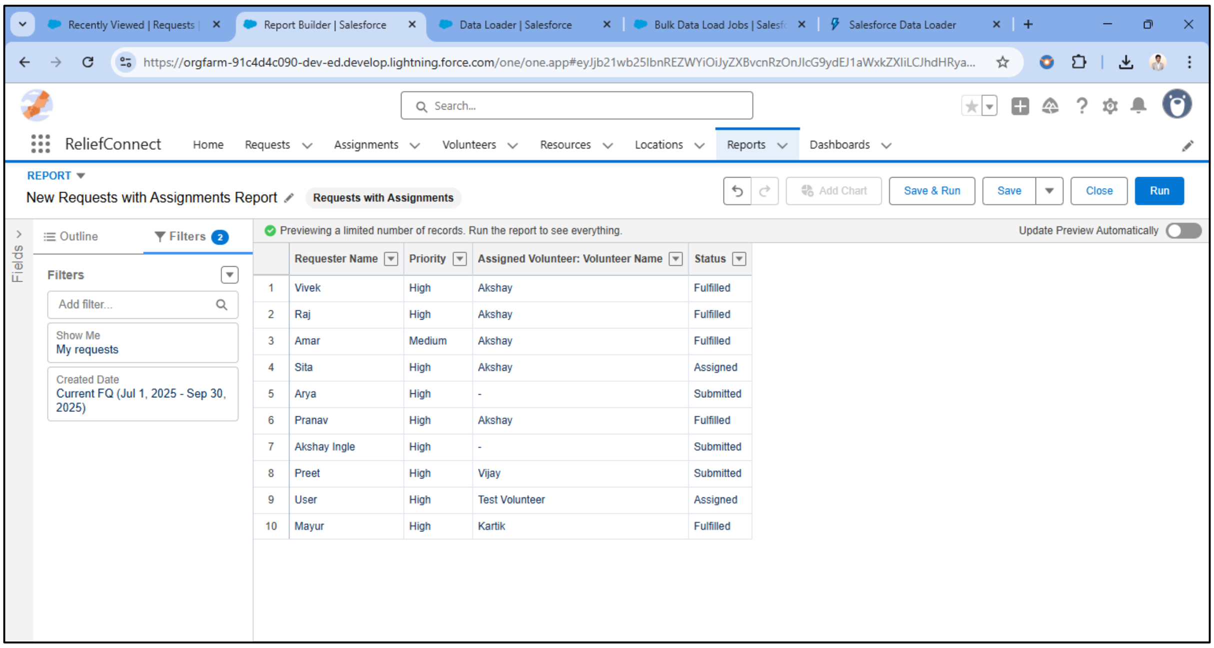


Figure 1 Before Applying filter

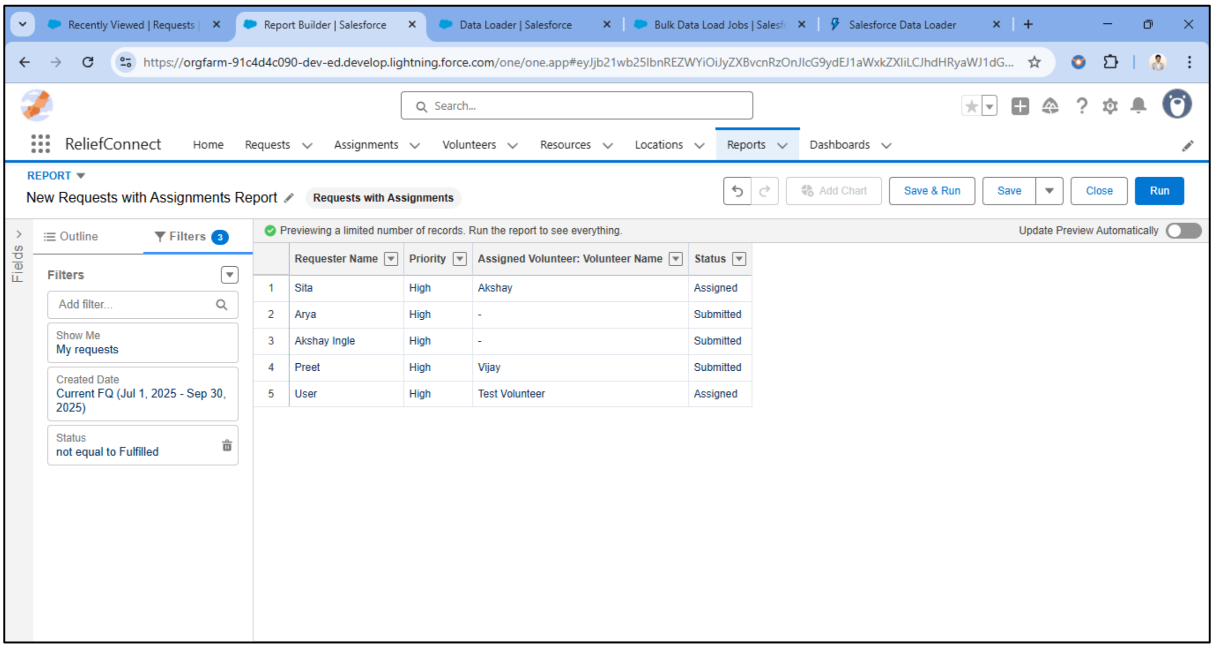
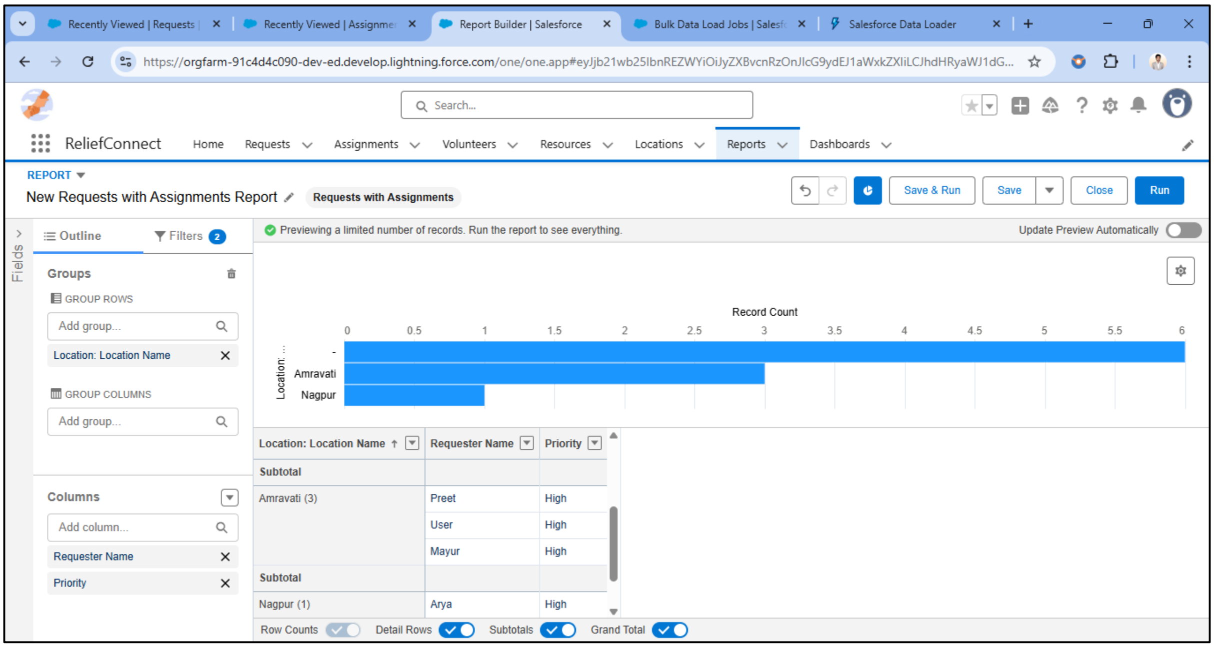


Figure 2 After Applying filter

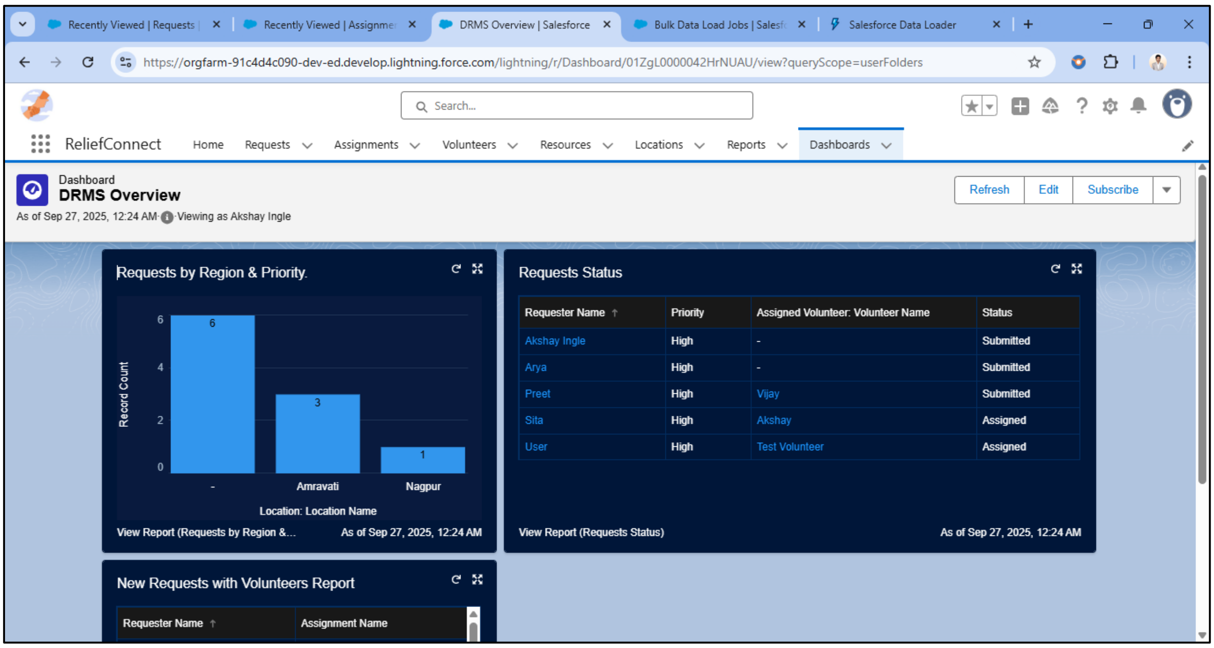
2. Matrix Report

1. App Launcher → Reports → New Report → select Requests with Assignments.
2. Rows = Region / Columns = Priority.
3. Aggregate = Count of Requests.
4. Save as **Requests by Region & Priority**.



C — Dashboards & Dynamic Dashboards

1. App Launcher → Dashboards → New Dashboard → Name: DRMS Overview.
2. Add Components:
    - **Bar Chart**: Open Requests by Location.
    - **Table**: Assignments by Volunteer.
3. Enable **Dynamic Dashboard**: View dashboard as Logged-in User (if org allows).
4. Save & Run.



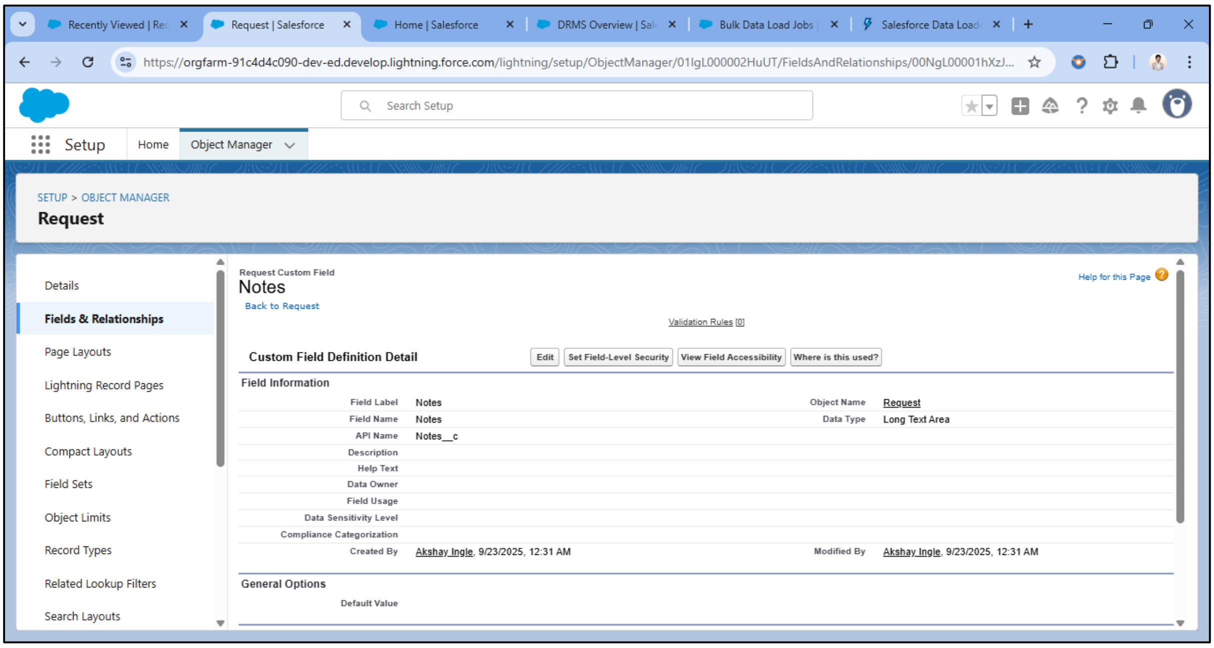
D — Sharing Settings & Field Level Security

1. Organization-Wide Defaults (OWD)

1. Setup → Quick Find → Sharing Settings.
2. Show OWD for Request__c (should be Private).

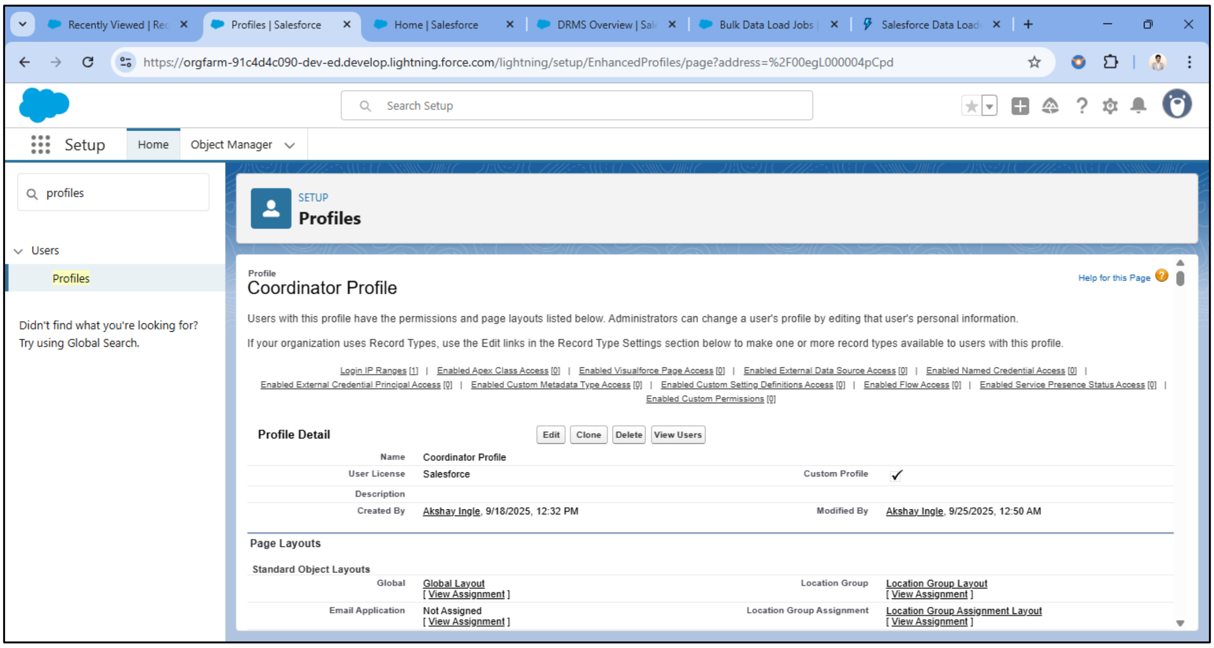
2. Field Level Security

1. Setup → Object Manager → Request__c → Fields & Relationships → select a field Notes.
2. Set **Field-Level Security** → hide from Volunteer profile if needed.



3. Profiles & Permission Sets

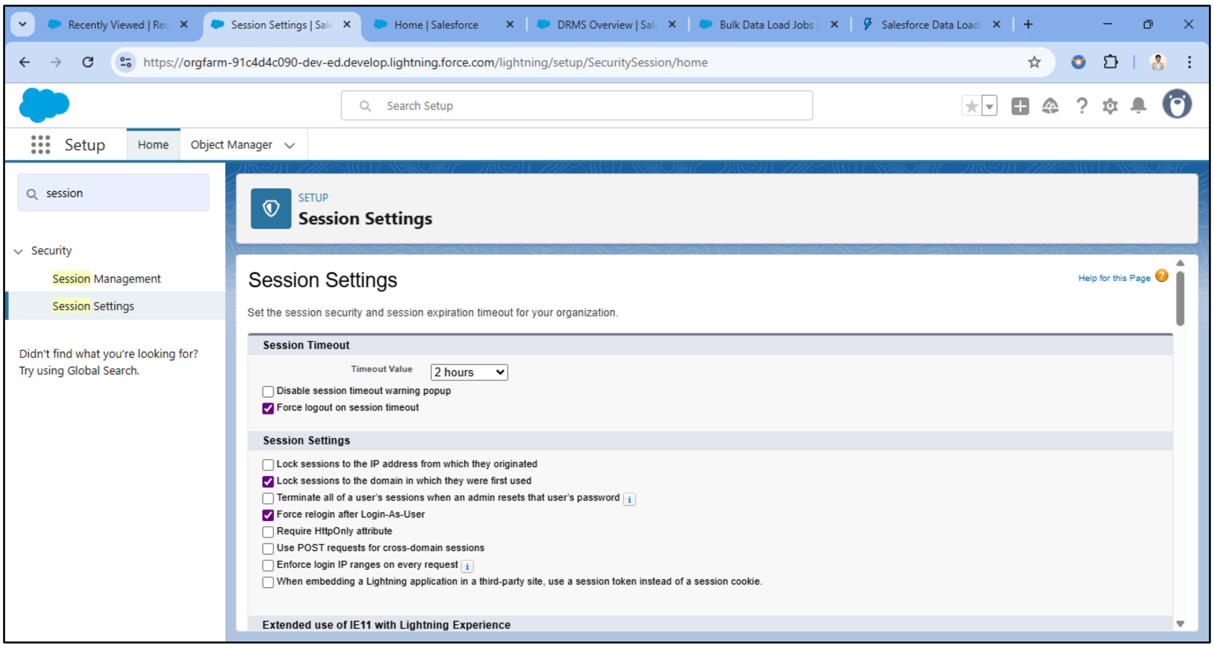
1. Ensure **Coordinator** has Read/Write on Requests.
2. Volunteers have limited access as planned.



E — Session Settings, Login IP Ranges & Audit Trail

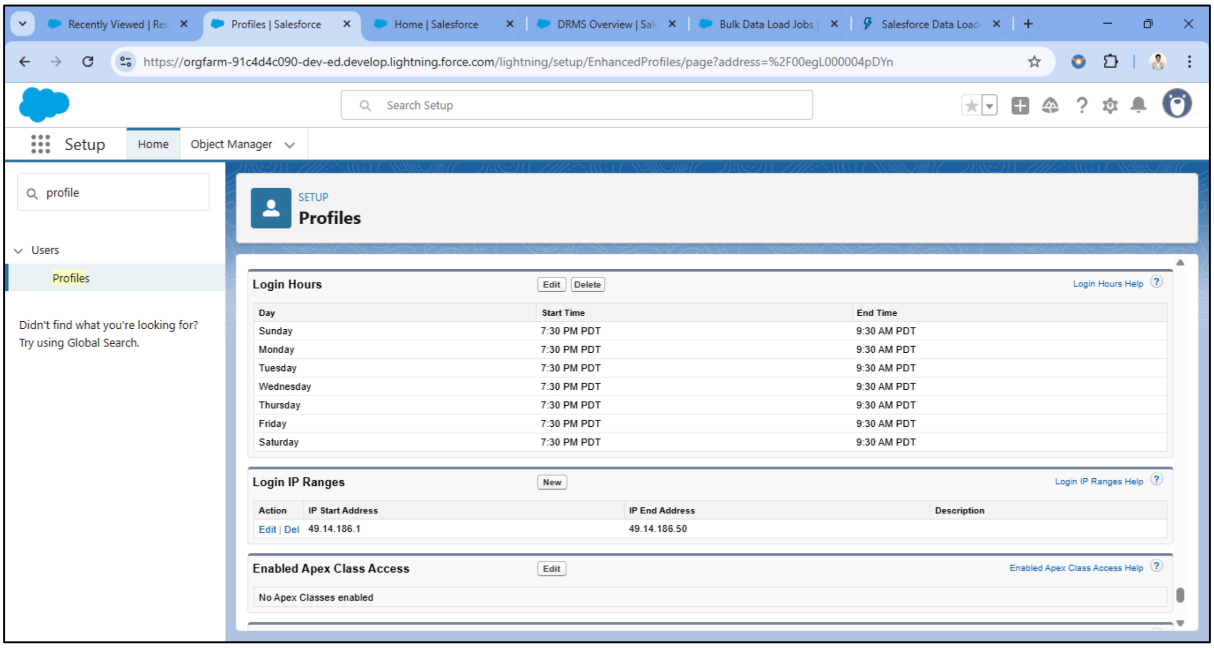
1. Session Settings

- Setup → Quick Find → Session Settings.
- Review timeout, secure cookie settings.



2. Login IP Ranges

- Profile → open Volunteer Profile → Login IP Ranges.
- Add if needed (demo: note none set if not configured).



3. Audit Trail

- Setup → Quick Find → View Setup Audit Trail → Download last 6 months.