

Organ Donation and Transplant Registry

Phase 1: Problem Understanding & Industry Analysis

1. Requirement Gathering

The organ donation and transplant process is often delayed due to the lack of a centralized digital platform that connects donors and recipients efficiently. Hospitals currently rely on manual data tracking, leading to errors, mismatches, and slow response times.

Identified Requirements:

- A unified system to store Donor and Recipient information.
 - Automatic matching of donors and recipients based on blood group and organ type.
 - Ability for administrators to monitor and manage the organ transplant workflow.
 - Automation to send notifications when a suitable donor is found.
 - Reports and dashboards for tracking donor statistics and transplant success rates.
-

2. Stakeholder Analysis

Stakeholder	Role / Responsibility
Hospital Admin	Manages all donor and recipient records, approves matches.
Doctor	Reviews donor-recipient compatibility and confirms organ availability.
Donor	Provides consent and information about available organs.
Recipient	Submits organ request and awaits matching.
System (Salesforce)	Automatically matches donor and recipient based on conditions defined in Flows or Triggers.

3. Business Process Mapping

The business process for the Organ Donation & Transplant Management System follows these key stages:

1. Donor Registration – Donor submits details such as name, organ available, blood group, and consent.
2. Recipient Registration – Recipient enters required organ and blood group.
3. Matching Process – The system automatically checks for compatible donors using Salesforce Flow.

4. Verification – Doctor verifies medical compatibility and approves the transplant.
5. Transplant Confirmation – Admin updates the status to “Transplanted” and generates reports.

 *This process reduces manual effort, improves match accuracy, and saves critical time for patients in need.*

4. Industry-Specific Use Case Analysis

Healthcare	Sector	Use	Case:
Organ donation management systems	are increasingly being adopted by hospitals and health authorities to manage donors, streamline organ allocation, and ensure ethical transparency.		
However, most existing systems are offline or semi-automated.			

Salesforce CRM offers a scalable, secure, and cloud-based solution that enables:

- Automation through Flows and Process Builders.
- Transparent tracking using Reports and Dashboards.
- Secure access through Profiles and Permission Sets.

This project bridges the gap by creating an end-to-end organ donation management system leveraging Salesforce’s low-code capabilities.

5. AppExchange Exploration

Before implementation, a search was conducted on Salesforce AppExchange for existing healthcare or organ management solutions. Findings showed that:

- Most healthcare apps focus on patient management or appointment scheduling.
- No existing app offered automated donor-recipient matching for organ donation.

Hence, this project introduces a custom-built Salesforce application specifically designed for organ donation and transplant coordination.

Phase 2: Org Setup & Configuration

Objective:

Configure the Salesforce org to enable development of a donor-recipient matching system with proper user access, security, and environment setup.

Key Configurations Implemented:

1. **Salesforce Edition:** Developer Edition used to access full customization.
2. **Company Profile Setup:** Company name, time zone, locale, and currency set to match hospital location.

3. **User Setup & Licenses:** Users created with profiles; system administrators and standard users assigned appropriate access.
4. **Login Access Policies:** Login hours and multi-factor authentication enabled for secure access.
5. **Developer Org Setup:** Prepared for creating custom objects, flows, and reports.
6. **Sandbox Usage:** All development done in the dev org; sandbox optional for future testing.
7. **Deployment Basics:** Configured for easy migration of objects, fields, flows, and reports.

The screenshot shows the Salesforce Setup interface under the Company Information section. The organization name is set to 'KSRM'. The primary contact is 'OrgFarm EPIC'. The fiscal year starts in January. The currency locale is set to English (United States) - USD. The default time zone is (GMT+05:30) India Standard Time (Asia/Kolkata). The organization ID is 00Dg5000000yQjc. The developer edition instance is IND168. The page also shows links for User Licenses, Permission Set Licenses, Feature Licenses, and Usage-based Entitlements.

Phase 3: Data Modeling & Relationships

Objective:

Design custom objects, fields, and relationships for donor and recipient data management.

Custom Objects Created:

1. **Donor** – stores donor details such as name, blood group, organs available, consent, city.
2. **Recipient** – stores recipient information like name, required organ, blood group, and city.
3. **Organ Request** – stores donor-recipient matching requests, status, and matched donor.

The image displays three separate screenshots of the Salesforce Object Manager setup screen, each showing the configuration for a different object: Donor, Recipient, and Organ Request.

Donor Object Setup:

- API Name:** Donor_c
- Custom:** ✓
- Singular Label:** Donor
- Plural Label:** Donors
- Description:** (empty)
- Enable Reports:** (unchecked)
- Track Activities:** (unchecked)
- Track Field History:** (unchecked)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

Recipient Object Setup:

- API Name:** Recipient_c
- Custom:** ✓
- Singular Label:** Recipient
- Plural Label:** Recipients
- Description:** (empty)
- Enable Reports:** (unchecked)
- Track Activities:** (unchecked)
- Track Field History:** (unchecked)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

Organ Request Object Setup:

- API Name:** Organ_Request_c
- Custom:** ✓
- Singular Label:** Organ Request
- Plural Label:** Organ Requests
- Description:** (empty)
- Enable Reports:** ✓
- Track Activities:** (unchecked)
- Track Field History:** (unchecked)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

Fields and Relationships:

- Lookup relationships between Organ Request and Donor & Recipient

- picklist for organs available in Donor object
- Status field for Organ Request to indicate progress (New, Matched, Completed)

Donor

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Group	Blood_Group__c	Picklist		
City	City__c	Picklist		
Consent	Consent__c	Checkbox		
Created By	CreatedById	Lookup(User)		
Donor Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Organ Available	Organ_Available__c	Picklist		
Owner	OwnerId	Lookup(User,Group)		✓

Recipient

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Group	Blood_Group__c	Picklist		
City	City__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Recipient Name	Name	Text(80)		✓
Required Organ	Required_Organ__c	Picklist		

Organ Request

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Matched Donor	Matched_Donor__c	Lookup(Donor)		✓
Organ Request Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Recipient	Recipient__c	Lookup(Recipient)		✓
Required Organ	Required_Organ__c	Picklist		
Status	Status__c	Picklist		

Page Layouts:

- Custom page layouts created for Donor, Recipient, and Organ Request for user-friendly entry and viewing.

Donor

Page Layouts		
PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Donor Layout	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:42 AM	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:53 AM

Recipient

Page Layouts		
PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Recipient Layout	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:46 AM	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:50 AM

Organ Request

Page Layouts		
PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Organ Request Layout	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:49 AM	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:53 AM

Schema Builder:

- Visual representation of relationships between Donor, Recipient, and Organ Request objects.

Phase 4: Process Automation (Admin)

Objective:

The goal of this phase is to reduce manual effort, prevent errors, and accelerate the organ matching process by using Salesforce automation tools. All steps aim to ensure that donor-recipient matching happens automatically based on predefined rules.

1. Validation Rules

Use Case:

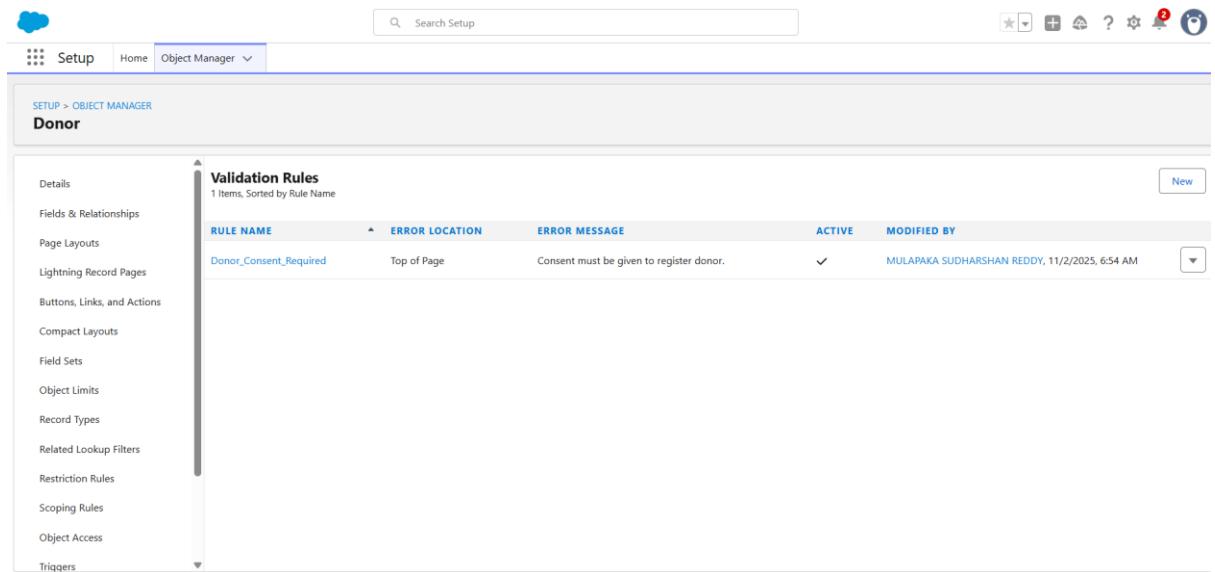
Before a donor can be considered for matching, they must have explicitly given consent. This prevents unauthorized or incomplete records from affecting the matching process.

Implementation:

- **Object:** Donor
- **Field Validated:** Consent__c (Checkbox)
- **Rule Logic:** Donor record cannot be marked “Available” unless Consent__c = True
- **Error Message:** “Donor consent must be provided before they can be matched.”

Benefits:

- Prevents matching of donors who have not given consent
- Maintains compliance with ethical and legal standards

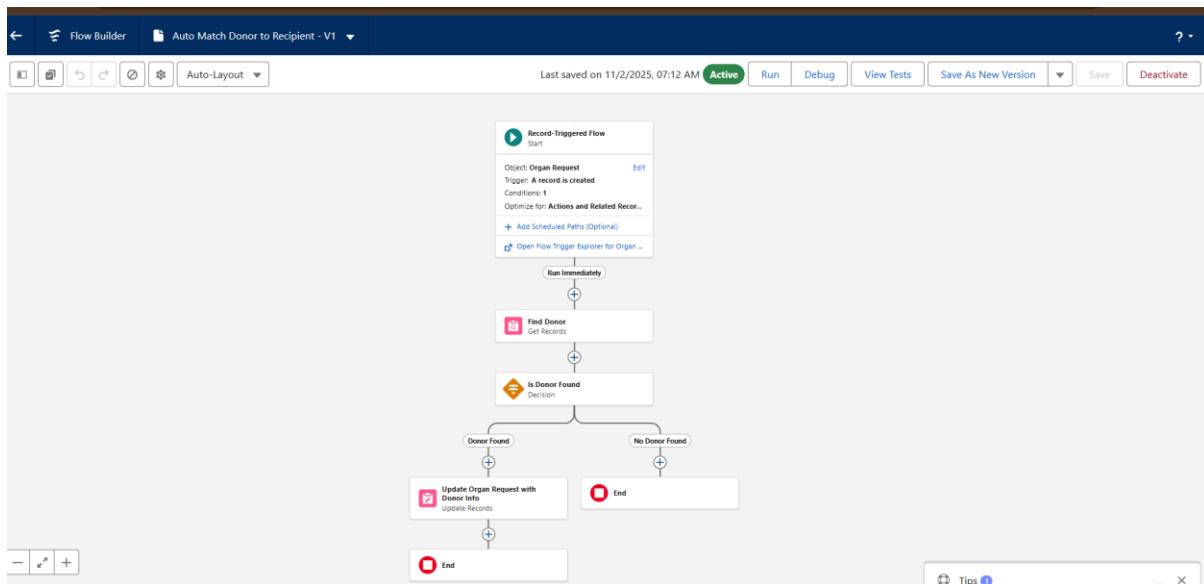


The screenshot shows the Salesforce Object Manager for the 'Donor' object. The left sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main content area is titled 'Validation Rules' and shows a single item: 'Donor_Consent_Required'. The table has columns for RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The rule details are: Rule Name: 'Donor_Consent_Required', Error Location: 'Top of Page', Error Message: 'Consent must be given to register donor.', Active status: checked, and Modified By: 'MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:54 AM'.

2. Record-Triggered Flow (Auto-Launched)

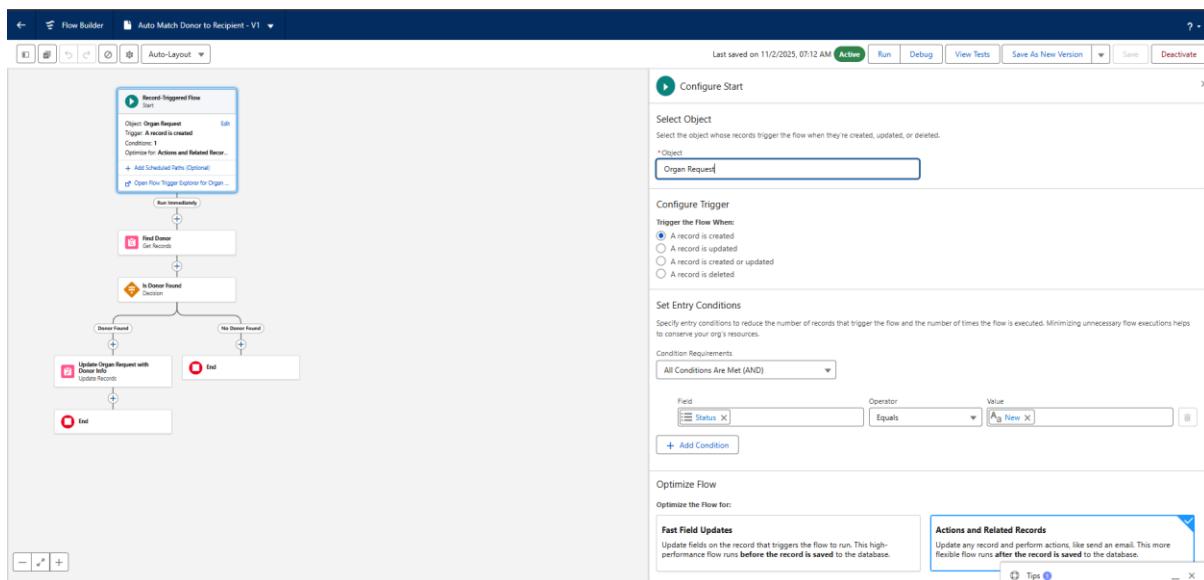
Use Case:

When a new Organ Request is created, the system automatically finds a matching donor without manual intervention.



Flow Details:

- Trigger Object:** Organ Request
- Trigger Event:** When a record is created
- Optimization:** Actions and Related Records (runs after record is saved)



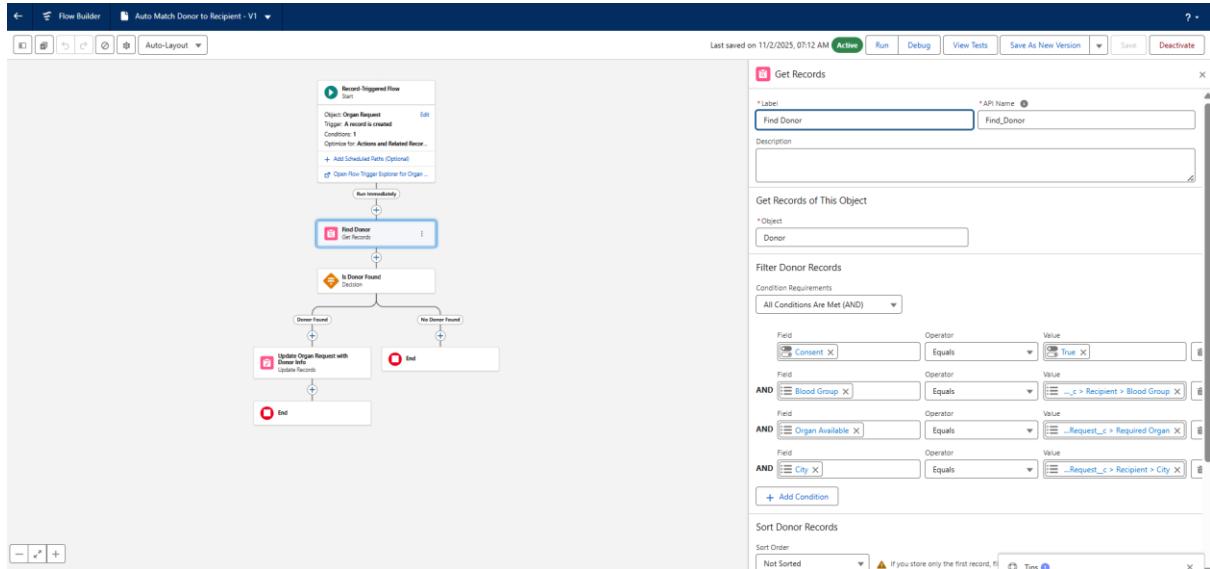
• Steps:

1. Get Records – Find Donor:

- Searches the **Donor** object for records where:

- `Consent__c = True`

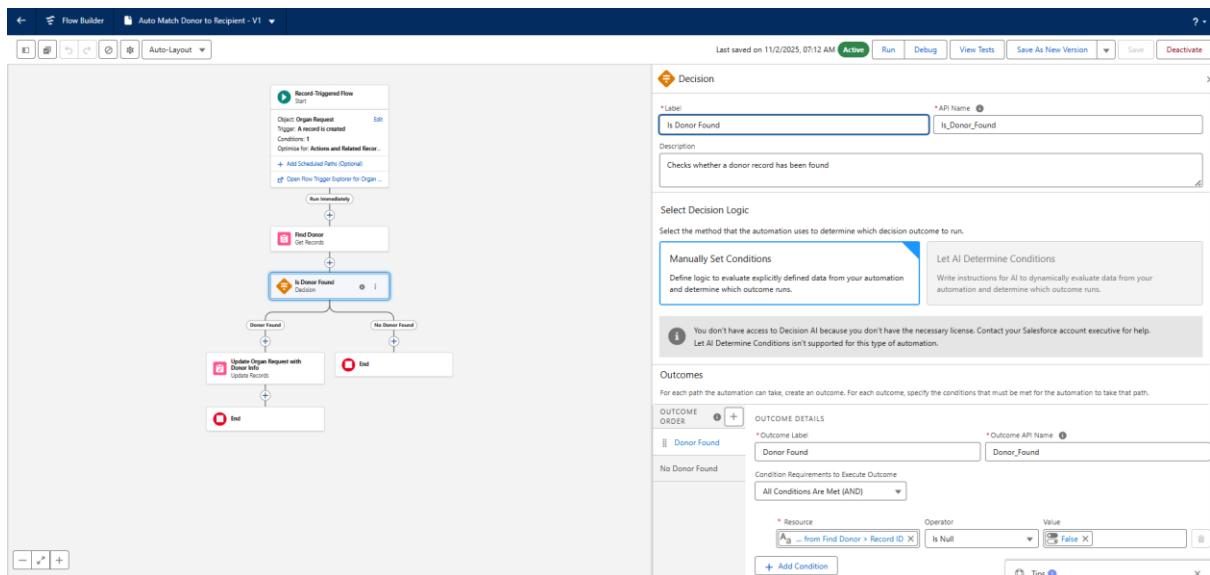
- Blood Group matches Recipient's Blood Group
- Organs Available includes Required Organ (using multi-select picklist)
- City matches Recipient's City



2. Decision Element – Is Donor Found?

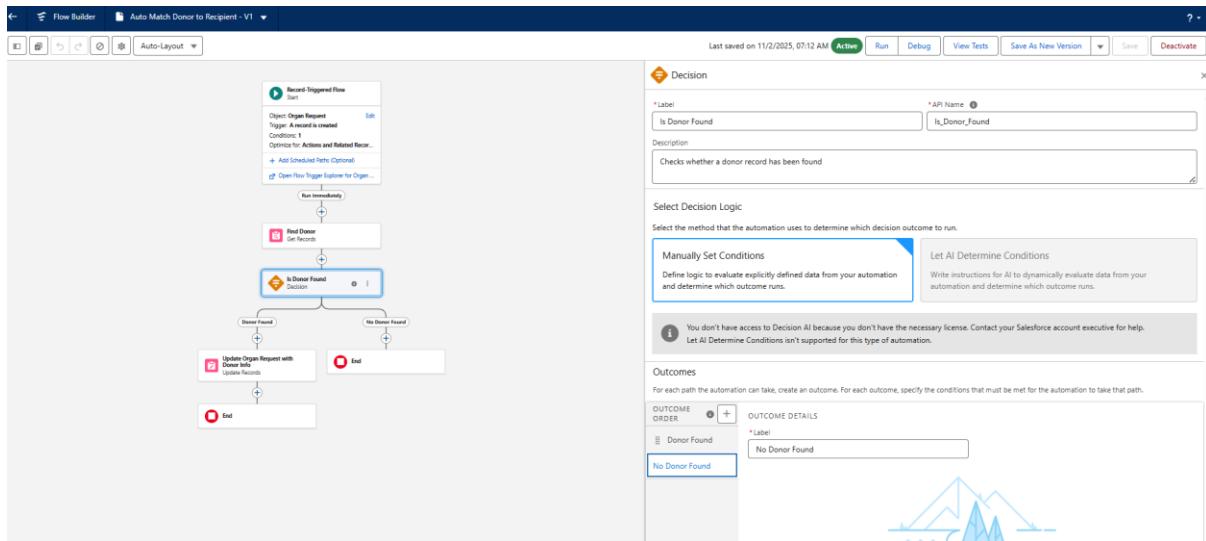
- **Outcome 1 (Donor Found):**

- Condition: `{!Find_Donor.Id}` Is Null = False
- Proceeds to update Organ Request with donor info



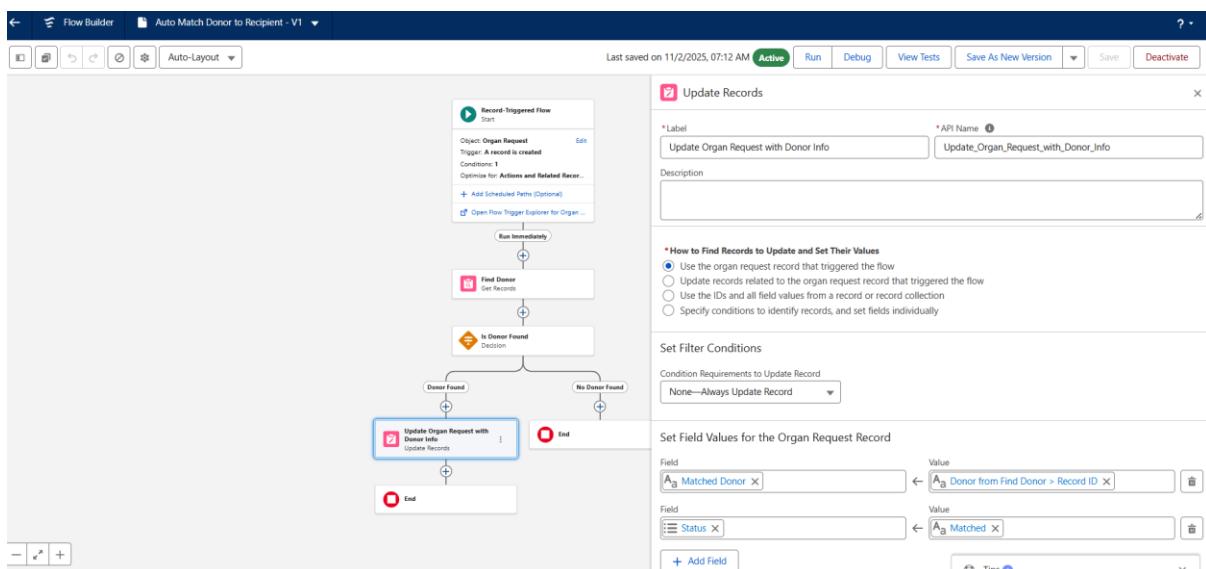
- **Outcome 2 (No Donor Found – Default):**

- No matching donor available
- Status remains "New" or optional notification sent



3. Update Records – Update Organ Request:

- Fields Updated:
 - **Matched_Donor__c = {!Find_Donor.Id}**
 - **Status__c = "Matched"**



Benefits:

- Eliminates manual matching and errors
- Real-time status updates
- Ensures proper donor-recipient pairing based on all criteria

Phase 5: Apex Programming (Developer)

Objective:

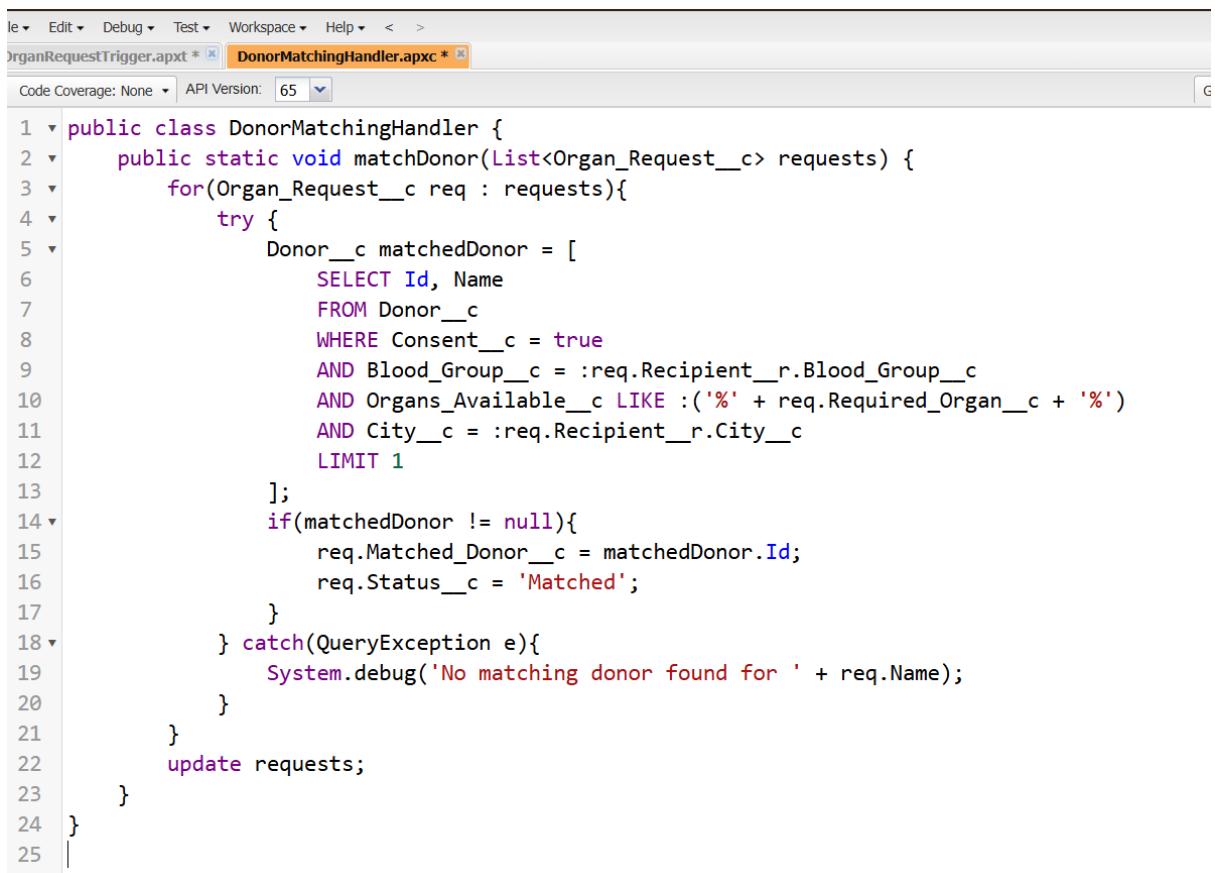
Automate donor-recipient matching programmatically, complementing Flow automation, and ensure real-time updates on Organ Request records when a new request is created.

1 Apex Classes & Objects

- **DonorMatchingHandler:**

A handler class to encapsulate the logic for finding a matching donor based on blood group, required organ, consent, and city.
Ensures best practices by separating logic from triggers (Trigger Handler Pattern).

- Updates the Matched_Donor__c field on Organ_Request__c.
- Updates the Status__c field to "Matched" when a donor is found.



The screenshot shows the Salesforce IDE interface with the code editor open. The title bar indicates the file is 'DonorMatchingHandler.apxc'. The code itself is a Apex class named 'DonorMatchingHandler' with a static method 'matchDonor'. This method takes a list of 'Organ_Request__c' objects and iterates through them. For each request, it performs a query to find a matching donor based on blood group, required organ, and city. If a donor is found, it updates the request's 'Matched_Donor__c' and 'Status__c' fields. If no donor is found, it logs a debug message. Finally, it updates the list of requests.

```
1 public class DonorMatchingHandler {  
2     public static void matchDonor(List<Organ_Request__c> requests) {  
3         for(Organ_Request__c req : requests){  
4             try {  
5                 Donor__c matchedDonor = [  
6                     SELECT Id, Name  
7                     FROM Donor__c  
8                     WHERE Consent__c = true  
9                     AND Blood_Group__c = :req.Recipient__r.Blood_Group__c  
10                    AND Organs_Available__c LIKE :('%' + req.Required_Organ__c + '%')  
11                    AND City__c = :req.Recipient__r.City__c  
12                    LIMIT 1  
13                ];  
14                if(matchedDonor != null){  
15                    req.Matched_Donor__c = matchedDonor.Id;  
16                    req.Status__c = 'Matched';  
17                }  
18            } catch(QueryException e){  
19                System.debug('No matching donor found for ' + req.Name);  
20            }  
21        }  
22        update requests;  
23    }  
24 }  
25 |
```

2 Apex Triggers

- **OrganRequestTrigger:**

Trigger runs **after insert** on Organ_Request__c.
Calls DonorMatchingHandler.matchDonor() to automatically assign a donor and update the status.

Trigger Type: After Insert

Purpose: Automate donor assignment without manual intervention.

```
trigger OrganRequestTrigger on Organ_Request__c (after insert) {
    DonorMatchingHandler.matchDonor(Trigger.new);
}
```

3 SOQL & SOSL Queries

- Used in the handler class to find **matching donors**.
- SOQL query filters on:
 - Consent__c = true
 - Blood_Group__c matches recipient
 - Organs_Available__c contains required organ
 - City__c matches recipient city

Example:

```
Donor__c matchedDonor = [
    SELECT Id, Name
    FROM Donor__c
    WHERE Consent__c = true
    AND Blood_Group__c = :req.Recipient__r.Blood_Group__c
    AND Organs_Available__c LIKE :('%' + req.Required_Organ__c + '%')
    AND City__c = :req.Recipient__r.City__c
    LIMIT 1
];
```

4 Collections: List, Set, Map

- **List<Organ_Request__c> requestsToUpdate:** Stores Organ Request records to be updated in bulk.
 - Bulk-safe design ensures that multiple Organ Requests can be processed efficiently in a single transaction.
-

5 Exception Handling

- Wraps SOQL query in **try-catch** to handle cases where no donor is found.
- Prevents runtime errors and allows flow to continue.

Example:

```
try {  
    // Query for donor  
}  
} catch(QueryException e) {  
    System.debug('No matching donor found for ' + req.Name);  
}
```

6 Test Class

- **DonorMatchingHandlerTest:** Ensures 100% code coverage.
- Inserts sample **recipient**, **donor**, and **organ request** records.
- Validates:
 - Status__c updated to "Matched"
 - Matched_Donor__c populated with the correct donor

Phase 6: User Interface Development (Detailed)

Objective

Create a user-friendly interface for managing Donors, Recipients, and Organ Requests. The UI should allow hospital staff or admins to:

- View and manage records
 - See matched donor information automatically
 - Navigate easily between objects and reports
-

Components Implemented in Your Project

1 Lightning App Builder

- **Use Case:** To build a custom Salesforce app for Organ Donation & Transplant Management.
- **What it does:** Combines tabs, record pages, and custom components in one interface for users.
- **Steps to View:**
 1. Go to **Setup → Lightning App Builder**.
 2. Find your app (likely named **Organ Donation & Transplant Management**).
 3. Click **Edit** → see the app canvas showing all included pages.

The screenshot shows the Lightning App Builder interface with the following details:

App Details & Branding:

- App Name:** Organ Donation & Transplant Management
- Developer Name:** Organ_Donation_Transplant_Management
- Description:** Enter a description...
- Image:** A placeholder image for the app icon, featuring a kidney and a heart.
- Primary Color Hex Value:** #000000
- Org Theme Options:** Use the app's image and color instead of the org's custom theme (unchecked)

App Launcher Preview:

App Options:

- Navigation and Form Factor:**
 - Navigation Style:** Standard navigation (selected)
 - Console navigation
- Supported Form Factors:**
 - Desktop and phone (selected)
 - Desktop
 - Phone
- Setup and Personalization:**
 - Setup Experience:** Setup (full set of Setup options) (selected)
 - Service Setup
 - Data Cloud Setup
- App Personalization Settings:**
 - Disable end user personalization of nav items in this app
 - Disable temporary tabs for items outside of this app
 - Use Omni-Channel sidebar

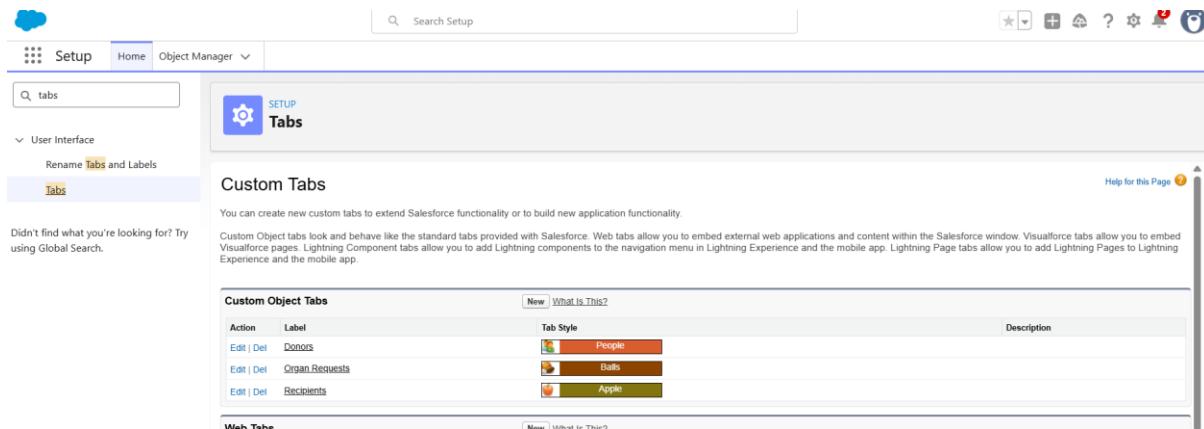
2 Record Pages

- Use Case:** Display fields and related lists for Donor, Recipient, and Organ Request records.
- What it does:** Shows matched donor, recipient details, and status updates on a single record page.
- Steps to View:**
 1. **Setup → Object Manager → Organ Request → Lightning Record Pages**
 2. Click **Edit** on the default record page.
 3. Check placement of fields like:
 - Recipient

- Required Organ
 - Status
 - Matched Donor
-

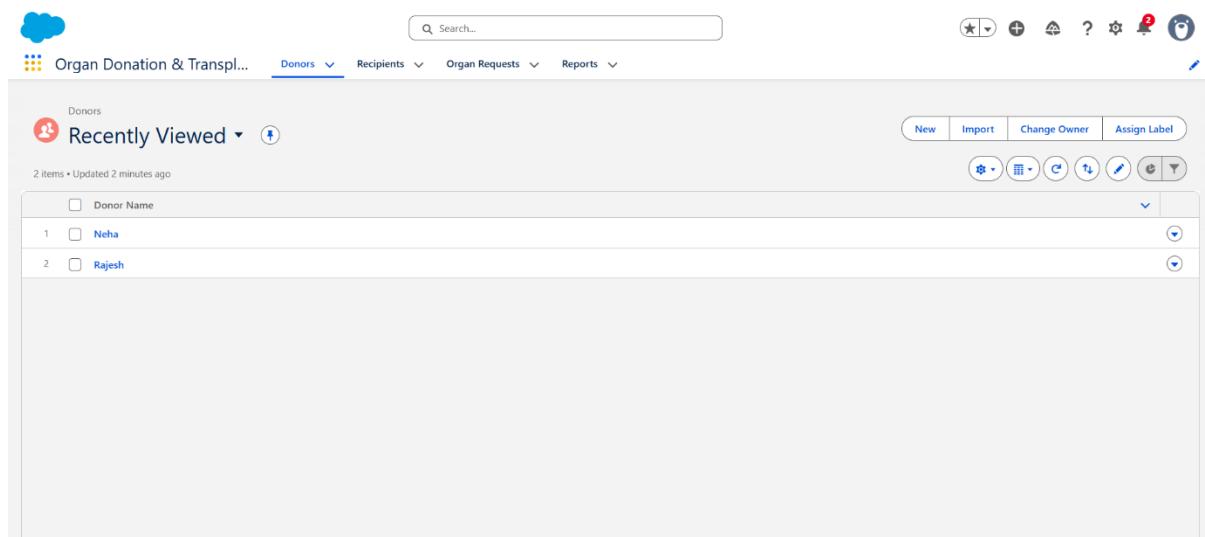
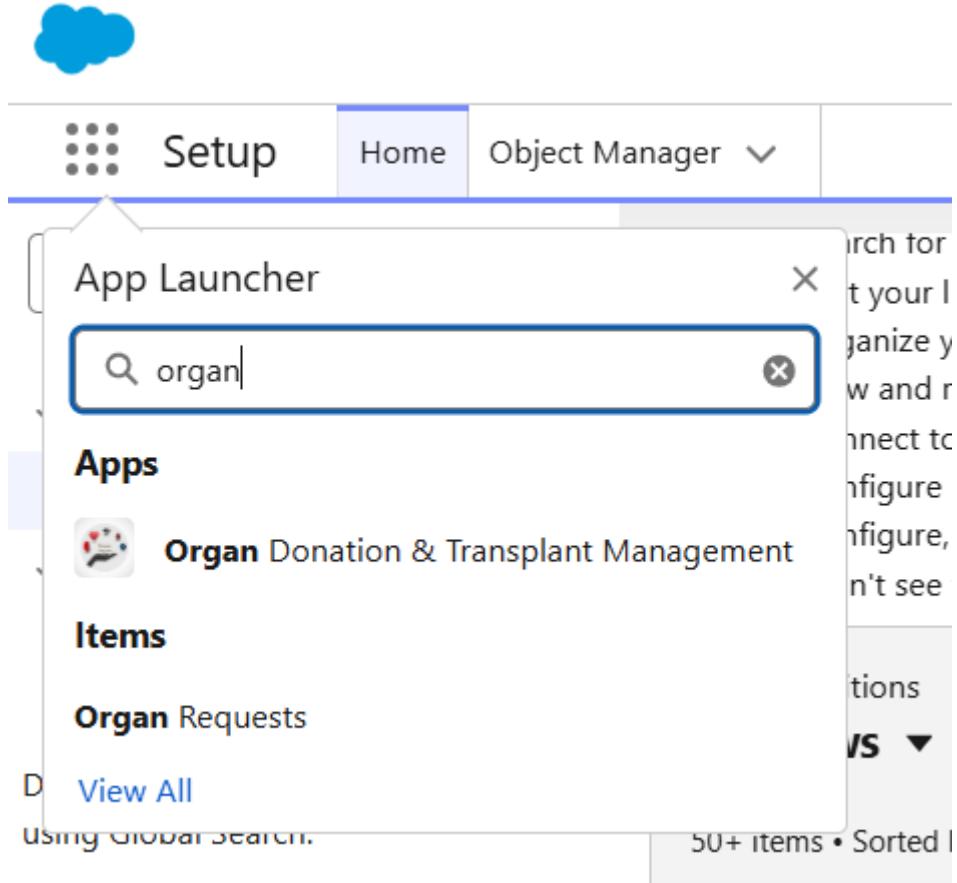
3 Tabs

- **Use Case:** Navigate easily between different objects.
- **What it does:** Provides access to **Donors, Recipients, and Organ Requests** in the same app.
- **Steps to View:**
 1. Click **App Launcher** ( icon) → **View All** → **Organ Donation & Transplant Management**
 2. Check visible tabs.
 3. Ensure tabs match the objects in your project.



The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The left sidebar shows 'User Interface' with 'Tabs' highlighted. The main area is titled 'Custom Tabs' and contains a sub-section 'Custom Object Tabs'. It lists three tabs: 'People' (red), 'Balls' (brown), and 'Apple' (green). Each tab has an 'Edit | Del' link next to its name. The 'People' tab is currently selected. A note at the bottom says: '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.' There is also a 'Help for this Page' link.

Action	Label	Tab Style	Description
Edit Del	Donors	 People	
Edit Del	Organ Requests	 Balls	
Edit Del	Recipients	 Apple	



Phase 7: Integration & External Access

Remote Site Settings

- **Use Case:** Configured to allow Salesforce to securely communicate with external systems if needed. This ensures that any future callouts to external donor databases or hospital APIs are authenticated and allowed by Salesforce security policies.

- **Implementation:** The external endpoints are added to Remote Site Settings to enable safe and authorized HTTP callouts.

Phase 8: Data Management & Deployment

Objective:

Manage donor, recipient, and organ request data efficiently while ensuring smooth deployment from development to production environments.

1. Data Import Wizard

- **Use Case:**
Used to import sample or bulk data for **Donor**, **Recipient**, and **Organ Request** objects into Salesforce without writing code.
 - **Implementation:**
 1. Navigate to Setup → Data → Data Import Wizard.
 2. Select the object (e.g., Donor).
 3. Upload a CSV file containing donor information such as Name, Blood Group, Organs Available, Consent, and City.
 4. Map the CSV fields to Salesforce fields.
 5. Start the import.
 - **Benefit:** Quickly populates the system for testing flows and reports.
-

2. Data Loader (Optional for Large Data)

- **Use Case:**
For large-scale data import or updates where Data Import Wizard is insufficient. Useful for migrating donor and recipient data from external systems.
 - **Implementation:**
 - Install Salesforce Data Loader.
 - Connect using Salesforce credentials.
 - Insert, Update, or Upsert records for Donor, Recipient, and Organ Request objects using CSV files.
-

3. Duplicate Rules

- **Use Case:**
Prevents creation of duplicate Donor or Recipient records in the system.
- **Implementation:**

1. Setup → Duplicate Management → Duplicate Rules.
 2. Create a rule for Donor object using matching criteria such as Name + Blood Group + City.
 3. Set action: Block or Alert user on duplicate detection.
- **Benefit:** Maintains clean and accurate donor-recipient data for matching flows.
-

4. Data Export & Backup

- **Use Case:**
Ensure backup of all donor-recipient records to prevent data loss.
 - **Implementation:**
 1. Setup → Data → Data Export.
 2. Schedule export weekly or monthly.
 3. Select objects: Donor, Recipient, Organ Request.
 4. Download CSV files for local backup.
-

5. Change Sets

- **Use Case:**
Deploy customizations such as Flows, Validation Rules, and Object fields from Sandbox to Production.
 - **Implementation:**
 1. Setup → Outbound Change Sets in Sandbox.
 2. Add all components (Objects, Flows, Validation Rules).
 3. Upload to Production org.
 4. In Production, deploy the change set.
-

6. VS Code & SFDX (Optional Advanced)

- **Use Case:**
Developers can use Salesforce CLI and Visual Studio Code for version control and advanced deployment.
- **Benefit:** Ideal for maintaining large projects and automated deployments.

Phase 9: Reporting, Dashboards & Security Review

Objective:

Provide a clear overview of donor-recipient matches, organ request status, and ensure data security and access control in the Salesforce system.

1. Reports

- **Use Case:**
Track all organ requests, donor matches, and recipient details efficiently.
- **Implementation:**
 1. Navigate to App Launcher → Reports → New Report.
 2. Select **Report Type:** Organ Requests.
 3. Add Columns: Request Number, Recipient, Required Organ, Status, Matched Donor.
 4. Run and save as **Organ Request Summary**.
- **Benefit:** Provides real-time insight into matched donors and pending organ requests.

The screenshots illustrate the Power BI interface for managing and previewing reports.

Screenshot 1: Reports Management

This screen shows the 'Reports' section under 'Organ Donation & Transpl...'. It displays a table of recent reports, including 'Organ Requests Report' and 'Organ Request with Matched Donor'. The sidebar on the left provides navigation links for Reports, Folders, and Favorites.

Report Name	Description	Folder	Created By	Created On	Subscribed
Organ Requests Report	Public Reports	MULAPAKA SUDHARSHAN REDDY	11/2/2025, 7:41 AM		
Organ Request with Matched Donor	Public Reports	MULAPAKA SUDHARSHAN REDDY	11/2/2025, 7:38 AM		

Screenshot 2: Report Preview

This screen shows the preview of the 'Organ Requests Report'. It displays a table with two rows: 'Liver' and 'Kidney'. The preview indicates that only a limited number of records are shown.

Organ Request: Organ Request Name
1 Liver
2 Kidney

Screenshot 3: Report Results

This screen shows the final report results for the 'Organ Requests Report'. It displays a table with two rows: 'Liver' and 'Kidney'. The columns include 'Organ Request: Organ Request Name', 'Required Organ', 'Recipient', 'Matched Donor', 'Status', and 'Organ Request: Created Date'.

	Organ Request: Organ Request Name	Required Organ	Recipient	Matched Donor	Status	Organ Request: Created Date
1	Liver	Liver	Priya	Neha	Matched	11/2/2025
2	Kidney	Kidney	Suresh	Rajesh	Matched	11/2/2025

2. Dashboards

- Use Case:**
Visual representation of organ donation statistics for administrators.
- Implementation:**
 - Setup → Dashboards → New Dashboard.
 - Add components showing metrics like total requests, matched donors, and requests by city.

- **Benefit:** Quick, visual analysis for decision-making.

The screenshot shows a web-based application interface for organ transplantation management. At the top, there's a navigation bar with links for 'Donors', 'Recipients', 'Organ Requests', 'Reports', and 'Dashboards'. The 'Dashboards' link is currently selected, indicated by an underline. Below the navigation is a search bar and a user profile icon. The main content area is titled 'Organ Donation Dashboard' and shows a timestamp 'As of Nov 2, 2025, 10:46 AM' and a note 'Viewing as MULAPAKA SUDHARSHAN REDDY'. A sub-section titled 'Organ Requests Report' contains a table with the following data:

Organ Request	Organ Request Name	Required	Recipient	Matched	Status
Kidney	Kidney	Suresh	Rajesh	Match	
Liver	Liver	Priya	Neha	Match	

At the bottom of the report section, there are links 'View Report (Organ Requests Report)' and 'As of Nov 2, 2025, 10:46 AM'.

3. Profiles & Roles

- **Use Case:**
Control access to sensitive donor and recipient information.
- **Implementation:**
 1. Profiles define permissions:
 2. Roles define hierarchy for record visibility: e.g., Regional Coordinators can see requests in their city.

4. Permission Sets

- **Use Case:**
Grant additional access to specific users without changing their profile.
- **Implementation:**
 - Example: Temporary access for a volunteer to view organ requests.

5. Organization-Wide Defaults (OWD)

- **Use Case:**
Set baseline record visibility for all objects.
- **Implementation:**
 - Donor, Recipient, and Organ Request objects set to **Private**, so users only see records they own unless shared.

Phase 10: Quality Assurance Testing

Objective:

Ensure that all Salesforce components (Objects, Flows, Validation Rules, Triggers, Reports, and Dashboards) work correctly and match the intended business logic.

1. Test Case Format

Each test case will include:

Use Case / Scenario	Test Steps (Input)	Expected Result	Actual Result
----------------------------	---------------------------	------------------------	----------------------

- Include **screenshots** of the record input and resulting output.
-

2. Test Cases

Test Case 1: Create Organ Request

- **Use Case:** Verify new organ request creation and auto-matching with donor.
- **Test Steps (Input):**
 1. Go to App Launcher → Organ Requests → New.
 2. Enter Recipient: Suresh
 3. Required Organ: Kidney
 4. Status: New
 5. Click Save.

Organ Donation & Transpl...

Donor
Rajesh

Related **Details**

Donor Name	Rajesh	Owner	MULAPAKA SUDHARSHAN REDDY
Blood Group	O+		
Organ Available	Kidney		
Consent	<input checked="" type="checkbox"/>		
City	Hyderabad		
Created By	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:58 AM	Last Modified By	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:58 AM

Organ Donation & Transpl...

Recipient
Suresh

Related **Details**

Recipient Name	Suresh	Owner	MULAPAKA SUDHARSHAN REDDY
Blood Group	O+		
Required Organ	Kidney		
City	Hyderabad		
Created By	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:59 AM	Last Modified By	MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 6:59 AM

- **Expected Result:**

- Status automatically updates to Matched.
- Matched Donor field shows Rajesh.

- **Actual Result:**

The screenshot shows a CRM interface for 'Organ Donation & Transplantation'. A specific organ request for a 'Kidney' is displayed. The request is owned by 'MULAPAKA SUDHARSHAN REDDY'. The recipient is 'Suresh' and the required organ is 'Kidney'. The status is 'Matched' and the donor is 'Rajesh'. The request was created by 'MULAPAKA SUDHARSHAN REDDY' on 11/2/2025, 7:17 AM, and last modified by the same user on 11/2/2025, 7:39 AM.

Test Case 2: Donor Consent Validation

- Use Case:** Ensure donors without consent are not matched.
- Test Steps (Input):**
 - Create a donor without checking the **Consent** checkbox.
 - Create an organ request that matches this donor's blood group and organ.

The screenshot shows a 'New Donor' form. The 'Information' section includes fields for 'Donor Name' (Arjun), 'Blood Group' (AB-), and 'Organ Available' (Heart). The 'Consent' field is empty. The 'City' field contains 'Delhi'. A validation error message 'We hit a snag.' is displayed, stating 'Consent must be given to register donor.' The form includes standard buttons for 'Cancel', 'Save & New', 'Save', and 'Save'.

- Expected Result:**
 - Flow does not match donor. Status remains New.

- **Actual Result:**

The screenshot shows a CRM interface for 'Organ Donation & Transpl...'. A specific 'Organ Request' record for 'Heart' is displayed. The details include:

- Organ Request Name:** Heart
- Recipient:** Suresh
- Required Organ:** Heart
- Status:** New
- Owner:** MULAPAKA SUDHARSHAN REDDY
- Created By:** MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 10:55 AM
- Last Modified By:** MULAPAKA SUDHARSHAN REDDY, 11/2/2025, 10:56 AM

Test Case 3: Report Accuracy

- **Use Case:** Verify Organ Requests Summary Report shows correct matched donors.
- **Test Steps (Input):**
 1. Go to Reports → Organ Requests Report.
 2. Run report.
- **Expected Result:**
 - Matched Donor column shows all successful matches.
 - Status column reflects Matched or New.
- **Actual Result:**

The screenshot shows a report titled 'Organ Requests Report' with the following data:

Organ Request	Required Organ	Recipient	Matched Donor	Status	Created Date
1 Liver	Liver	Priya	Neha	Matched	11/2/2025
2 Kidney	Kidney	Suresh	Rajesh	Matched	11/2/2025
3 Heart	Heart	Suresh	-	New	11/2/2025

Test Case 4: Dashboard Display

- **Use Case:** Ensure dashboard correctly reflects organ requests, matched/unmatched stats.
- **Test Steps (Input):**
 1. Go to Dashboards → Organ Donation Dashboard → Run.
 2. Verify metrics, charts, and tables.
- **Actual Result:**

The screenshot shows the Salesforce interface with the 'Organ Donation & Transplant Management System' logo at the top. The navigation bar includes 'Search...', 'Donors', 'Recipients', 'Organ Requests', 'Reports', and 'Dashboards'. The 'Dashboards' menu is currently selected. Below the navigation is the 'Organ Donation Dashboard' title, with a timestamp 'As of Nov 2, 2025, 10:46 AM' and a note 'Viewing as MULAPAKA SUDHARSHAN REDDY'. A sub-section titled 'Organ Requests Report' contains a table with the following data:

Organ Request	Organ Request Name	Required	Recipient	Matched	Status
Kidney	Kidney	Suresh	Rajesh	Match	
Liver	Liver	Priya	Neha	Match	

At the bottom of the report section, there are links 'View Report (Organ Requests Report)' and 'As of Nov 2, 2025, 10:46 AM'.

Conclusion

The Salesforce-based Organ Donation & Transplant Management System successfully streamlines the complex process of matching organ donors with recipients. By leveraging Salesforce's robust features—custom objects, flows, validation rules, triggers, reports, and dashboards—the system automates donor-recipient matching, updates request statuses in real time, and provides actionable insights through reporting.

Key achievements of this project include:

- **Automation of Critical Processes:** Record-triggered flows automatically match donors and recipients based on blood group, organ availability, and city, reducing manual effort and errors.
- **Data Management and Accuracy:** Custom objects and fields ensure organized storage of donor and recipient data, with validations to maintain data integrity.
- **Enhanced Reporting and Monitoring:** Real-time dashboards and reports allow administrators to monitor organ requests, matched donors, and overall trends effectively.

- **Scalability for Future Enhancements:** The system is designed to incorporate AI-driven suggestions, chatbot integration, approval workflows, and mobile access in future iterations.

Overall, this project demonstrates the power of Salesforce in transforming traditional organ donation workflows into a **smart, automated, and reliable system** that improves operational efficiency, supports timely decision-making, and ultimately contributes to saving lives.

Future Enhancements

The current Salesforce-based Organ Donation & Transplant Management System provides an automated donor-recipient matching system, real-time status updates, and reporting dashboards. In the future, the system can be enhanced further with advanced features to improve efficiency, user engagement, and patient outcomes:

1. Chatbot Integration

- Implement a Salesforce chatbot using **Einstein Bots** or **Service Cloud Chat** to:
 - Guide users in creating organ requests or donor records.
 - Provide instant information on available donors and organ availability.
 - Answer frequently asked questions about consent, eligibility, or donation procedures.

2. AI-Powered Donor Suggestions

- Integrate **Salesforce Einstein AI** to:
 - Suggest the best donor based on historical success rates, urgency, or compatibility beyond just blood group and city.
 - Prioritize matching based on criticality and donor health data.

3. Automated Notifications & Alerts

- Expand email and SMS notifications to:
 - Notify hospitals and coordinators when a donor becomes available.
 - Send reminders for donor follow-ups or organ collection schedules.

4. Approval Workflows for Hospitals

- Introduce **Approval Processes** for hospital administrators to verify donor-recipient matches before final confirmation.

5. Mobile App Integration

- Develop a **Salesforce mobile app** interface for field coordinators:
 - Allow real-time access to donor and recipient data.
 - Enable creation and updates of organ requests on the go.

6. Advanced Analytics & Dashboards

- Include predictive dashboards to analyze:
 - Organ demand trends by city or type.
 - Donor availability patterns.
 - Success rates of matched transplants.

7. External System Integration

- Connect with hospital databases, EMR systems, or national organ registries for:
 - Automatic data updates.
 - Cross-platform donor-recipient matching.
 - Real-time organ availability tracking.