Project Title: Blood Donor Management System

Problem Statement

Blood donation and availability are often managed manually through registers and phone calls.

- Lack of centralized system leads to delays in finding compatible donors during emergencies.

- Miscommunication between hospitals, blood banks, and donors results in shortages.

- Difficulty in maintaining donor history, eligibility, and tracking regular donors.

- Emergency blood requests are not communicated effectively to potential donors.

- No proper reporting system to analyze donation trends and shortages.

Project Overview

- A centralized, automated platform for managing donors, blood inventory, and requests.

- Real-time donor matching based on blood group and location.

- Hospitals and blood banks can raise urgent blood requests and notify eligible donors instantly.

- Dashboards and reports for monitoring donor activity, blood availability, and hospital demand.

- Secure storage of donor and patient data with privacy compliance.

Objective

• Digitize donor registration, hospital requests, and blood availability tracking.

• Provide dashboards for hospitals, blood banks, and administrators to monitor real-time data.

• Allow donors to view their donation history, eligibility, and receive alerts.

• Generate real-time alerts for urgent requests and donor reminders.

• Ensure data accuracy, privacy, and security at all levels.

Use Cases

• Donor Registration & Management: Donors register, update health status, and check eligibility for donation.

• Blood Inventory Management: Hospitals and blood banks update available blood units with expiry

tracking.

• Blood Request & Allocation: Hospitals raise requests; system matches with nearest eligible donors.

• Notifications & Alerts: Donors get alerts for urgent requests, upcoming eligibility, and reminders.

Phase 1: Problem Understanding & Industry Analysis

Requirement Gathering

Collect requirements from hospitals, donors, patients, and blood banks.

Identify key pain points: manual processes, delays in finding donors, miscommunication, lack of real-time updates.

Document system requirements: donor registration, blood request, alerts, inventory tracking, and reporting.

Stakeholder Analysis

Donors: Register, update health status, track donation history, and receive alerts.

Hospitals: Request blood urgently and manage inventory efficiently.

Blood Banks: Update stock, manage expiry dates, and ensure timely donor-hospital matches.

Healthcare Administrators: Monitor availability, donor activity, and compliance with standards.

Patients: Benefit indirectly from faster donor matching and emergency response.

Business Process Mapping

Current Process: Manual donor identification via phone calls or registers.

Challenges: Slow emergency response, no centralized donor history, limited reporting.

Future Process: Centralized donor database, automated matching, dashboards for hospitals and admins, real-time donor notifications.

Industry-Specific Use Case Analysis

Donor Management: Digital registration, eligibility monitoring, health checks, and reminders.

Emergency Response: Instant donor matching for hospital requests.

Inventory Tracking: Blood unit expiry monitoring and shortage alerts.

Compliance: Ensure privacy, security, and adherence to government guidelines for blood donation.

AppExchange Exploration

Explore existing healthcare and blood bank management apps for Salesforce.

Identify reusable components such as donor/volunteer management, healthcare CRMs, and alert systems.

Evaluate whether customization or integration of existing apps provides better efficiency.

Phase 2: Org Setup & Configuration

Step 1. Salesforce Editions

- Using Salesforce Developer Edition Org.
- Developer Edition provides Enterprise-level features free with limited licenses and storage.
- Suitable for student projects and practice

Step 2. Company Profile Setup

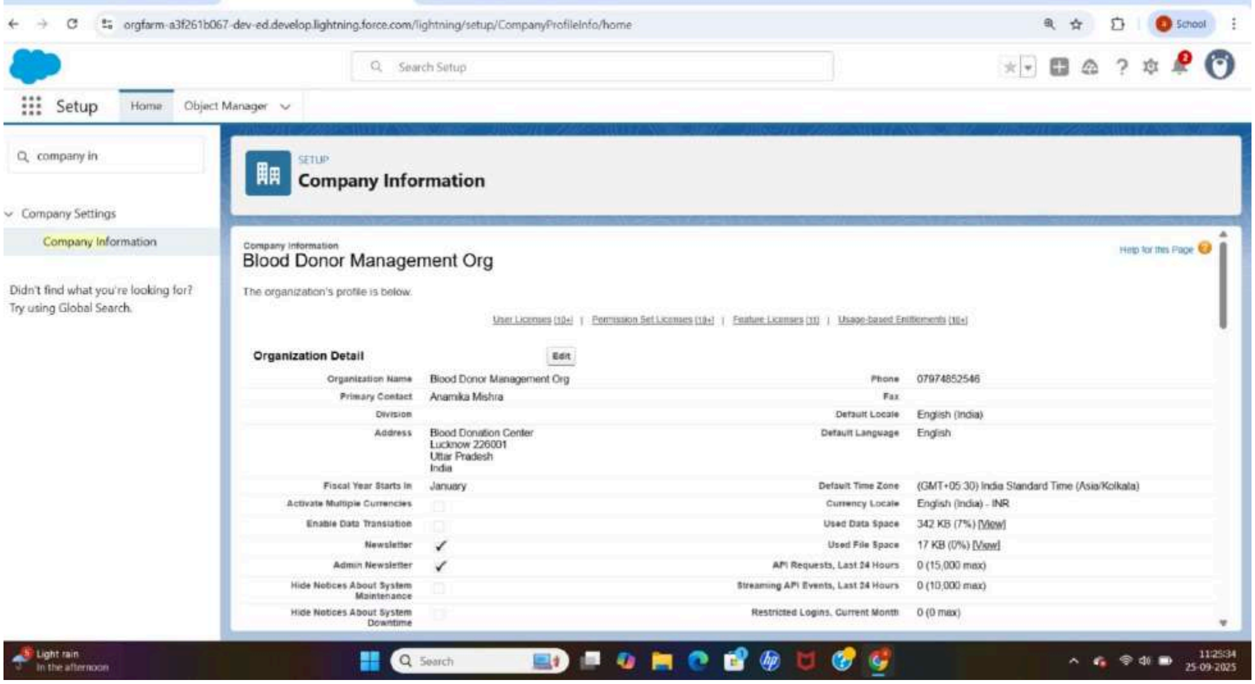
Organization Name: Blood Donor Management Org

Default Locale: English (India)

Default Language: English

Default Time Zone: (GMT +05:30) India Standard Time (Asia/Kolkata)

Default Currency: INR (Indian Rupee)



Company Information

Blood Donor Management Org

The organization's profile is below.

User Licenses [10+] | Permission Set Licenses [10+] | Feature Licenses [11] | Usage-based Entitlements [10+]

Organization Detail

Organization Name	Blood Donor Management Org	Phone	07974852546
Primary Contact	Anamika Mishra	Fax	
Division		Default Locale	English (India)
Address	Blood Donation Center Lucknow 226001 Uttar Pradesh India	Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	□	Currency Locale	English (India) - INR
Enable Data Translation	□	Used Data Space	342 KB (7%) [View]
Newsletter	✓	Used File Space	17 KB (0%) [View]
Admin Newsletter	✓	API Requests, Last 24 Hours	0 (15,000 max)
Hide Notices About System Maintenance	□	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	□	Restricted Logins, Current Month	0 (0 max)

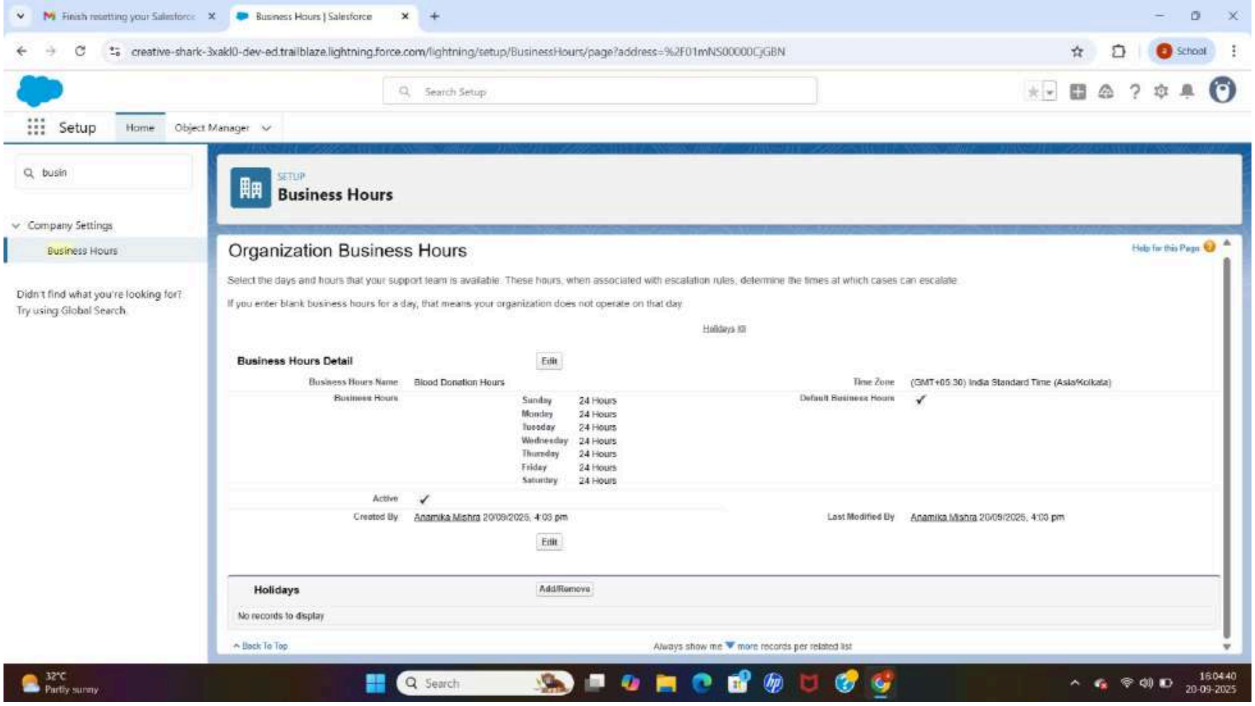
Step 3. Business Hours

Name: Blood Donation Hours

Time Zone: GMT +05:30 India Standard Time

Start: 9:00 AM

End: 6:00 PM


Finish resetting your Salesforce
Business Hours | Salesforce
creative-shark-3xakl0-dev-ed.trailblaze.lightning.force.com/lightning/setup/BusinessHours/page?address=%2F01mNS00000CjGBN
School
Search Setup
Setup
Home
Object Manager
busin
Company Settings
Business Hours
Didn't find what you're looking for?
Try using Global Search.
SETUP
Business Hours
Help for this Page
Edit
Add/Remove
Back To Top
32°C
Partly sunny
Search
16:04:40
20-09-2025

Organization Business Hours

Select the days and hours that your support team is available. These hours, when associated with escalation rules, determine the times at which cases can escalate.

If you enter blank business hours for a day, that means your organization does not operate on that day.

Holidays [0]

Business Hours Detail

Business Hours Name	Blood Donation Hours	Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Business Hours	Sunday 24 Hours Monday 24 Hours Tuesday 24 Hours Wednesday 24 Hours Thursday 24 Hours Friday 24 Hours Saturday 24 Hours	Default Business Hours	✓
Active	✓		
Created By	Anamika Mishra 20/09/2025, 4:03 pm	Last Modified By	Anamika Mishra 20/09/2025, 4:03 pm

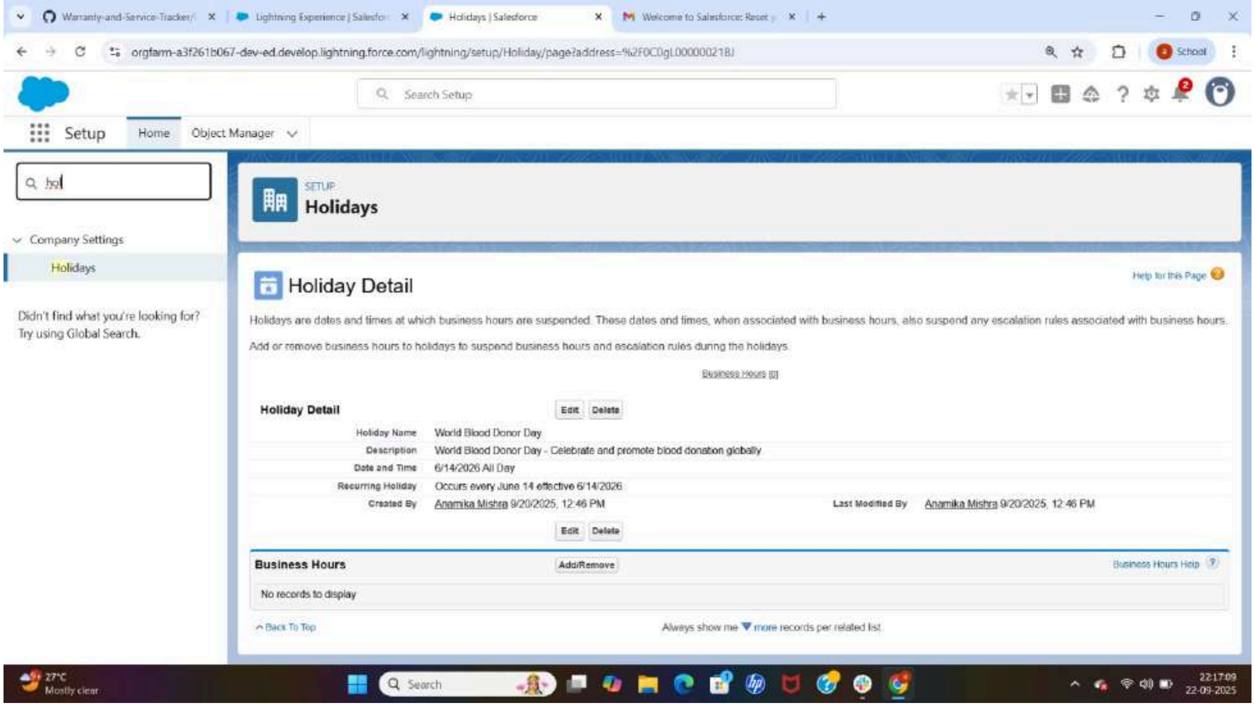
Holidays

No records to display

Always show me more records per related list

Holiday Name: World Blood Donor Day

Date: 6/14/2026 (All Day)


Warranty-and-Service-Tracker/
Lightning Experience | Salesforce
Holidays | Salesforce
Welcome to Salesforce: Reset
orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/Holiday/page?address=%2F0C0gL00000021BJ
School
Search Setup
Setup
Home
Object Manager
hol
Company Settings
Holidays
Didn't find what you're looking for?
Try using Global Search.
SETUP
Holidays
Help for this Page
Edit
Delete
Add/Remove
Business Hours Help
Back To Top
27°C
Mostly clear
Search
22:17:09
22-09-2025

Holiday Detail

Holidays are dates and times at which business hours are suspended. These dates and times, when associated with business hours, also suspend any escalation rules associated with business hours.

Add or remove business hours to holidays to suspend business hours and escalation rules during the holidays.

Business Hours [0]

Holiday Detail

Holiday Name	World Blood Donor Day		
Description	World Blood Donor Day - Celebrate and promote blood donation globally		
Date and Time	6/14/2026 All Day		
Recurring Holiday	Occurs every June 14 effective 6/14/2026		
Created By	Anamika Mishra 9/20/2025, 12:46 PM	Last Modified By	Anamika Mishra 9/20/2025, 12:46 PM

Business Hours

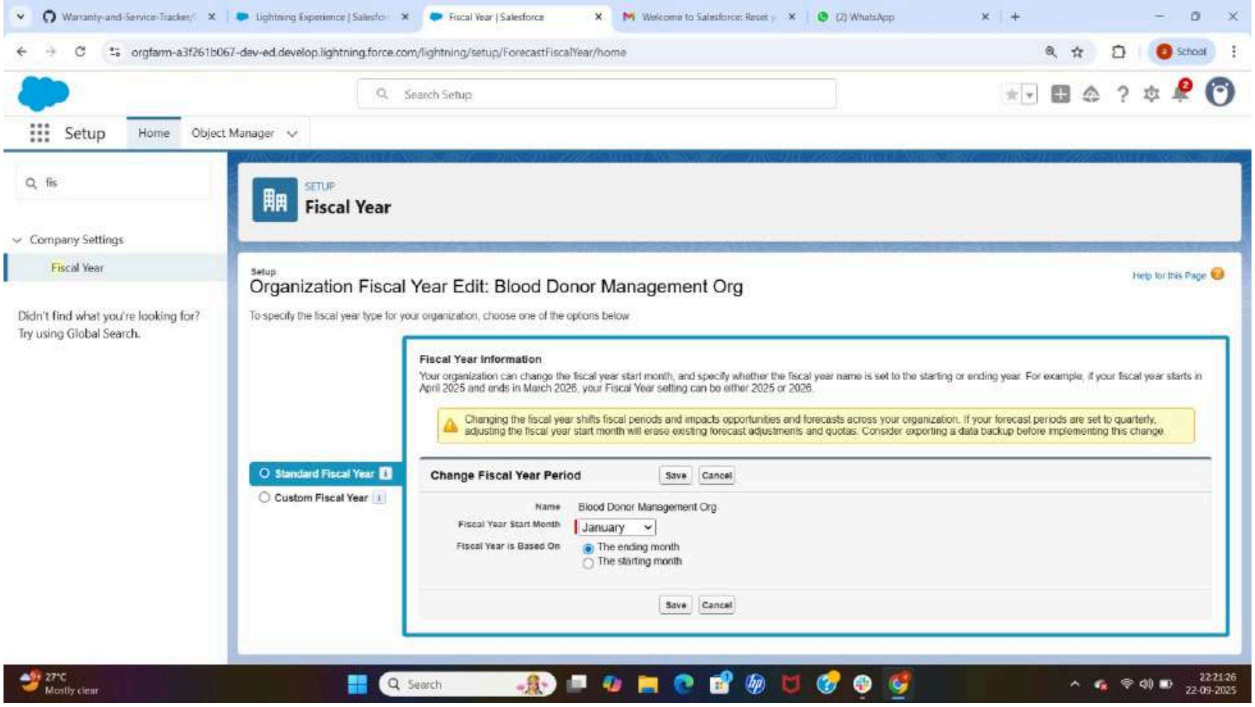
No records to display

Always show me more records per related list

Step 4. Fiscal Year Settings

Fiscal Year: Standard (January-December).

Custom Fiscal Year not enabled (not needed for this project).


Warranty-and-Service-Tracker/
Lightning Experience | Salesforce
Fiscal Year | Salesforce
Welcome to Salesforce: Reset
(2) WhatsApp
orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/ForecastFiscalYear/home
School
Search Setup
Setup
Home
Object Manager
fis
Company Settings
Fiscal Year
Didn't find what you're looking for? Try using Global Search.
SETUP
Fiscal Year
Help for this Page
27°C
Mostly clear
Search
22:21:26
22-09-2025

Setup

Organization Fiscal Year Edit: Blood Donor Management Org

To specify the fiscal year type for your organization, choose one of the options below

○ Standard Fiscal Year

○ Custom Fiscal Year

Fiscal Year Information

Your organization can change the fiscal year start month, and specify whether the fiscal year name is set to the starting or ending year. For example, if your fiscal year starts in April 2025 and ends in March 2026, your Fiscal Year setting can be either 2025 or 2026.

Changing the fiscal year shifts fiscal periods and impacts opportunities and forecasts across your organization. If your forecast periods are set to quarterly, adjusting the fiscal year start month will erase existing forecast adjustments and quotas. Consider exporting a data backup before implementing this change.

Change Fiscal Year Period Save Cancel

Name: Blood Donor Management Org

Fiscal Year Start Month: January

Fiscal Year is Based On: ● The ending month ○ The starting month

Save Cancel

Step 5. User Setup & Licenses

Created 2 Test Users

User 1:

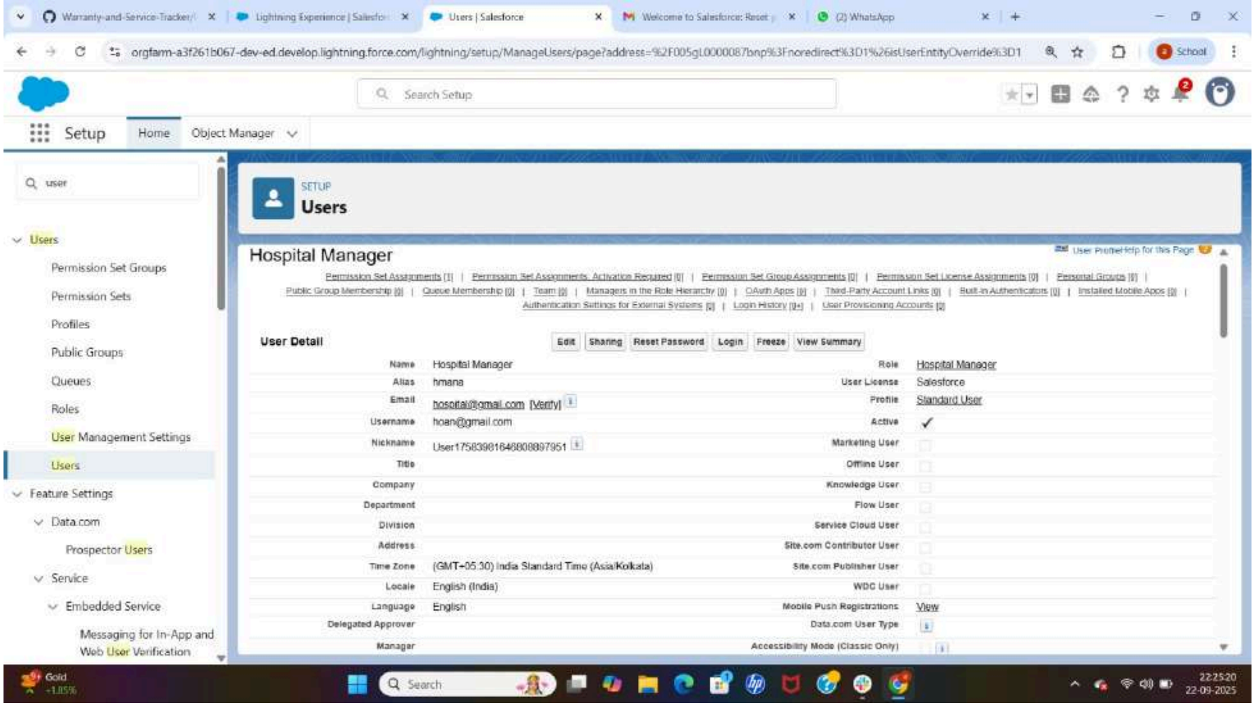
First Name: Hospital

Last Name: Manager

Role: Hospital Manager

User License: Salesforce

Profile: Standard User


Setup
Home
Object Manager
Search Setup
user
Users
Permission Set Groups
Permission Sets
Profiles
Public Groups
Queues
Roles
User Management Settings
Users
Feature Settings
Data.com
Prospector Users
Service
Embedded Service
Messaging for In-App and Web User Verification
SETUP
Users
User Profile Help for this Page
Permission Set Assignments [1] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] |
Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [0] | OAuth Apps [0] | Third-Party Account Links [0] | Built-in Authenticators [0] | Installed Mobile Apps [0] |
Authentication Settings for External Systems [0] | Login History [9+] | User Provisioning Accounts [0]
Edit
Sharing
Reset Password
Login
Freeze
View Summary

Hospital Manager

User Detail

Name	Hospital Manager	Role	Hospital Manager
Alias	hmana	User License	Salesforce
Email	hospital@gmail.com [Verify]	Profile	Standard User
Username	hoan@gmail.com	Active	✓
Nickname	User17583981646808897951	Marketing User	☐
Title		Offline User	☐
Company		Knowledge User	☐
Department		Flow User	☐
Division		Service Cloud User	☐
Address		Site.com Contributor User	☐
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	☐
Locale	English (India)	WDC User	☐
Language	English	Mobile Push Registrations	View
Delegated Approver		Data.com User Type	
Manager		Accessibility Mode (Classic Only)	☐

User 2:

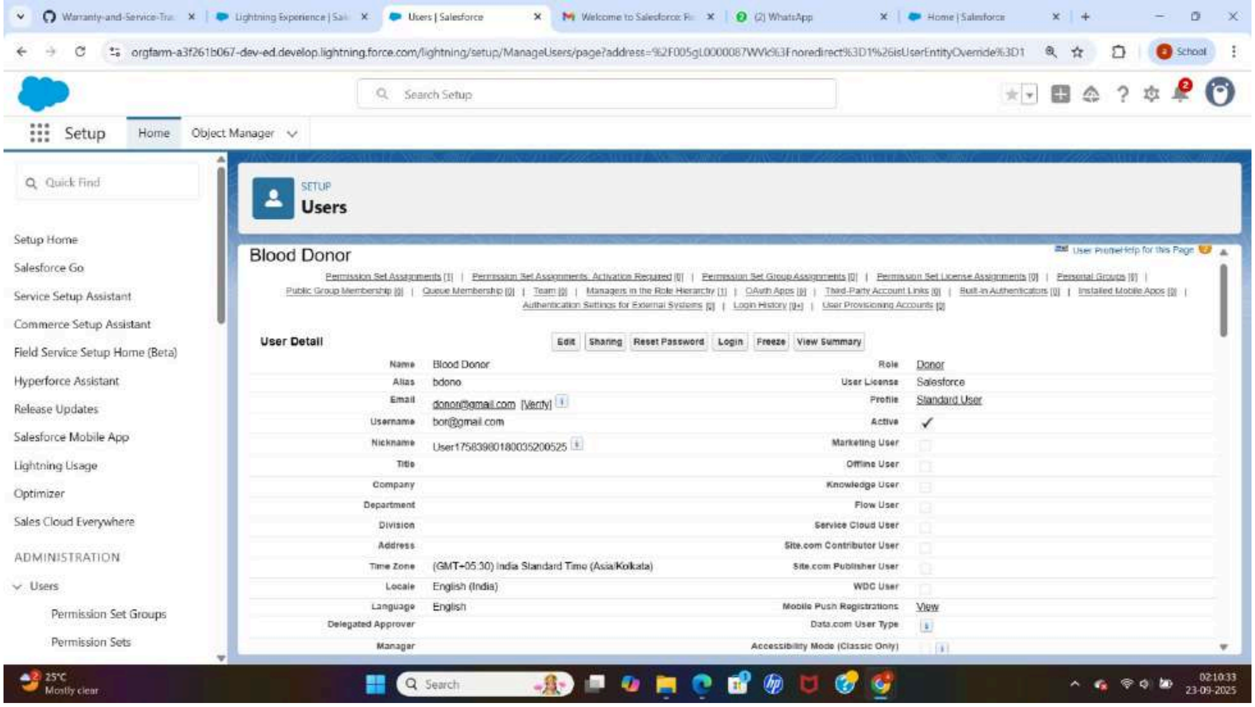
First Name: Blood

Last Name: Donor

Role: Donor

User License: Salesforce

Profile: Standard User



SETUP
Users

Blood Donor

Permission Set Assignments [1] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] | Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [1] | OAuth Apps [0] | Third-Party Account Links [0] | Built-in Authenticators [0] | Installed Mobile Apps [0] | Authentication Settings for External Systems [0] | Login History [0+] | User Provisioning Accounts [0]

User Detail

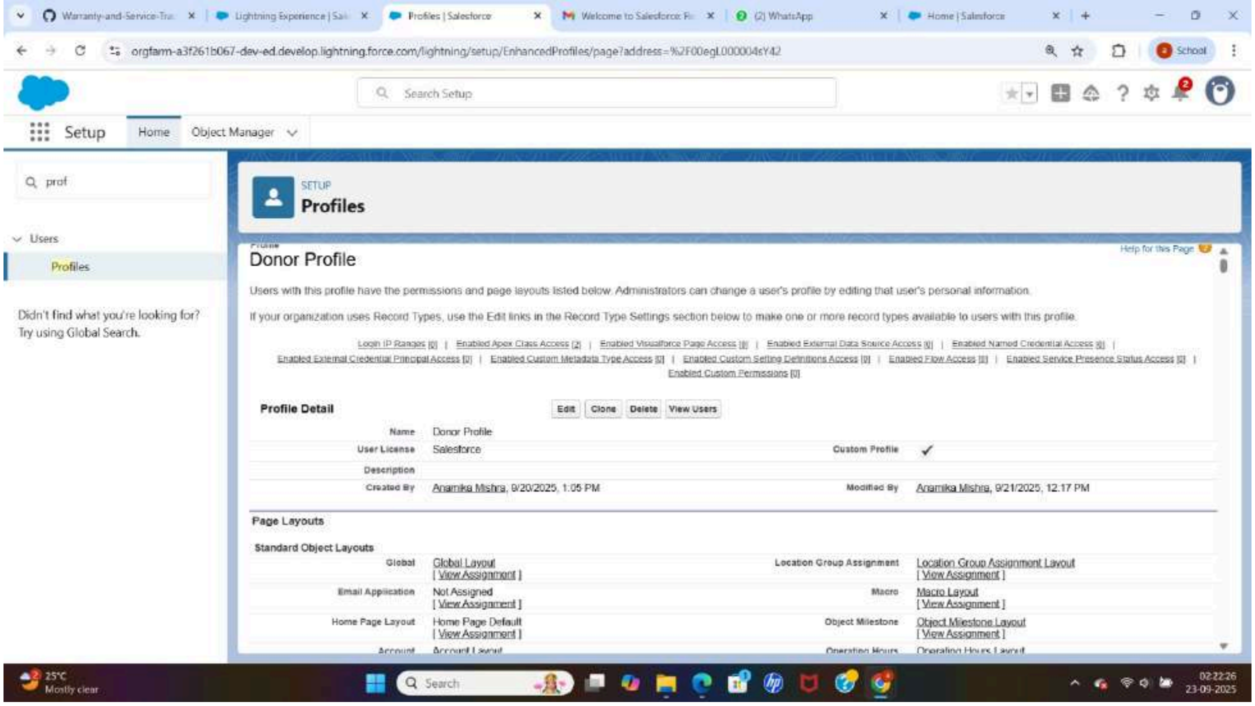
Name	Blood Donor	Role	Donor
Alias	bdono	User License	Salesforce
Email	donor@gmail.com [Verify]	Profile	Standard User
Username	bor@gmail.com	Active	✓
Nickname	User17583980180035200525	Marketing User	□
Title		Offline User	□
Company		Knowledge User	□
Department		Flow User	□
Division		Service Cloud User	□
Address		Site.com Contributor User	□
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	□
Locale	English (India)	WDC User	□
Language	English	Mobile Push Registrations	View
Delegated Approver		Data.com User Type	
Manager		Accessibility Mode (Classic Only)	□

Step 6. Profiles

1. Go to Setup → Profiles.

2. Clone a profile:

Name: Donor Profile
Permission: Donor__c → Read/Edit


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/EnhancedProfiles/page?address=%2F00egL000004sY42
Search Setup
Setup
Home
Object Manager
prof
Users
Profiles
Didn't find what you're looking for? Try using Global Search.
Edit
Clone
Delete
View Users
25°C Mostly clear
Search
02:22:26 23-09-2025

SETUP
Profiles

Donor Profile

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges [0] | Enabled Apex Class Access [2] | Enabled Visualforce Page Access [0] | Enabled External Data Source Access [0] | Enabled Named Credential Access [0] | Enabled External Credential Principal Access [0] | Enabled Custom Metadata Type Access [0] | Enabled Custom Setting Definitions Access [0] | Enabled Flow Access [0] | Enabled Service Presence Status Access [0] | Enabled Custom Permissions [0]

Profile Detail

Name	Donor Profile		
User License	Salesforce	Custom Profile	✓
Description			
Created By	Anamika Mishra, 9/20/2025, 1:05 PM	Modified By	Anamika Mishra, 9/21/2025, 12:17 PM

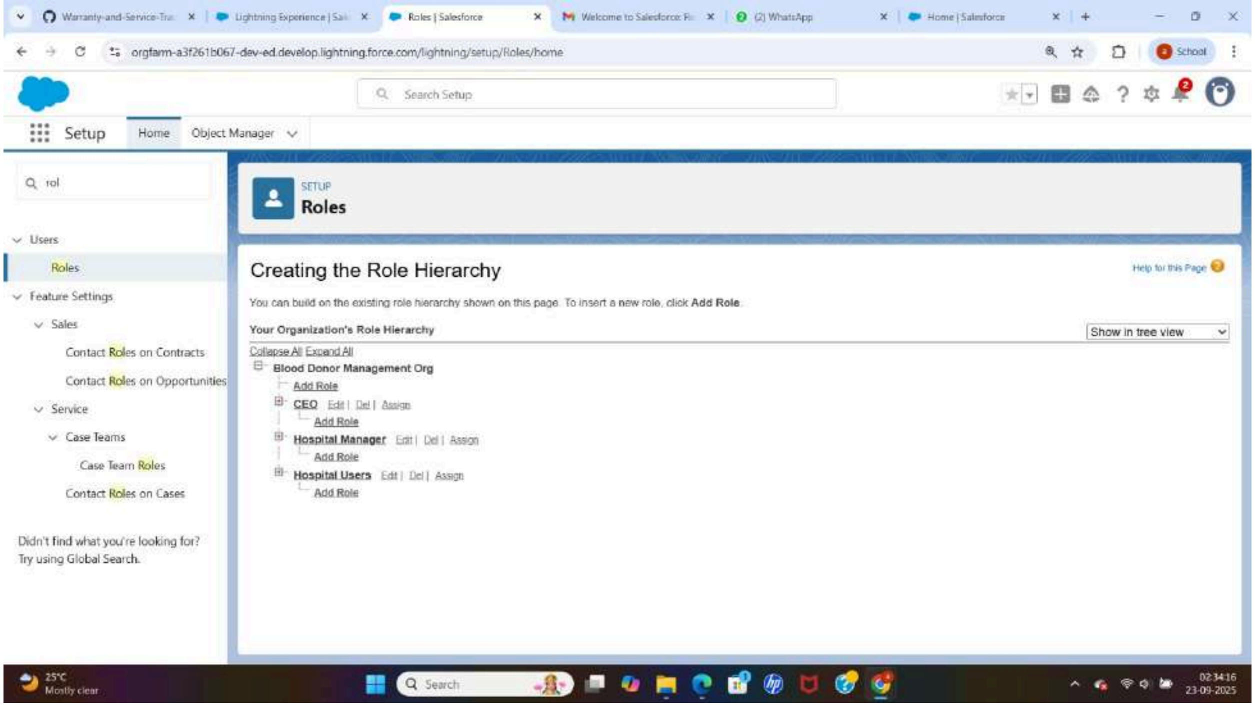
Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Macro	Macro Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]
Account	Account Layout	Operating Hours	Operating Hours Layout

Step 7. Roles

Role hierarchy created
Blood Donor Management Org (top)
Hospital Manager



SETUP
Roles

Creating the Role Hierarchy

Help for this Page

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

Show in tree view

Collapse All Expand All

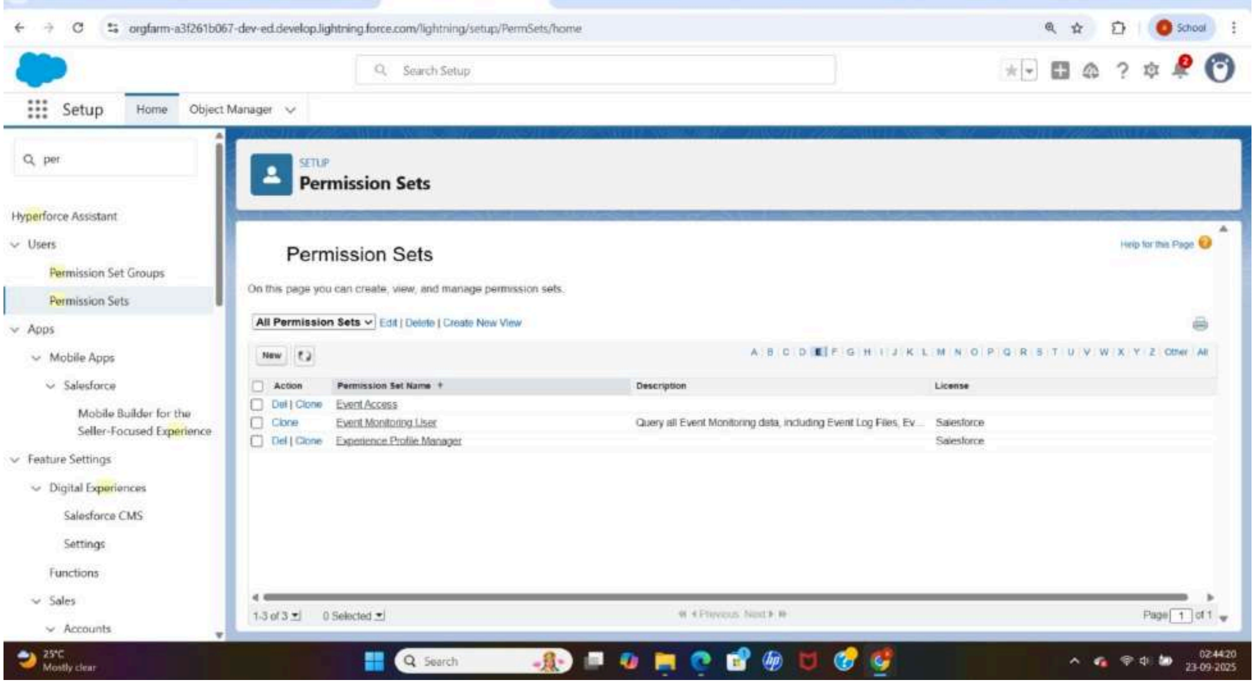
- **Blood Donor Management Org**
  - Add Role
  - **CEO** Edit | Del | Assign
    - Add Role
  - **Hospital Manager** Edit | Del | Assign
    - Add Role
  - **Hospital Users** Edit | Del | Assign
    - Add Role

Step 8. Permission sets

Name: Event Access

Object Settings: Donation_Event__c → Read/Write (enabled)

Assign this permission set to Hospital User



Permission Sets

On this page you can create, view, and manage permission sets.

	Action	Permission Set Name	Description	License
□	Del | Clone	Event Access		
□	Clone	Event Monitoring User	Query all Event Monitoring data, including Event Log Files, Ev...	Salesforce
□	Del | Clone	Experience Profile Manager		Salesforce

Step 9. OWD (Organization-Wide Defaults)

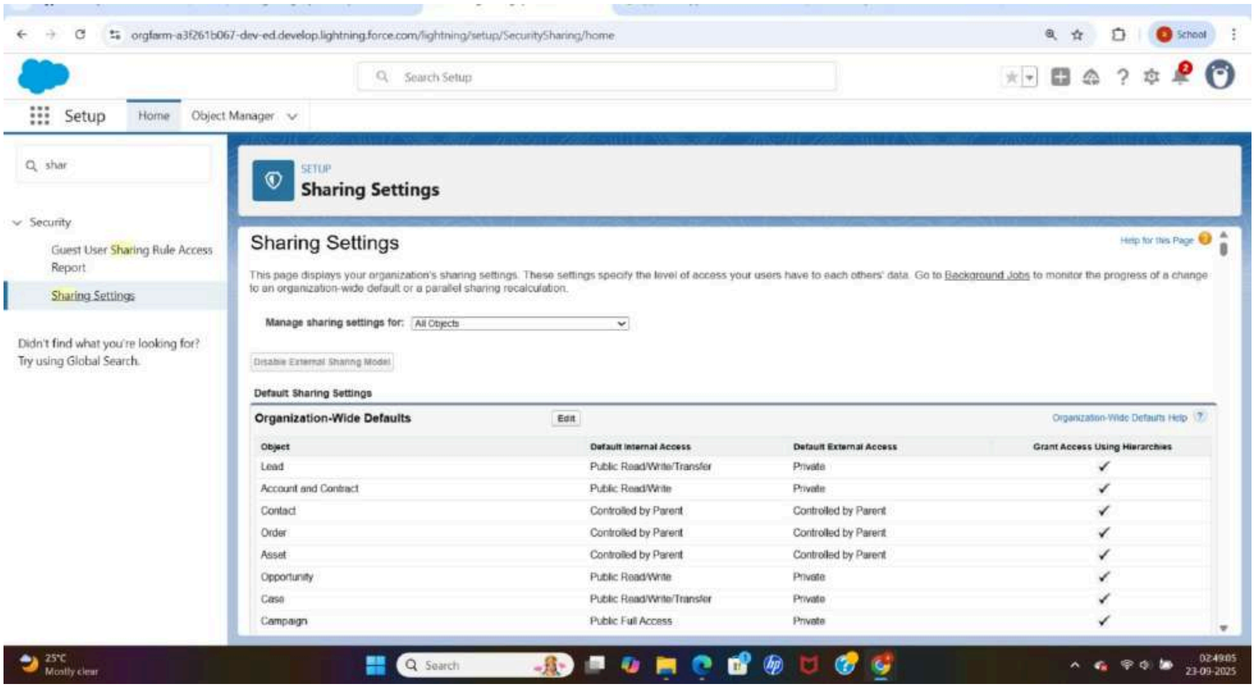
1. Go to Setup → Sharing Settings.

2. Set object access:

Donor__c → Private

Donation_Request__c → Public Read-Only

Donation_Event__c → Public Read/Write


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home
School
Search Setup
Setup
Home
Object Manager
shar
Security
Guest User Sharing Rule Access Report
Sharing Settings
Didn't find what you're looking for? Try using Global Search.
SETUP
Sharing Settings
Help for this Page
Search
25°C
Mostly clear
02:49:05
23-09-2025

Sharing Settings

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to Background Jobs to monitor the progress of a change to an organization-wide default or a parallel sharing recalculation.

Manage sharing settings for: All Objects

Disable External Sharing Model

Default Sharing Settings

Organization-Wide Defaults Edit Organization-Wide Defaults Help

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
Campaign	Public Full Access	Private	✓

Step 10. Sharing Rules

1. Go to Setup → Sharing Settings.

2. For Donor__c, create a new sharing rule:

Name: Share Donors with Hospital Manager

Rule Type: Based on Role

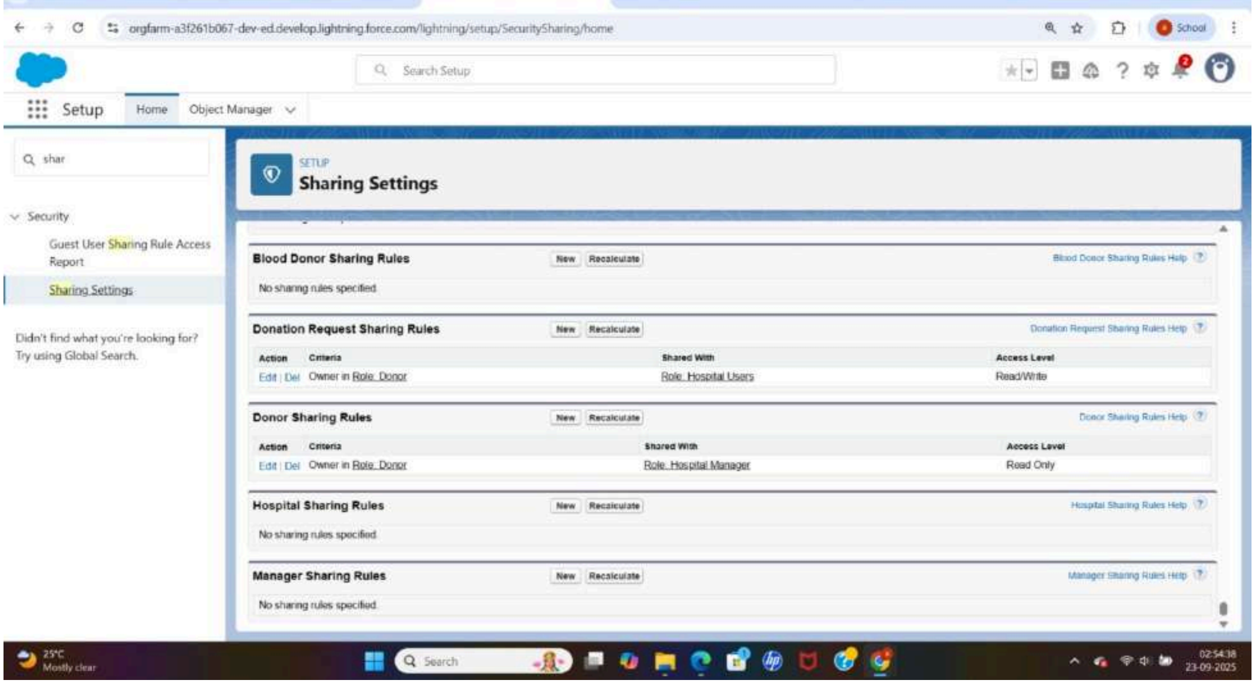
Share with: Hospital Manager

Access: Read Only

3. For Donation_Request__c, create a new sharing rule:

Share with: All Hospital Users

Access: Read/Write


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home
Search Setup
Setup
Home
Object Manager
shar
Security
Guest User Sharing Rule Access Report
Sharing Settings
Didn't find what you're looking for? Try using Global Search.
SETUP

Sharing Settings

Blood Donor Sharing Rules New Recalculate Blood Donor Sharing Rules Help

No sharing rules specified

Donation Request Sharing Rules New Recalculate Donation Request Sharing Rules Help

Action	Criteria	Shared With	Access Level
Edit | Del	Owner in Role: Donor	Role: Hospital Users	Read/Write

Donor Sharing Rules New Recalculate Donor Sharing Rules Help

Action	Criteria	Shared With	Access Level
Edit | Del	Owner in Role: Donor	Role: Hospital Manager	Read Only

Hospital Sharing Rules New Recalculate Hospital Sharing Rules Help

No sharing rules specified

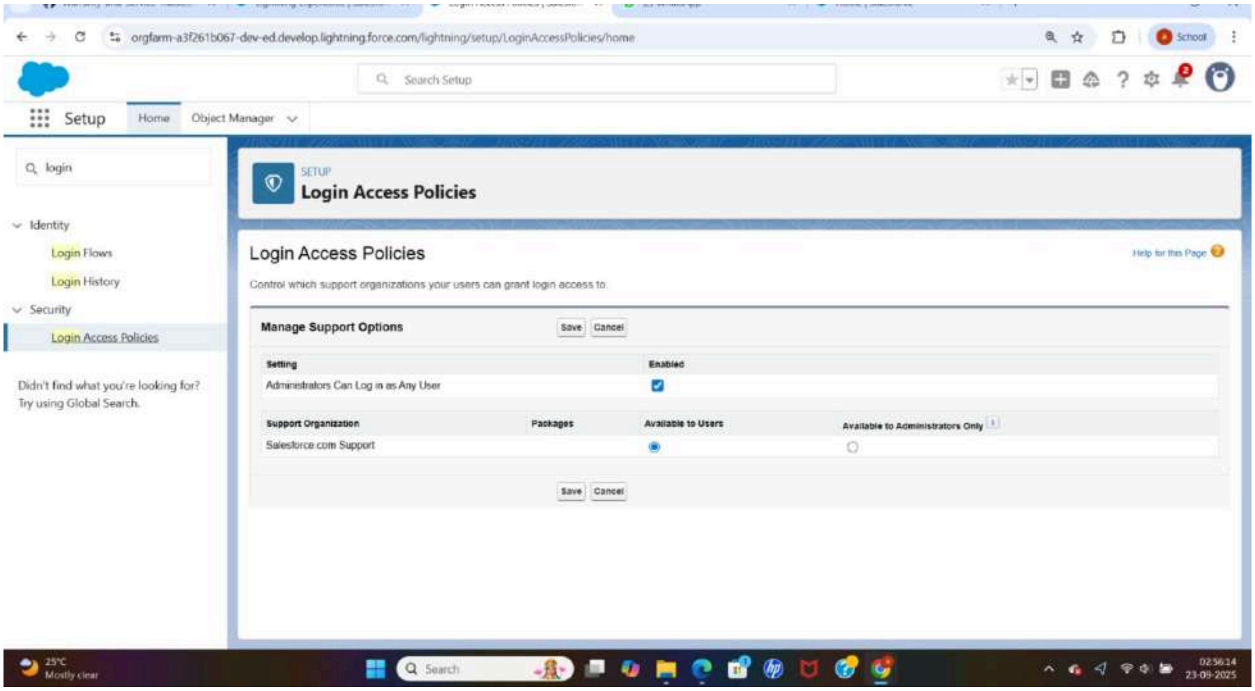
Manager Sharing Rules New Recalculate Manager Sharing Rules Help

No sharing rules specified

Step 11: Login Access Policies

1. Setup Login Access Policies

2. Enable: Administrators Can Log in as Any User



Login Access Policies

Control which support organizations your users can grant login access to.

Manage Support Options

Setting	Enabled
Administrators Can Log in as Any User	☑

Support Organization	Packages	Available to Users	Available to Administrators Only
Salesforce.com Support		●	○

Step 12. Developer Org Setup

Project developed in Salesforce Developer Edition Org.

Provides free Enterprise-level features suitable for learning and development purposes.

Step 13. Sandbox Usage

Sandboxes are not available in Developer Edition.

In real-world companies, sandboxes are used for testing, development, and training without affecting Production.

Step 14. Deployment Basics

Deployment methods include Change Sets or Salesforce CLI / VS Code.

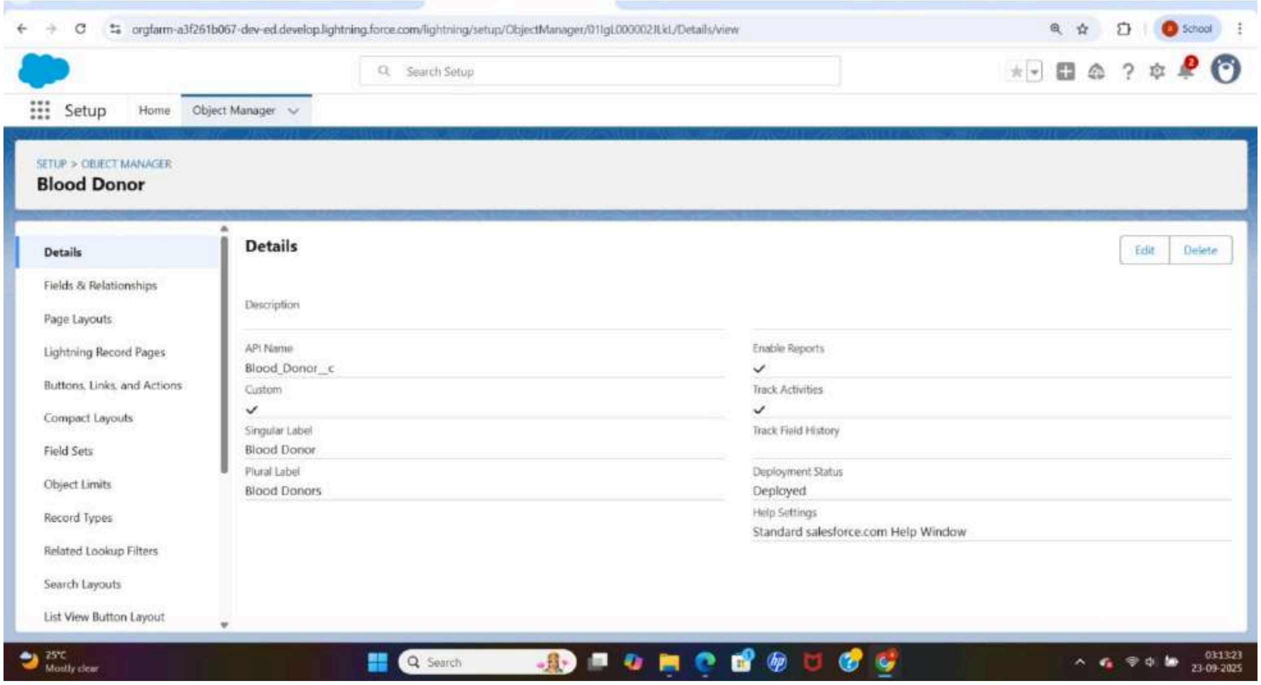
Not required here since the project is fully developed and tested within a single Developer Org.

Phase 3: Data Modeling & Relationships

Step 1. Standard & Custom Objects

Go to Setup → Object Manager → Create Custom Object.
Create these objects:

- Blood Donor
- Donation Event
- Donation Registration



Blood Donor

Details

Description

API Name
Blood_Donor__c

Custom
✓

Singular Label
Blood Donor

Plural Label
Blood Donors

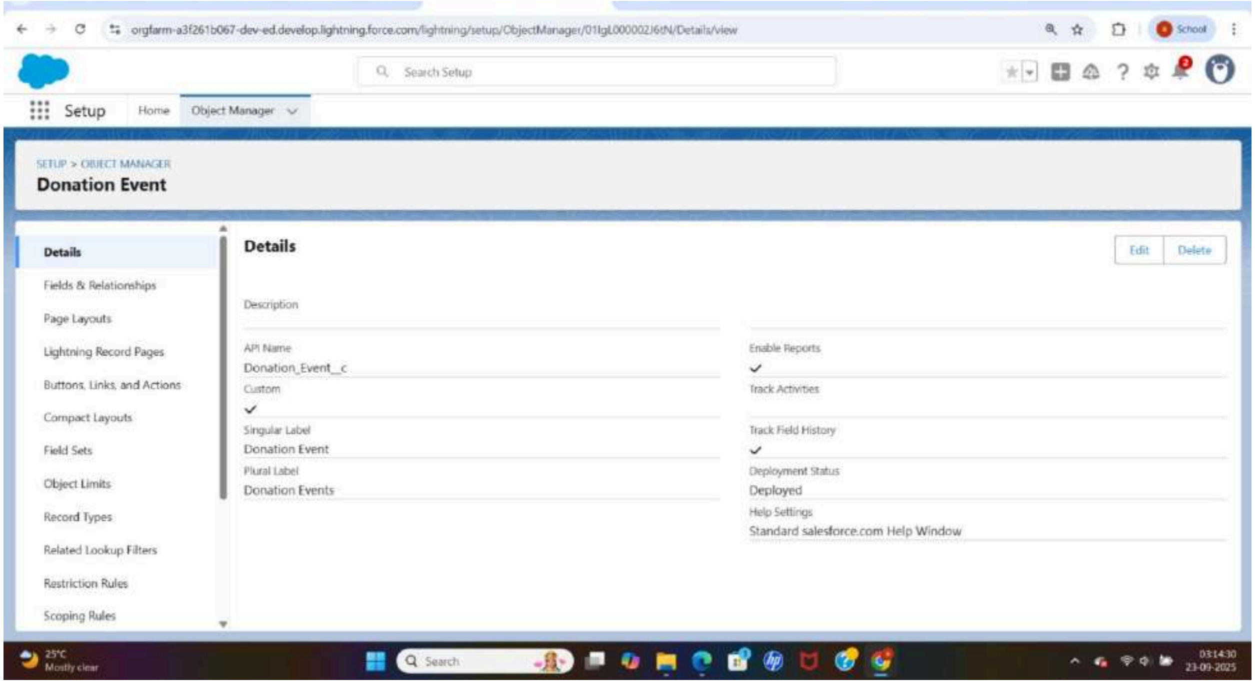
Enable Reports
✓

Track Activities
✓

Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window



SETUP > OBJECT MANAGER

Donation Event

Details

Description

API Name
Donation_Event__c

Custom
✓

Singular Label
Donation Event

Plural Label
Donation Events

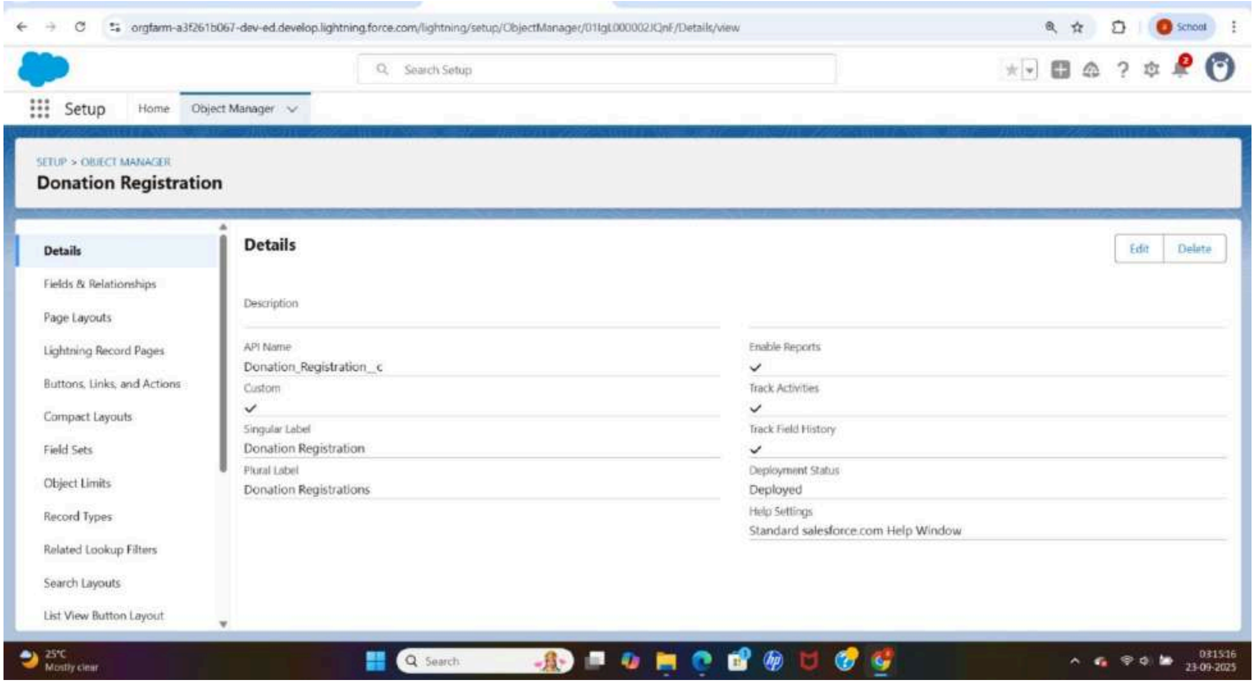
Enable Reports
✓

Track Activities

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window



SETUP > OBJECT MANAGER

Donation Registration

Details

Description

API Name
Donation_Registration__c

Custom
✓

Singular Label
Donation Registration

Plural Label
Donation Registrations

Enable Reports
✓

Track Activities
✓

Track Field History
✓

Deployment Status
Deployed

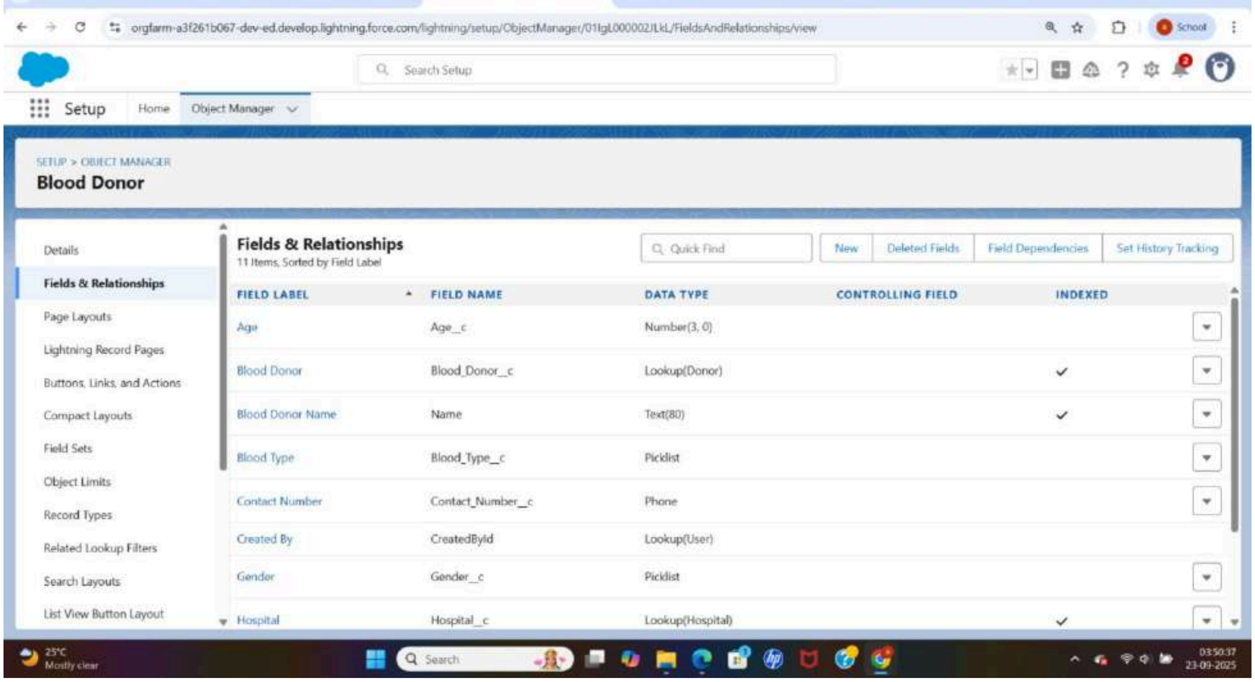
Help Settings
Standard salesforce.com Help Window

Step 2. Fields

1. Go to Object Manager → [Object] → Fields & Relationships → New.

2. Blood Donor fields:

- Blood Type → Picklist
- Age → Number
- Gender → Picklist
- Last Donation Date → Date
- Contact Number → Phone
- Hospital → Lookup



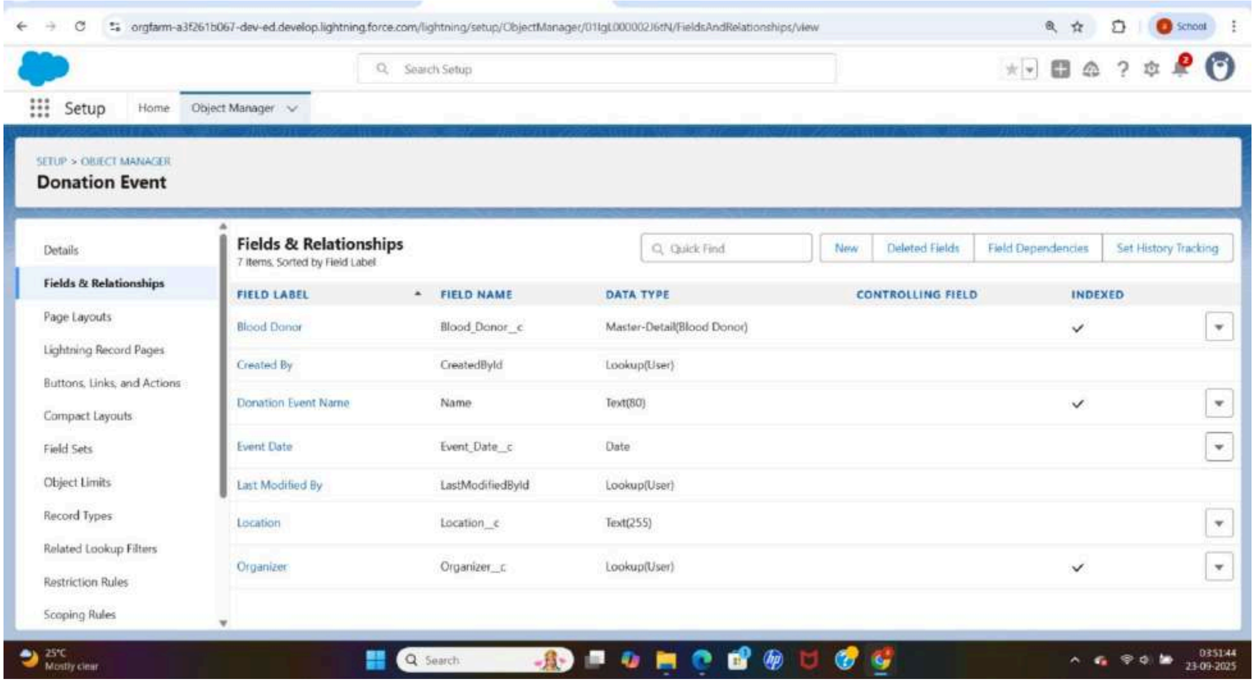
Blood Donor

Fields & Relationships
11 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age__c	Number(3, 0)		
Blood Donor	Blood_Donor__c	Lookup(Donor)		✓
Blood Donor Name	Name	Text(80)		✓
Blood Type	Blood_Type__c	Picklist		
Contact Number	Contact_Number__c	Phone		
Created By	CreatedById	Lookup(User)		
Gender	Gender__c	Picklist		
Hospital	Hospital__c	Lookup(Hospital)		✓

3. Donation Event fields:

- Event Date → Date
- Location → Text.
- Organizer → Lookup



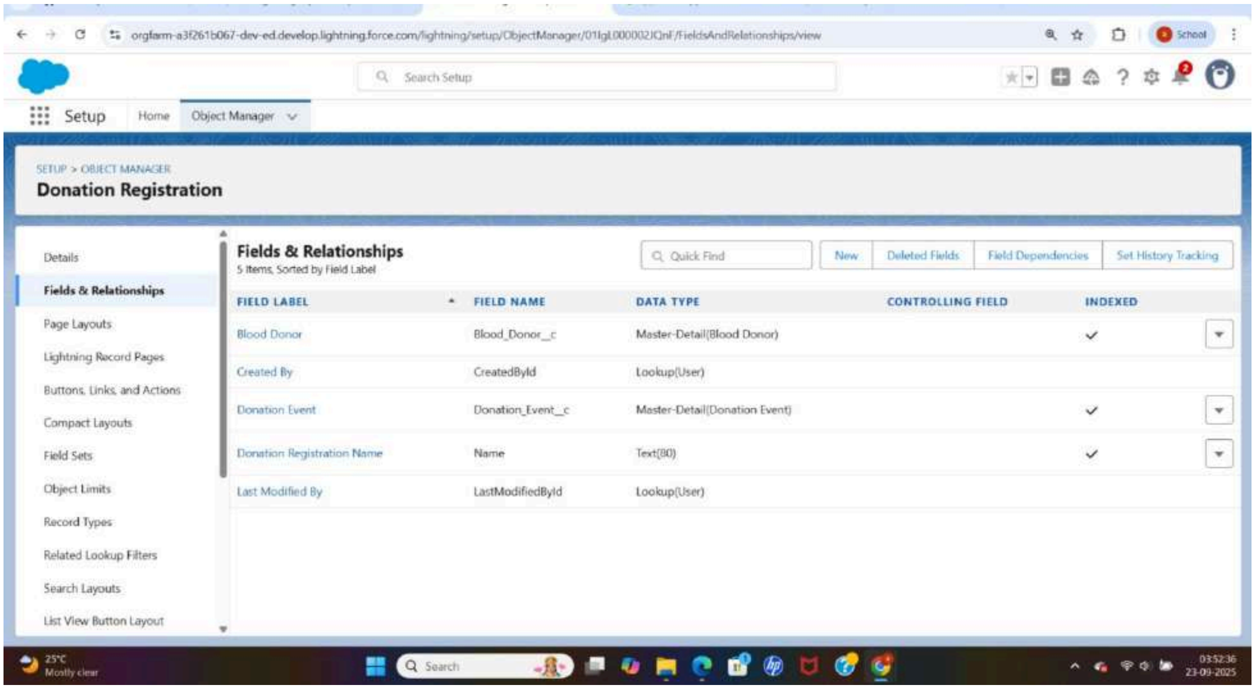
Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Donor	Blood_Donor__c	Master-Detail(Blood Donor)		✓
Created By	CreatedById	Lookup(User)		
Donation Event Name	Name	Text(80)		✓
Event Date	Event_Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Text(255)		
Organizer	Organizer__c	Lookup(User)		✓

4. Donation Registration fields:

- Donor → Master-Detail (Blood Donor)
- Event → Master-Detail (Donation Event)



Fields & Relationships

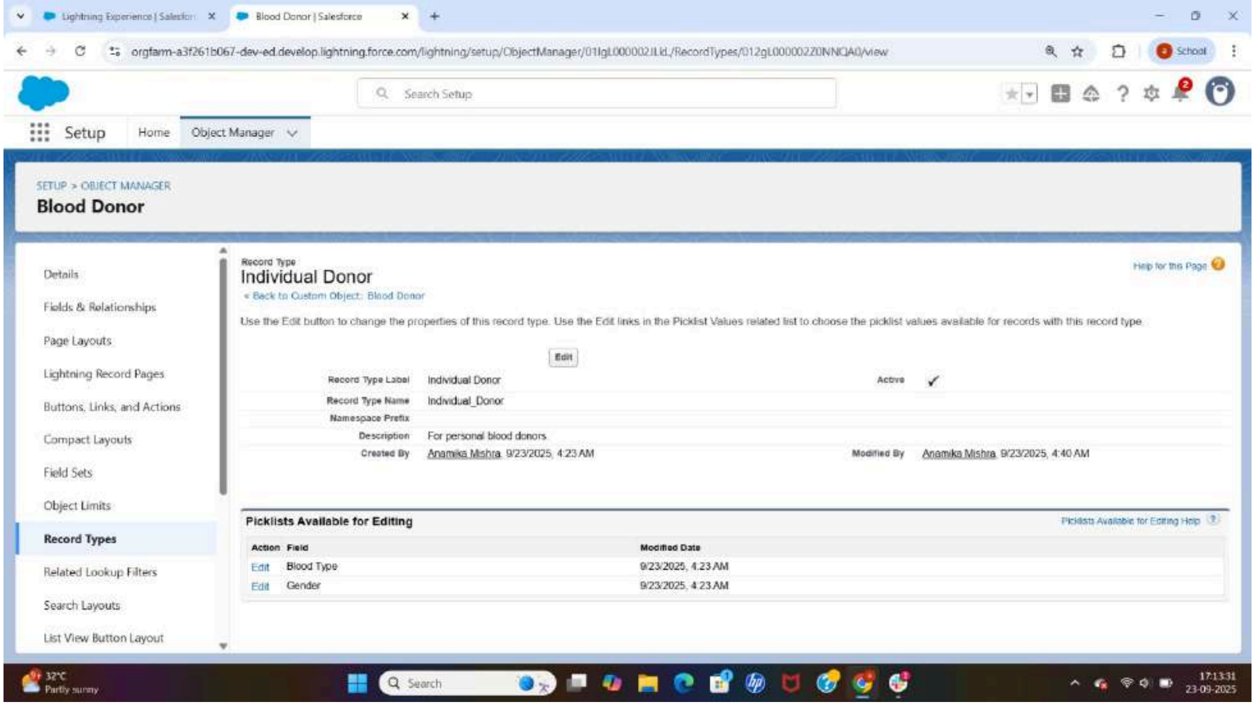
5 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Donor	Blood_Donor__c	Master-Detail(Blood Donor)		✓
Created By	CreatedById	Lookup(User)		
Donation Event	Donation_Event__c	Master-Detail(Donation Event)		✓
Donation Registration Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		

Step 3. Record Types

1. Record Types = different layouts/picklists.

2. Blood Donor → Individual & Corporate Donor.
3. Setup → Object Manager → Record Types → New.



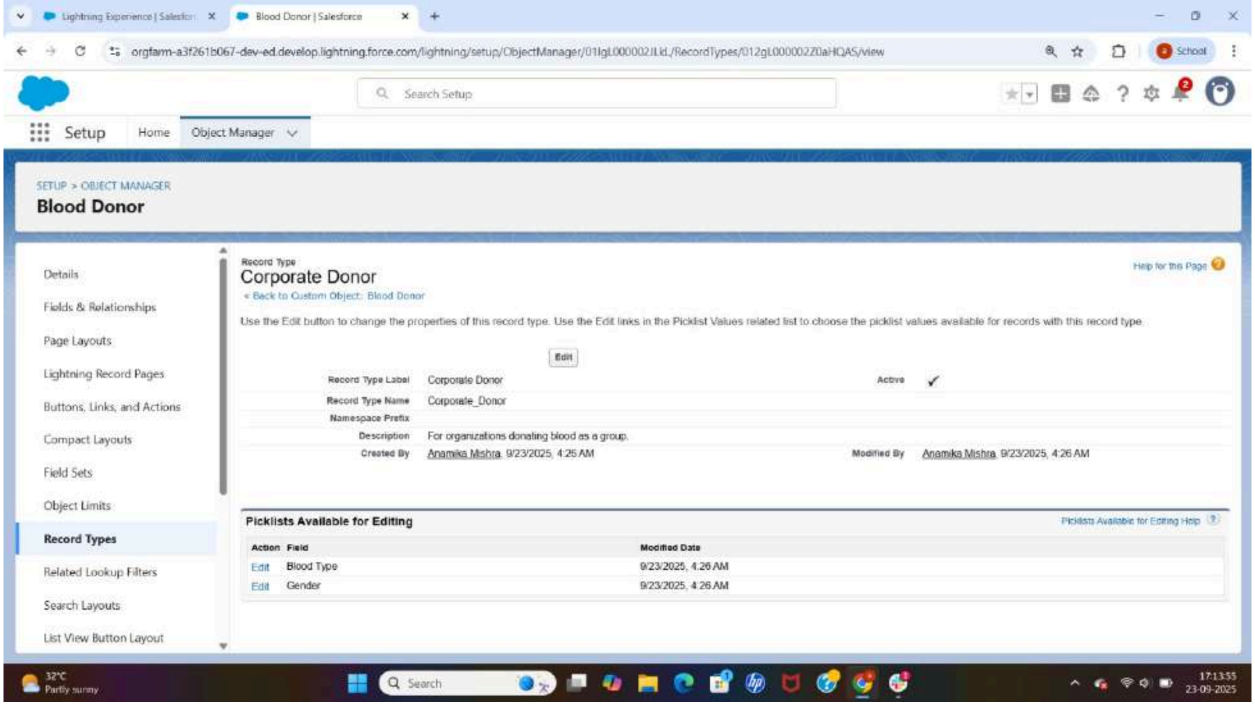
Record Type
Individual Donor
« Back to Custom Object: Blood Donor

Use the Edit button to change the properties of this record type. Use the Edit links in the Picklist Values related list to choose the picklist values available for records with this record type.

Record Type Label: Individual Donor
Active: ✓
Record Type Name: Individual_Donor
Namespace Prefix:
Description: For personal blood donors
Created By: Anamika Mishra, 9/23/2025, 4:23 AM
Modified By: Anamika Mishra, 9/23/2025, 4:40 AM

Picklists Available for Editing

Action	Field	Modified Date
Edit	Blood Type	9/23/2025, 4:23 AM
Edit	Gender	9/23/2025, 4:23 AM



Record Type
Corporate Donor
« Back to Custom Object: Blood Donor

Use the Edit button to change the properties of this record type. Use the Edit links in the Picklist Values related list to choose the picklist values available for records with this record type.

Record Type Label: Corporate Donor
Active: ✓
Record Type Name: Corporate_Donor
Namespace Prefix:
Description: For organizations donating blood as a group.
Created By: Anamika Mishra, 9/23/2025, 4:25 AM
Modified By: Anamika Mishra, 9/23/2025, 4:26 AM

Picklists Available for Editing

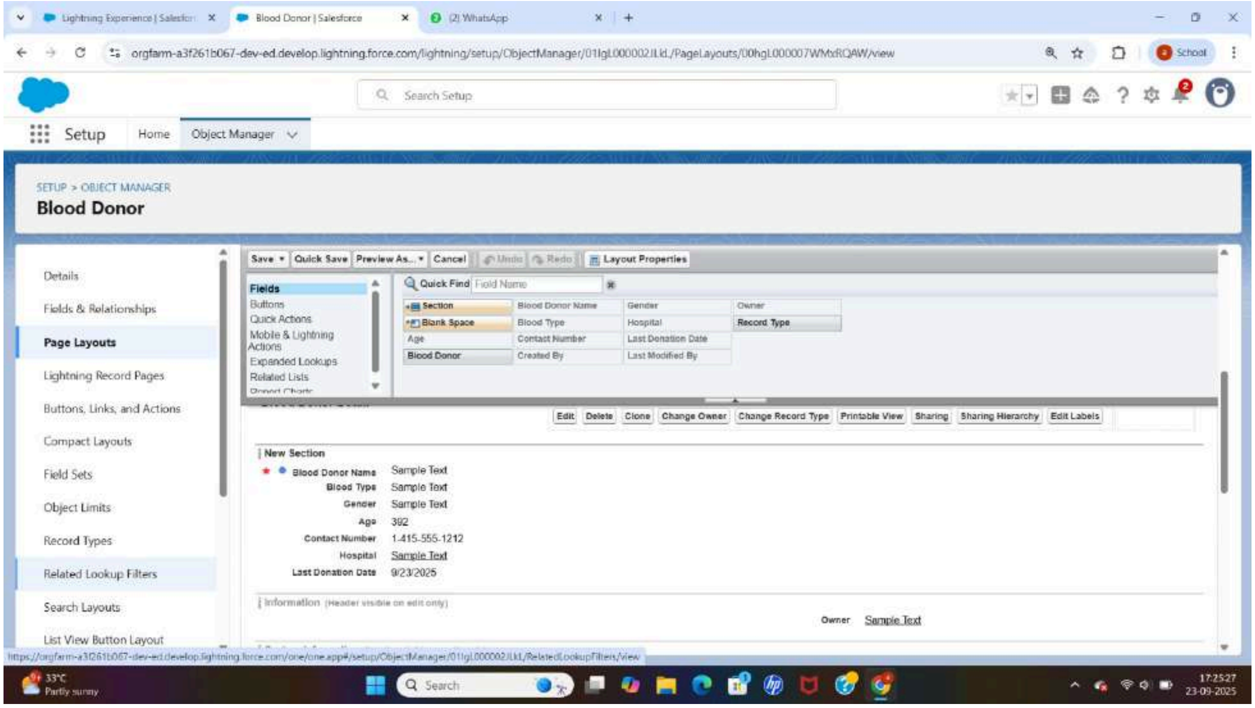
Action	Field	Modified Date
Edit	Blood Type	9/23/2025, 4:26 AM
Edit	Gender	9/23/2025, 4:26 AM

Step 4. Page Layouts

Setup: Object Manager → Object → Page Layouts → New

Sections

Blood Donor - Personal Information, Blood Type , Hospital, Last Donation Date



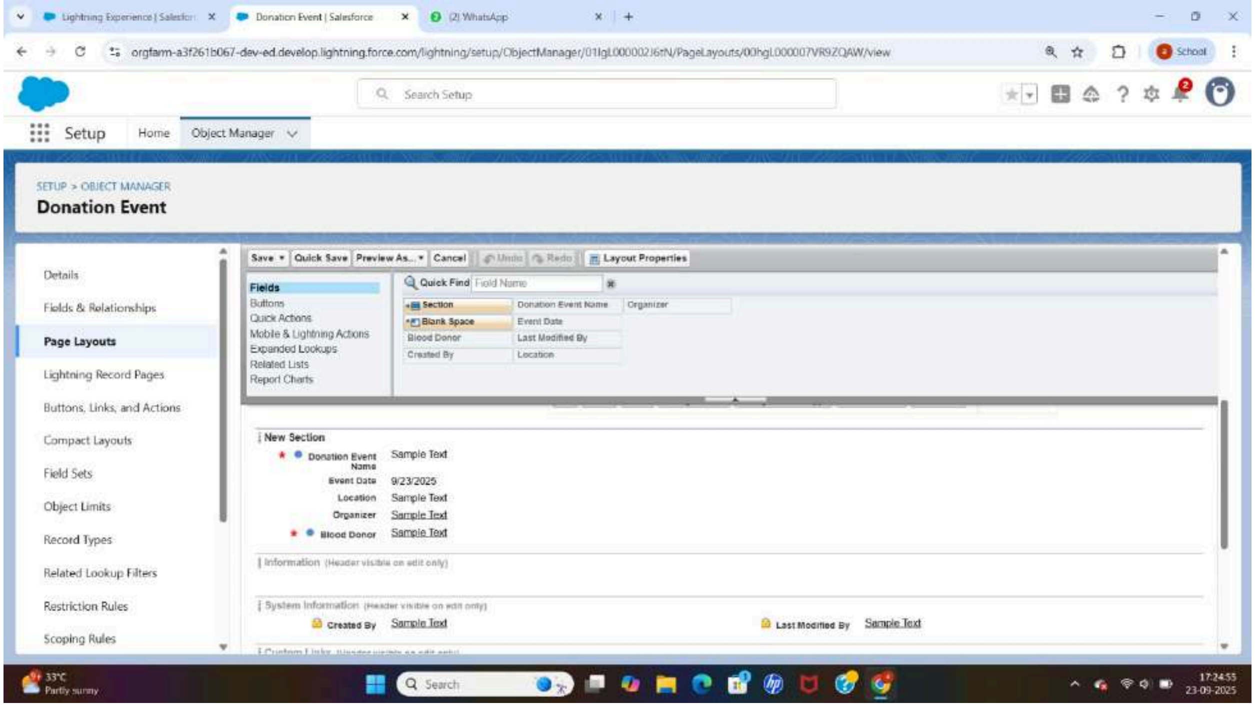
New Section

Blood Donor Name	Sample Text
Blood Type	Sample Text
Gender	Sample Text
Age	392
Contact Number	1-415-555-1212
Hospital	Sample Text
Last Donation Date	9/23/2025

Information (Header visible on edit only)

Owner Sample Text

Donation Event - Event Information, Organizer, Location



New Section

Donation Event Name	Sample Text
Event Date	9/23/2025
Location	Sample Text
Organizer	Sample Text
Blood Donor	Sample Text

Information (Header visible on edit only)

System Information (Header visible on edit only)

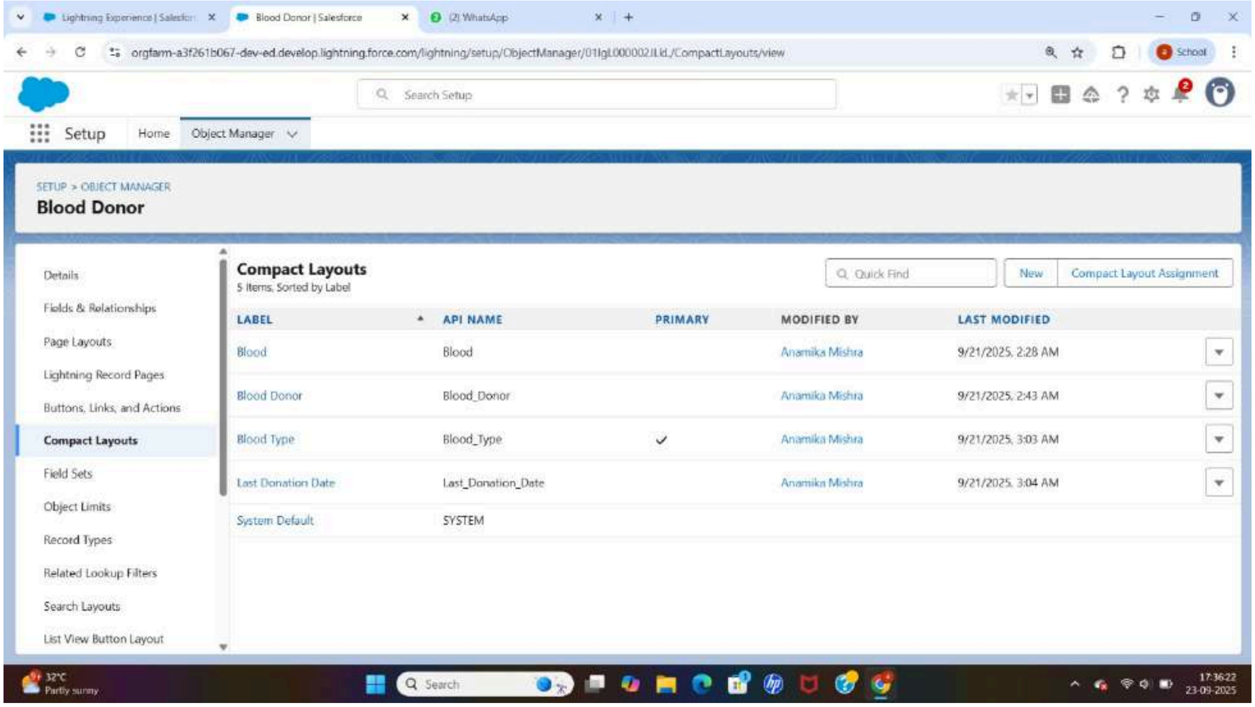
Created By Sample Text

Last Modified By Sample Text

Custom Links (Header visible on edit only)

Step 5. Compact Layouts

Setup: Object Manager → [Object] → Compact Layouts → New
Example: Blood Donor Last Donation Date
Fields: Name, Blood Type



Blood Donor

Compact Layouts
5 Items, Sorted by Label

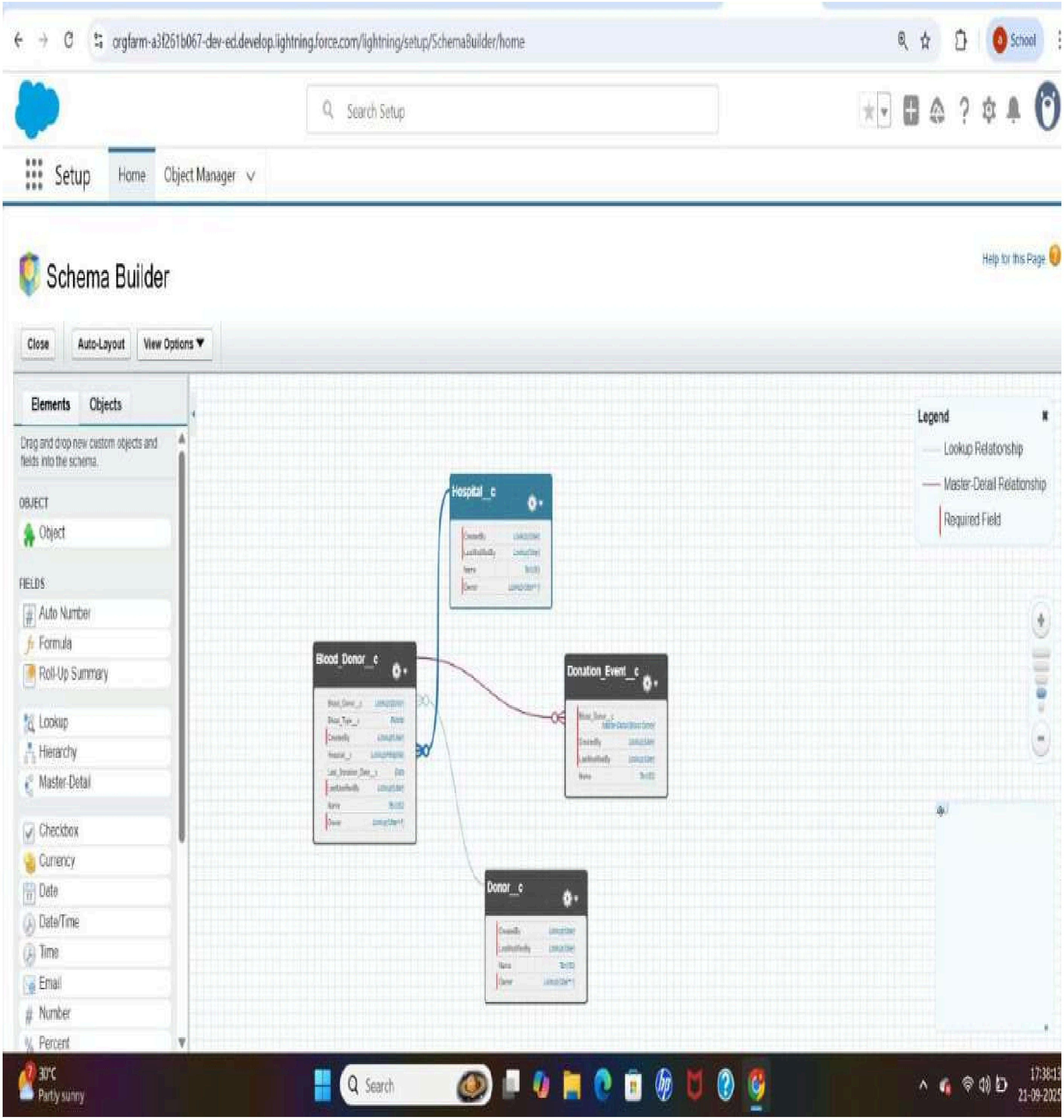
LABEL	API NAME	PRIMARY	MODIFIED BY	LAST MODIFIED
Blood	Blood		Anamika Mishra	9/21/2025, 2:28 AM
Blood Donor	Blood_Donor		Anamika Mishra	9/21/2025, 2:43 AM
Blood Type	Blood_Type	✓	Anamika Mishra	9/21/2025, 3:03 AM
Last Donation Date	Last_Donation_Date		Anamika Mishra	9/21/2025, 3:04 AM
System Default	SYSTEM			

Step 6. Schema Builder

Go to Setup → Schema Builder.

Drag objects onto canvas.

Create Master-Detail and Lookup relationships visually.



Step 7. Lookup vs Master-Detail vs Hierarchical Relationships

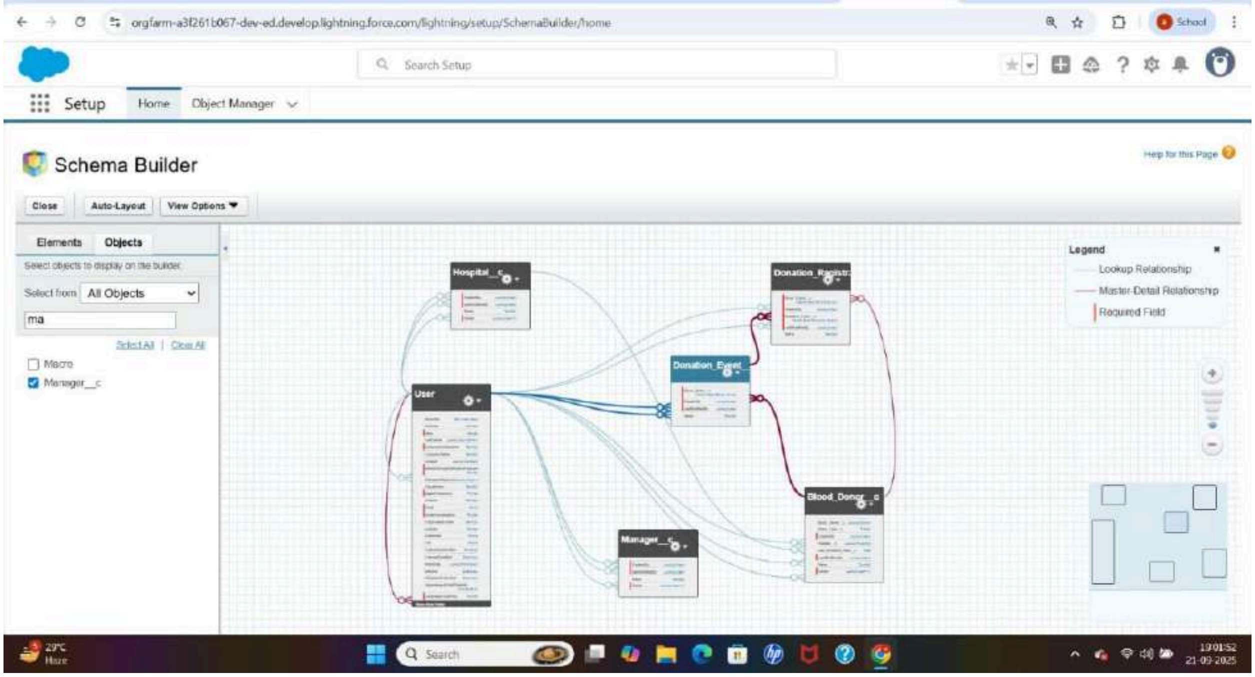
- Relationship Type - Lookup

- Example - Blood Donor Hospital
- Reason for Choice - Donor can exist independently

- Relationship Type - Master-Detail
- Example - Donation Registration   Donor
- Reason for Choice - Strong dependency

- Relationship Type - Master-Detail
- Example - Donation Registration Event
- Reason for Choice - Strong dependency

- Relationship Type - Hierarchical
- Example - User Manager
- Reason for Choice - Only for User reporting



Step 8. Junction Objects

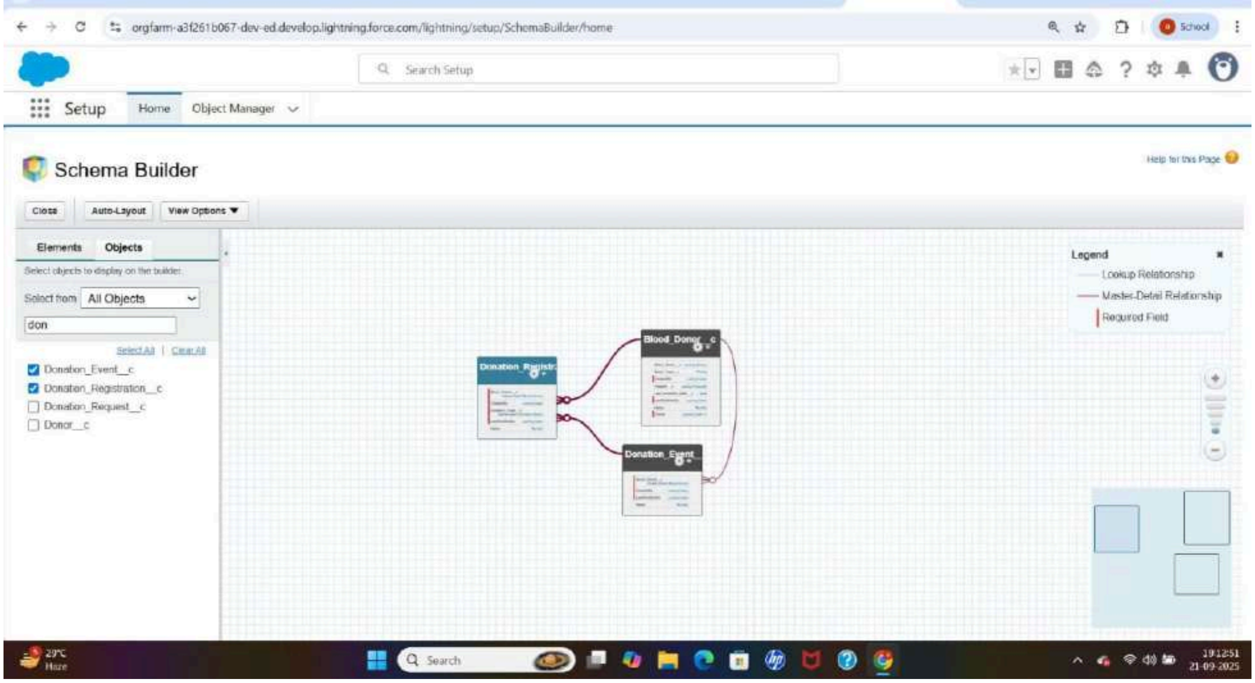
Used for many-to-many relationships

Donor - Donation Event

Junction Object - Donation Registration

Master-Detail to Blood Donor

Master-Detail to Donation Event

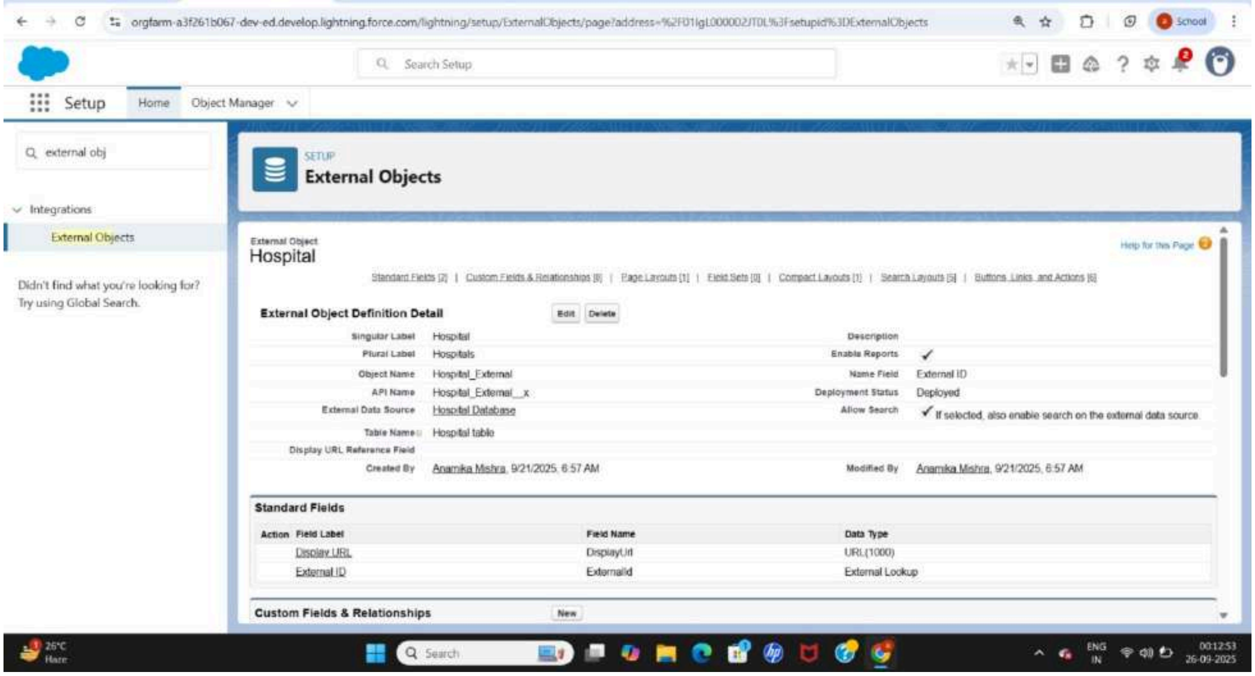


Step 9. External Objects

Use to access data from external systems without importing it into Salesforce.

Example: Connect to a hospital database outside Salesforce.

Setup: Setup → External Objects → Create → Connect using Salesforce Connect.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/ExternalObjects/page?address=%2F01IgL000002JT0L%3Fsetupid%3DExternalObjects
Search Setup
Setup
Home
Object Manager
external obj
Integrations
External Objects
Didn't find what you're looking for? Try using Global Search.
SETUP
External Objects
Help for this Page
Standard Fields [2] | Custom Fields & Relationships [8] | Page Layouts [1] | Field Sets [0] | Compact Layouts [1] | Search Layouts [5] | Buttons, Links, and Actions [6]
Edit
Delete
New

External Object
Hospital

External Object Definition Detail

Singular Label	Hospital	Description	
Plural Label	Hospitals	Enable Reports	✓
Object Name	Hospital_External	Name Field	External ID
API Name	Hospital_External__x	Deployment Status	Deployed
External Data Source	Hospital Database	Allow Search	✓ If selected, also enable search on the external data source.
Table Name	Hospital table		
Display URL Reference Field			
Created By	Anamika Mishra, 9/21/2025, 6:57 AM	Modified By	Anamika Mishra, 9/21/2025, 6:57 AM

Standard Fields

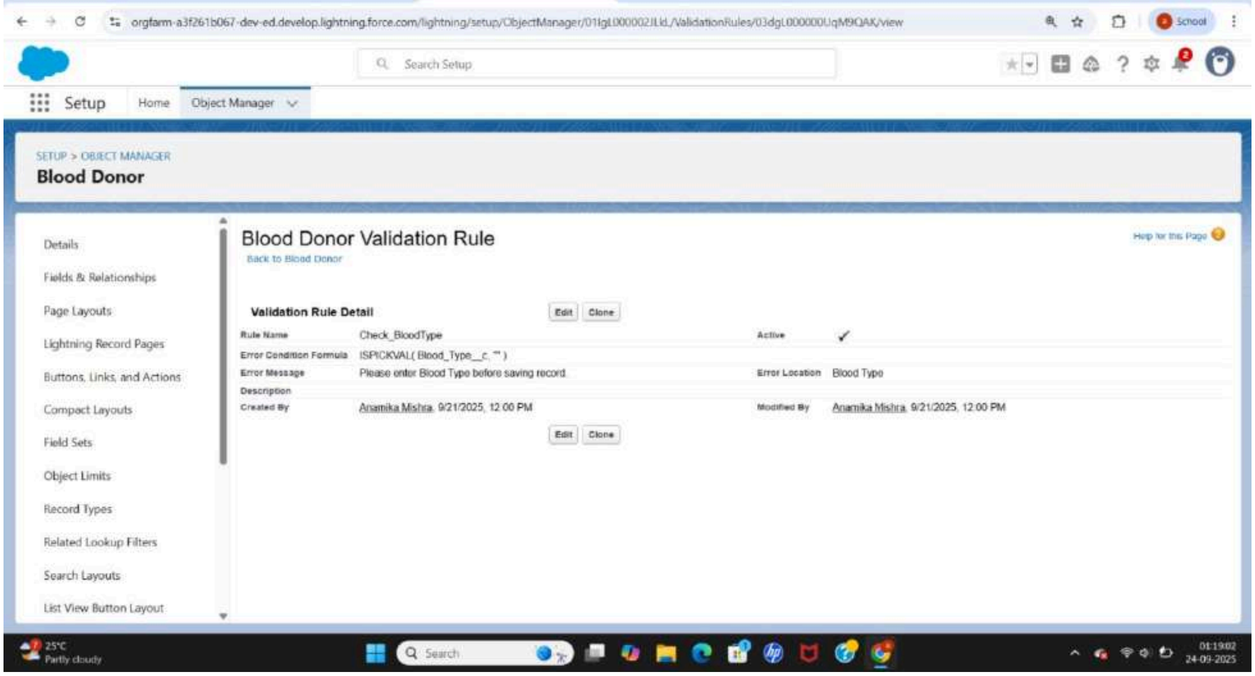
Action	Field Label	Field Name	Data Type
	Display URL	DisplayUrl	URL(1000)
	External ID	ExternalId	External Lookup

Custom Fields & Relationships

Phase 4: Process Automation (Admin)

Step 1. Validation Rules

- Rule Name: Check_BloodType
- Formula: ISPICKVAL(Blood_Type__c, "")
- Error Message: "Please enter Blood Type before saving."
- Error Location: Field → Blood Type

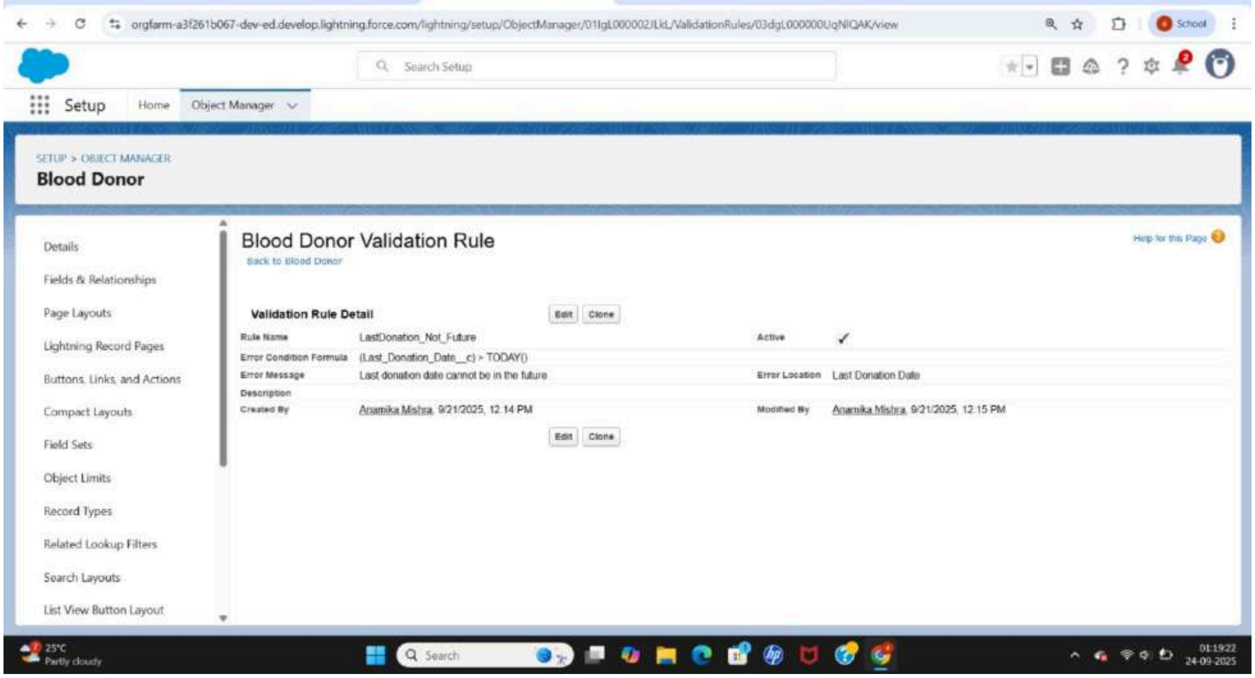


Blood Donor Validation Rule

Validation Rule Detail

Rule Name	Check_BloodType	Active	✓
Error Condition Formula	ISPICKVAL(Blood_Type__c, "")		
Error Message	Please enter Blood Type before saving record.	Error Location	Blood Type
Description			
Created By	Anamika Mishra, 9/21/2025, 12:00 PM	Modified By	Anamika Mishra, 9/21/2025, 12:00 PM

- Rule Name: LastDonation_Not_Future
- Formula: Last_Donation_Date__c > TODAY()
- Error Message: Last donation date cannot be in the future.
- Error Location: Field → Last Donation Date



Blood Donor Validation Rule

Validation Rule Detail

Rule Name	LastDonation_Not_Future	Active	✓
Error Condition Formula	(Last_Donation_Date__c) > TODAY()		
Error Message	Last donation date cannot be in the future	Error Location	Last Donation Date
Description			
Created By	Anamika Mishra, 9/21/2025, 12:14 PM	Modified By	Anamika Mishra, 9/21/2025, 12:15 PM

Step 2. Workflow Rules

Rule Name: Donor_Reminder_90days

Evaluation: Created, and edited to meet criteria

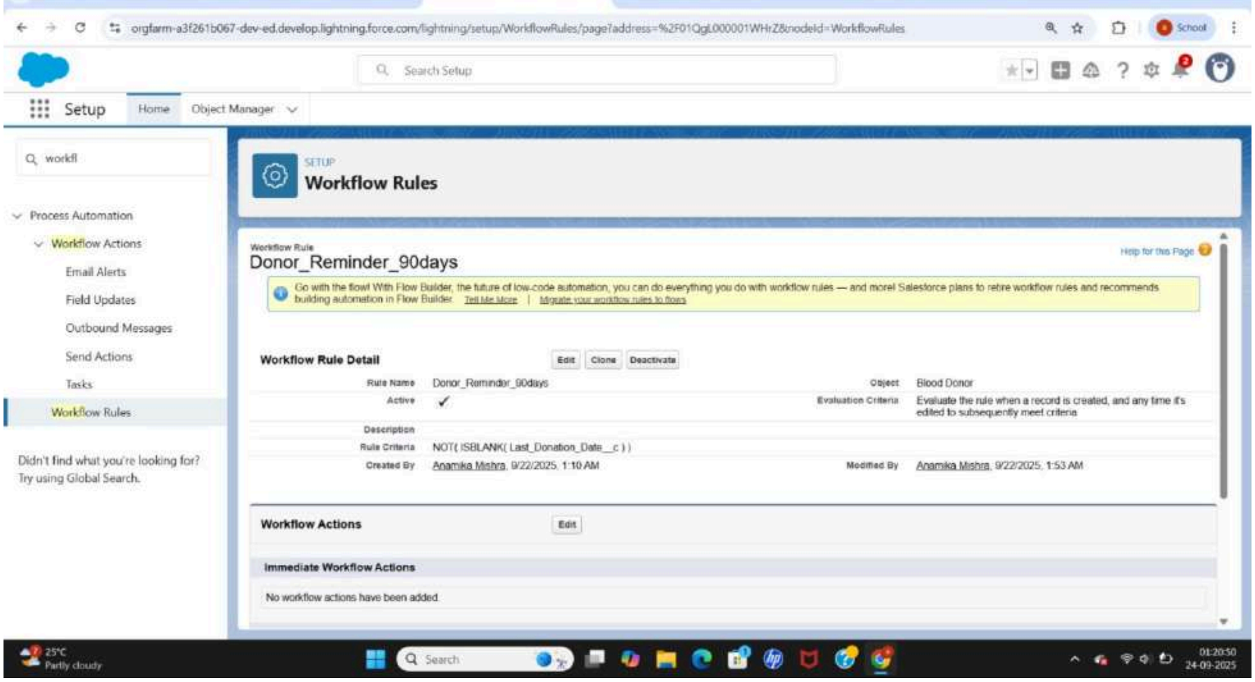
Criteria (Formula): NOT(ISBLANK(Last_Donation_Date__c))

Time Trigger: 90 days after Last_Donation_Date__c

Action: Email Alert → Template: Donor_Reminder_Template

Subject: Time for Your Next Blood Donation

Recipient: Donor Email (Email__c)



Workflow Rule

Donor_Reminder_90days

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder. Tell Me More | Migrate your workflow rules to flows

Workflow Rule Detail

Rule Name	Donor_Reminder_90days	Object	Blood Donor
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created, and any time it's edited to subsequently meet criteria
Description			
Rule Criteria	NOT(ISBLANK(Last_Donation_Date__c))		
Created By	Anamika Mishra, 9/22/2025, 1:10 AM	Modified By	Anamika Mishra, 9/22/2025, 1:53 AM

Workflow Actions

Immediate Workflow Actions

No workflow actions have been added.

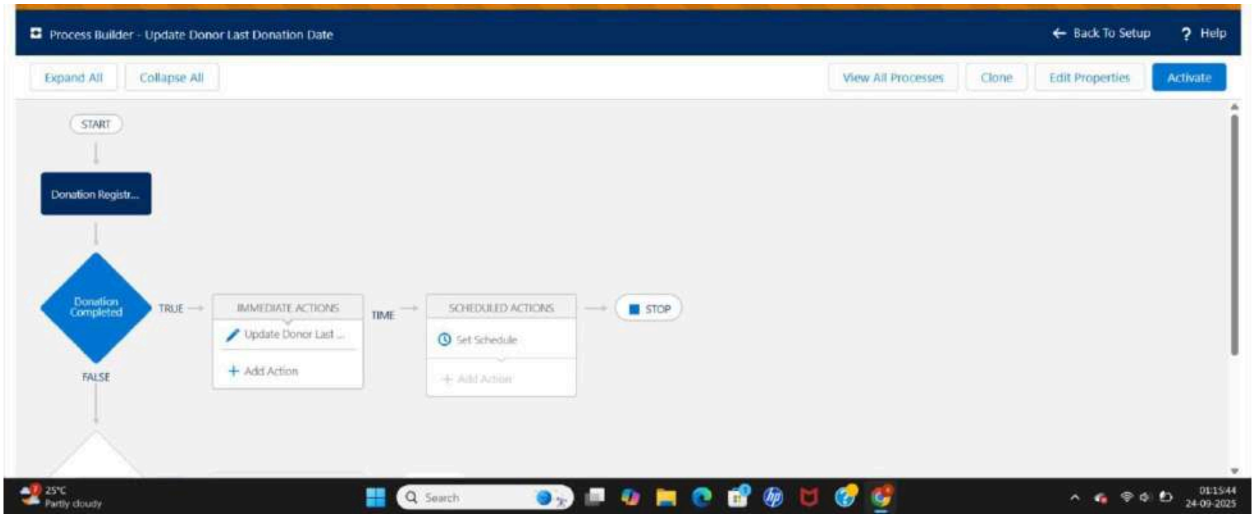
Step 3. Process Builder

1. Create Process
   Name: Update Donor Last Donation Date
   Starts: When a record changes

2. Select Object
   Object: Donation Registration
   Start when: Record is created or edited

3. Define Criteria
   Name: Donation Completed
   Condition: Donation Status = Completed
   Execute only when specified changes are made: Yes

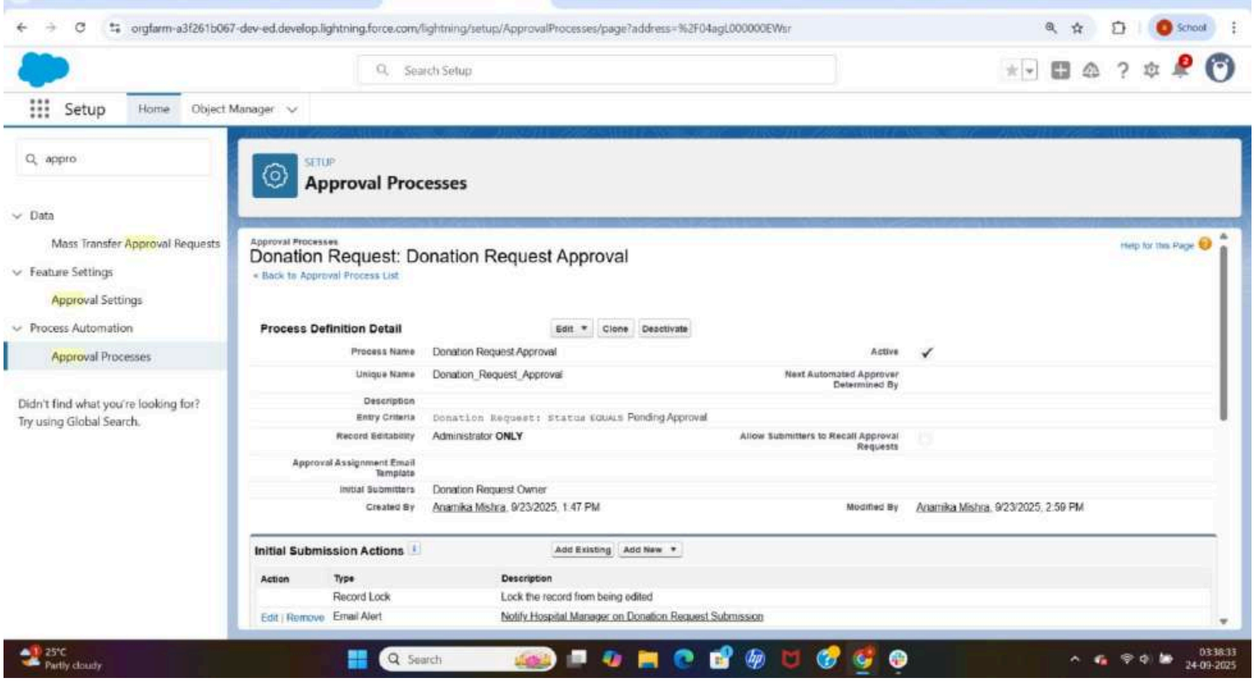
4. Define Action
   Action Type: Update Records
   Update: Related Donor record
   Field: Last Donation Date = Donation Registration.Donation Date



Step 4. Approval Process

Donation Request Approval Process

1. Setup: Approval Processes → Donation_Request__c → Jump Start Wizard.
2. Name & Entry: Donation Request Approval; records with Status__c = "Pending Approval".
3. Approver: Hospital Manager.
4. Actions:
On submit: Lock record
If approved: Status → Approved, notify owner
If rejected: Status → Rejected, notify requester
5. Activate: Process ready for use.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04agL000000EWsr
Search Setup
Setup
Home
Object Manager
appro
Data
Mass Transfer Approval Requests
Feature Settings
Approval Settings
Process Automation
Approval Processes
Didn't find what you're looking for? Try using Global Search.
SETUP
Approval Processes

Approval Processes

Donation Request: Donation Request Approval

« Back to Approval Process List

Process Definition Detail

Process Name	Donation Request Approval	Active	✓
Unique Name	Donation_Request_Approval	Next Automated Approver Determined By	
Description			
Entry Criteria	Donation Request: Status EQUALS Pending Approval		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	□
Approval Assignment Email Template			
Initial Submitters	Donation Request Owner		
Created By	Anamika Mishra, 9/23/2025, 1:47 PM	Modified By	Anamika Mishra, 9/23/2025, 2:59 PM

Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited
Edit | Remove	Email Alert	Notify Hospital Manager on Donation Request Submission

Step 5. Flow Builder (Screen, Record-Triggered, Scheduled, Auto-Launched)

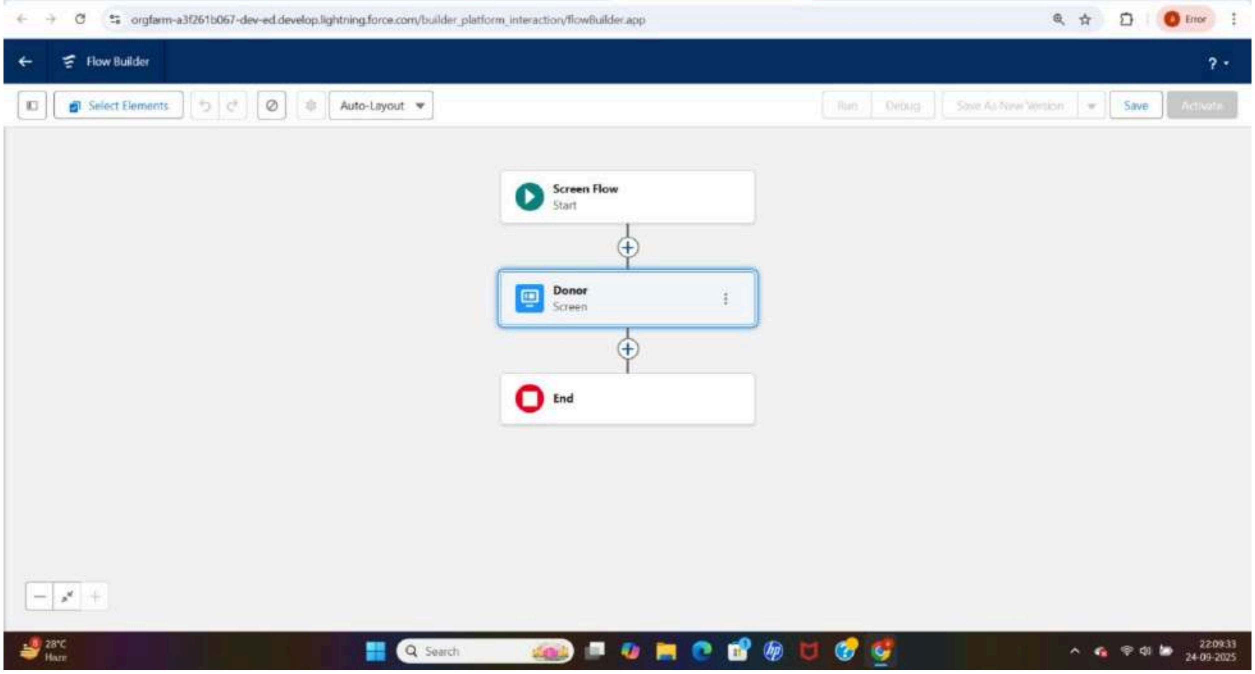
Use Cases in Blood Donor Management

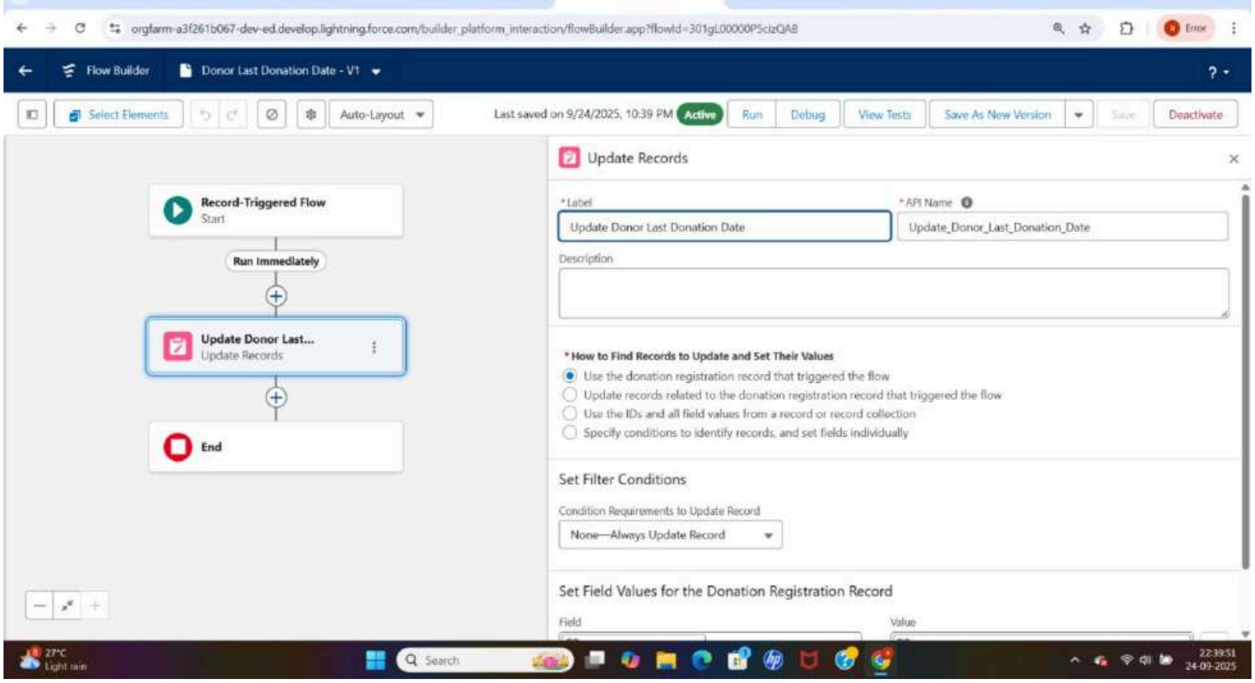
1. Screen Flow: Register a new donor (collect name, age, blood group, contact info).

2. Record-Triggered Flow: Update donor’s Last Donation Date whenever a donation record is added.

3. Scheduled Flow: Send reminder emails after 90 days of last donation.

4. Auto-Launched Flow: Assign donors to nearest hospital automatically.





Update Records

*Label: Update Donor Last Donation Date

*API Name: Update_Donor_Last_Donation_Date

Description

*How to Find Records to Update and Set Their Values

● Use the donation registration record that triggered the flow
○ Update records related to the donation registration record that triggered the flow
○ Use the IDs and all field values from a record or record collection
○ Specify conditions to identify records, and set fields individually

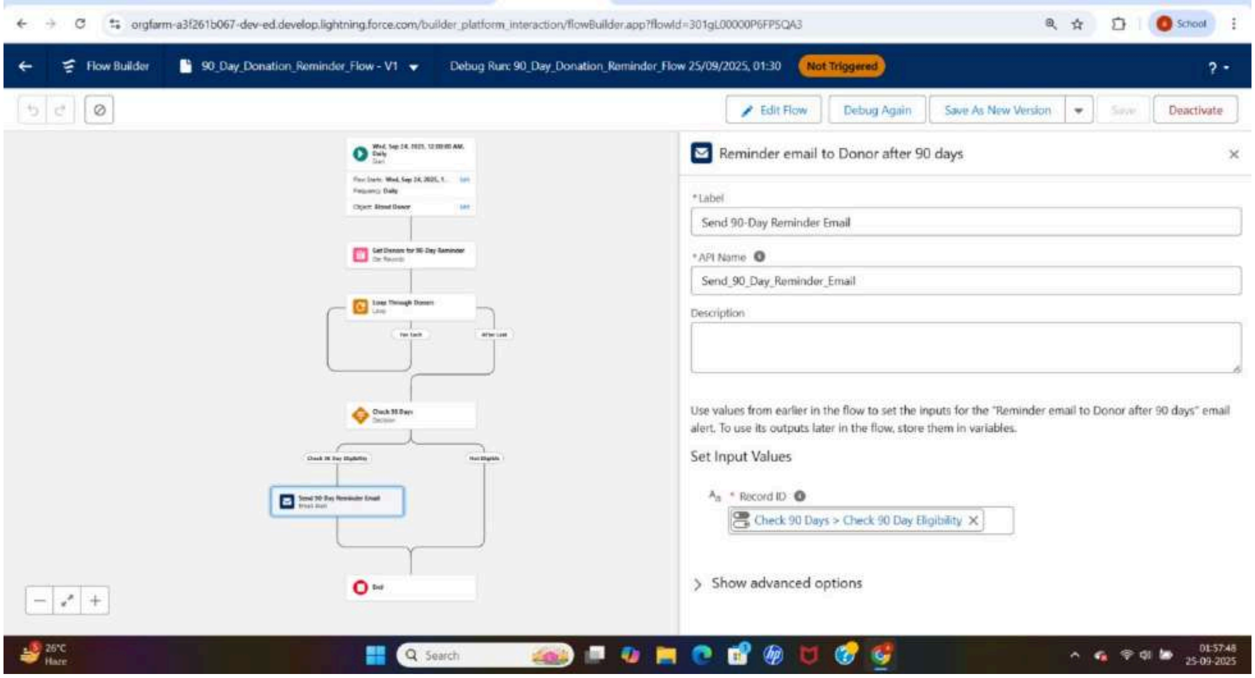
Set Filter Conditions

Condition Requirements to Update Record: None—Always Update Record

Set Field Values for the Donation Registration Record

Field

Value



orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gL00000P6FPSQA3
School
Flow Builder
90_Day_Donation_Reminder_Flow - V1
Debug Run: 90_Day_Donation_Reminder_Flow 25/09/2025, 01:30
Not Triggered
Edit Flow
Debug Again
Save As New Version
Save
Deactivate
Wed, Sep 24, 2025, 12:00:00 AM, Daily
Start
Flow Starts: Wed, Sep 24, 2025, 1...
Frequency: Daily
Object: Blood Donor
Edit
Get Donors for 90-Day Reminder
Get Records
Loop Through Donors
Loop
For Each
After Last
Check 90 Days
Decision
Check 90 Day Eligibility
Not Eligible
Send 90-Day Reminder Email
Email Alert
End
Search
26°C
Haze
01:57:48
25-09-2025

Reminder email to Donor after 90 days

*Label: Send 90-Day Reminder Email

*API Name: Send_90_Day_Reminder_Email

Description:

Use values from earlier in the flow to set the inputs for the "Reminder email to Donor after 90 days" email alert. To use its outputs later in the flow, store them in variables.

Set Input Values

* Record ID: Check 90 Days > Check 90 Day Eligibility

Show advanced options

Step 6. Email Alerts

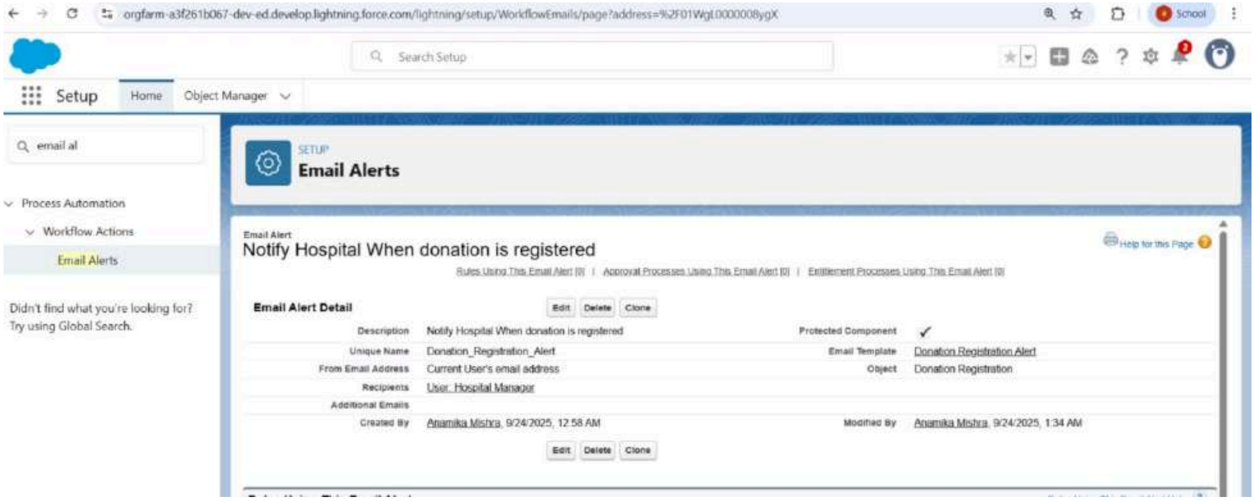
1. Setup → Email Alerts → New Email Alert

Description: Notify hospital when donation is registered

Unique Name: Donation_Registration_Alert

Object: Donation Registration

Choose recipients: User

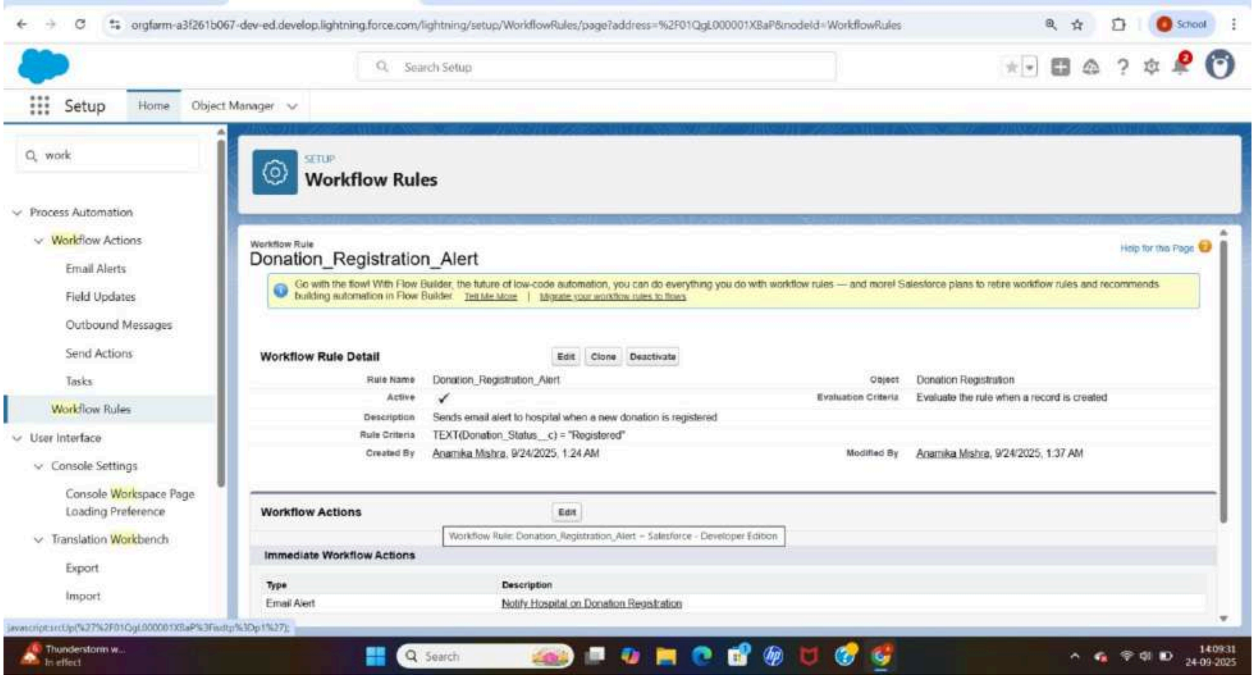

orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/WorkflowEmails/page?address=%2F01WgL000000ByqX
Search Setup
Setup
Home
Object Manager
email al
Process Automation
Workflow Actions
Email Alerts
Didn't find what you're looking for? Try using Global Search.
SETUP
Email Alerts
Help for this Page
Rules Using This Email Alert [0] | Approval Processes Using This Email Alert [0] | Entitlement Processes Using This Email Alert [0]
Edit
Delete
Clone

Email Alert

Notify Hospital When donation is registered

Email Alert Detail

Description	Notify Hospital When donation is registered	Protected Component	✓
Unique Name	Donation_Registration_Alert	Email Template	Donation Registration Alert
From Email Address	Current User's email address	Object	Donation Registration
Recipients	User: Hospital Manager		
Additional Emails			
Created By	Anamika Mishra, 9/24/2025, 12:58 AM	Modified By	Anamika Mishra, 9/24/2025, 1:34 AM


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/WorkflowRules/page?address=%2F01QgL000001XBaP&nodeId=WorkflowRules
Search Setup
Setup
Home
Object Manager
work
Process Automation
Workflow Actions
Email Alerts
Field Updates
Outbound Messages
Send Actions
Tasks
Workflow Rules
User Interface
Console Settings
Console Workspace Page Loading Preference
Translation Workbench
Export
Import
SETUP
Workflow Rules
Help for this Page
Tell Me More
Migrate your workflow rules to flows
Edit
Clone
Deactivate
Edit
Workflow Rule: Donation_Registration_Alert ~ Salesforce - Developer Edition
Search
Thunderstorm w... In effect
14:09:31
24-09-2025

Workflow Rule

Donation_Registration_Alert

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder.

Workflow Rule Detail

Rule Name	Donation_Registration_Alert	Object	Donation Registration
Active	✓	Evaluation Criteria	Evaluate the rule when a record is created
Description	Sends email alert to hospital when a new donation is registered		
Rule Criteria	TEXT(Donation_Status__c) = "Registered"		
Created By	Anamika Mishra, 9/24/2025, 1:24 AM	Modified By	Anamika Mishra, 9/24/2025, 1:37 AM

Workflow Actions

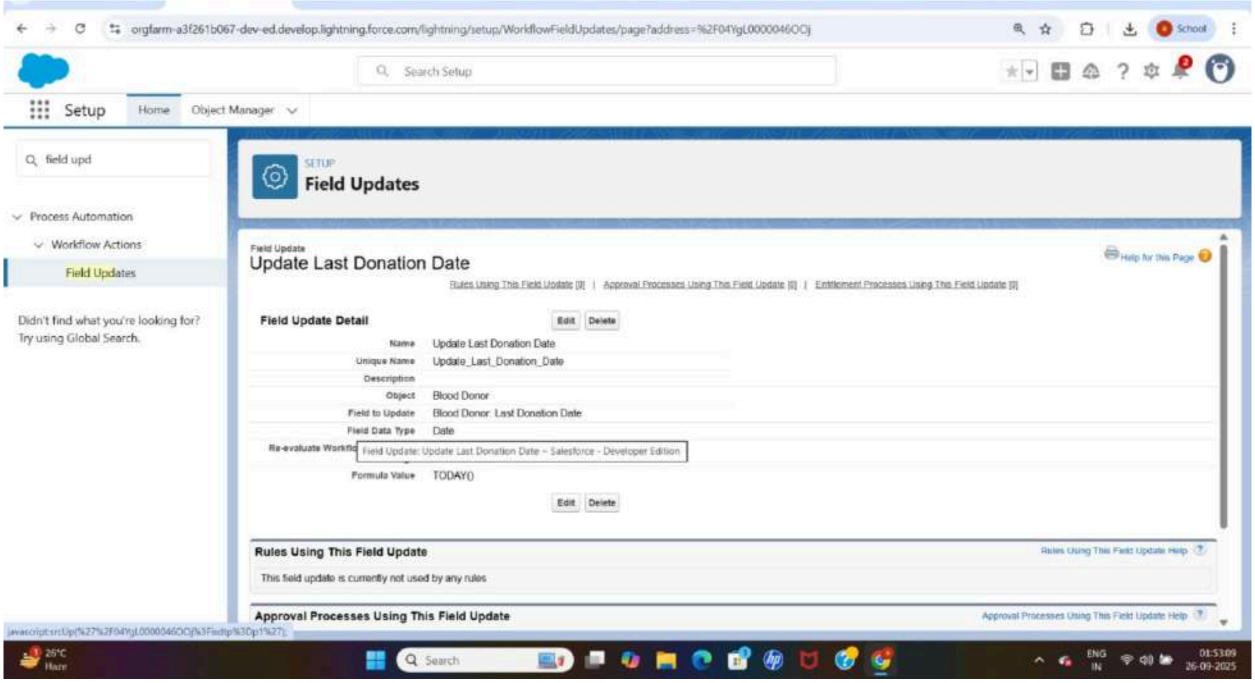
Immediate Workflow Actions

Type	Description
Email Alert	Notify Hospital on Donation Registration

Step 7. Field Updates

Name: Update_Last_Donation_Date

Select the field to update: Choose Last Donation Date on Blood Donor object.



SETUP

Field Updates

Field Update

Update Last Donation Date

Rules Using This Field Update [0] | Approval Processes Using This Field Update [0] | Entitlement Processes Using This Field Update [0]

Field Update Detail

Name	Update Last Donation Date
Unique Name	Update_Last_Donation_Date
Description	
Object	Blood Donor
Field to Update	Blood Donor: Last Donation Date
Field Data Type	Date
Re-evaluate Workflo	Field Update: Update Last Donation Date ~ Salesforce - Developer Edition
Formula Value	TODAY()

Rules Using This Field Update

This field update is currently not used by any rules

Approval Processes Using This Field Update

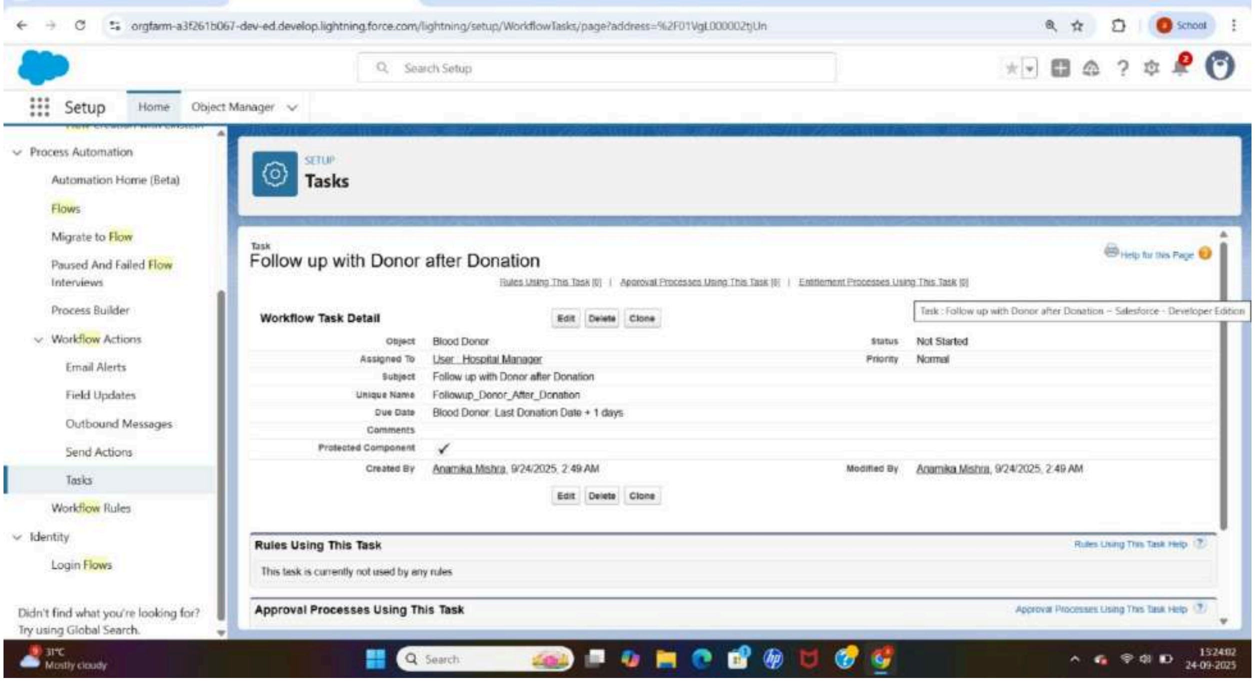
orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/WorkflowFieldUpdates/page?address=%2F04YgL0000046OOj
Search Setup
Setup
Home
Object Manager
field upd
Process Automation
Workflow Actions
Field Updates
Didn't find what you're looking for? Try using Global Search.
Help for this Page
Edit
Delete
Rules Using This Field Update Help
Approval Processes Using This Field Update Help
26°C Haze
Search
ENG IN
01:53:09
26-09-2025

Step 8. Tasks

Object: Blood Donor

Assigned To: Hospital Manager

Subject: Follow up with Donor after Donation


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/WorkflowTasks/page?address=%2F01VgL000002tjUn
Search Setup
Setup
Home
Object Manager
Process Automation
Automation Home (Beta)
Flows
Migrate to Flow
Paused And Failed Flow Interviews
Process Builder
Workflow Actions
Email Alerts
Field Updates
Outbound Messages
Send Actions
Tasks
Workflow Rules
Identity
Login Flows
Didn't find what you're looking for? Try using Global Search.
SETUP
Tasks
Help for this Page
Task : Follow up with Donor after Donation ~ Salesforce - Developer Edition

Task
Follow up with Donor after Donation

Rules Using This Task [0] | Approval Processes Using This Task [0] | Entitlement Processes Using This Task [0]

Workflow Task Detail Edit Delete Clone

Object	Blood Donor	Status	Not Started
Assigned To	User: Hospital Manager	Priority	Normal
Subject	Follow up with Donor after Donation		
Unique Name	Followup_Donor_After_Donation		
Due Date	Blood Donor: Last Donation Date + 1 days		
Comments			
Protected Component	✓		
Created By	Anamika Mishra, 9/24/2025, 2:49 AM	Modified By	Anamika Mishra, 9/24/2025, 2:49 AM

Edit Delete Clone

Rules Using This Task Rules Using This Task Help

This task is currently not used by any rules

Approval Processes Using This Task Approval Processes Using This Task Help

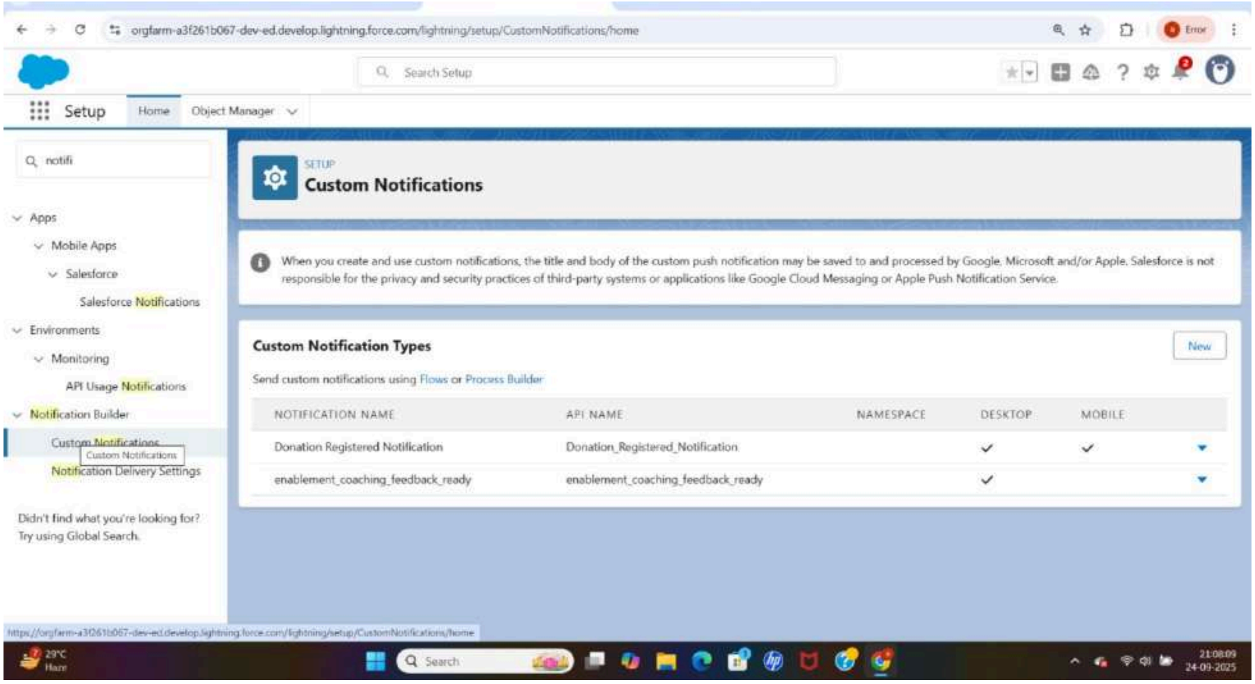
Step 9. Custom Notifications

Name: Donation Registered Notification

Object: Donation Registration

Supported Channels -

- Desktop (for Salesforce users)
- Mobile (for Salesforce Mobile App)



Custom Notifications

When you create and use custom notifications, the title and body of the custom push notification may be saved to and processed by Google, Microsoft and/or Apple. Salesforce is not responsible for the privacy and security practices of third-party systems or applications like Google Cloud Messaging or Apple Push Notification Service.

Custom Notification Types

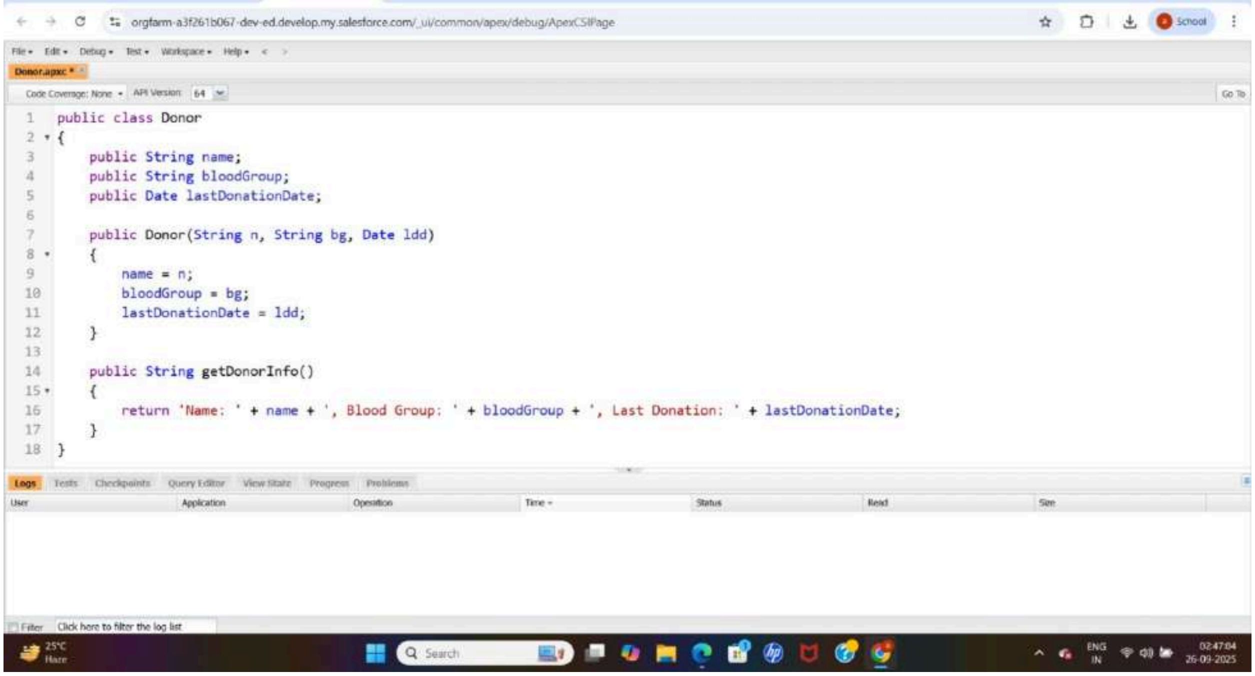
Send custom notifications using Flows or Process Builder

NOTIFICATION NAME	API NAME	NAMESPACE	DESKTOP	MOBILE
Donation Registered Notification	Donation_Registered_Notification		✓	✓
enablement_coaching_feedback_ready	enablement_coaching_feedback_ready		✓	

Phase 5: Apex Programming (Developer)

Step 1. Classes & Objects

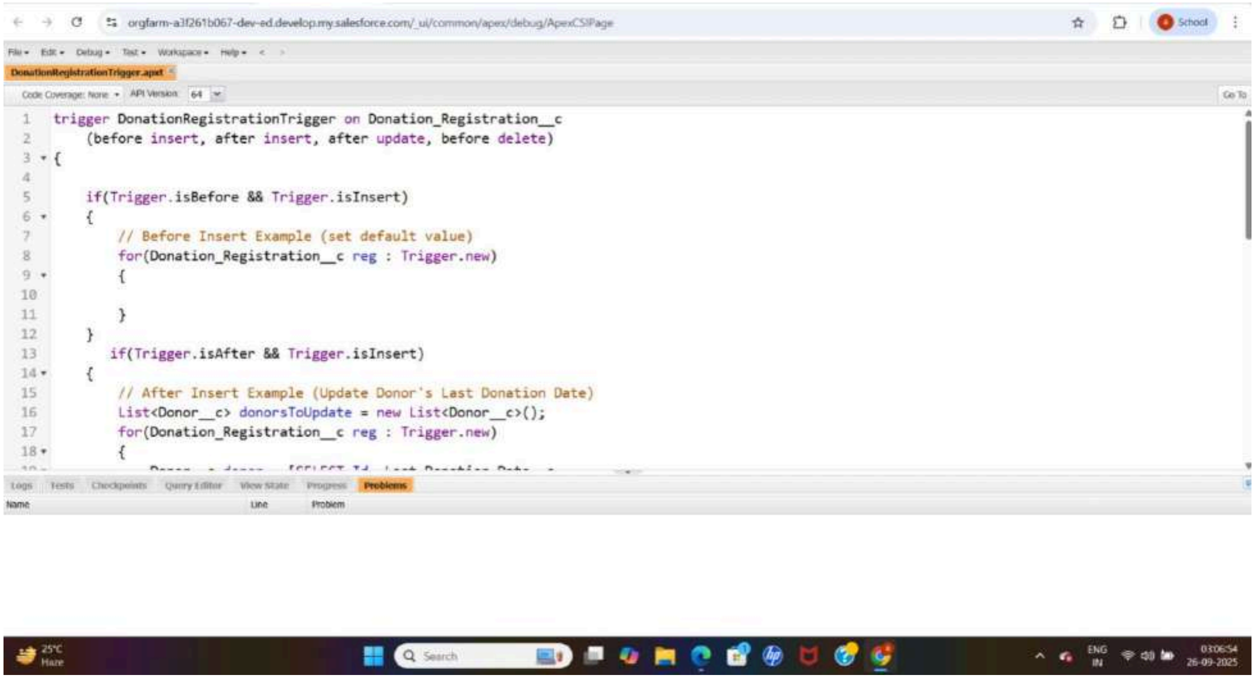
In Apex, a class is a blueprint that defines variables (data) and methods (functions).
An object is an instance of a class, which represents a real-world entity.
In your Blood Donor Management Project, you can use classes to represent Donors, Hospitals, Donations, and Events.
This makes the system more organized, reusable, and scalable.



```
public class Donor
{
    public String name;
    public String bloodGroup;
    public Date lastDonationDate;

    public Donor(String n, String bg, Date ldd)
    {
        name = n;
        bloodGroup = bg;
        lastDonationDate = ldd;
    }

    public String getDonorInfo()
    {
        return 'Name: ' + name + ', Blood Group: ' + bloodGroup + ', Last Donation: ' + lastDonationDate;
    }
}
```

```
public class Donor
{
   public String name;
   public String bloodGroup;
   public Date lastDonationDate;

   public Donor(String n, String bg, Date ldd)
   {
      name = n;
      bloodGroup = bg;
      lastDonationDate = ldd;
   }

   public String getDonorInfo()
   {
      return 'Name: ' + name + ', Blood Group: ' + bloodGroup + ', Last Donation: ' + lastDonationDate;
   }
}
```

Step 2. Apex Triggers (before/after insert/update/delete)



```
trigger DonationRegistrationTrigger on Donation_Registration__c
    (before insert, after insert, after update, before delete)
{

    if(Trigger.isBefore && Trigger.isInsert)
    {
        // Before Insert Example (set default value)
        for(Donation_Registration__c reg : Trigger.new)
        {

        }
    }
      if(Trigger.isAfter && Trigger.isInsert)
    {
        // After Insert Example (Update Donor's Last Donation Date)
        List<Donor__c> donorsToUpdate = new List<Donor__c>();
        for(Donation_Registration__c reg : Trigger.new)
        {
            Donor__c donor = [SELECT Id, Last_Donation_Date__c
```

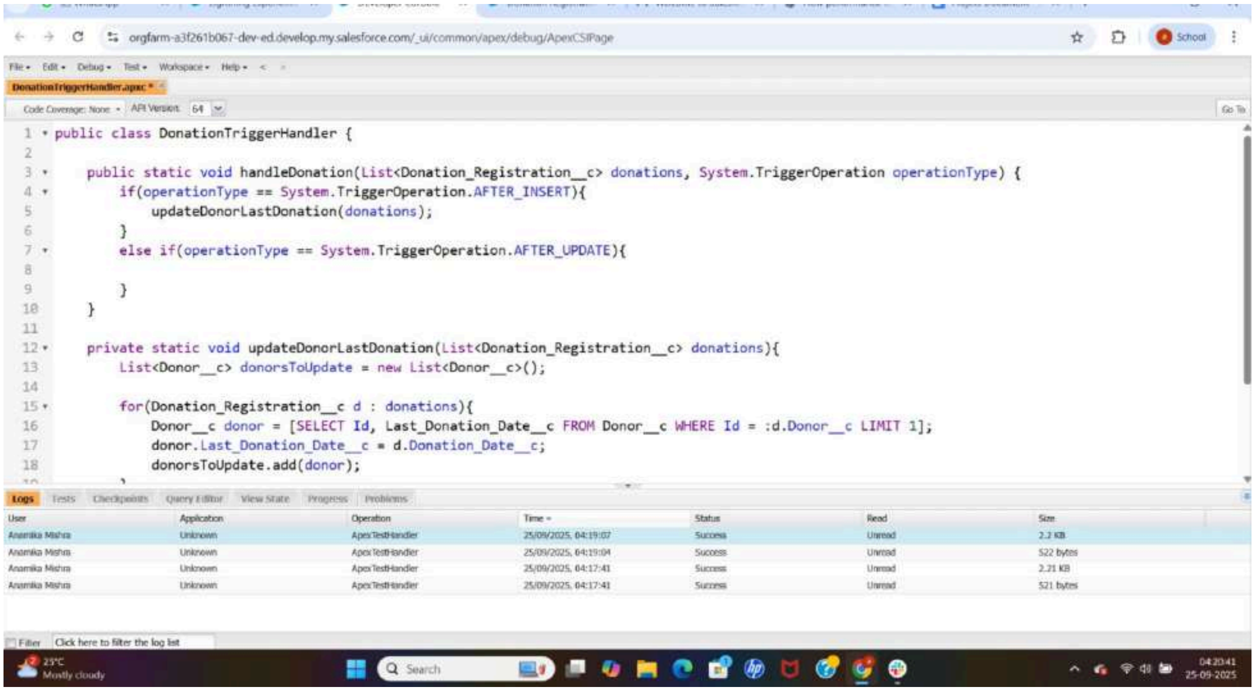
```
trigger DonationRegistrationTrigger on Donation_Registration__c
   (before insert, after insert, after update, before delete)
{
  if(Trigger.isBefore && Trigger.isInsert)
   {
      // Before Insert Example (set default value)
      for(Donation_Registration__c reg : Trigger.new)
      {
      }
   }
     if(Trigger.isAfter && Trigger.isInsert)
   {
      // After Insert Example (Update Donor's Last Donation Date)
      List<Donor__c> donorsToUpdate = new List<Donor__c>();
      for(Donation_Registration__c reg : Trigger.new)
      {
         Donor__c donor = [SELECT Id, Last_Donation_Date__c
                      FROM Donor__c
                      WHERE Id = :reg.Donor__c LIMIT 1];
         donor.Last_Donation_Date__c = reg.Donation_Date__c;
         donorsToUpdate.add(donor);
      }
```

```
    if(!donorsToUpdate.isEmpty())
    {
      update donorsToUpdate;
    }
  }

  if(Trigger.isAfter && Trigger.isUpdate)
  {
    // After Update Example
    for(Donation_Registration__c reg : Trigger.new)
    {
      System.debug('Donation record updated: ' + reg.Id);
    }
  }

  if(Trigger.isBefore && Trigger.isDelete)
  {
    // Before Delete Example
    List<Donor__c> donorsToUpdate = new List<Donor__c>();
    for(Donation_Registration__c reg : Trigger.old)
    {

    }
    if(!donorsToUpdate.isEmpty())
    {
      update donorsToUpdate;
    }
  }
}
```

Step 3. Trigger Design Pattern

```
public class DonationTriggerHandler {
 public static void handleDonation(List<Donation_Registration__c> donations,
System.TriggerOperation operationType) {
    if(operationType == System.TriggerOperation.AFTER_INSERT){
      updateDonorLastDonation(donations)}
```

```
   else if(operationType == System.TriggerOperation.AFTER_UPDATE){
        }} private static void
updateDonorLastDonation(List<Donation_Registration__c> donations){
      List<Donor__c> donorsToUpdate = new List<Donor__c>();
      for(Donation_Registration__c d : donations){
         Donor__c donor = [SELECT Id, Last_Donation_Date__c FROM Donor__c
WHERE Id = :d.Donor__c LIMIT 1];
         donor.Last_Donation_Date__c = d.Donation_Date__c;
         donorsToUpdate.add(donor); }
       if(!donorsToUpdate.isEmpty()){
         update donorsToUpdate;
      } }}
```


orgfarm-a3f261b067-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
School
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
DonationTriggerHandler.apxc *
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Filter Click here to filter the log list
23°C Mostly cloudy
Search
04:20:41 25-09-2025

```
public class DonationTriggerHandler {

    public static void handleDonation(List<Donation_Registration__c> donations, System.TriggerOperation operationType) {
        if(operationType == System.TriggerOperation.AFTER_INSERT){
            updateDonorLastDonation(donations);
        }
        else if(operationType == System.TriggerOperation.AFTER_UPDATE){

        }
    }

    private static void updateDonorLastDonation(List<Donation_Registration__c> donations){
        List<Donor__c> donorsToUpdate = new List<Donor__c>();

        for(Donation_Registration__c d : donations){
            Donor__c donor = [SELECT Id, Last_Donation_Date__c FROM Donor__c WHERE Id = :d.Donor__c LIMIT 1];
            donor.Last_Donation_Date__c = d.Donation_Date__c;
            donorsToUpdate.add(donor);
```

User	Application	Operation	Time ▾	Status	Read	Size
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 04:19:07	Success	Unread	2.2 KB
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 04:19:04	Success	Unread	522 bytes
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 04:17:41	Success	Unread	2.21 KB
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 04:17:41	Success	Unread	521 bytes

Step 4. SOQL & SOSL

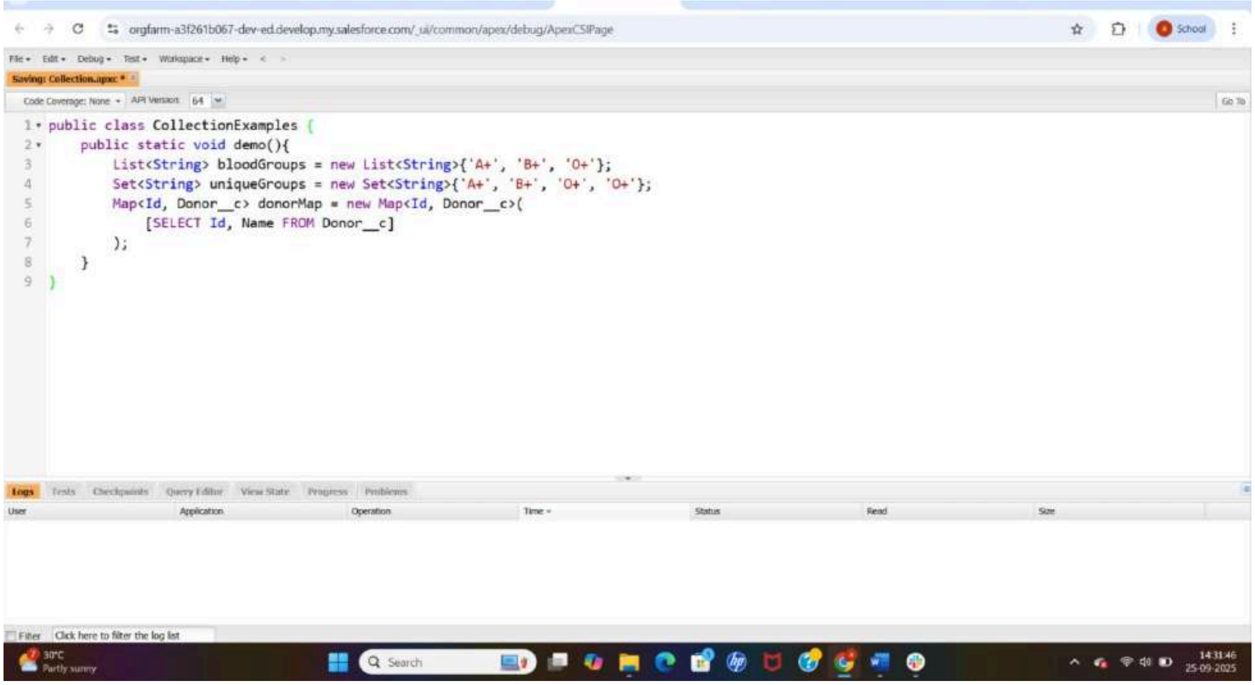
SOQL: Salesforce Object Query Language → fetch records from Salesforce objects.

```
List<Donor__c> oPositiveDonors = [SELECT Name, Phone__c FROM Donor__c
WHERE Blood_Group__c = 'O+'];
```

SOSL: Salesforce Object Search Language → search text across multiple objects.
List<List<SObject>> searchResults = [FIND 'Anamika*' IN ALL FIELDS RETURNING Donor__c(Name, Phone__c)];

Step 5. Collections: List,Set,Map

List: Ordered collection.
Set: Unique elements.
Map: Key-value pairs.

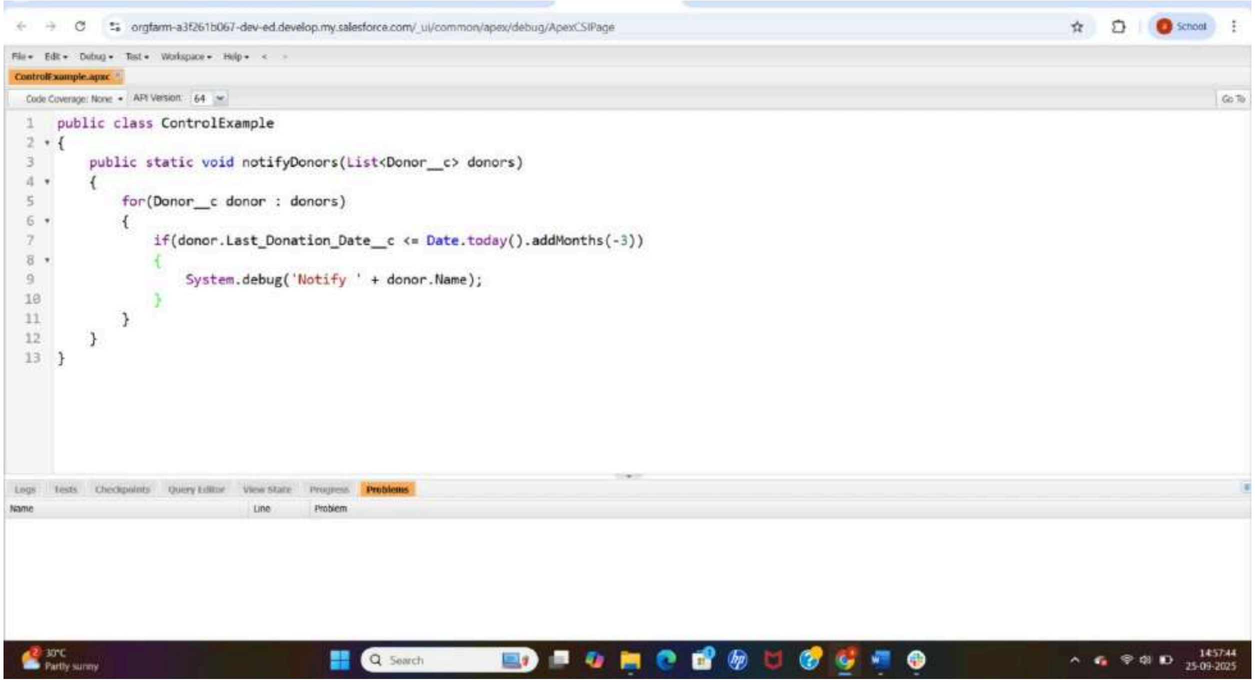


```
public class CollectionExamples {
    public static void demo(){
        List<String> bloodGroups = new List<String>{'A+', 'B+', 'O+'};
        Set<String> uniqueGroups = new Set<String>{'A+', 'B+', 'O+', 'O+'};
        Map<Id, Donor__c> donorMap = new Map<Id, Donor__c>(
            [SELECT Id, Name FROM Donor__c]
        );
    }
}
```

```
public class CollectionExamples {
   public static void demo(){
      List<String> bloodGroups = new List<String>{'A+', 'B+', 'O+'};
      Set<String> uniqueGroups = new Set<String>{'A+', 'B+', 'O+', 'O+'};
      Map<Id, Donor__c> donorMap = new Map<Id, Donor__c>(
         [SELECT Id, Name FROM Donor__c]
      );
   }
}
```

Step 6. Control Statements

Send SMS only if donor hasn't donated in 3 months.


orgfarm-a3f261b067-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
ControlExample.apxc
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name
Line
Problem

```
public class ControlExample
{
    public static void notifyDonors(List<Donor__c> donors)
    {
        for(Donor__c donor : donors)
        {
            if(donor.Last_Donation_Date__c <= Date.today().addMonths(-3))
            {
                System.debug('Notify ' + donor.Name);
            }
        }
    }
}
```

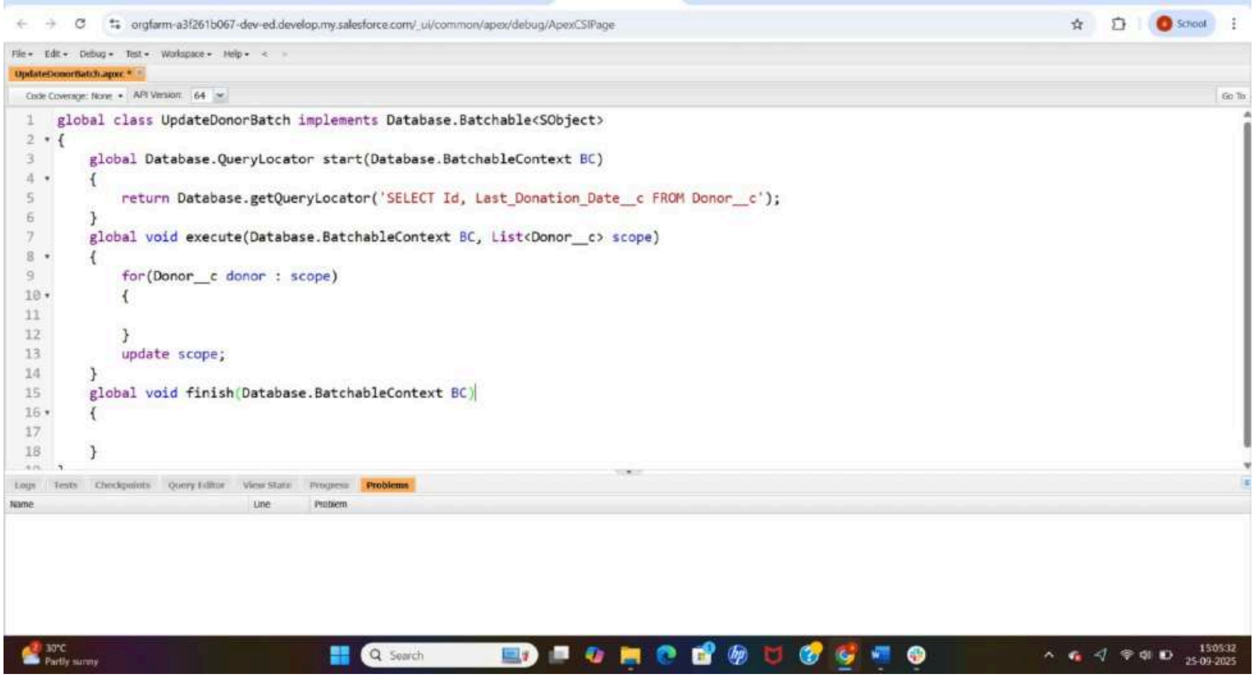
```
public class ControlExample {
   public static void notifyDonors(List<Donor__c> donors){
      for(Donor__c donor : donors){
         if(donor.Last_Donation_Date__c <= Date.today().addMonths(-3)){
            System.debug('Notify ' + donor.Name);
         }
      }
   }
}
```

Step 7. Batch Apex

Update all donors' last donation date based on donation records

```
global class UpdateDonorBatch implements Database.Batchable<SObject>{
   global Database.QueryLocator start(Database.BatchableContext BC){
```

```
return Database.getQueryLocator('SELECT Id, Last_Donation_Date__c FROM
Donor__c');  }
   global void execute(Database.BatchableContext BC, List<Donor__c> scope){
      for(Donor__c donor : scope){ }
      update scope;}
   global void finish(Database.BatchableContext BC){}}
```



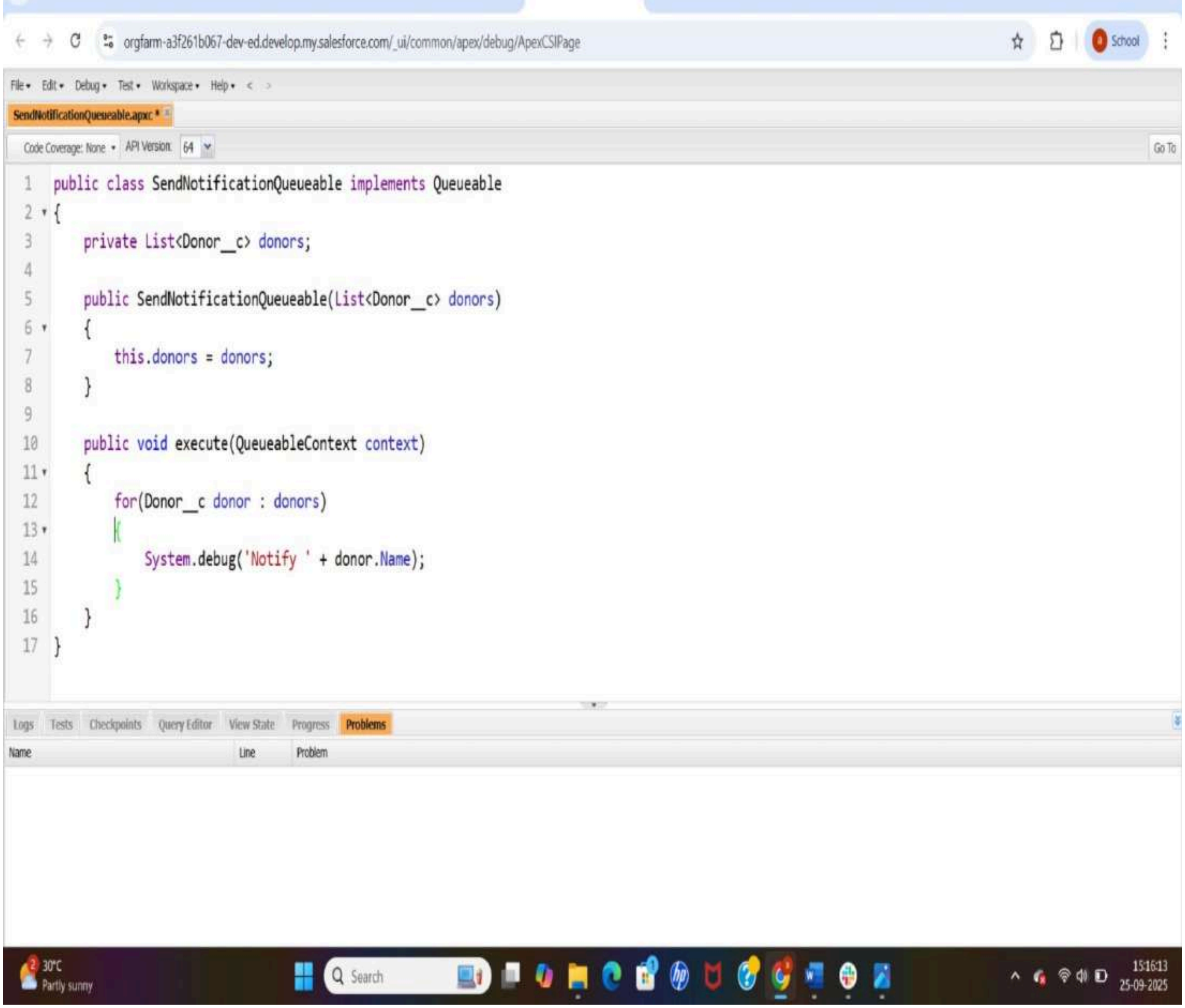
orgfarm-a3f261b067-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
School
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾
UpdateDonorBatch.apxc *
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
30°C Partly sunny
Search
15:05:32 25-09-2025

```
global class UpdateDonorBatch implements Database.Batchable<SObject>
{
    global Database.QueryLocator start(Database.BatchableContext BC)
    {
        return Database.getQueryLocator('SELECT Id, Last_Donation_Date__c FROM Donor__c');
    }
    global void execute(Database.BatchableContext BC, List<Donor__c> scope)
    {
        for(Donor__c donor : scope)
        {

        }
        update scope;
    }
    global void finish(Database.BatchableContext BC)
    {

    }
```

Step 8. Queueable Apex

Send notifications to donors after donation recorded.

```
public class SendNotificationQueueable implements Queueable {
   private List<Donor__c> donors;
   public SendNotificationQueueable(List<Donor__c> donors){
      this.donors = donors;  }
   public void execute(QueueableContext context){
      for(Donor__c donor : donors){
         System.debug('Notify ' + donor.Name);  }}}
```


orgfarm-a3f261b067-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
School
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
SendNotificationQueueable.apxc *
Code Coverage: None ▾ API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
30°C Partly sunny
Search
15:16:13
25-09-2025

```
public class SendNotificationQueueable implements Queueable
{
    private List<Donor__c> donors;

    public SendNotificationQueueable(List<Donor__c> donors)
    {
        this.donors = donors;
    }

    public void execute(QueueableContext context)
    {
        for(Donor__c donor : donors)
        {
            System.debug('Notify ' + donor.Name);
        }
    }
}
```

Step 9. Scheduled Apex

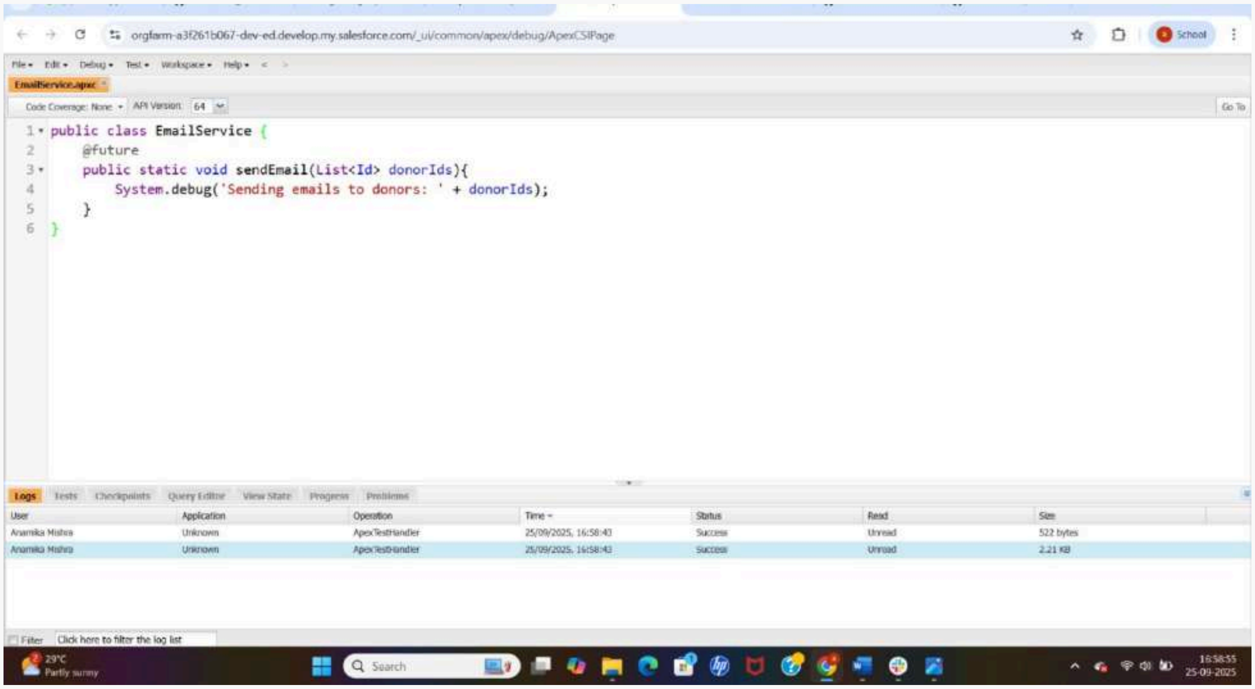
Notify Donors weekly
Class Name: weeklyNotification

```
global class WeeklyNotification implements Schedulable{
   global void execute(SchedulableContext sc)   {
      List<Donor__c> donors = [SELECT Id, Name FROM Donor__c
```

WHERE Last_Donation_Date__c <= :Date.today().addMonths(-3)]; }}

Step 10. Future Methods

Class Name: EmailService



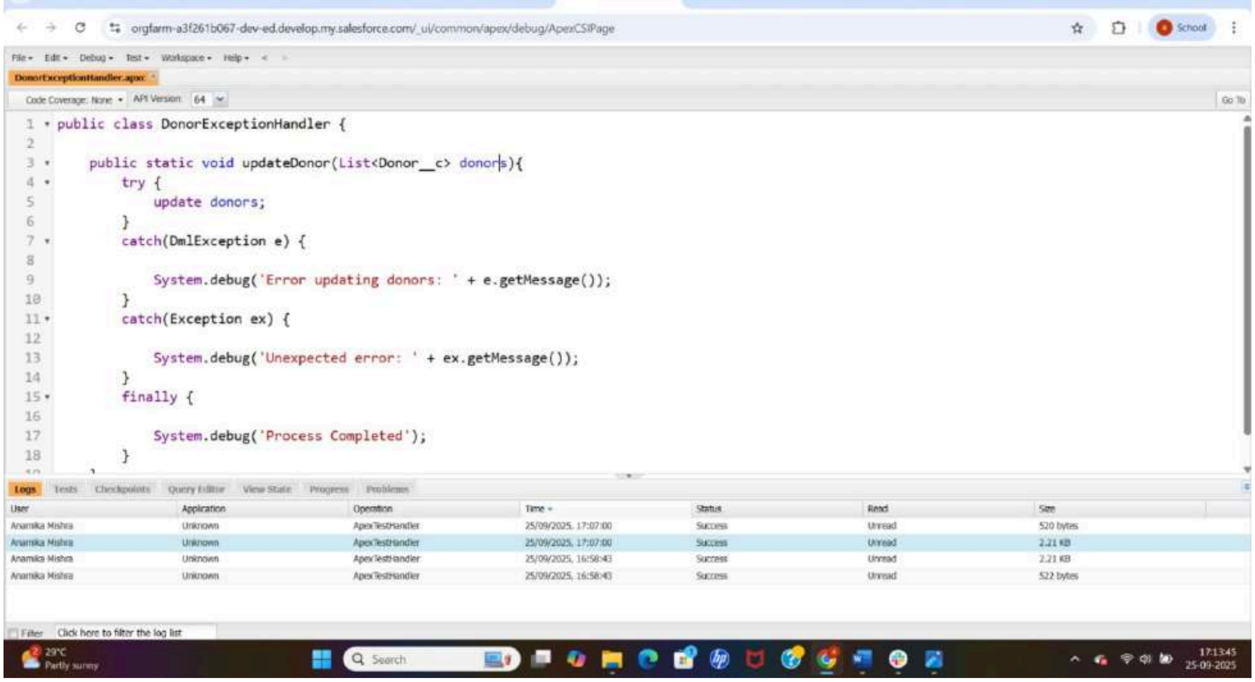
```
public class EmailService {
    @future
    public static void sendEmail(List<Id> donorIds){
        System.debug('Sending emails to donors: ' + donorIds);
    }
}
```

User	Application	Operation	Time	Status	Read	Size
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 16:58:43	Success	Unread	522 bytes
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 16:58:43	Success	Unread	2.21 KB

```
public class EmailService {
   @future
   public static void sendEmail(List<Id> donorIds){
      System.debug('Sending emails to donors: ' + donorIds);
   }
}
```

Step 11. Exception Handling

Class Name : DonorExceptionHandler



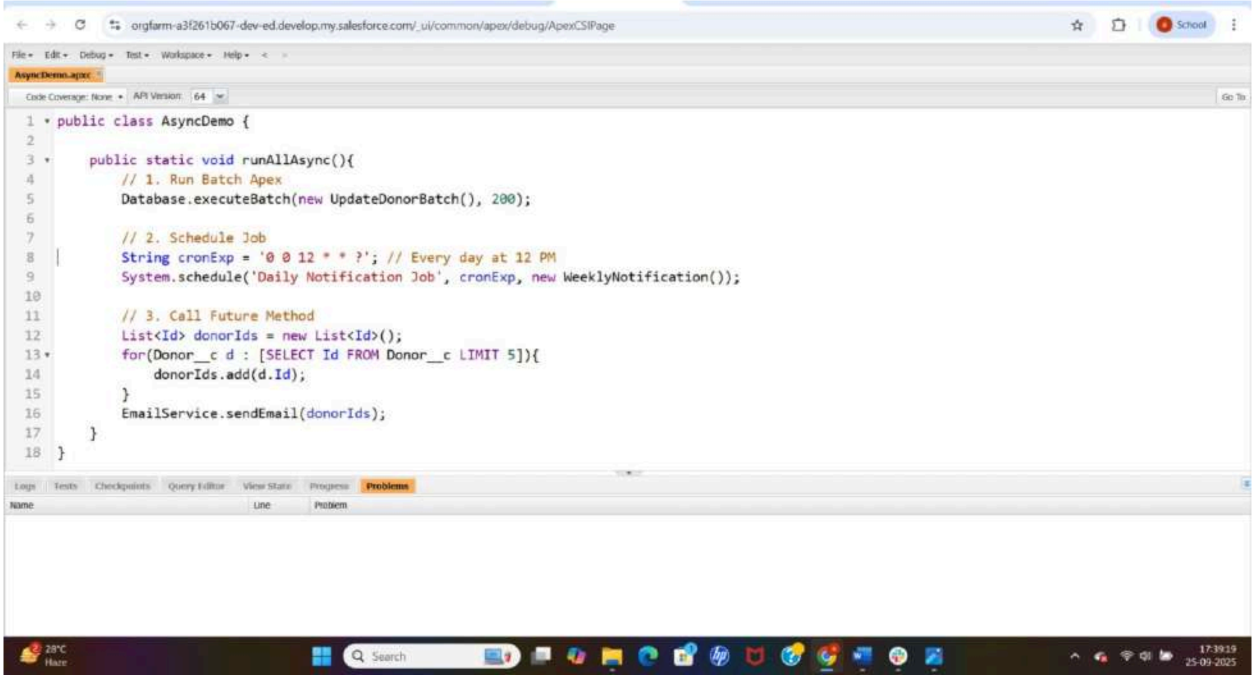
```
public class DonorExceptionHandler {

    public static void updateDonor(List<Donor__c> donors){
        try {
            update donors;
        }
        catch(DmlException e) {

            System.debug('Error updating donors: ' + e.getMessage());
        }
        catch(Exception ex) {

            System.debug('Unexpected error: ' + ex.getMessage());
        }
        finally {

            System.debug('Process Completed');
        }
```

User	Application	Operation	Time	Status	Read	Size
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 17:07:00	Success	Unread	520 bytes
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 17:07:00	Success	Unread	2.21 KB
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 16:58:43	Success	Unread	2.21 KB
Anamika Mishra	Unknown	ApexTestHandler	25/09/2025, 16:58:43	Success	Unread	522 bytes

```
public class DonorExceptionHandler {

    public static void updateDonor(List<Donor__c> donors){
        try {
            update donors;
        }
        catch(DmlException e) {

            System.debug('Error updating donors: ' + e.getMessage());
        }
        catch(Exception ex) {

            System.debug('Unexpected error: ' + ex.getMessage());
        }
        finally {

            System.debug('Process Completed');
        }
    }
}
```

Step 12. Asynchronous Processing

Class Name - AsyncDemo


orgfarm-a3f261b067-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
School
File Edit Debug Test Workspace Help
AsyncDemo.apxc
Code Coverage: None API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
28°C Haze
Search
17:39:19 25-09-2025

```
public class AsyncDemo {

    public static void runAllAsync(){
        // 1. Run Batch Apex
        Database.executeBatch(new UpdateDonorBatch(), 200);

        // 2. Schedule Job
        String cronExp = '0 0 12 * * ?'; // Every day at 12 PM
        System.schedule('Daily Notification Job', cronExp, new WeeklyNotification());

        // 3. Call Future Method
        List<Id> donorIds = new List<Id>();
        for(Donor__c d : [SELECT Id FROM Donor__c LIMIT 5]){
            donorIds.add(d.Id);
        }
        EmailService.sendEmail(donorIds);
    }
}
```

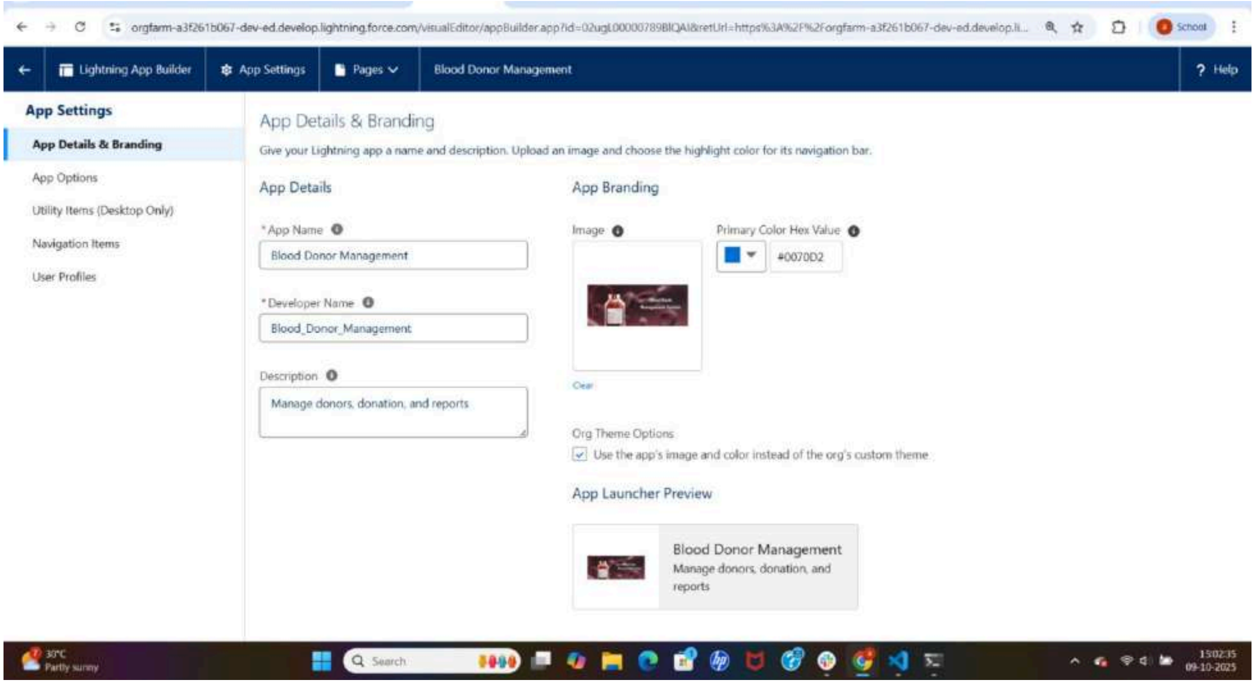
```
public class AsyncDemo {

  public static void runAllAsync(){
    // 1. Run Batch Apex
    Database.executeBatch(new UpdateDonorBatch(), 200);

    // 2. Schedule Job
    String cronExp = '0 0 12 * * ?'; // Every day at 12 PM
    System.schedule('Daily Notification Job', cronExp, new WeeklyNotification());

    // 3. Call Future Method
    List<Id> donorIds = new List<Id>();
    for(Donor__c d : [SELECT Id FROM Donor__c LIMIT 5]){
      donorIds.add(d.Id);
    }
    EmailService.sendEmail(donorIds);
  }
}
```

Phase 6: User Interface Development

Step 1. Lighting App Builder

Lightning App Builder is a point-and-click tool in Salesforce that allows administrators to design and customize user interfaces without code.
It provides drag-and-drop components to create pages for apps, records, and home screens.
For the Blood Donor Management project, it helps to design screens for **Donor, Donation, and Blood Camp

Purpose - The purpose of Lightning App Builder in this project is to design donor and donation pages, dashboards, and workflows that make blood donor management faster, simpler, and more effective for all users.



orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02ugL00000789BIQAI&retUrl=https%3A%2F%2Forgfarm-a3f261b067-dev-ed.develop.li...

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Blood Donor Management

*Developer Name: Blood_Donor_Management

Description: Manage donors, donation, and reports

App Branding

Image

Primary Color Hex Value: #0070D2

Org Theme Options

☑ Use the app's image and color instead of the org's custom theme

App Launcher Preview

Blood Donor Management
Manage donors, donation, and reports

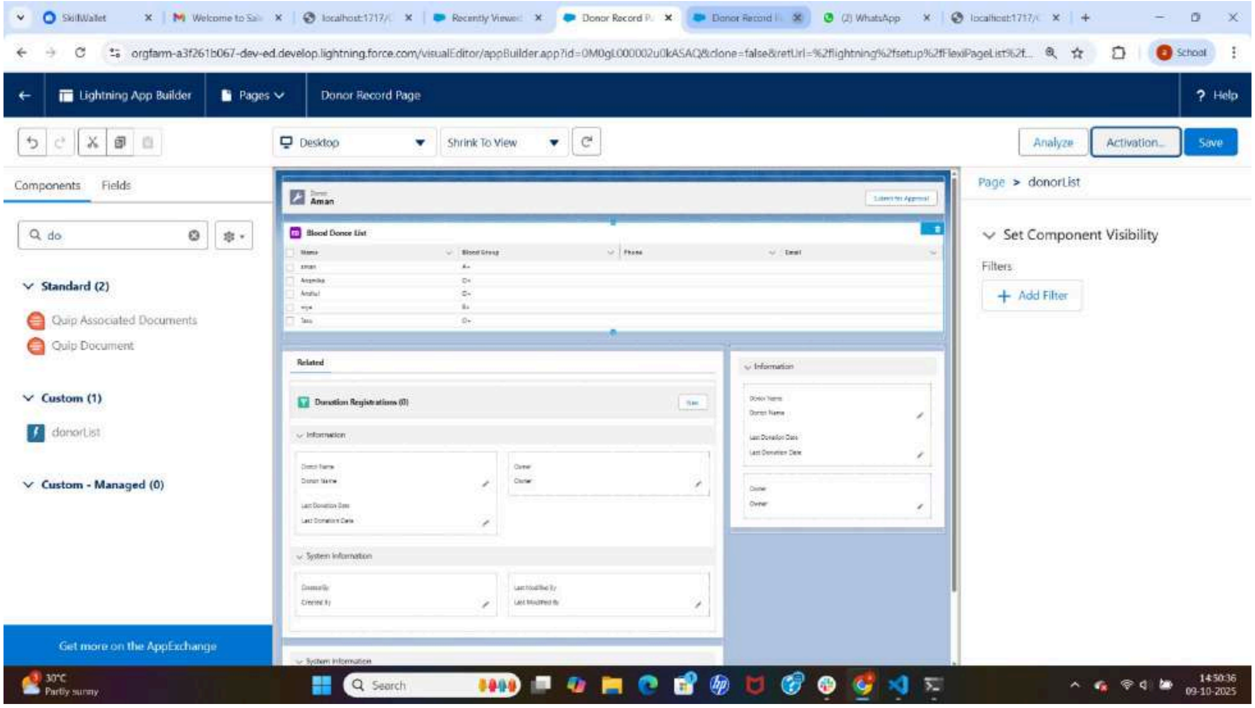
Step 2. Record Pages

A Record Page in Salesforce Lightning App Builder is a customized layout that controls how a single record (like a Donor or a Donation) is displayed.

It allows administrators to add components such as record details, related lists, charts, and flows on one page.

Purpose of Record Pages

To give users a consolidated view of all important information related to a record.
To make data entry and updates faster and easier.
To integrate flows, charts, and quick actions directly into the record screen.



Name	Blood Group	Phone	Email
aman	A+		
Anamika	O+		
Anshul	O+		
riya	B+		
Tanu	O+		

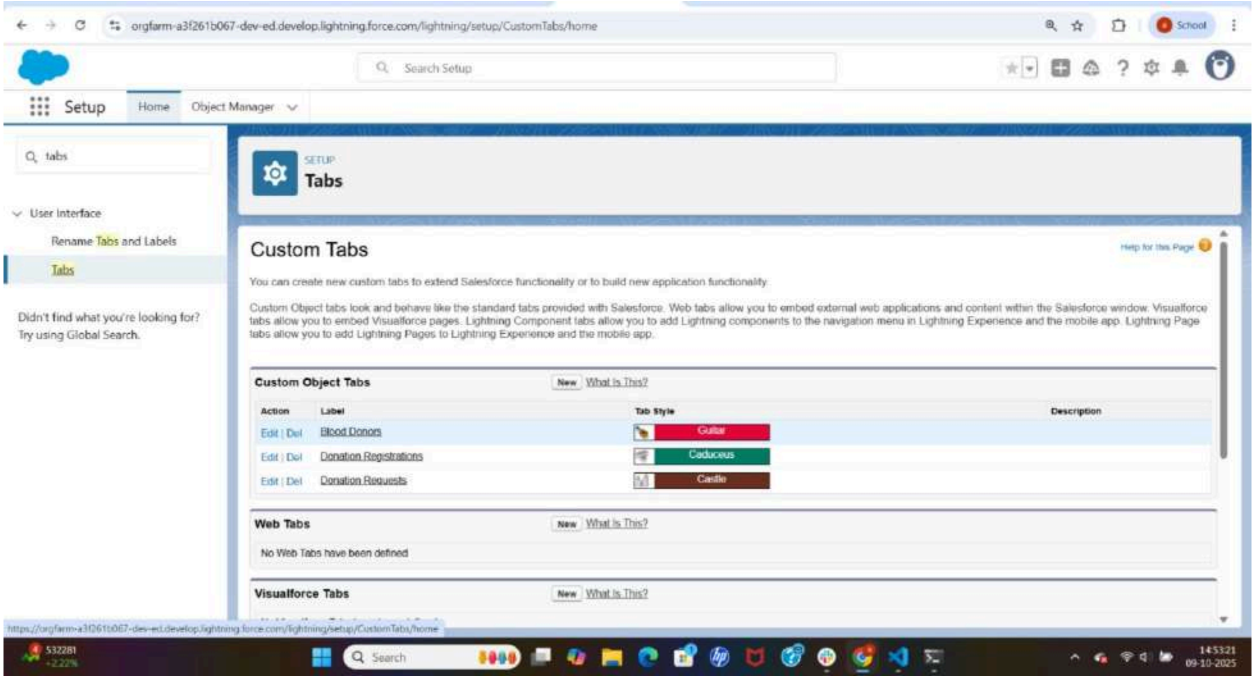
Step 3. Tabs

Tabs in Salesforce are navigation elements that allow users to access different objects, pages, or components within an app.
They make it easier for users to move between various parts of the application without confusion.
Each tab represents a specific object or functionality.

In the Blood Donor Management Project, tabs are used to separate and organize data related to donors, donation registrations, and donation requests.

Purpose of Tabs

To provide easy navigation between different sections of the application.
To make the user interface simple and organized.
To allow users to quickly access records of donors, donations, and requests.
To improve the usability of the Lightning App by dividing it into clear parts.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home
School
Search Setup
Setup
Home
Object Manager
tabs
User Interface
Rename Tabs and Labels
Tabs
Didn't find what you're looking for? Try using Global Search.
SETUP
Tabs
Help for this Page
Search
14:53:21
09-10-2025

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	Blood Donors	Guitar	
Edit | Del	Donation Registrations	Caduceus	
Edit | Del	Donation Requests	Castle	

Web Tabs New What Is This?

No Web Tabs have been defined

Visualforce Tabs New What Is This?

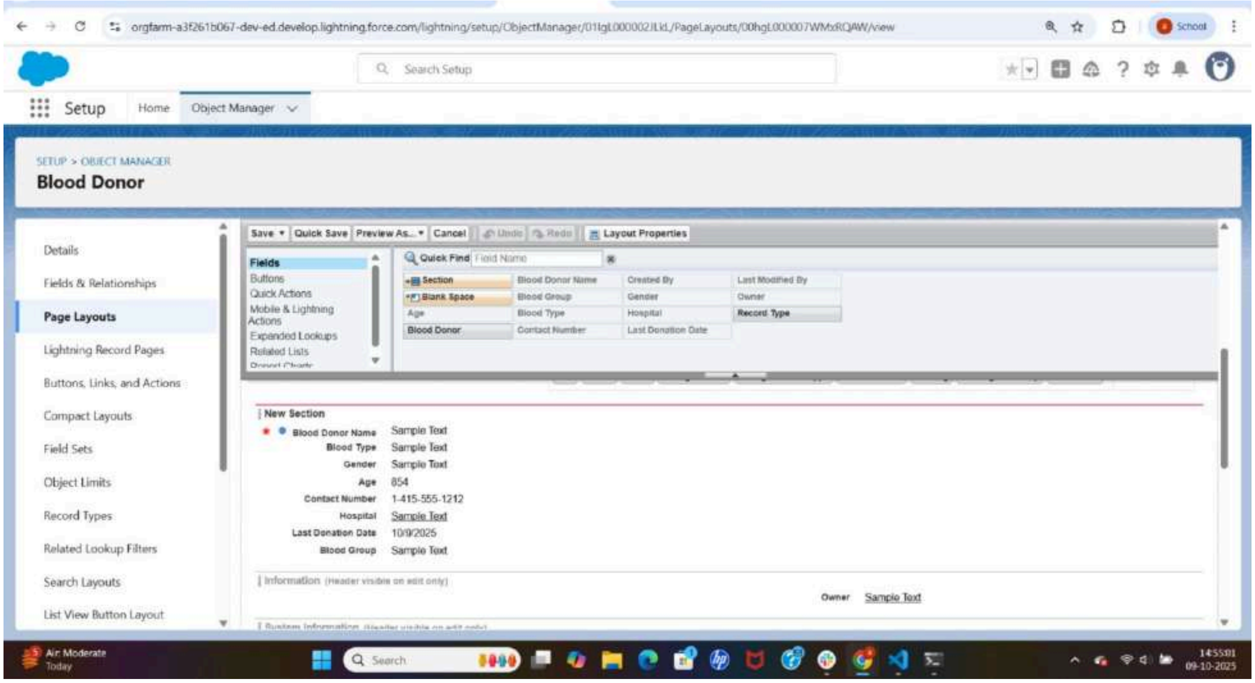
Step 4. Home Page Layouts

In Salesforce, Home Page Layouts define what users see when they first open an app.
It provides an overview of key information, quick actions, reports, and dashboards — all in one place.
A well-designed home page layout helps users quickly access important data and perform frequent actions efficiently.

Purpose of Home Page Layout

To display important donor statistics and real-time information.

To provide quick access to frequently used records and actions.
To create a dashboard-like view for users (Admin, Volunteer, or Staff).
To make the system interactive and user-friendly.
To show charts, reports, and key metrics for better decision-making.



New Section

Blood Donor Name Sample Text
Blood Type Sample Text
Gender Sample Text
Age 854
Contact Number 1-415-555-1212
Hospital Sample Text
Last Donation Date 10/9/2025
Blood Group Sample Text

Information (Header visible on edit only)

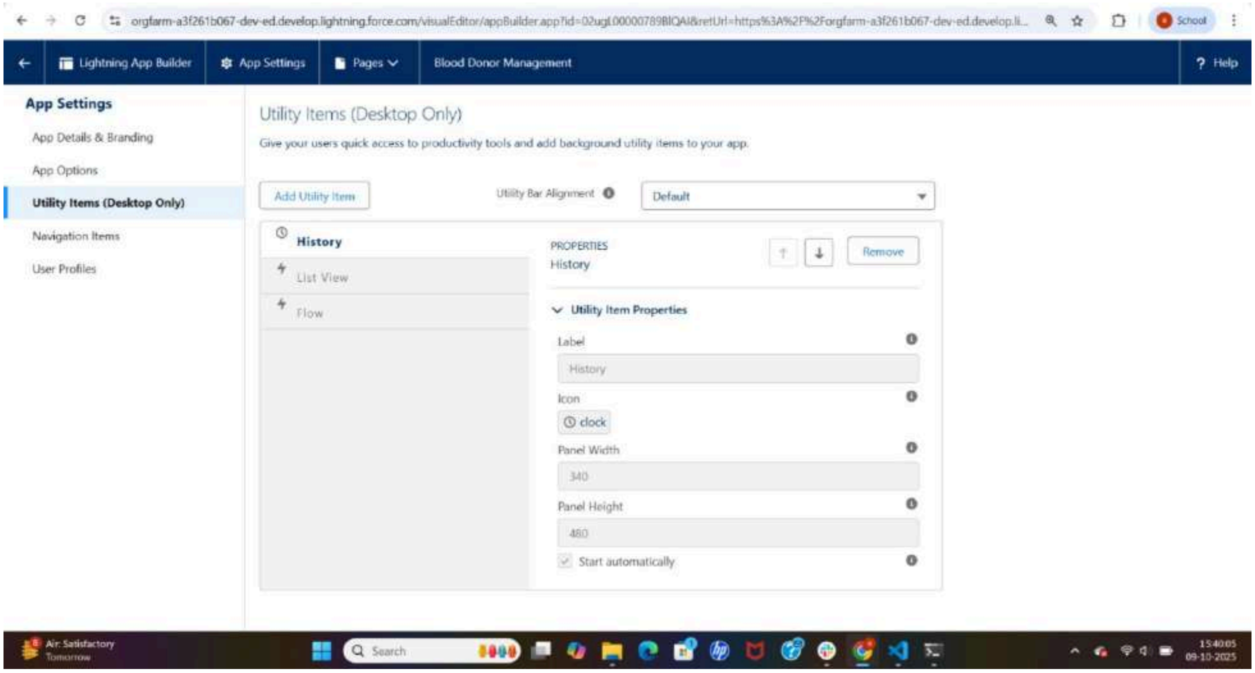
Owner Sample Text

System Information (Header visible on edit only)

Step 5. Utility Bar

The Utility Bar in Salesforce Lightning is a fixed footer bar that provides quick access to important tools and utilities in your app. It allows users to perform frequent tasks or access key features without leaving their current page.

1. Open Setup → App Manager → Edit Lightning App.
2. Select Utility Items (Desktop Only) from the left panel.
3. Click Add Utility Item → Choose options like History, Flow, or List View.
4. Set the Label, Icon, Panel Width, and Panel Height (e.g., 340 x 480).
5. Check the option “Start automatically” if you want it to open automatically when the app loads.
6. Click Save and Activate.



Utility Items (Desktop Only)

Give your users quick access to productivity tools and add background utility items to your app.

Utility Bar Alignment: Default

PROPERTIES
History

Utility Item Properties

Label: History

Icon: clock

Panel Width: 340

Panel Height: 480

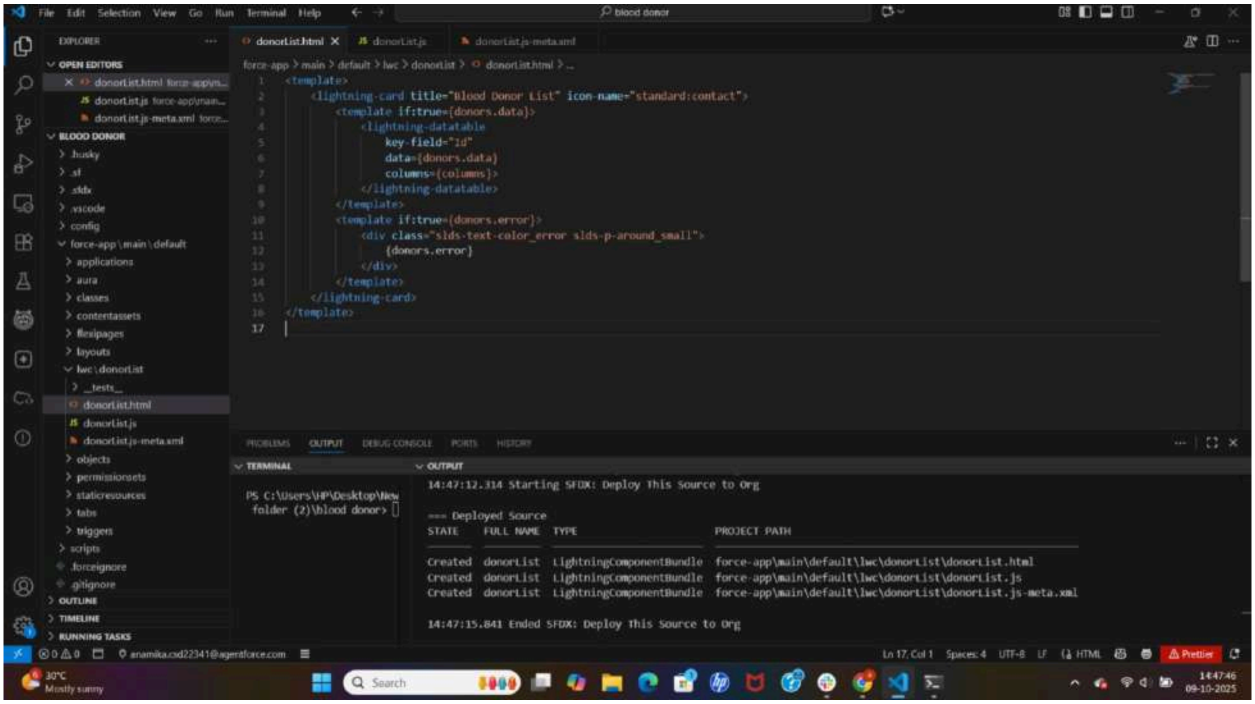
☑ Start automatically

Step 6. LWC (Lightning Web Components)

Lightning Web Components (LWC) is a modern framework introduced by Salesforce for building fast, lightweight, and reusable components for the Salesforce platform. LWC is based on standard web technologies like HTML, CSS, and JavaScript, which makes it easier for developers to create responsive and efficient user interfaces. LWC works seamlessly with the Salesforce data model, allowing developers to access Salesforce objects, fields, and business logic through Apex classes and Lightning Data Service.

Key Features of LWC

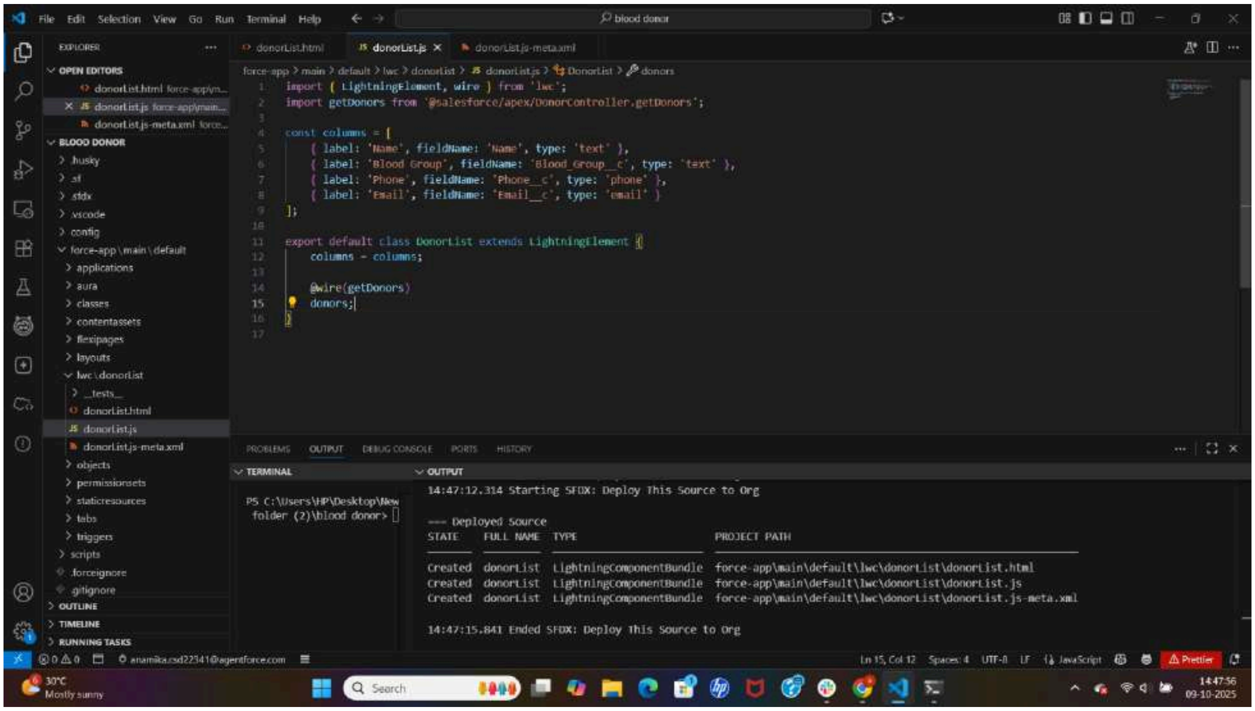
Uses standard web components for better performance.
Lightweight and fast compared to Aura Components.
Can be integrated with Salesforce Lightning pages, apps, and record pages.
Supports event-driven architecture to communicate between components.
Enables reusability and modular design.



```html
<template>
    <lightning-card title="Blood Donor List" icon-name="standard:contact">
        <template if:true={donors.data}>
            <lightning-datatable
                key-field="Id"
                data={donors.data}
                columns={columns}>
            </lightning-datatable>
        </template>
        <template if:true={donors.error}>
            <div class="slds-text-color_error slds-p-around_small">
                {donors.error}
            </div>
        </template>
    </lightning-card>
</template>
```

```
PS C:\Users\HP\Desktop\New
 folder (2)\blood donor>
```

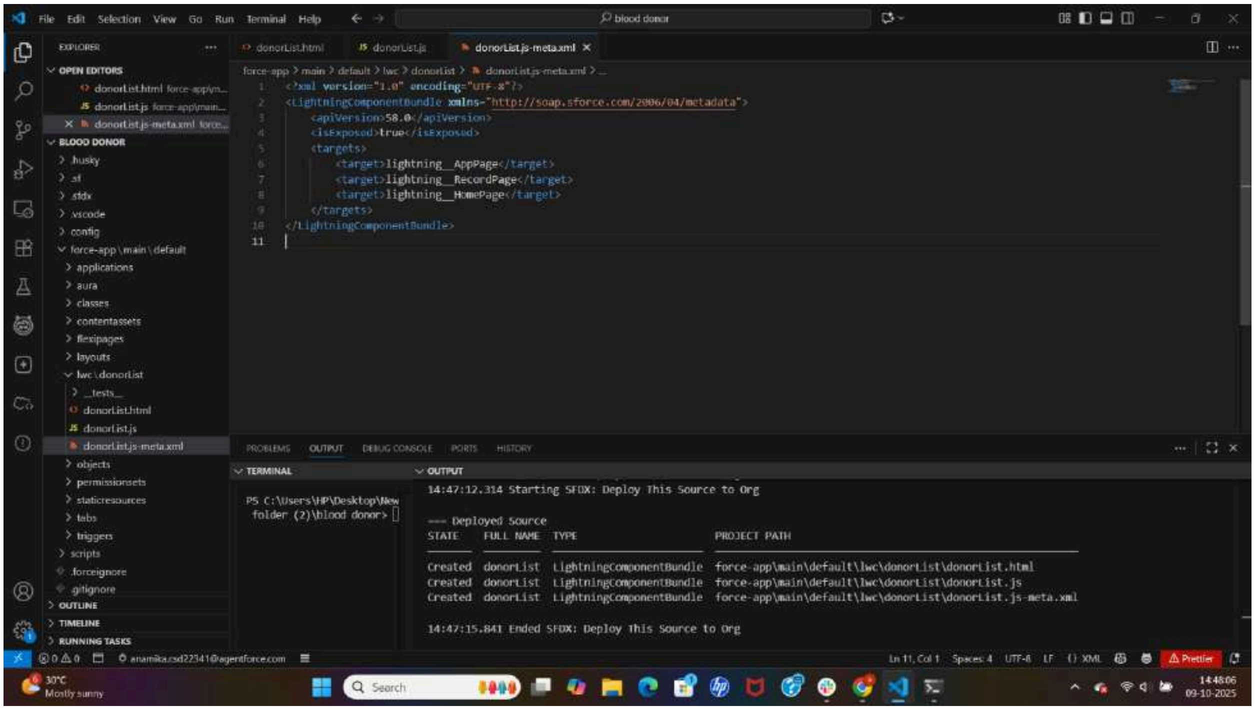
```
14:47:12.314 Starting SFDX: Deploy This Source to Org

=== Deployed Source
STATE    FULL NAME  TYPE                      PROJECT PATH
───────  ─────────  ────────────────────────  ──────────────────────────────────────────────────────
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.html
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js-meta.xml

14:47:15.841 Ended SFDX: Deploy This Source to Org
```



```javascript
import { LightningElement, wire } from 'lwc';
import getDonors from '@salesforce/apex/DonorController.getDonors';

const columns = [
    { label: 'Name', fieldName: 'Name', type: 'text' },
    { label: 'Blood Group', fieldName: 'Blood_Group__c', type: 'text' },
    { label: 'Phone', fieldName: 'Phone__c', type: 'phone' },
    { label: 'Email', fieldName: 'Email__c', type: 'email' }
];

export default class DonorList extends LightningElement {
    columns = columns;

    @wire(getDonors)
    donors;
}
```

```
PS C:\Users\HP\Desktop\New
 folder (2)\blood donor>
```

```
14:47:12.314 Starting SFDX: Deploy This Source to Org

=== Deployed Source
STATE    FULL NAME  TYPE                      PROJECT PATH
───────  ─────────  ────────────────────────  ──────────────────────────────────────────────────────
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.html
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js-meta.xml

14:47:15.841 Ended SFDX: Deploy This Source to Org
```



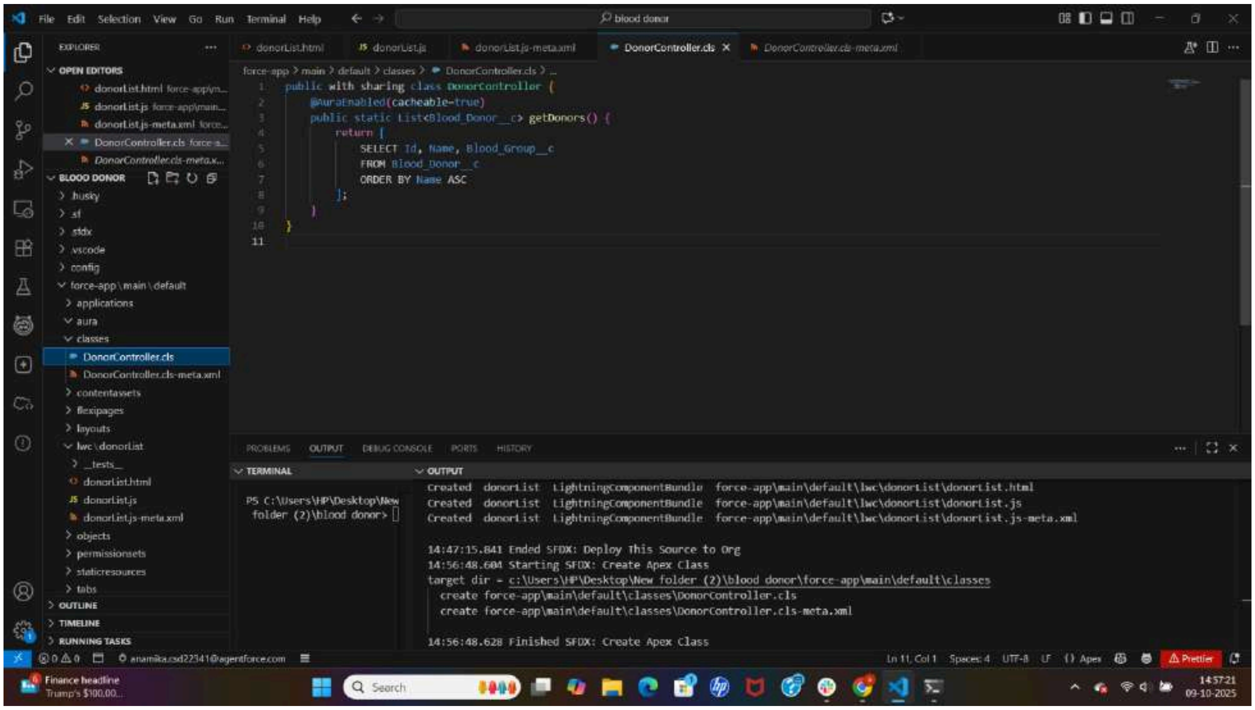
```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__AppPage</target>
        <target>lightning__RecordPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```

```
PS C:\Users\HP\Desktop\New
 folder (2)\blood donor>
```

```
14:47:12.314 Starting SFDX: Deploy This Source to Org

=== Deployed Source
STATE    FULL NAME  TYPE                       PROJECT PATH
───────  ─────────  ─────────────────────────  ───────────────────────────────────────────────────────
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.html
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js-meta.xml

14:47:15.841 Ended SFDX: Deploy This Source to Org
```

Step 7. Apex With LWC

Apex is the server-side language of Salesforce, used for database operations and business logic. Lightning Web Components (LWC) are used for building modern, fast, and reusable user interfaces. By combining both, LWC handles the user interface while Apex fetches or updates data from Salesforce objects, enabling dynamic and interactive applications (e.g., searching donors by blood group in a Blood Donor Management project).



```
public with sharing class DonorController {
    @AuraEnabled(cacheable=true)
    public static List<Blood_Donor__c> getDonors() {
        return [
            SELECT Id, Name, Blood_Group__c
            FROM Blood_Donor__c
            ORDER BY Name ASC
        ];
    }
}
```

```
PS C:\Users\HP\Desktop\New
 folder (2)\blood donor>
```

```
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.html
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js
Created  donorList  LightningComponentBundle  force-app\main\default\lwc\donorList\donorList.js-meta.xml

14:47:15.841 Ended SFDX: Deploy This Source to Org
14:56:48.604 Starting SFDX: Create Apex Class
target dir - c:\Users\HP\Desktop\New folder (2)\blood donor\force-app\main\default\classes
  create force-app\main\default\classes\DonorController.cls
  create force-app\main\default\classes\DonorController.cls-meta.xml

14:56:48.628 Finished SFDX: Create Apex Class
```

Step 8. Events in LWC

Events in Lightning Web Components (LWC) are custom or standard signals used to communicate between components or to trigger actions.

Purpose: To allow components to communicate in a structured way without tight coupling.
Types of Events in LWC

Standard DOM Events
Built-in events like click, change, input.

Example: <button onclick={handleClick}>Click Me</button>.
Custom Events

Used to send data between components.
Created using CustomEvent in JavaScript.
Can carry data in the detail property.

Step 9. Wire Adapters

Wire adapters are reactive mechanisms in LWC that automatically fetch data from Salesforce and pass it to your component.
They react to changes in the component’s properties and refresh the data automatically.
Typically used to get data from Salesforce objects, Apex methods, or Lightning Data Service without manually writing complex logic.

Purpose

In Blood Donor Management System, wire adapters can be used to:
Fetch the list of all donors.
Display blood requests dynamically.
Show blood inventory status in real-time.

Advantages

Automatic reactivity: updates UI when data changes.
Reduces manual code for data fetching.
Easier maintenance and cleaner code.

```
import { LightningElement, wire } from 'lwc';
import getAllDonors from '@salesforce/apex/DonorController.getAllDonors';

export default class DonorList extends LightningElement {
   donors;
   error;

   // Wire the Apex method
   @wire(getAllDonors)
   wiredDonors({ error, data }) {
      if (data) {
         this.donors = data;
         this.error = undefined;
      } else if (error) {
         this.error = error;
         this.donors = undefined;
```

```
    }
  }
}
```

Step 10. Imperative Apex Calls

Imperative Apex calls allow an LWC component to explicitly call an Apex method when needed, rather than automatically fetching data like a wire adapter.

Provides more control over when and how data is fetched.

Used when data is dependent on dynamic input parameters or needs processing before display.

Purpose

Imperative Apex calls are useful to:

Submit a new blood request after a form is filled.

Update a donor’s status after a donation.

Approve or reject urgent blood requests.

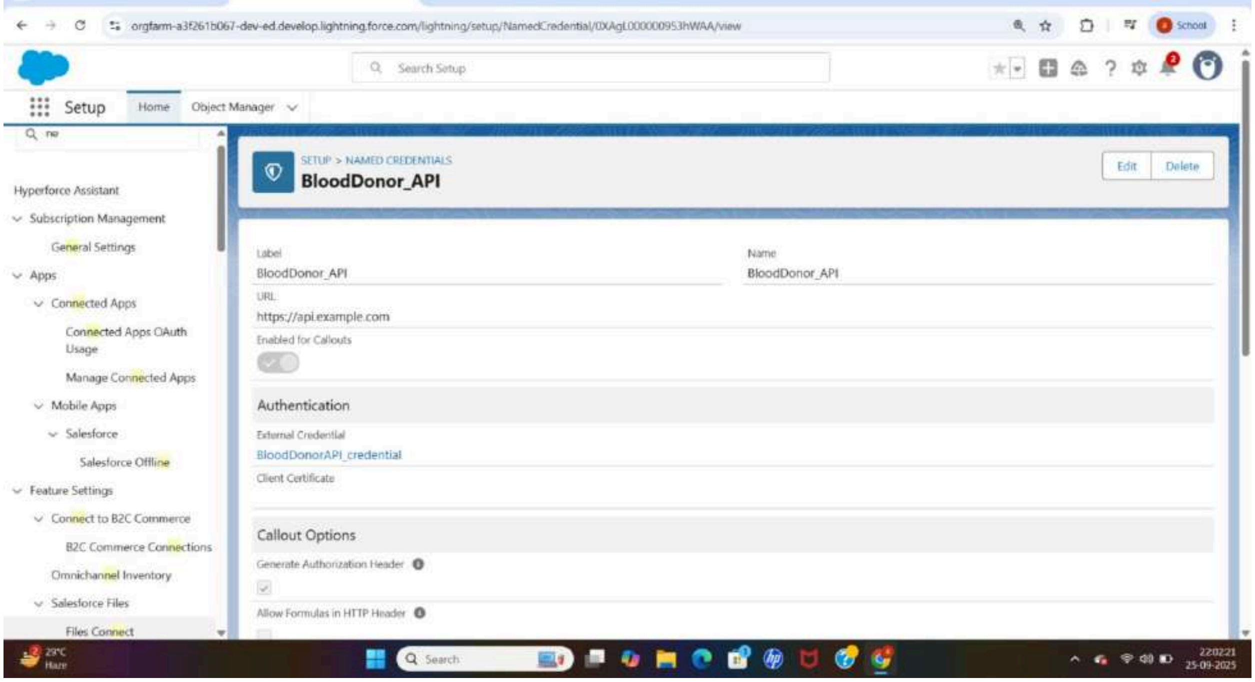
```
import { LightningElement, track } from 'lwc';
import createBloodRequest from
'@salesforce/apex/BloodRequestController.createBloodRequest';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';

export default class BloodRequestForm extends LightningElement {
  @track donorId;
  @track units;

  handleDonorChange(event) {
    this.donorId = event.target.value;
  }

  handleUnitsChange(event) {
    this.units = event.target.value;
  }
```

```
    handleSubmit() {
        // Imperative Apex call
        createBloodRequest({ donorId: this.donorId, units: parseInt(this.units) })
            .then(result => {
                this.dispatchEvent(
                    new ShowToastEvent({
                        title: 'Success',
                        message: result,
                        variant: 'success'
                    })
                );
            })
            .catch(error => {
                this.dispatchEvent(
                    new ShowToastEvent({
                        title: 'Error',
                        message: error.body.message,
                        variant: 'error'
                    })
                );
            });
    }
}
```

Phase 7. Integration & External Access

Step 1. Named Credentials

Label - BloodDonor_API
Name - BloodDonor_API



SETUP > NAMED CREDENTIALS

BloodDonor_API

Label: BloodDonor_API

Name: BloodDonor_API

URL: https://api.example.com

Enabled for Callouts

Authentication

External Credential: BloodDonorAPI_credential

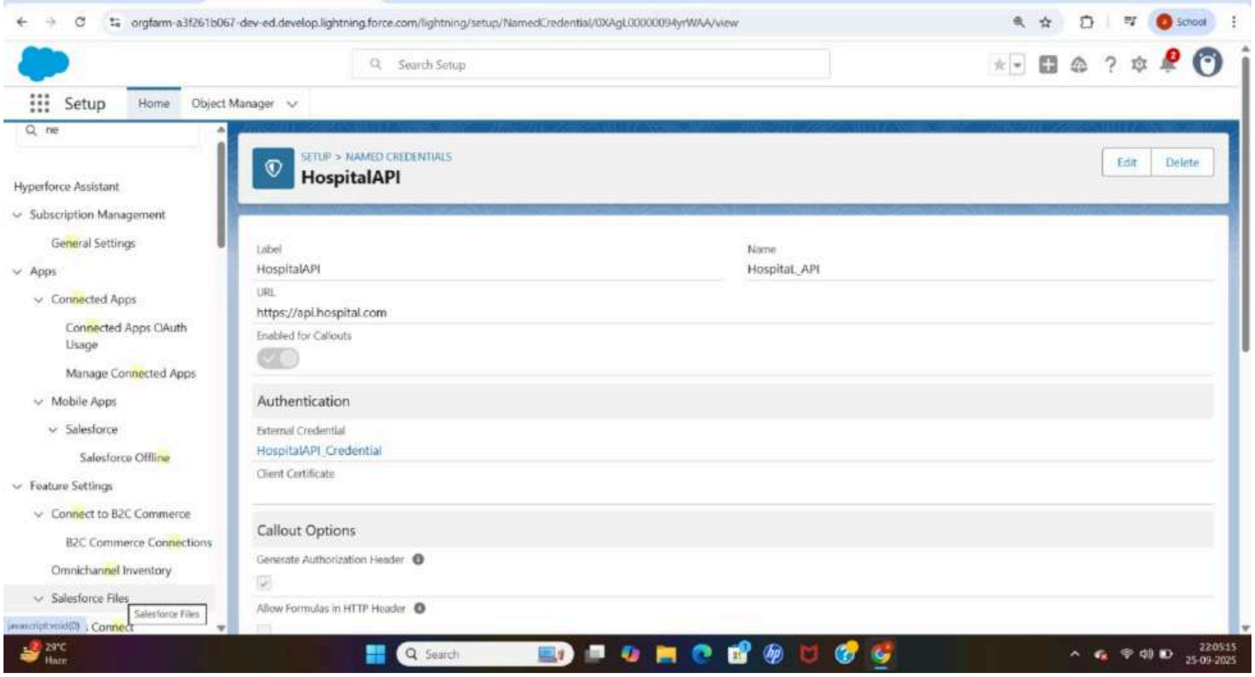
Client Certificate

Callout Options

Generate Authorization Header ☑

Allow Formulas in HTTP Header

Label - HospitalAPI
Name - Hospital_API



SETUP > NAMED CREDENTIALS

HospitalAPI

Label: HospitalAPI

Name: Hospital_API

URL: https://api.hospital.com

Enabled for Callouts

Authentication

External Credential: HospitalAPI_Credential

Client Certificate

Callout Options

Generate Authorization Header ☑

Allow Formulas in HTTP Header

Step 2. External Service Name

BloodDonorService

. Description

Service to manage Blood Donor records through API integration.

. Service Schema

Uploaded JSON schema (blooddonor_service_schema.json) defining:

Get Donors

Add Donor

Get Donor by ID

Update Donor by ID

. Named Credential

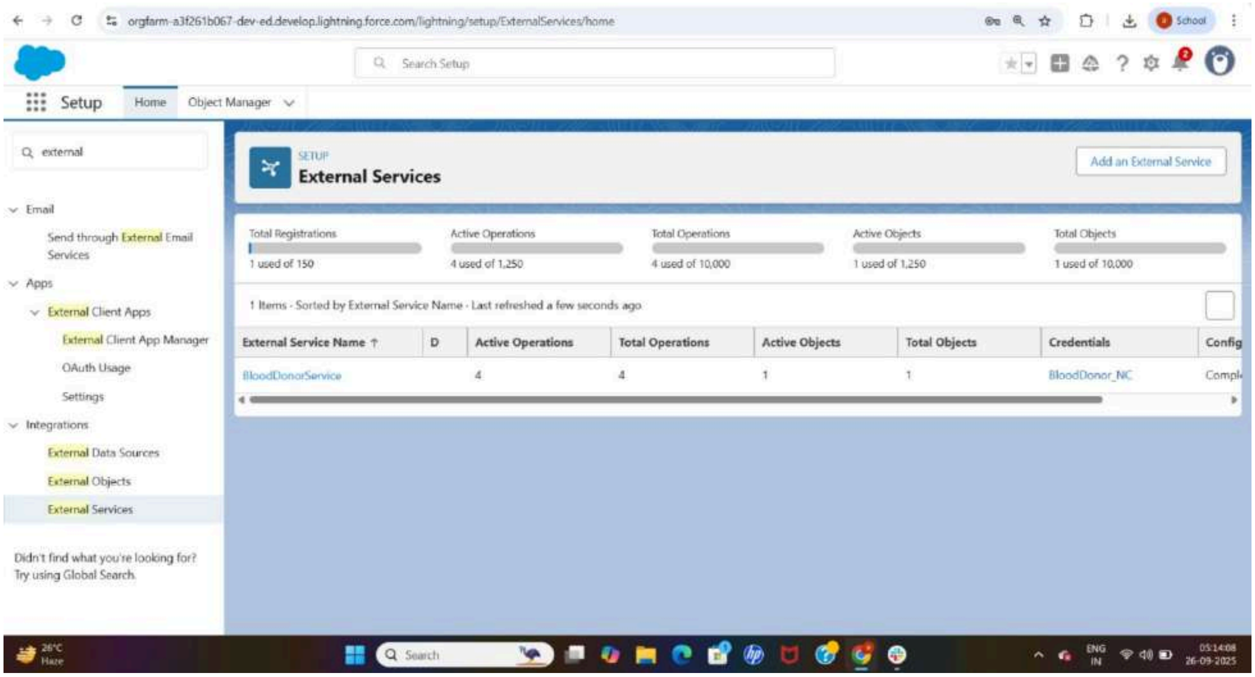
BloodDonor_NC (used for authentication and endpoint configuration)

. Result

Registration Completed

Active Operations: 4

Total Objects Created: 1


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/ExternalServices/home
Search Setup
Setup
Home
Object Manager
external
Email
Send through External Email Services
Apps
External Client Apps
External Client App Manager
OAuth Usage
Settings
Integrations
External Data Sources
External Objects
External Services
Didn't find what you're looking for? Try using Global Search.
SETUP
External Services
Add an External Service

Total Registrations	Active Operations	Total Operations	Active Objects	Total Objects
1 used of 150	4 used of 1,250	4 used of 10,000	1 used of 1,250	1 used of 10,000

1 Items · Sorted by External Service Name · Last refreshed a few seconds ago

External Service Name ↑	D	Active Operations	Total Operations	Active Objects	Total Objects	Credentials	Config
BloodDonorService		4	4	1	1	BloodDonor_NC	Compl

Step 3. Callouts

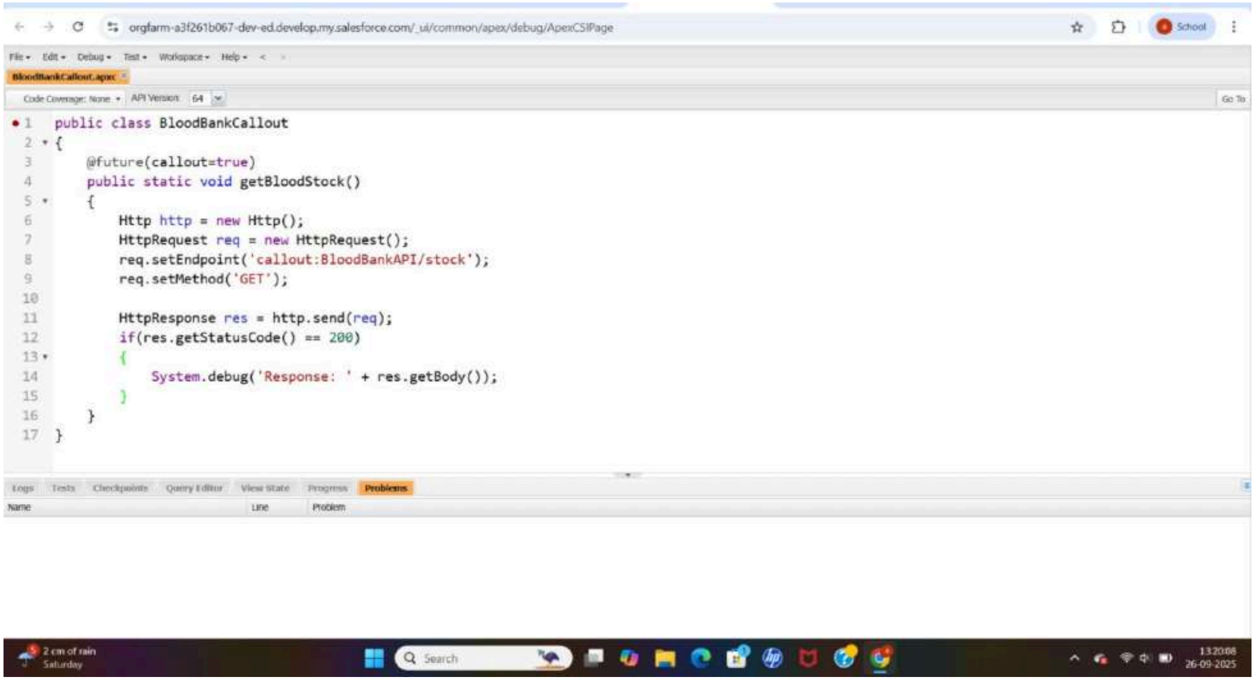
Go to Setup → Remote Site Settings → New Remote Site

Remote Site Name: Blood_Bank_API

Setup → Named Credentials → New

Label: BloodBankAPI

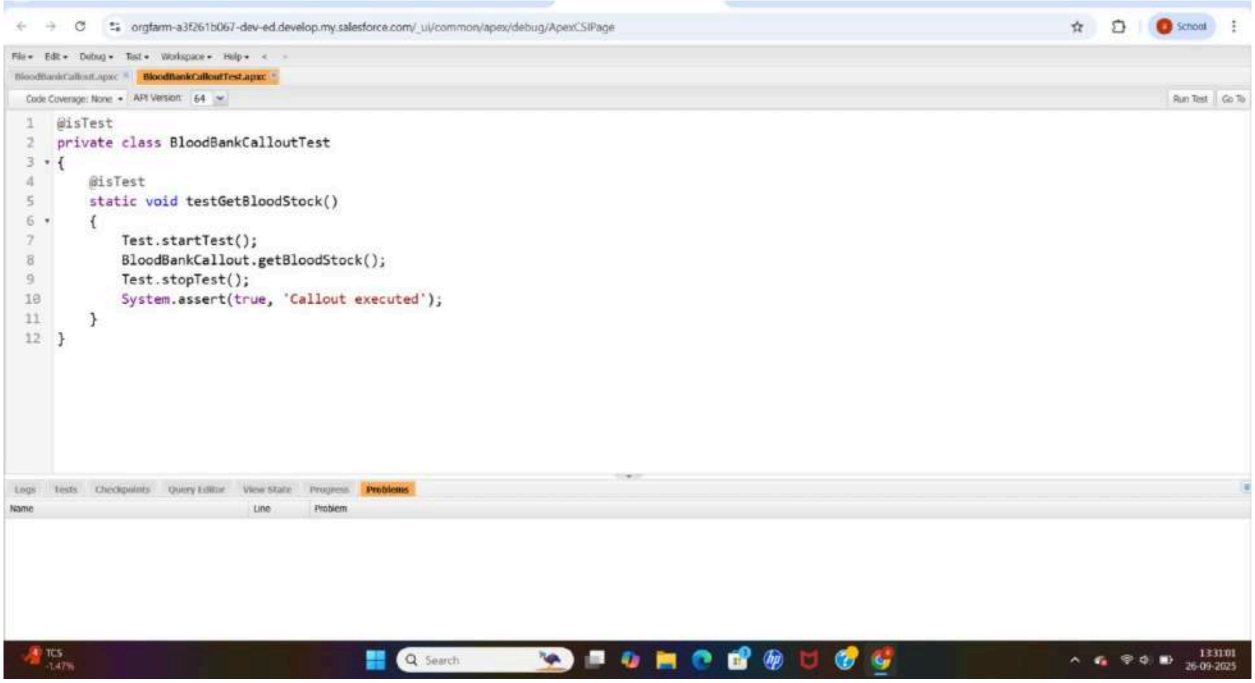
Class Name - BloodBankCallout



```
public class BloodBankCallout
{
    @future(callout=true)
    public static void getBloodStock()
    {
        Http http = new Http();
        HttpRequest req = new HttpRequest();
        req.setEndpoint('callout:BloodBankAPI/stock');
        req.setMethod('GET');

        HttpResponse res = http.send(req);
        if(res.getStatusCode() == 200)
        {
            System.debug('Response: ' + res.getBody());
        }
    }
}
```

```
public class BloodBankCallout
{
   @future(callout=true)
   public static void getBloodStock()
   {
      Http http = new Http();
      HttpRequest req = new HttpRequest();
      req.setEndpoint('callout:BloodBankAPI/stock');
      req.setMethod('GET');

      HttpResponse res = http.send(req);
      if(res.getStatusCode() == 200)
      {
         System.debug('Response: ' + res.getBody());
      }
   }
}
```

Class Name - BloodBankCalloutTest



```
@isTest
private class BloodBankCalloutTest
{
    @isTest
    static void testGetBloodStock()
    {
        Test.startTest();
        BloodBankCallout.getBloodStock();
        Test.stopTest();
        System.assert(true, 'Callout executed');
    }
}
```

```
@isTest
private class BloodBankCalloutTest
{
  @isTest
  static void testGetBloodStock()
  {
    Test.startTest();
    BloodBankCallout.getBloodStock();
    Test.stopTest();
    System.assert(true, 'Callout executed');
  }
}
```

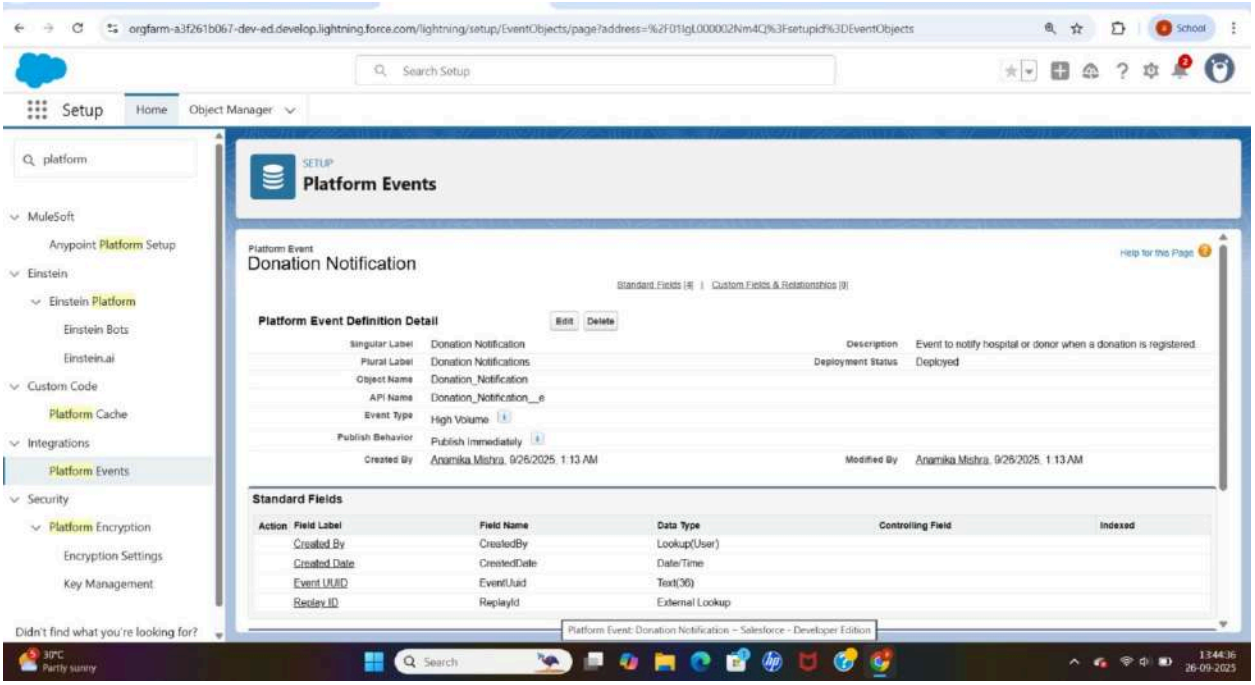
Step 4. Platform Events

Go to Setup → Platform Events → New Platform Event.

Label: Donation Notification

Plural Label: Donation Notifications

API Name: Donation_Notification__e



Platform Event
Donation Notification

Standard Fields [4] | Custom Fields & Relationships [0]

Platform Event Definition Detail

Singular Label	Donation Notification	Description	Event to notify hospital or donor when a donation is registered
Plural Label	Donation Notifications	Deployment Status	Deployed
Object Name	Donation_Notification		
API Name	Donation_Notification__e		
Event Type	High Volume		
Publish Behavior	Publish Immediately		
Created By	Anamika Mishra, 9/26/2025, 1:13 AM	Modified By	Anamika Mishra, 9/26/2025, 1:13 AM

Standard Fields

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Event UUID	EventUuid	Text(36)		
	Replay ID	ReplayId	External Lookup		

```
@isTest
private class BloodBankCalloutTest
{
   @isTest
   static void testGetBloodStock()
   {
      Test.startTest();
      BloodBankCallout.getBloodStock();
      Test.stopTest();
      System.assert(true, 'Callout executed');
   }
}
```

Step 5. Change Data Capture

Go to Setup → Quick Find → Change Data Capture.

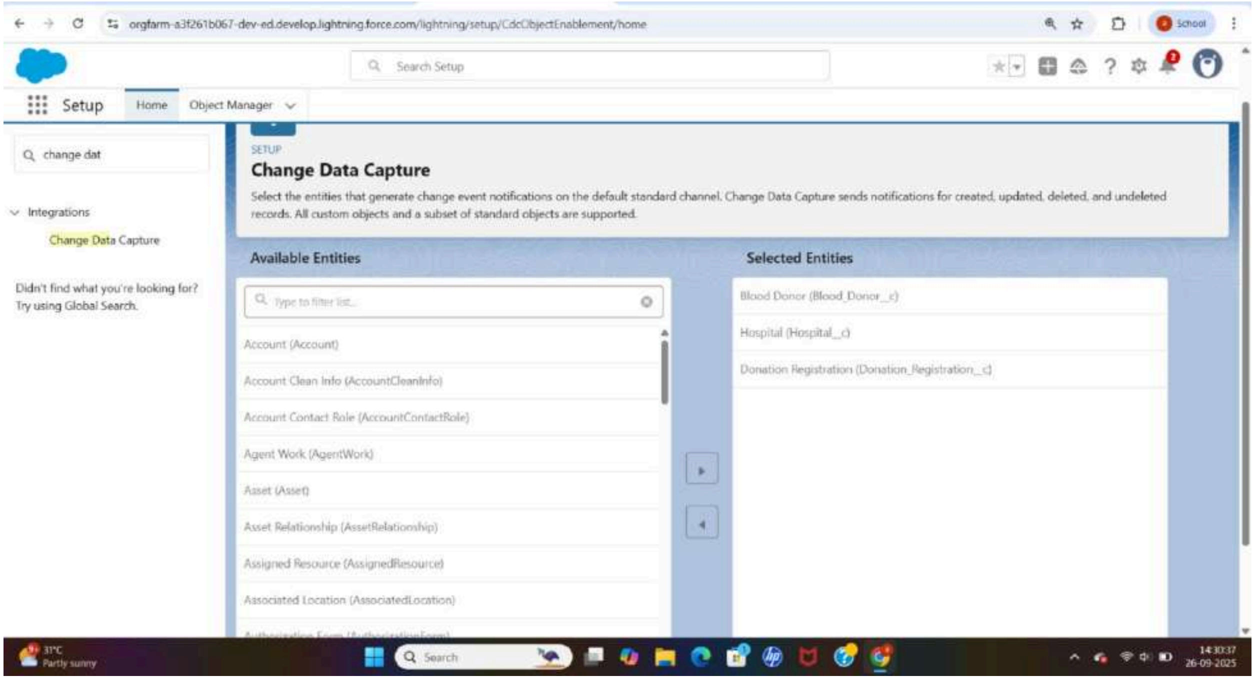
Click Enable Entities.

Select the objects you want to track (for example):

Blood_Donor__c (Donor object)

Donation_Registration__c (Donation records)

Hospital__c (Hospital details)



orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/CdcObjectEnablement/home
Search Setup
Setup
Home
Object Manager
change dat
Integrations
Change Data Capture
Didn't find what you're looking for? Try using Global Search.
SETUP

Change Data Capture

Select the entities that generate change event notifications on the default standard channel. Change Data Capture sends notifications for created, updated, deleted, and undeleted records. All custom objects and a subset of standard objects are supported.

Available Entities

- Account (Account)
- Account Clean Info (AccountCleanInfo)
- Account Contact Role (AccountContactRole)
- Agent Work (AgentWork)
- Asset (Asset)
- Asset Relationship (AssetRelationship)
- Assigned Resource (AssignedResource)
- Associated Location (AssociatedLocation)
- Authorization Form (AuthorizationForm)

Selected Entities

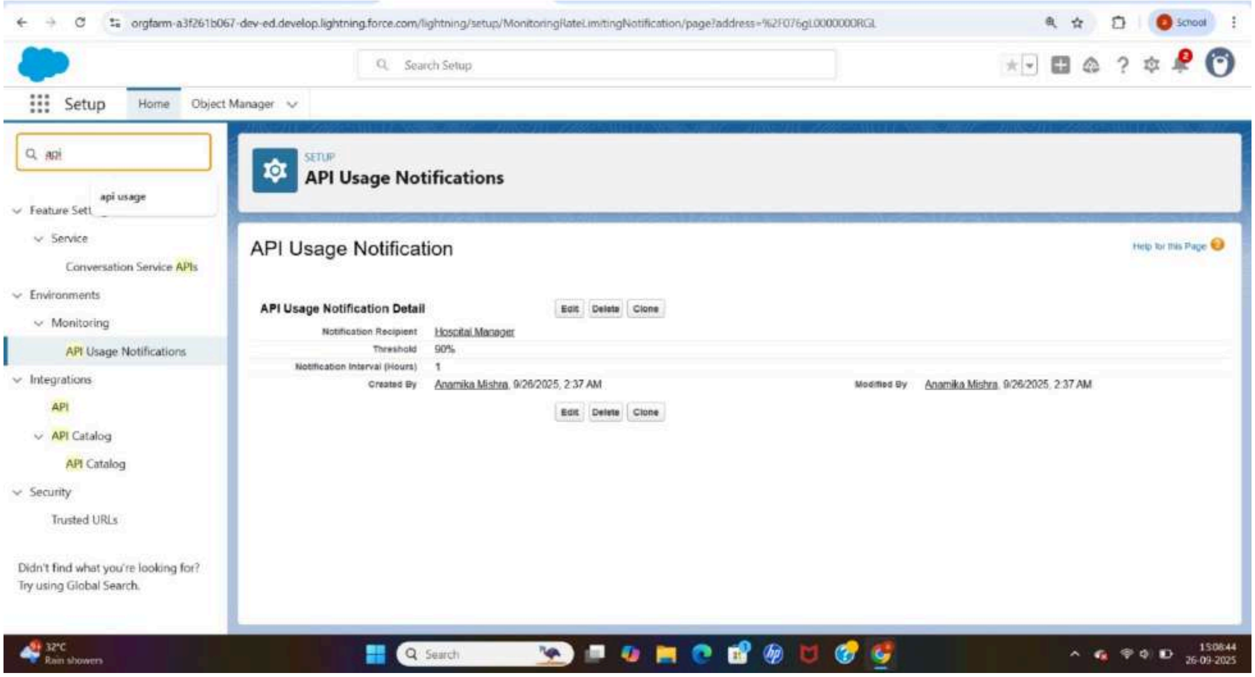
- Blood Donor (Blood_Donor__c)
- Hospital (Hospital__c)
- Donation Registration (Donation_Registration__c)

Step 6. API Limits

Recipient: Hospital Manager

Threshold: 90%

Interval: 1 hour



API Usage Notifications

API Usage Notification

API Usage Notification Detail

Notification Recipient	Hoscital Manager
Threshold	90%
Notification Interval (Hours)	1
Created By	Anamika Mishra, 9/26/2025, 2:37 AM
Modified By	Anamika Mishra, 9/26/2025, 2:37 AM

Step 7. OAuth & Authentication

Setup → App Manager → New Connected App

Fill details:

Connected App Name: BloodDonorApp

API Name: BloodDonorApp

Contact Email: aapka email

Enable OAuth Settings:

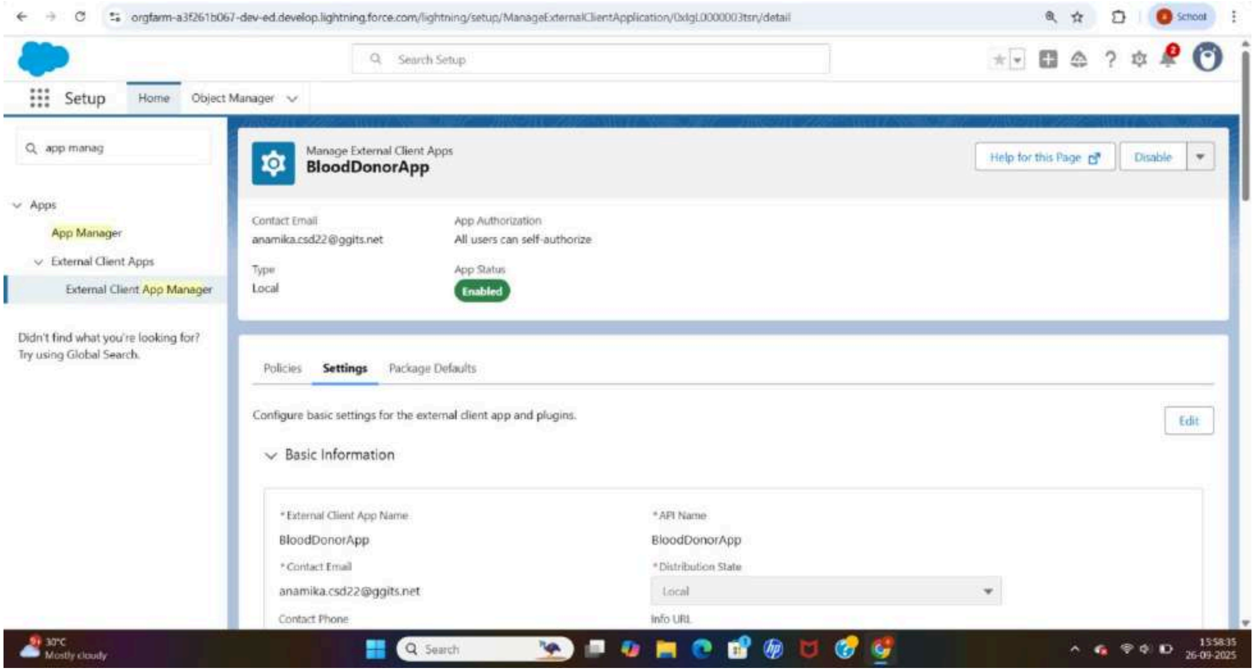
Enable OAuth Settings → check

Callback URL: https://login.salesforce.com/services/oauth2/callback (testing)

Selected OAuth Scopes:

Full access (full)

Perform requests on your behalf at any time (refresh_token, offline_access)



Manage External Client Apps
BloodDonorApp

Contact Email
anamika.csd22@ggits.net

App Authorization
All users can self-authorize

Type
Local

App Status
Enabled

Configure basic settings for the external client app and plugins.

Basic Information

*External Client App Name
BloodDonorApp

*API Name
BloodDonorApp

*Contact Email
anamika.csd22@ggits.net

*Distribution State
Local

Contact Phone

Info URL

Step 8. Remote Site Settings

Purpose:

Remote Site Settings in Salesforce allow your org to make callouts to external APIs securely. For the Blood Donor Management project, this is required to access external blood donor data.

1. Navigate to Remote Site Settings

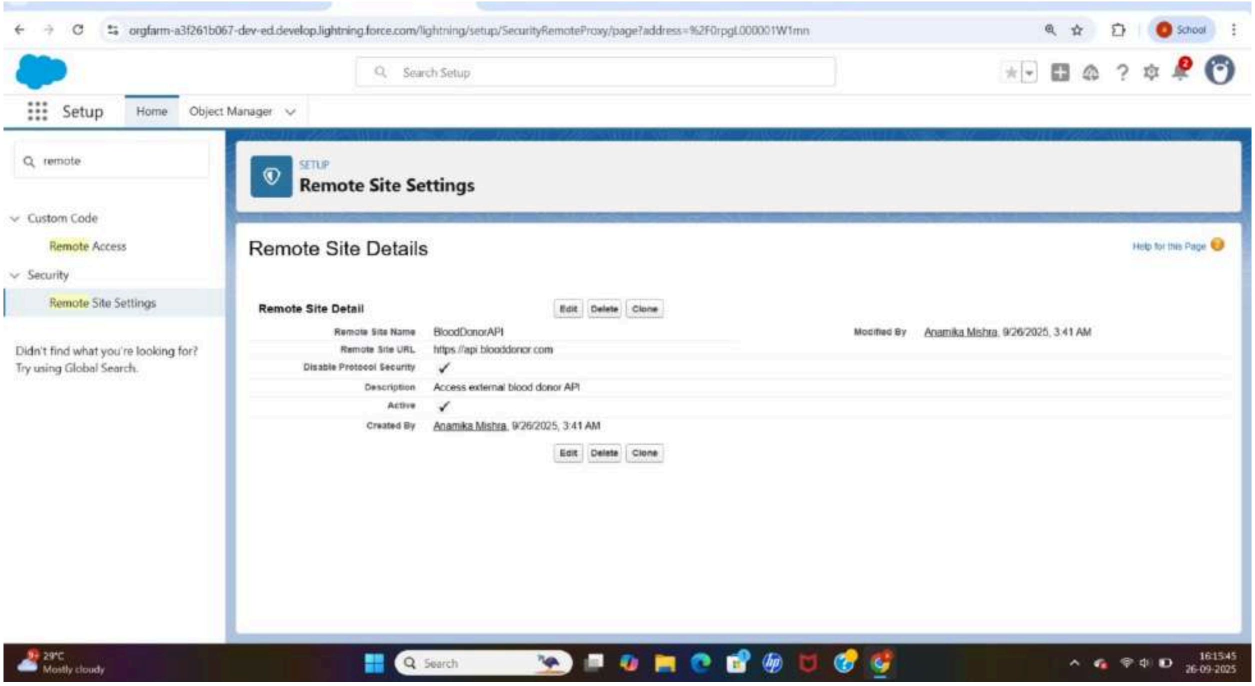
Go to Setup → Quick Find → Remote Site Settings → New Remote Site.

2. Enter Remote Site Details

Remote Site Name: BloodDonorAPI

Remote Site URL: https://api.blooddonor.com

Description: Access external blood donor API



Remote Site Details

Remote Site Detail

Remote Site Name	BloodDonorAPI
Remote Site URL	https://api.blooddonor.com
Disable Protocol Security	✓
Description	Access external blood donor API
Active	✓
Created By	Anamika Mishra, 9/26/2025, 3:41 AM
Modified By	Anamika Mishra, 9/26/2025, 3:41 AM

Phase 8: Data Management & Deployment

Step 1. Data Loader

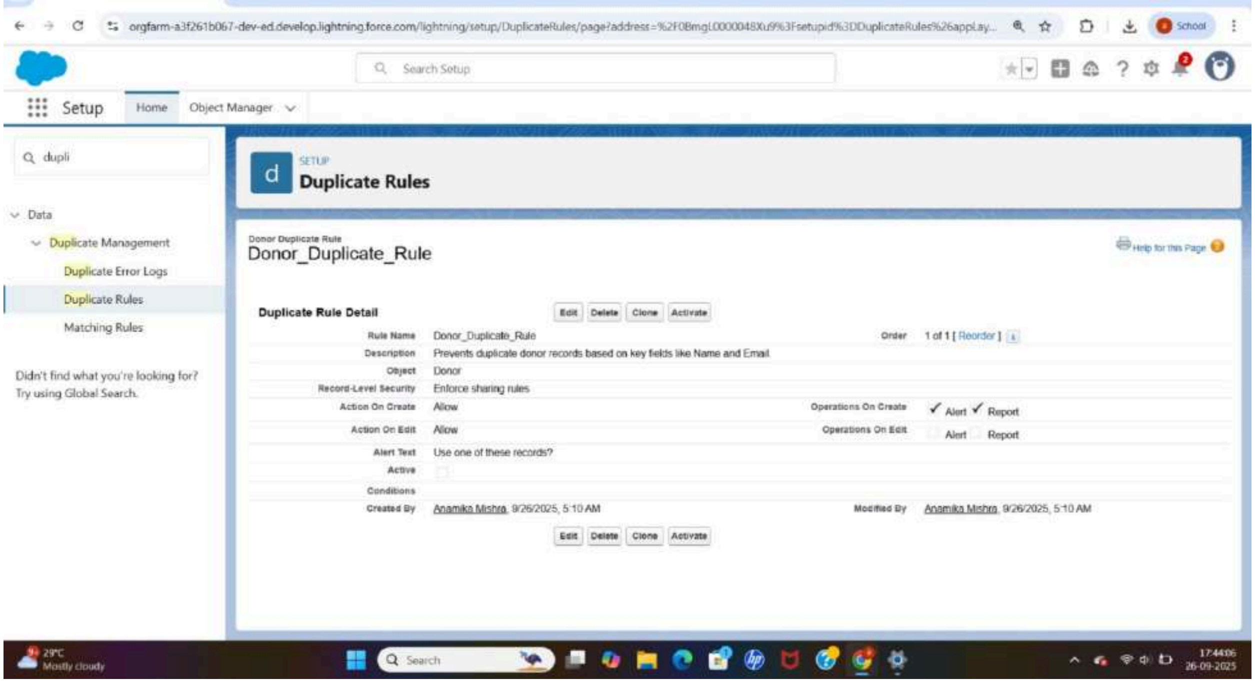
Purpose
Bulk import, export, update, delete Salesforce records.
Useful for large data operations (thousands or millions of records).
Supports CSV file format and automation via command line.

Steps (Process)
Download & Install Data Loader → From Salesforce Setup → Data Loader.
Login → Use Salesforce credentials (or OAuth).
Select Operation → Insert, Update, Upsert, Delete, Export, Export All.
Choose Object → Select the Salesforce object to work on (e.g., Contact, Account).

Select CSV File → Map CSV columns to Salesforce fields.
Run Operation → Process the data.
Check Results → Review success and error files generated by Data Loader.

Step 2. Duplicate Rules

Rule Name - Donor_Duplicate_Rule
Description - Prevents duplicate donor records based on key fields like Name and Email.
Object - Donor



Donor Duplicate Rule
Donor_Duplicate_Rule

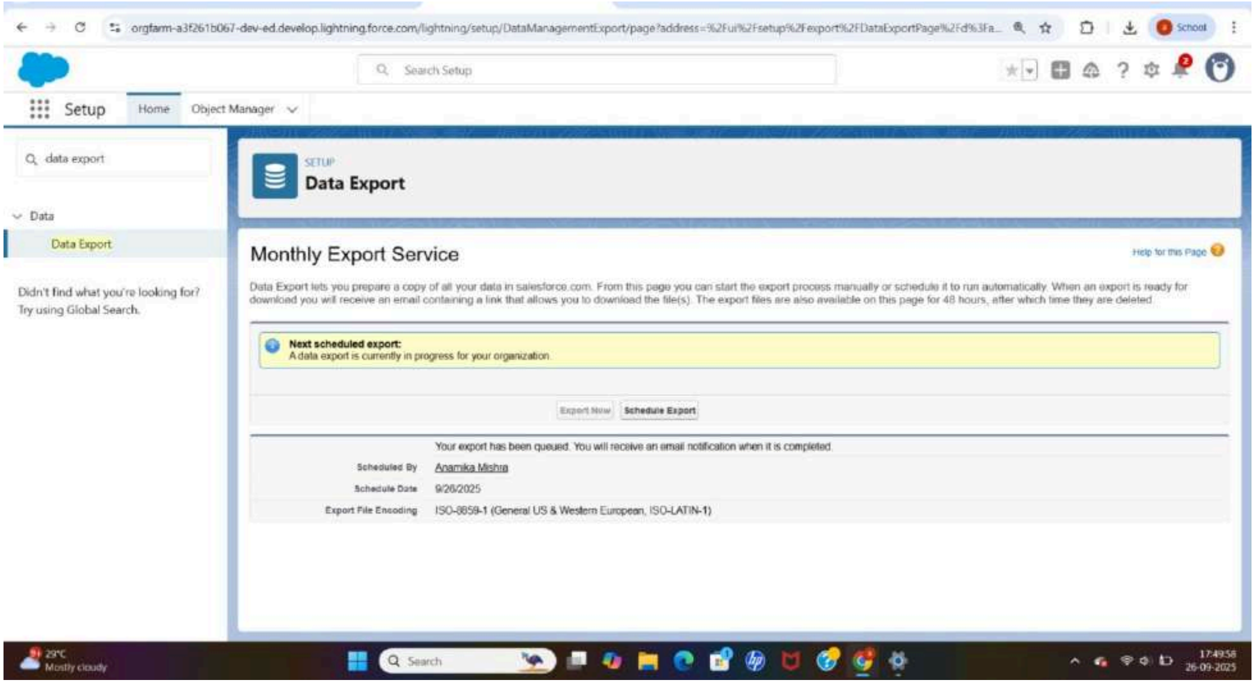
Duplicate Rule Detail

Rule Name	Donor_Duplicate_Rule	Order	1 of 1 [Reorder]
Description	Prevents duplicate donor records based on key fields like Name and Email		
Object	Donor		
Record-Level Security	Enforce sharing rules		
Action On Create	Allow	Operations On Create	✓ Alert ✓ Report
Action On Edit	Allow	Operations On Edit	□ Alert □ Report
Alert Text	Use one of these records?		
Active	□		
Conditions			
Created By	Anamika Mishra, 9/26/2025, 5:10 AM	Modified By	Anamika Mishra, 9/26/2025, 5:10 AM

Step 4: Data Export & Backup

Purpose: Backup donor data regularly to avoid loss.

1. Select objects:
BloodDonor__c
DonationRegistration__c
4. Export & download ZIP file with CSVs.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/DataManagementExport/page?address=%2Fui%2Fsetup%2Fexport%2FDataExportPage%2Fd%3Fa...
Search Setup
Setup
Home
Object Manager
data export
Data
Data Export
Didn't find what you're looking for? Try using Global Search.
SETUP
Data Export
Help for this Page
Export Now
Schedule Export
29°C Mostly cloudy
17:49:58 26-09-2025

Monthly Export Service

Data Export lets you prepare a copy of all your data in salesforce.com. From this page you can start the export process manually or schedule it to run automatically. When an export is ready for download you will receive an email containing a link that allows you to download the file(s). The export files are also available on this page for 48 hours, after which time they are deleted.

Next scheduled export:
A data export is currently in progress for your organization.

Your export has been queued. You will receive an email notification when it is completed.

Scheduled By: Anamika Mishra
Schedule Date: 9/26/2025
Export File Encoding: ISO-8859-1 (General US & Western European, ISO-LATIN-1)

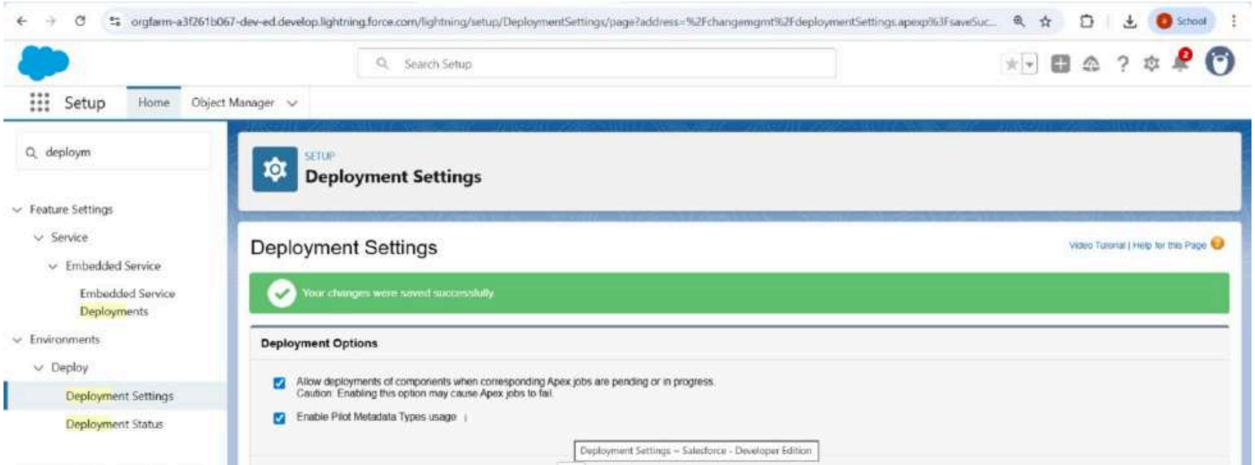
Step 5: Change Sets

A Change Set is a Salesforce tool used to deploy customizations and configurations from one Salesforce environment (usually Sandbox) to another (usually Production) without manually recreating them. It is mainly used for moving metadata like objects, fields, flows, Apex classes, triggers, and other components.

Types:

1. Outbound Change Set – Created in the source org (Sandbox) to send components to another org.

2. Inbound Change Set – Received in the target org (Production) to deploy the components.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/DeploymentSettings/page?address=%2Fchangemgmt%2FdeploymentSettings.apexp%3FsaveSuc...
Search Setup
Setup
Home
Object Manager
deploym
Feature Settings
Service
Embedded Service
Embedded Service Deployments
Environments
Deploy
Deployment Settings
Deployment Status
SETUP
Deployment Settings
Video Tutorial | Help for this Page
Deployment Settings ~ Salesforce - Developer Edition

Deployment Settings

Your changes were saved successfully.

Deployment Options

☑ Allow deployments of components when corresponding Apex jobs are pending or in progress.
Caution: Enabling this option may cause Apex jobs to fail.

☑ Enable Pilot Metadata Types usage

Step 6. ANT Migration Tool

Purpose
Automate deployment of metadata between Salesforce orgs.
Useful for continuous integration (CI) and version control.
Works well for large projects with many components.
Steps (Process)
Install Java & ANT → Ensure Java and Apache ANT are installed on your system.
Download Salesforce ANT Migration Tool → From Salesforce Setup → Tools → Force.com Migration Tool.
Set Up project folder → Create a folder for project and build files.
Configure build.xml & package.xml → Define metadata to retrieve/deploy.
Retrieve Metadata → Use ANT command:
ant retrieve
Deploy Metadata → Use ANT command:
ant deploy

Phase 9. Reporting, Dashboards & Security Review

Step 1. Reports(Tabular, Summary, Matrix,Joined)

Purpose
Define which objects and fields are available for reports in Salesforce.
Control how records are related in reports.
Allow creation of custom reports for specific business needs.
Small Steps (Process)
Go to Setup → Search Report Types.
Click “New Custom Report Type”.
Select Primary Object → Choose the main object (e.g., Accounts, Orders).
Define Related Objects → Add related objects if needed.
Set Deployment Status → In Development (editable) or Deployed (available to all users).
Save → Use this report type when creating reports.

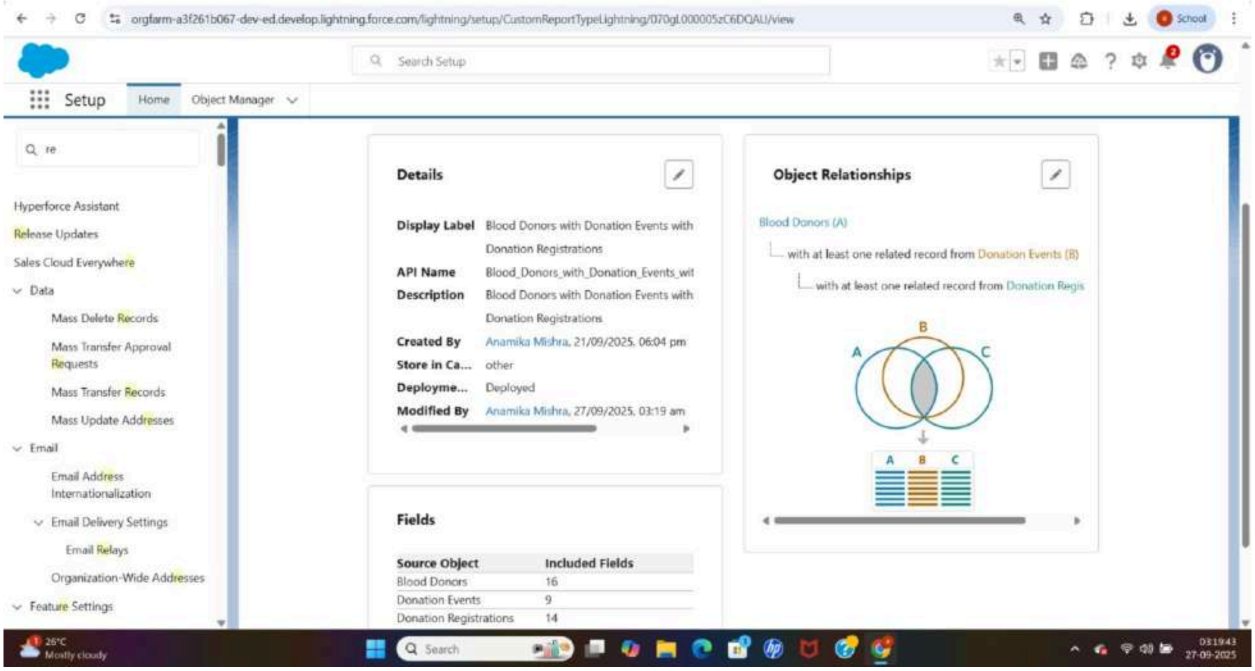
Step 2. Report Types

Purpose

Visually display Salesforce data in charts, graphs, tables, and gauges.
Provide quick insights and analytics for decision-making.
Summarize reports for managers and teams.

Steps (Process)

Go to Dashboards Tab → Click New Dashboard
Name Dashboard → Give a meaningful name and select a folder.
Add Components → Charts, graphs, tables, gauges.
Select Source Reports → Choose reports that will feed data to dashboard components.
Customize Components → Adjust chart types, filters, labels.
Save & Refresh → Ensure data is up-to-date.



Details

Display Label Blood Donors with Donation Events with Donation Registrations
API Name Blood_Donors_with_Donation_Events_wit
Description Blood Donors with Donation Events with Donation Registrations
Created By Anamika Mishra, 21/09/2025, 06:04 pm
Store in Ca... other
Deployme... Deployed
Modified By Anamika Mishra, 27/09/2025, 03:19 am

Object Relationships

Blood Donors (A)
└ with at least one related record from Donation Events (B)
└ with at least one related record from Donation Regis

Fields

Source Object	Included Fields
Blood Donors	16
Donation Events	9
Donation Registrations	14

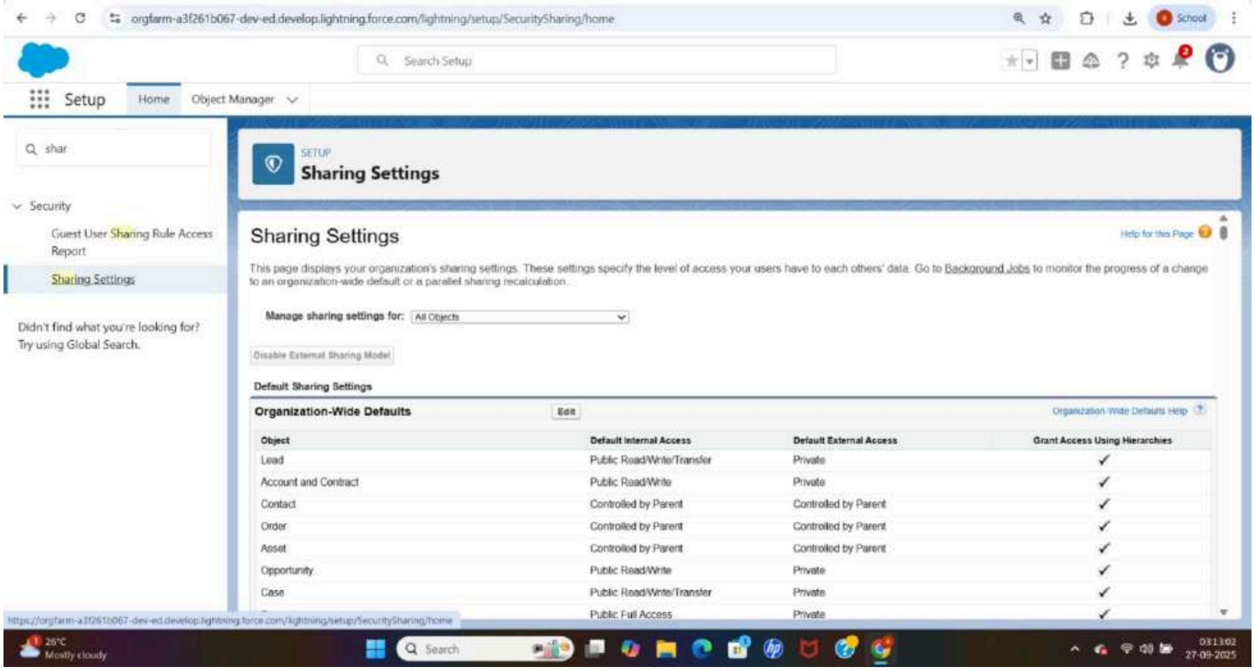
Step 3.Sharing Settings

Purpose

Control record-level access in Salesforce.
Define who can view, edit, or delete records.
Maintain data security while enabling collaboration.

Steps (Process)

Go to Setup → Search Sharing Settings.
Set Organization-Wide Defaults (OWD) → Define baseline access for each object (Public, Private, Controlled).
Create Sharing Rules → Grant access to specific groups/roles.
Use Role Hierarchies → Inherit access based on role levels.
Set Manual Sharing → Allow record owners to share records individually.
Test Access → Log in as a user to verify permissions.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySharing/home
Search Setup
Setup
Home
Object Manager
shar
Security
Guest User Sharing Rule Access Report
Sharing Settings
Didn't find what you're looking for? Try using Global Search.
SETUP
Sharing Settings
Help for this Page
Disable External Sharing Model
Edit
Organization-Wide Defaults Help
Search
26°C Mostly cloudy
03:13:02
27-09-2025

Sharing Settings

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to Background Jobs to monitor the progress of a change to an organization-wide default or a parallel sharing recalculation.

Manage sharing settings for: All Objects

Default Sharing Settings

Organization-Wide Defaults

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
	Public Full Access	Private	✓

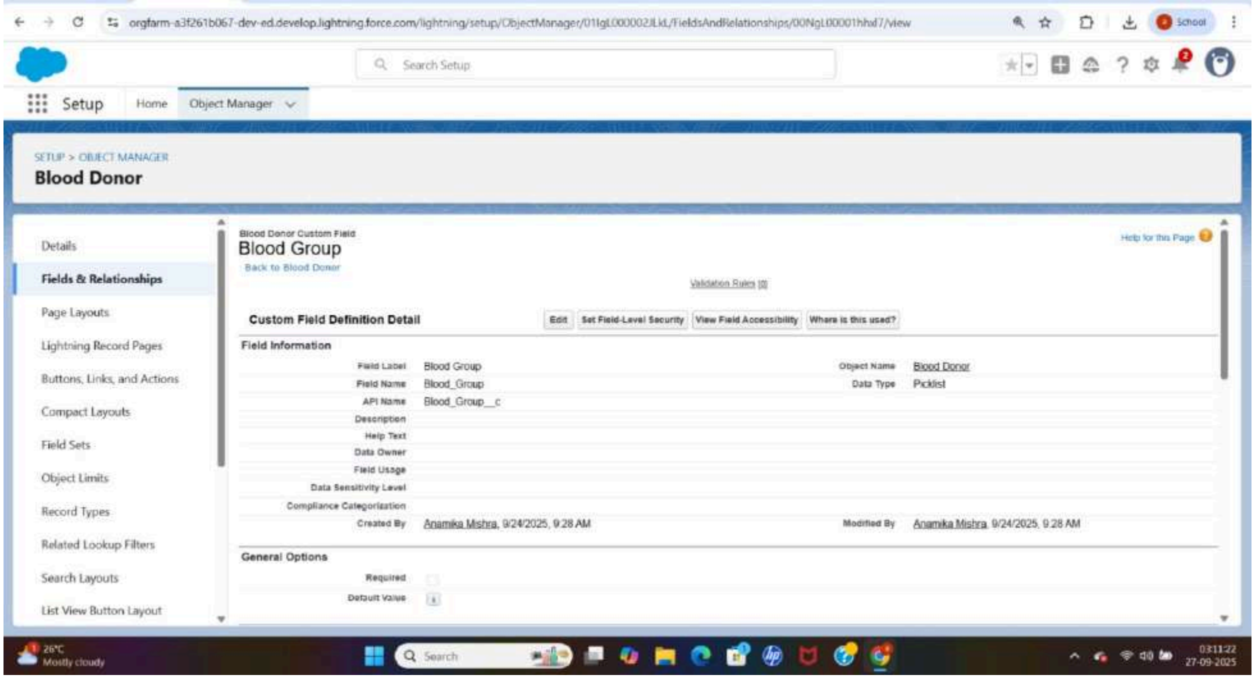
Step 4.Field Level Security

Purpose

Control visibility and editability of individual fields in Salesforce objects.
Protect sensitive data while allowing access to necessary information.
Apply security at the profile or permission set level.

Steps (Process)

Go to Setup → Search Field Accessibility or Profiles.
Select Object → Choose the object containing the field.
Select Field → Click on the field to edit security.
Set Access Levels →
Visible → User can see the field.
Read-Only → User can view but not edit.
Hidden → Field is not visible.
Save Changes → Apply settings for profiles or permission sets.
Test Access → Log in as user to confirm field visibility.



Custom Field Definition Detail

Field Information

Field Label	Blood Group	Object Name	Blood Donor
Field Name	Blood_Group	Data Type	Picklist
API Name	Blood_Group__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Anamika Mishra, 9/24/2025, 9:28 AM	Modified By	Anamika Mishra, 9/24/2025, 9:28 AM

General Options

Required □
Default Value

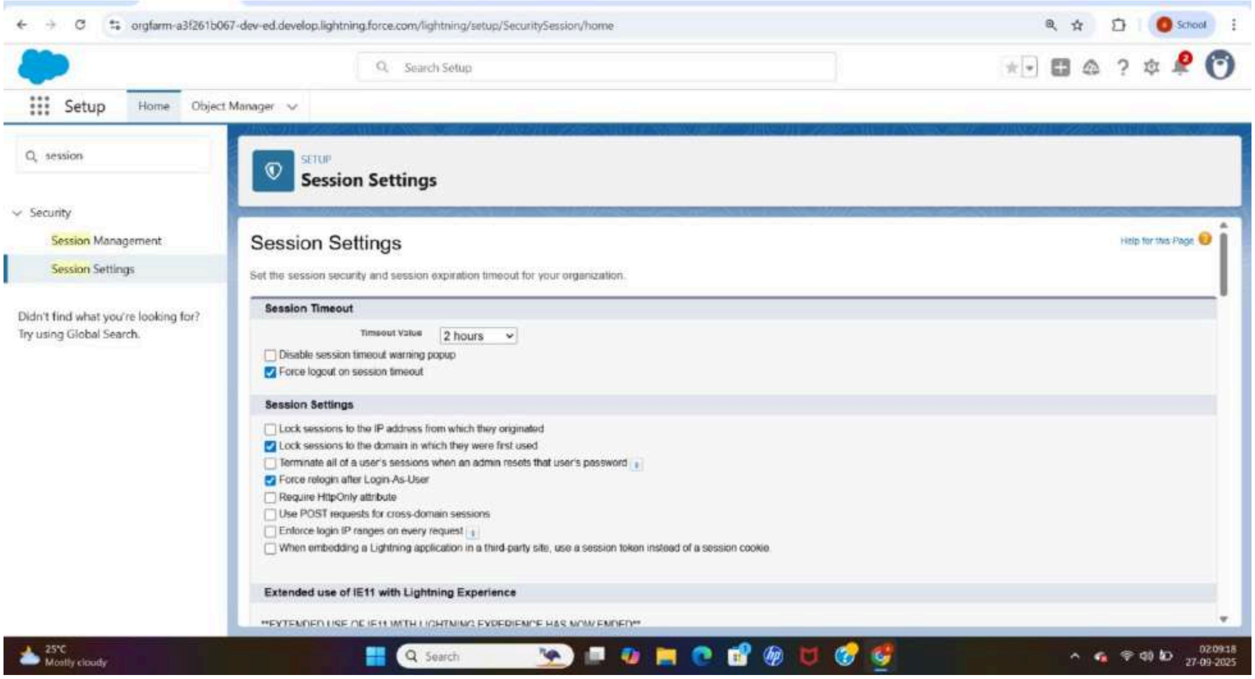
Step 5. Session Settings

Purpose

Control user session behavior in Salesforce for security and performance.
Define session timeout, security policies, and login access.
Protect org data by managing how and when sessions expire.

Steps (Process)

Go to Setup → Search Session Settings.
Set Session Timeout → Choose duration (e.g., 30 min, 2 hours).
Configure Security Policies → Enable/disable secure session settings (e.g., IPranges, login hours).
Enable/Disable Persistent Sessions → Decide whether users stay logged in across browser restarts.
Save Changes → Apply settings.
Test → Log in as a user to verify session behavior.


orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/SecuritySession/home
Search Setup
Setup
Home
Object Manager
session
Security
Session Management
Session Settings
Didn't find what you're looking for? Try using Global Search.
SETUP
Help for this Page

Session Settings

Set the session security and session expiration timeout for your organization.

Session Timeout

Timeout Value: 2 hours

□ Disable session timeout warning popup
☑ Force logout on session timeout

Session Settings

□ Lock sessions to the IP address from which they originated
☑ Lock sessions to the domain in which they were first used
□ Terminate all of a user's sessions when an admin resets that user's password
☑ Force relogin after Login-As-User
□ Require HttpOnly attribute
□ Use POST requests for cross-domain sessions
□ Enforce login IP ranges on every request
□ When embedding a Lightning application in a third-party site, use a session token instead of a session cookie

Extended use of IE11 with Lightning Experience

EXTENDED USE OF IE11 WITH LIGHTNING EXPERIENCE HAS NOW ENDED

Step 6 . Login IP Ranges

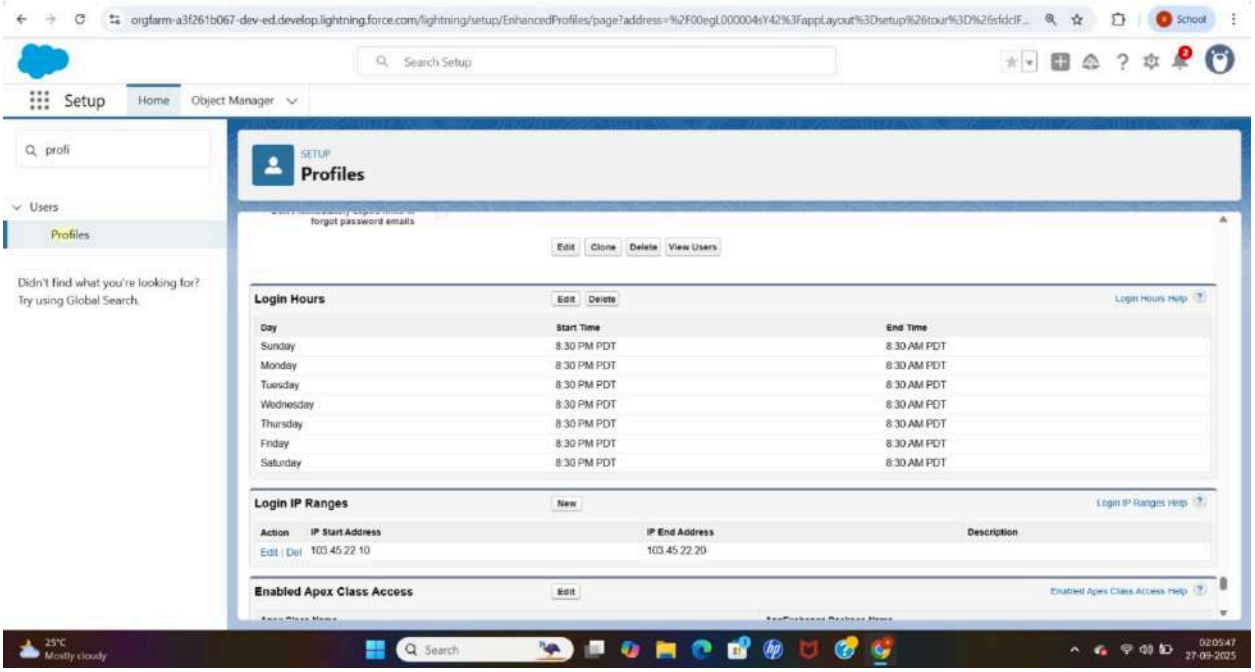
Purpose

Restrict Salesforce login access to specific IP address ranges.
Enhance org security by controlling where users can log in from.
Useful for enforcing network-based access control.

Steps (Process)

Go to Setup → Search Profiles (IP ranges are set per profile).
Select Profile → Click the profile you want to set IP ranges for.

Find Login IP Ranges Section → Click New.
Enter IP Range → Start IP and End IP addresses (e.g., 103.45.22.10 to 103.45.22.20).
Save Changes → Restriction applies immediately.



Login Hours

Day	Start Time	End Time
Sunday	8:30 PM PDT	8:30 AM PDT
Monday	8:30 PM PDT	8:30 AM PDT
Tuesday	8:30 PM PDT	8:30 AM PDT
Wednesday	8:30 PM PDT	8:30 AM PDT
Thursday	8:30 PM PDT	8:30 AM PDT
Friday	8:30 PM PDT	8:30 AM PDT
Saturday	8:30 PM PDT	8:30 AM PDT

Login IP Ranges

Action	IP Start Address	IP End Address	Description
Edit | Del	103.45.22.10	103.45.22.20	

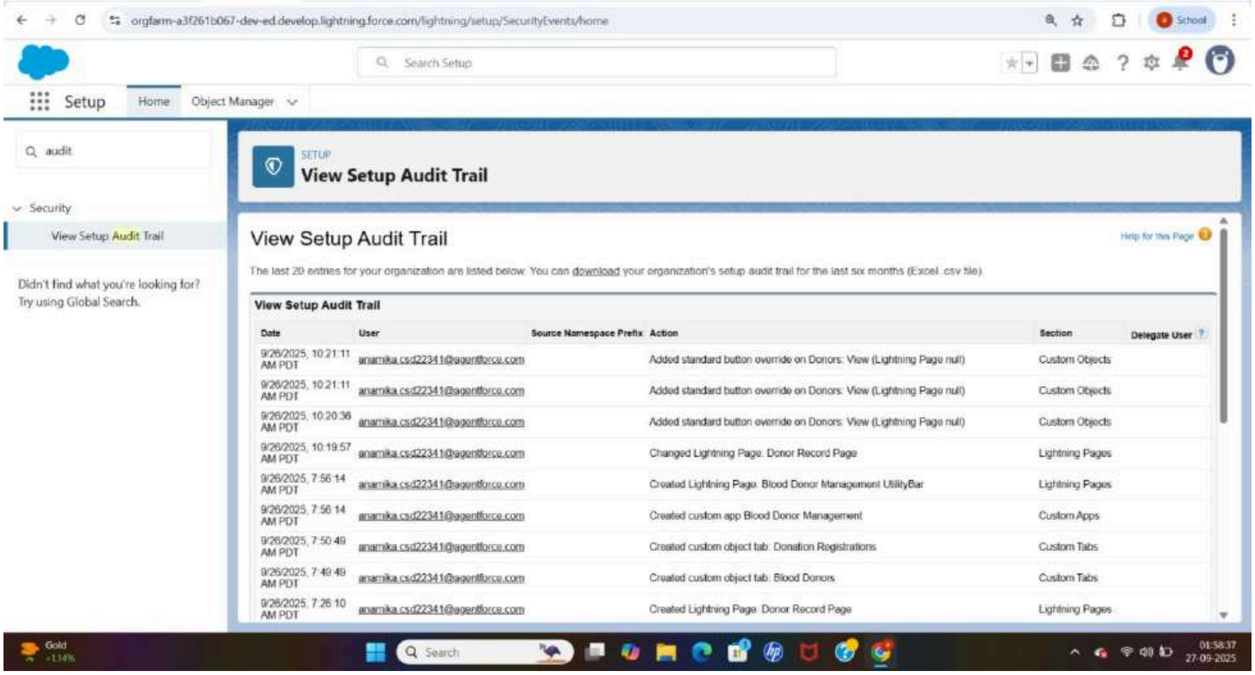
Step 7. Audit Trail

Purpose

Track administrative changes for accountability.
Support compliance and security audits.
Help troubleshoot unexpected behavior by reviewing configuration changes.

Steps (Process)

Go to Setup → Search View Setup Audit Trail.
Review Recent Changes → See changes for the last 180 days.
Download Audit Log → Export for analysis or compliance documentation.
Filter by Date/User → Focus on specific changes.



orgfarm-a3f261b067-dev-ed.develop.lightning.force.com/lightning/setup/SecurityEvents/home
Search Setup
Setup
Home
Object Manager
audit
Security
View Setup Audit Trail
Didn't find what you're looking for? Try using Global Search.
SETUP
View Setup Audit Trail
Help for this Page
Search

View Setup Audit Trail

The last 20 entries for your organization are listed below. You can download your organization's setup audit trail for the last six months (Excel .csv file).

View Setup Audit Trail

Date	User	Source Namespace Prefix	Action	Section	Delegate User
9/26/2025, 10:21:11 AM PDT	anamika.csd22341@agentforce.com		Added standard button override on Donors: View (Lightning Page null)	Custom Objects	
9/26/2025, 10:21:11 AM PDT	anamika.csd22341@agentforce.com		Added standard button override on Donors: View (Lightning Page null)	Custom Objects	
9/26/2025, 10:20:36 AM PDT	anamika.csd22341@agentforce.com		Added standard button override on Donors: View (Lightning Page null)	Custom Objects	
9/26/2025, 10:19:57 AM PDT	anamika.csd22341@agentforce.com		Changed Lightning Page: Donor Record Page	Lightning Pages	
9/26/2025, 7:56:14 AM PDT	anamika.csd22341@agentforce.com		Created Lightning Page: Blood Donor Management UtilityBar	Lightning Pages	
9/26/2025, 7:56:14 AM PDT	anamika.csd22341@agentforce.com		Created custom app Blood Donor Management	Custom Apps	
9/26/2025, 7:50:49 AM PDT	anamika.csd22341@agentforce.com		Created custom object tab: Donation Registrations	Custom Tabs	
9/26/2025, 7:49:49 AM PDT	anamika.csd22341@agentforce.com		Created custom object tab: Blood Donors	Custom Tabs	
9/26/2025, 7:26:10 AM PDT	anamika.csd22341@agentforce.com		Created Lightning Page: Donor Record Page	Lightning Pages	

Phase 10. Final Presentation & Demo Day

1. Pitch Presentation

Purpose: Introduce the project, its value, and impact.

Problem Statement: Challenges in blood donation and management.

Solution: How the Blood Donor Management System addresses these challenges.

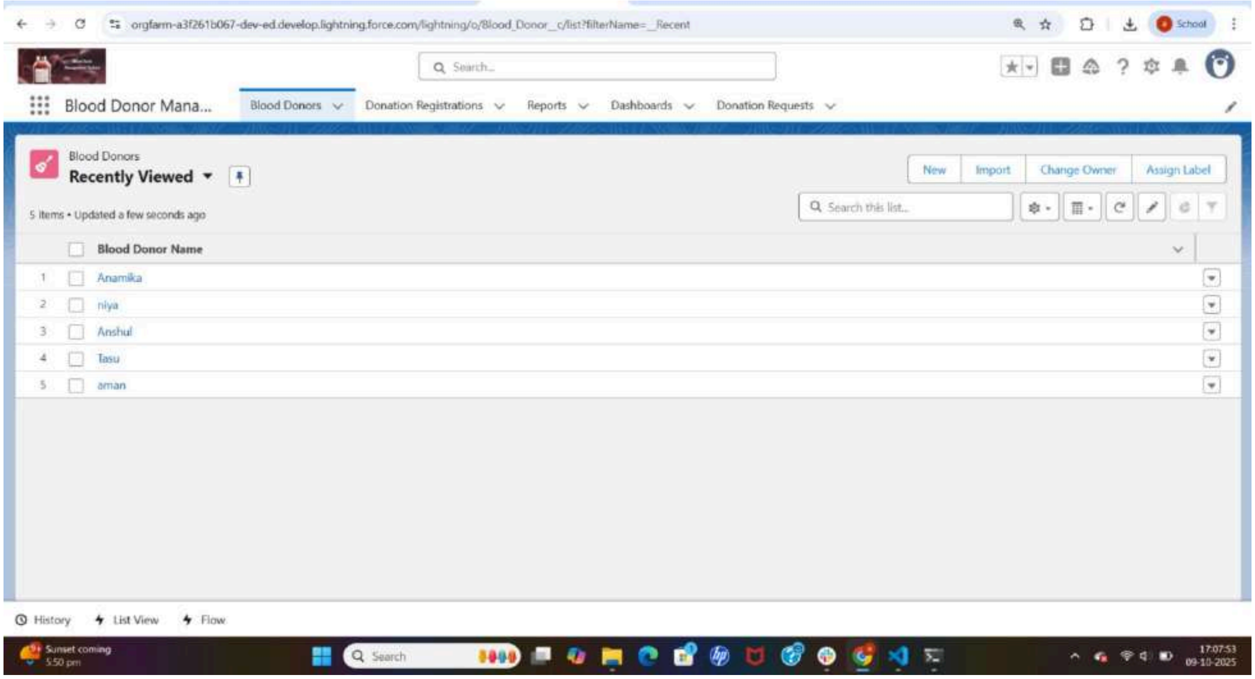
Impact: Efficiency, timely blood availability, better donor engagement.

2. Demo Walkthrough

Purpose: Showcase the project in action.

Steps:

1. Login & Home Page overview.
2. Donor registration process.
3. Blood inventory & donation tracking.
4. Reports & dashboards highlights.
5. Any automation (notifications, alerts).



Blood Donors

Recently Viewed

5 items • Updated a few seconds ago

	Blood Donor Name
1	Anamika
2	niya
3	Anshul
4	Tasu
5	aman

3. Feedback Collection

Purpose: Gather suggestions for improvement.

Methods:

Online forms (Google Forms, SurveyMonkey).

Verbal feedback during demo.

Focus Areas: Usability, completeness, design, functionality, real-world applicability.

4. Handoff Documentation

Purpose: Enable smooth continuation or deployment by other users or developers.

Content:

System overview and architecture.

Salesforce objects, fields, and relationships.

Reports, dashboards, and automation details.

User manual for donor registration, reporting, and notifications.

Deployment or setup instructions.

5. LinkedIn/Portfolio Project Showcase

Project Title: **Blood Donor Management System**

Summary: **Salesforce application for managing donors, blood requests, and inventory efficiently.**

Key Features:

Donor registration and management

Blood request submission using Imperative Apex

Real-time donor list updates with Wire Adapters

Blood inventory tracking

Lightning App Builder, Tabs, Home Page Layouts, and Utility Bar integration

Conclusion

The Blood Donor Management System built on Salesforce provides a comprehensive solution to streamline blood donation processes. By centralizing donor information, tracking donations, managing blood inventory, and enabling automated notifications, the system improves efficiency, reduces errors, and ensures timely availability of blood for patients in need.

The project demonstrates how Salesforce's powerful features—like Lightning Web Components, Apex, Reports, and Dashboards—can be leveraged to create a user-friendly, scalable, and impactful solution. Overall, this system not only enhances hospital operations but also fosters better engagement with donors, ultimately contributing to saving lives.