Phase 2: Org Setup & Configuration

1. Salesforce Editions

Before starting the actual setup, I needed to choose an appropriate Salesforce edition. Salesforce offers multiple editions such as Essentials, Professional, Enterprise, and Unlimited, each with different features and limits.

Since this project is meant to be a proof of concept for a nonprofit CRM, I decided to use the **Salesforce Developer Edition**. The Developer Edition provides full Enterprise-level features (including custom objects, automation tools, profiles, roles, and APIs) with limited storage and users, which is ideal for building and testing a solution like CauseConnect CRM.

I signed up at developer.salesforce.com and logged in to my Developer Org. This org acts as the base environment where I will configure and build the entire CRM application.

2. Company Profile Setup

Once the org was active, I began setting up the company profile to reflect the NGO context for this project.

- I navigated to **Setup → Company Information**.
- I edited the **Company Name**, **Primary Contact**, **Address**, and **Default Locale**.
- I set the **Default Locale** to *English (India)*, **Currency** to *INR – Indian Rupee*, and **Time Zone** to *Asia/Kolkata (GMT +05:30)*.

This setup ensures that all records, date formats, currency values, and time references within the org are aligned with the organization’s region and operational context.

3. Business Hours & Holidays

Business hours define when users are considered “working,” and they are used by Salesforce in features like escalations, case management, and reporting. Though this CRM will not initially use cases or escalations, it is good practice to define them properly.

- I navigated to **Setup → Company Settings → Business Hours**.
- I created a new set of business hours named **“NGO Standard Hours”**.
- I set the working days as **Monday to Friday**, from **9:00 AM to 6:00 PM IST**.
- I marked **Saturday and Sunday as non-working days**.

I also created a **Holiday Calendar** under **Setup → Company Settings → Holidays** and added common holidays (such as Republic Day and Independence Day) to ensure that future escalations or automations do not trigger on these non-working days.

4. Fiscal Year Settings

Since NGOs in India typically follow an April to March financial year, I configured the fiscal year settings accordingly. This ensures that all reports, dashboards, and donation analytics are grouped according to the actual financial year rather than the default January–December year.

- I went to **Setup → Company Settings → Fiscal Year**.
- I selected **Standard Fiscal Year**.
- I set the **Fiscal Year Start Month to April**.

This alignment will be useful later when generating donor contribution summaries and program impact reports based on the NGO's reporting cycle.

5. User Setup & Licenses

After setting up the company-level configuration, I created the core internal users who will represent different NGO staff roles within the CRM. These users will later be assigned to specific profiles and roles as the security model matures.

I navigated to **Setup → Users → Users → New User** and created the following internal users:

- **Fundraising Manager** : manages donors and campaigns
  - Username: fundraisingmanager@causeconnect.com
  - User License: Salesforce
  - Profile: Standard User
- **Program Manager** : manages programs and beneficiaries
  - Username: programmanager@causeconnect.com
  - User License: Salesforce
  - Profile: Standard User

I plan to add the remaining internal users (Field Officer, NGO Leadership, API User) later by reusing the available Salesforce licenses when needed.

I also created a **Public Group named Auditors** under **Setup → Public Groups**, which will be used later for sharing rules. This group is currently empty and will be populated once Experience Cloud is enabled and external users (Auditors) are created.

6. Profiles

Profiles control object-level and field-level permissions for users. I created custom profiles for each type of user who will use the system, based on the access they need.

I went to **Setup → Profiles → Clone Standard User** and created:

- **Fundraising Manager Profile** : Full CRUD on Donation, Contact, Campaign, Program
- **Program Manager Profile** : Full CRUD on Program and Beneficiary
- **Field Officer Profile** : Full CRUD on Beneficiary, Read on Program
- **NGO Leadership Profile** : Read-only on all custom objects (Donation, Program, Beneficiary), access to dashboards and reports
- **API User Profile** : Minimum access, API Enabled, no UI permissions

For now, I assigned the Fundraising Manager and Program Manager users to their respective custom profiles. The other profiles are created and will be used when the other users are added.

7. Roles

Roles define record visibility through the role hierarchy. I created a simple hierarchy that reflects an NGO's structure.

I went to **Setup → Roles → Set Up Roles → Add Role** and created:

- NGO Leadership (top level)
- Fundraising Manager (reports to Leadership)
- Program Manager (reports to Leadership)
- Field Officer (reports to Program Manager)

This ensures that users higher in the hierarchy automatically get visibility into the records owned by users below them.

8. Permission Sets

Permission sets provide additional permissions on top of profiles. At this stage, most access was handled directly through profiles, so I created only one general permission set:

- **Reports & Dashboards Access** : includes "Create and Customize Reports" and "Manage Dashboards" permissions

This permission set can be assigned to any internal user who needs to build or edit reports and dashboards regardless of their profile.

9. Organization-Wide Defaults (OWD)

OWD determines the baseline record access for each object before role hierarchy and sharing rules are applied. To ensure data security, I configured the following:

- **Program** : Public Read Only (everyone can see programs)
- **Donation** : Private
- **Beneficiary** : Private
- **Contact** : Private
- **Campaign** : Public Read/Write

I went to **Setup → Sharing Settings** and set these defaults under the "Default Internal Access" column.

10. Sharing Rules

Because Donations, Beneficiaries, and Contacts are private by default, users outside the hierarchy (like external Auditors) would not be able to see them. To fix this, I created sharing rules that grant read-only access to the Auditors group.

I went to **Setup → Sharing Settings → [Object] Sharing Rules → New** and created:

- **Donation Sharing Rule**
  - Share all Donation records owned by internal users
  - With Public Group: Auditors
  - Access: Read Only
- **Beneficiary Sharing Rule**
  - Share all Beneficiary records with Auditors
  - Access: Read Only
- **Contact Sharing Rule**
  - Share all Contact records with Auditors
  - Access: Read Only

This ensures that when Auditor users are later created and added to the Auditors group, they will be able to see all key data without having edit permissions.

11. Login Access Policies

To make it easier to troubleshoot issues during development, I enabled login access policies. This allows Salesforce support and delegated administrators to log in as users for debugging (if needed), and also allows me to log in as other users directly from their user records.

I went to **Setup → Login Access Policies** and:

- Checked **“Administrators Can Log in as Any User”**
- Saved the setting

This will help during testing when I want to view the org from another user’s perspective, especially to verify permissions and page layouts.

12. Developer Org Setup

Since I am using a Developer Edition org, most developer features are already available by default. I still did some initial configuration to prepare it as a development environment:

- Verified **My Domain** was enabled under **Setup → My Domain** (required later for enabling Experience Cloud)
- Enabled **Dev Hub** under **Setup → Dev Hub** to allow creation of scratch orgs in the future if needed
- Created a **custom app workspace** called **CauseConnect CRM** from **App Manager**, which will act as the main app container where all custom tabs, objects, and pages will be grouped

This gives me a structured space to begin building and testing the application components.

13. Sandbox Usage

Since the Developer Edition org does not support sandboxes, I will be using the same Developer org for development and testing. Currently, all changes are being developed and tested directly in the Developer org.

14. Deployment Basics

To ensure future scalability and maintainability, I planned a basic deployment strategy:

- All metadata changes (custom objects, fields, profiles, roles, flows, layouts) will be tracked through **Change Sets** or **Salesforce CLI (sfdx)** if moved to another org

- I also enabled **Dev Hub** for version control with source-driven development if needed later

This provides a basic deployment framework for when the project needs to be migrated from the current Developer org to a production environment.

With this, I have completed the entire **Org Setup & Configuration phase**.
The Salesforce org now has:

- A fully configured company profile, fiscal calendar, and working hours
- Users, profiles, roles, and permission structure
- Secure OWD and sharing rules
- Developer environment readiness for building the CRM application