**Phase 4: Process Automation (Admin)**

**1. Validation Rules**

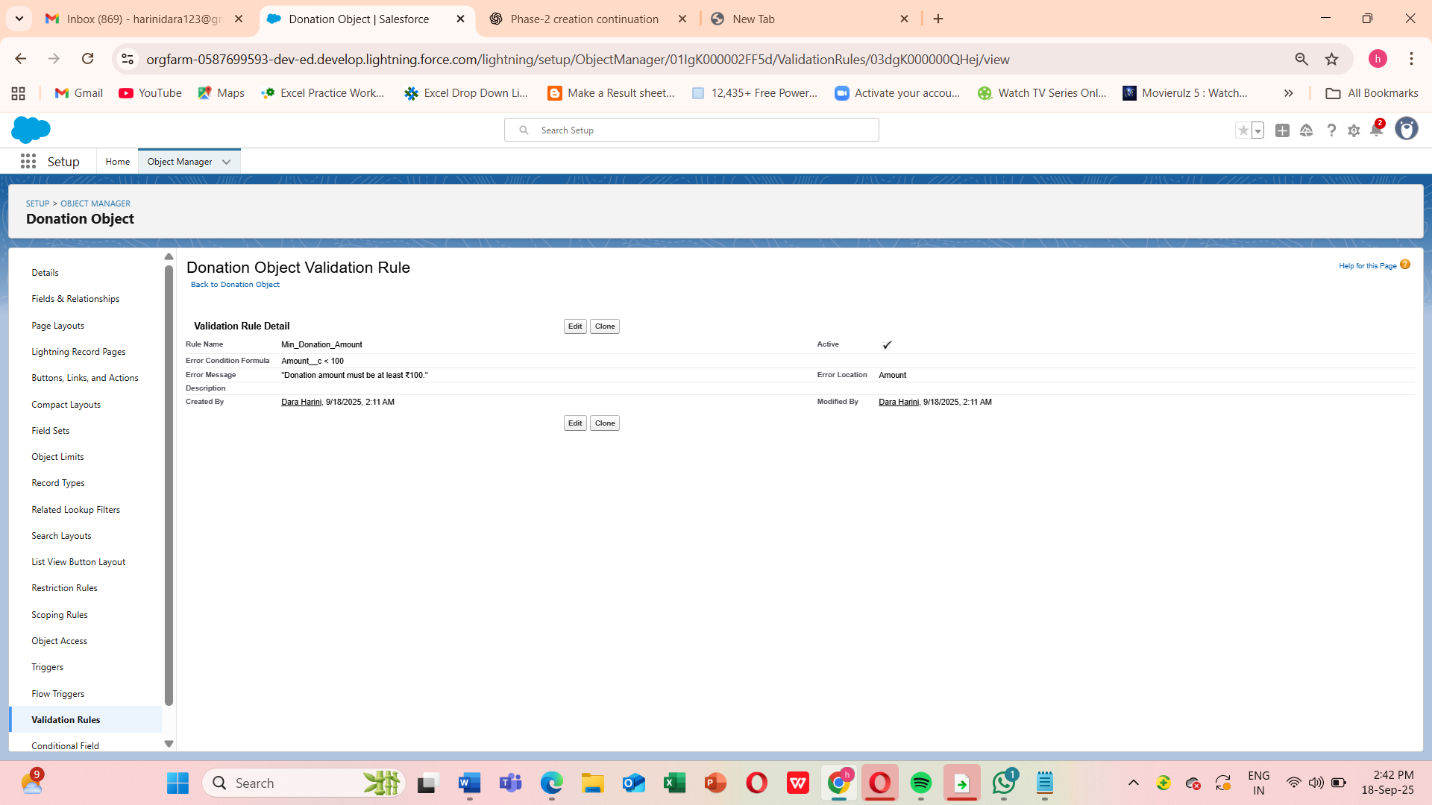
**Why?** To ensure clean, accurate data is entered by users. For NGOs, donations should have required info like amount > 0, donor email filled, etc.

**Example Use Case:** Prevent users from saving a donation record if the **Donation Amount is less than ₹100**.

**Steps:**

1. Go to **Setup → Object Manager → Donation\_\_c → Validation Rules**.
2. Click **New**.
3. Enter:
   * **Rule Name:** Min\_Donation\_Amount
   * **Error Condition Formula:**
   * Amount\_\_c < 100
   * **Error Message:** "Donation amount must be at least ₹100."
   * Error Location: **Field → Amount\_\_c**
4. Save → Test by creating a donation with amount ₹50 → error will stop you.

This ensures only valid donations are entered.



**2. Configure Workflow Rule (Donation\_\_c)**

To trigger simple automation like sending reminders or updating fields.

**Example Use Case:** Whenever a new donation is created, send a **“Thank You” email** to the donor

Steps:

1. Object:

Already set → Donation\_\_c (your custom object for Donations). ✅

2. Rule Name:

Send\_ThankYou\_Email

3. Description (optional but good practice):

This workflow sends a thank-you email to donors when a new donation record is created.

4. Evaluation Criteria:

Choose: Created  
Why? Because we only want to thank donors when the record is first created, not on every edit.

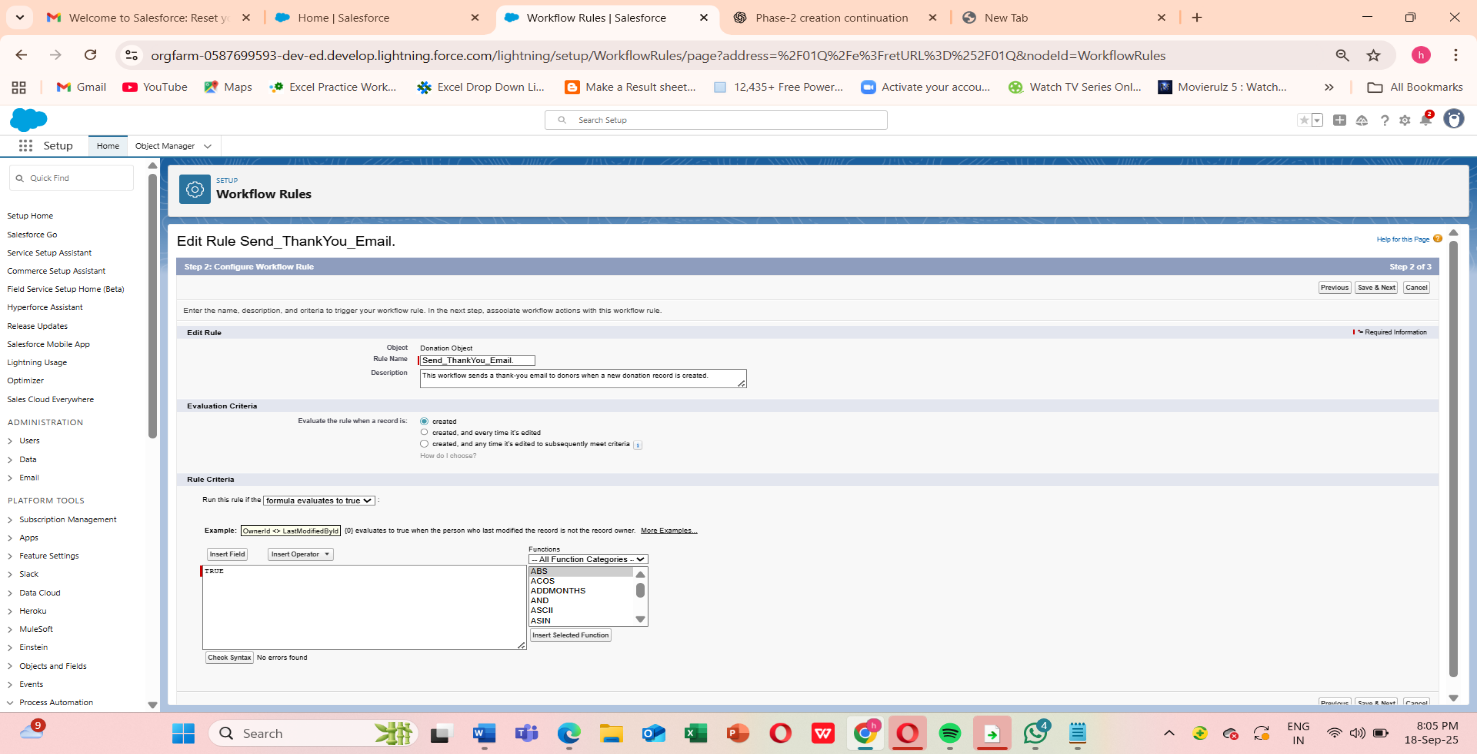
5. Rule Criteria:

We’ll set it so the rule always runs (whenever a donation is created).

* Select: Formula evaluates to true.
* Enter formula:
* TRUE

This ensures the workflow triggers for every new donation.  
Note : (If you wanted to restrict it — e.g., only when Amount\_\_c >= 100 — you could add conditions here, but for now we’ll thank every donor.)

6.Save & Next



**3. Process Builder**

1. On the Workflow Rule screen:

* After saving Step 2, you’ll see options to Add Workflow Action.
* Click Add Workflow Action → New Email Alert.

2. Configure the Email Alert:

Fill in the details:

* Description: Thank You Email Alert for Donors
* Unique Name: Thank\_You\_Email\_Alert
* Email Template:  
   Select an existing email template, or create one:
  + Go to Setup → Classic Email Templates (or Lightning → Email Templates).
  + Create a new HTML (Letterhead) template:
  + Subject: Thank You for Your Donation!
  + Body: Dear {!Donation\_\_c.Donor\_Name\_\_c},

Thank you for your generous donation of ₹{!Donation\_\_c.Amount\_\_c}.

Your support helps us achieve our mission.

Regards, NGO Team

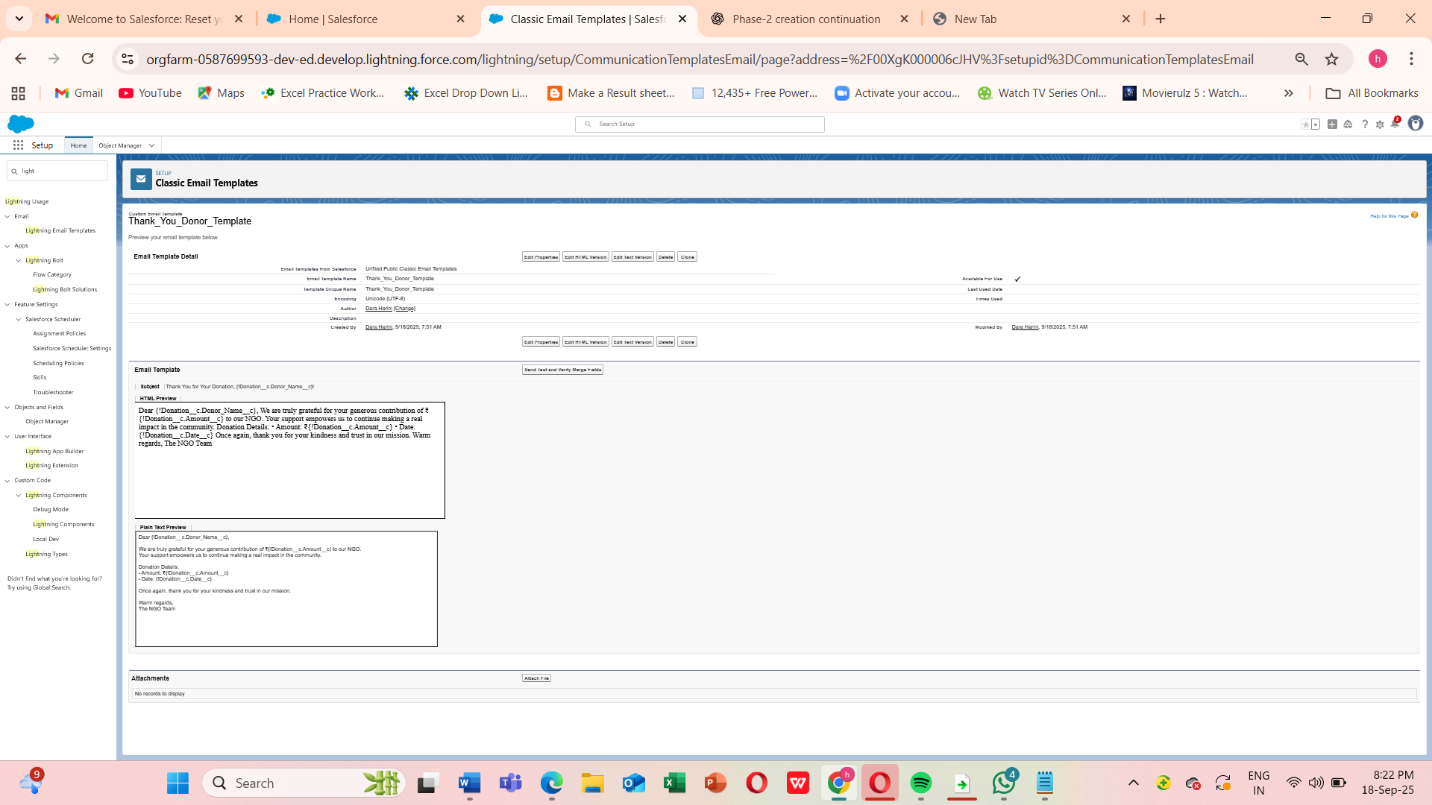
* + Save.
* Recipients:
  + Choose Email Field → Donor Email\_\_c (assuming you created a custom email field on Donation or Donor object).
  + You can also add additional recipients like fundraising officers.
* From Email Address:
  + Choose Organization-Wide Email Address (configure in Setup → Email → Org-Wide Addresses if not already).

3. Save the Email Alert

* Click Save → you’ll return to the Workflow Rule detail page.
* Now click Done.

4. Activate the Workflow

* On the Workflow Rule detail page, click Activate.



**4.Approval Process**

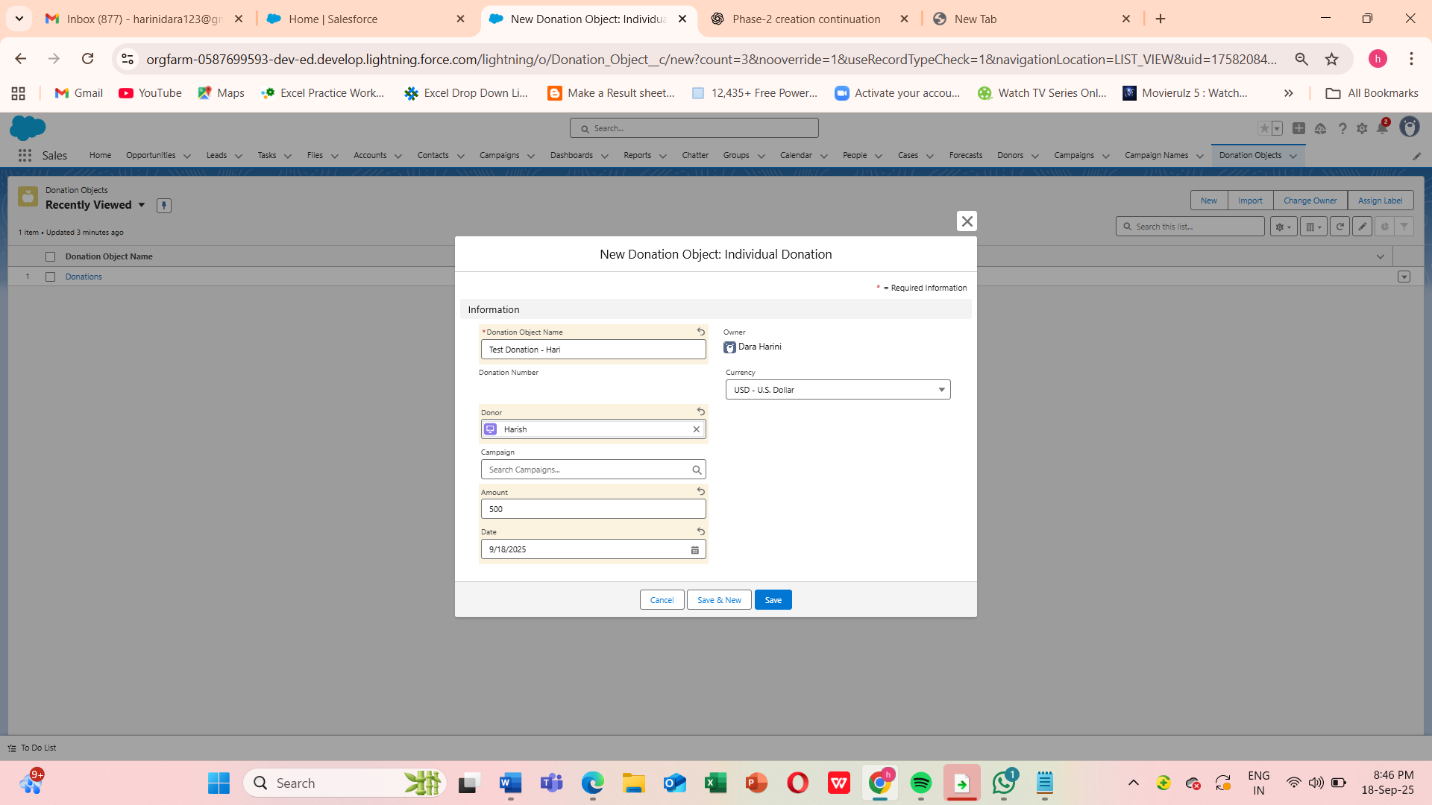
Some actions need approval before finalizing (e.g., large fund disbursements).

**Example Use Case:** If a **Grant Disbursement\_\_c** > ₹1,00,000, it must be approved by the NGO Director.

**Steps:**

1. Go to **Setup → Approval Processes → New Approval Process (Standard Wizard)**.
2. Object: Grant\_Disbursement\_\_c.
3. Entry Criteria: Amount\_\_c > 100000.
4. Approver: NGO Director (specific user or role).
5. Final Actions:
   * Approval → Update Status = “Approved”.
   * Rejection → Status = “Rejected”.
6. Save and Activate.

Ensures financial accountability and governance.



**5. Flow Builder**

Make sure you have:

* A **Donation** object with a **Fundraising\_Officer\_\_c** Lookup(User) field.
* A **Flow** (Record-Triggered Flow) that starts **when a Donation is created or updated**.

**Step 1: Create the Flow**

1. Go to **Setup → Flows → New Flow**.
2. Select **Record-Triggered Flow** → Click **Next**.
3. Object: **Donation Object** (your custom Donation object).
4. Trigger the Flow: **A record is created or updated**.
5. Condition Requirements (optional, e.g., only donations > ₹50,000):
   * Field: Amount\_\_c
   * Operator: Greater Than
   * Value: 50000
6. Optimize for: **Actions and Related Records** → Click **Done**.

**Step 2: Add a Decision (Optional, if you only want high-value donations)**

1. Drag **Decision** onto canvas → Label: High Value Donation?
2. Outcome 1: Yes
   * Condition: Amount\_\_c > 50000
3. Outcome 2: No → Default outcome.
4. Connect the Start element to this Decision.

**Step 3: Create Records (Task)**

1. Drag **Create Records** → Label: Create Follow-Up Task.
2. How Many Records to Create: **One**
3. How to Set Fields: **Manually**
4. Object: **Task**
5. Click **+ Add Field** for each:

| **Task Field** | **Value** |
| --- | --- |
| Subject | "Call Major Donor" |
| Status | Not Started |
| Priority | High |
| Due Date Only | TODAY() |
| Assigned To ID | {!$Record.Fundraising\_Officer\_\_c} |
| Related To ID | {!$Record.Id} |

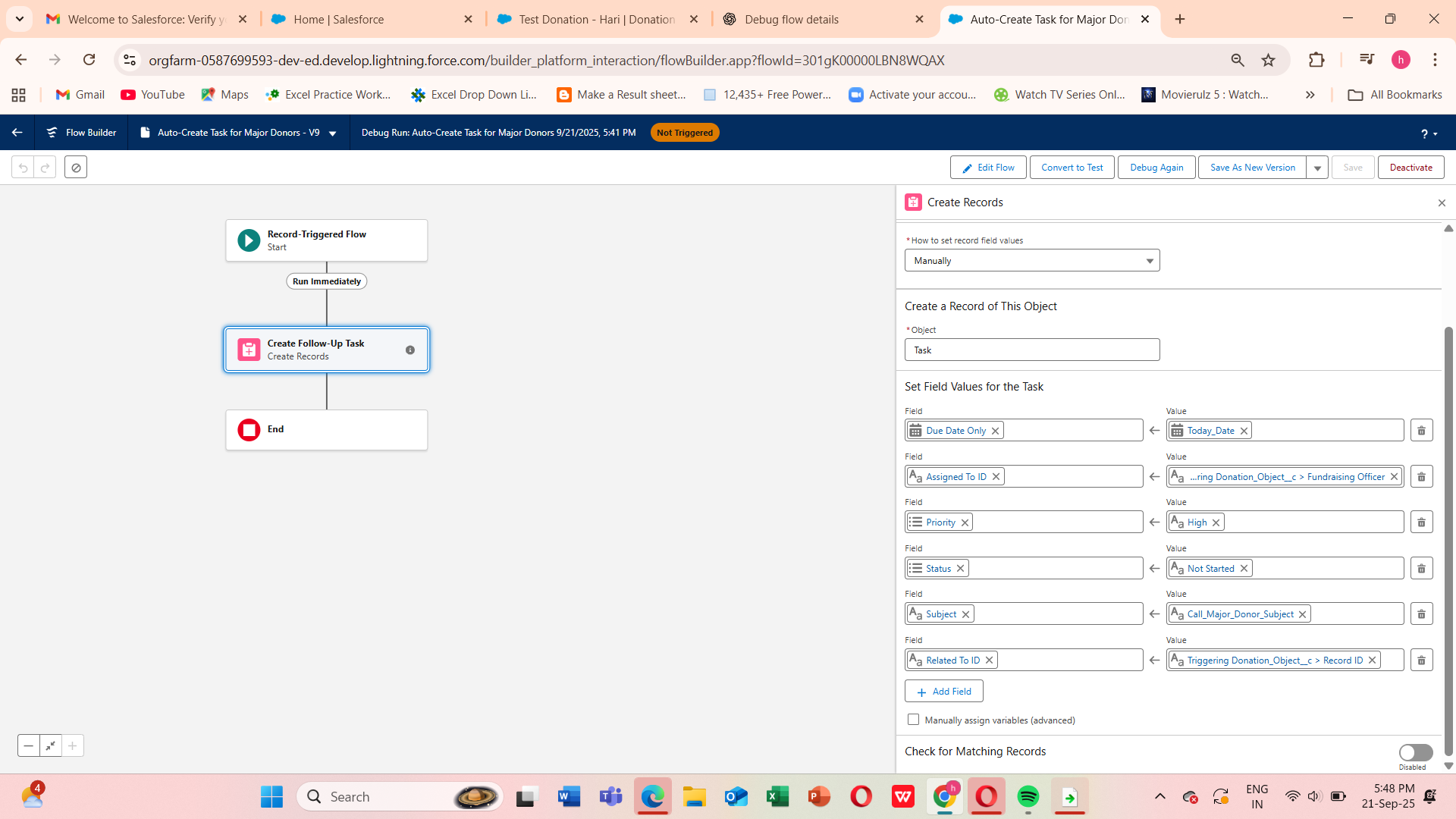
**Important:** For Assigned To ID, **do not type text**. Use the Resource Picker → $Record → Fundraising\_Officer\_\_c.

**Step 4: Connect Elements**

* Connect **Decision → Yes outcome → Create Follow-Up Task**.
* If no Decision, connect **Start → Create Follow-Up Task** directly.

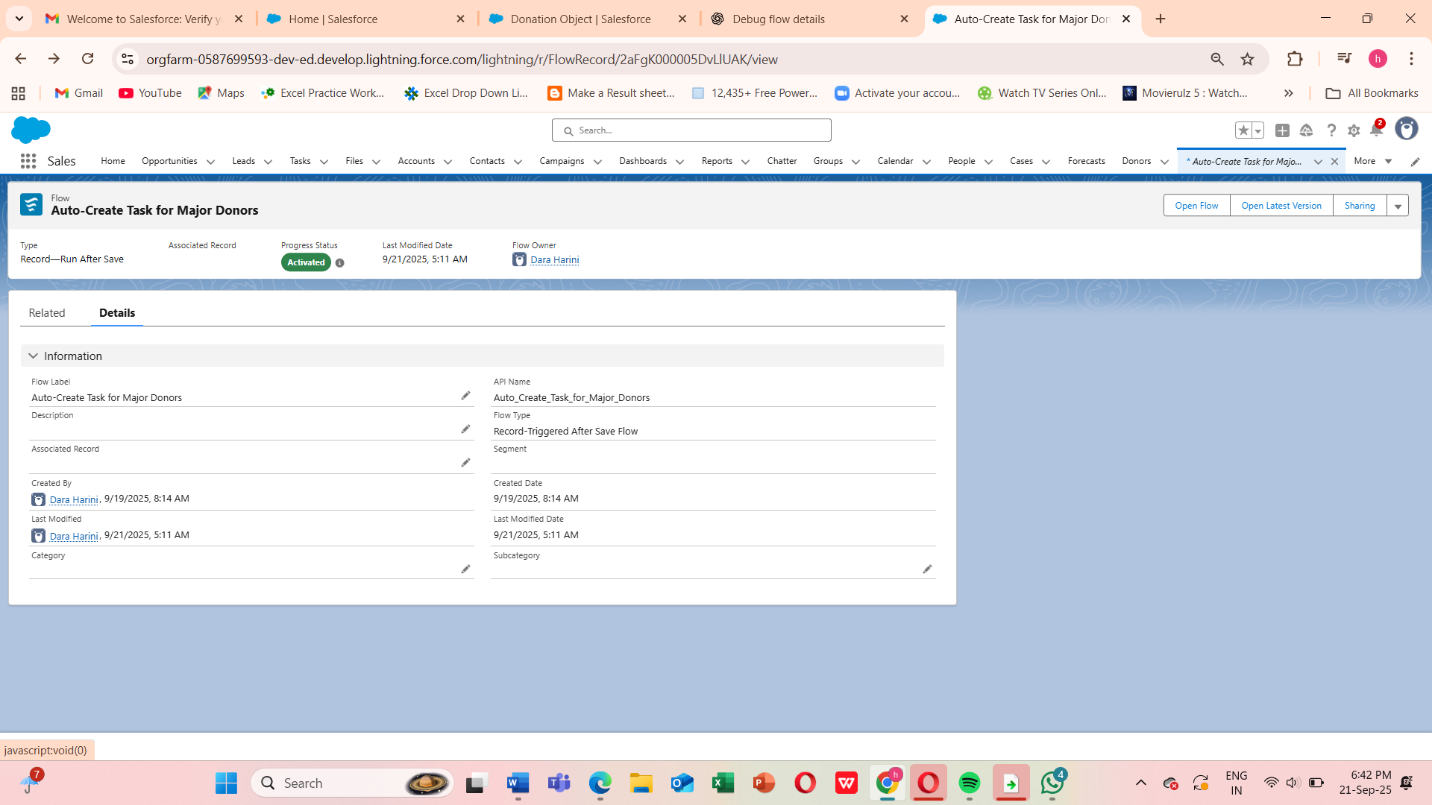
**Step 5: Save & Activate**

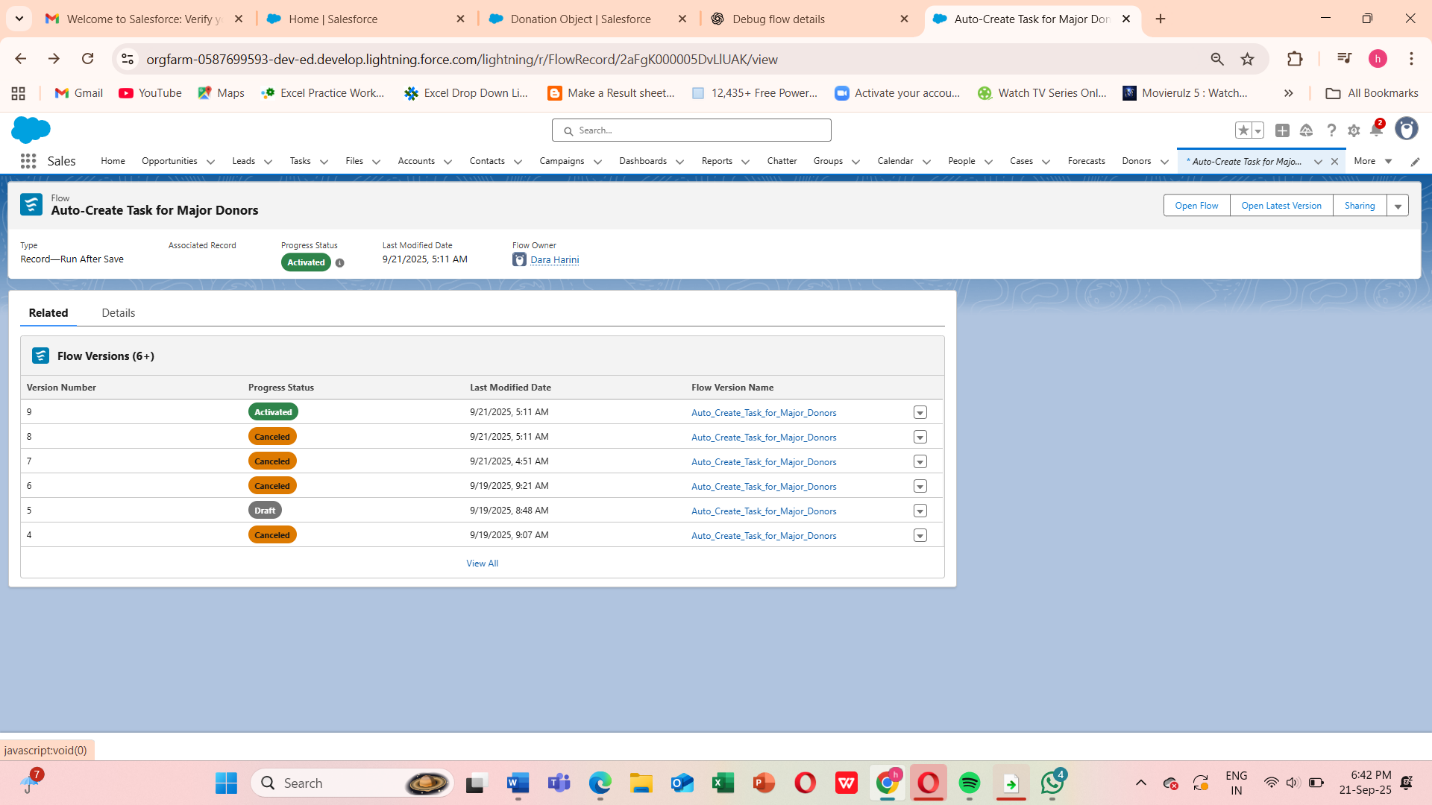
1. Click **Save** → Name: Donation Follow-Up Task.
2. Click **Activate**.



**Step 6: Test the Flow**

1. **Create a Test Donation Record**
   * Go to your **Donation object**.
   * Click **New**.
   * Fill in required fields (e.g., Donor Name, Amount).
   * **Important:** Set **Amount > ₹50,000** and choose a **Fundraising Officer** in the lookup field.
2. **Save the Donation Record**.
   * This should trigger the Flow automatically.
3. **Check if Task is Created**
   * Go to **Tasks** in Salesforce.
   * Look for a new Task:
     + **Subject:** Call Major Donor
     + **Assigned To:** The Fundraising Officer you selected
     + **Related To:** The Donation record you just created
     + **Status / Priority / Due Date:** Matches what you set in Flow





**Screen Flow — Donor Intake Form**

**Goal:** Capture Donor info and create a Donation\_\_c record.

**Steps:**

1. **Create Flow**
   * Setup → Flows → New Flow → Screen Flow → Create
   * **Flow Name:** Donor Intake Flow
2. **Add Screen**
   * Label: Donor Intake Screen
   * Components to add:

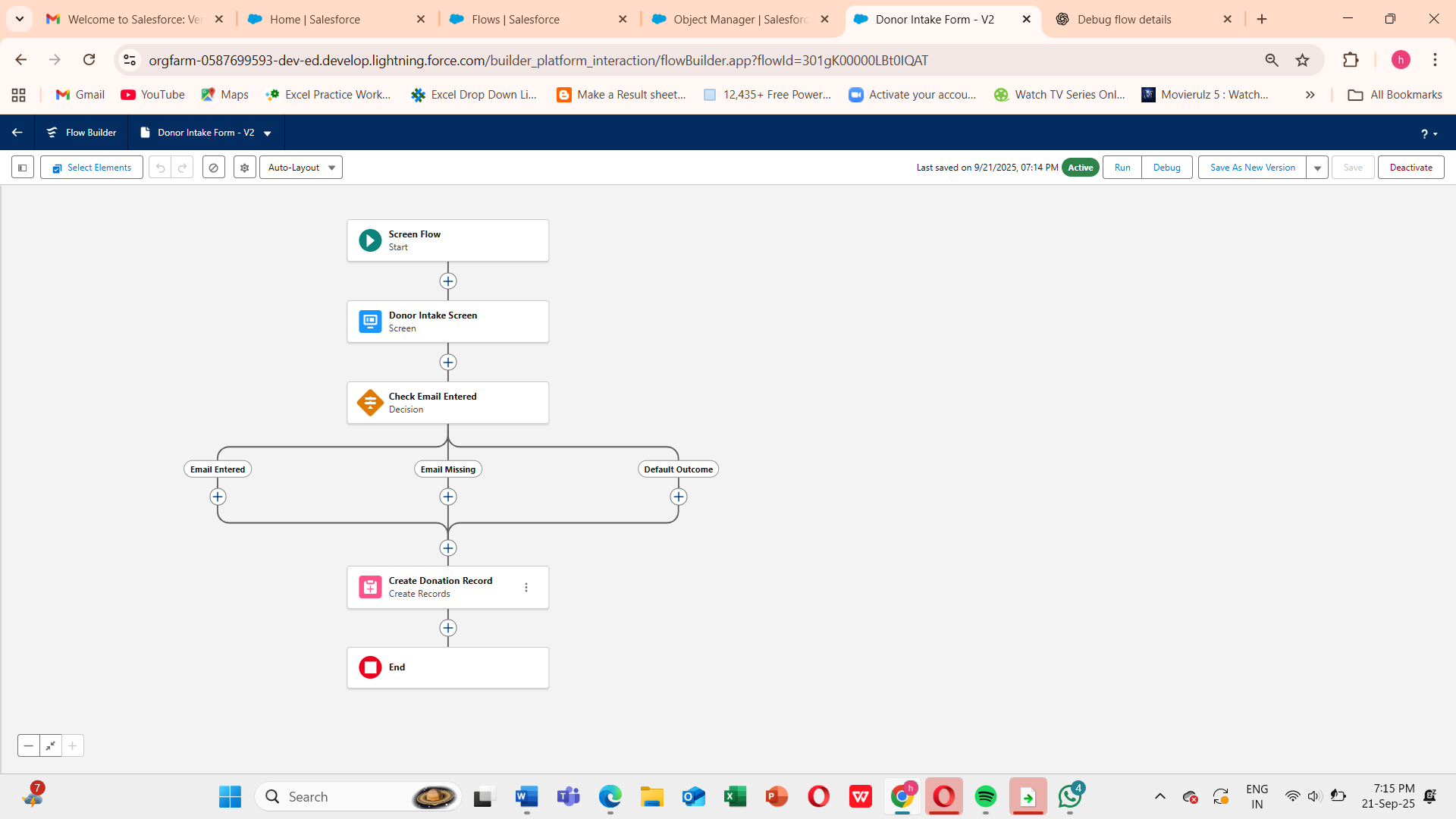
| **Component** | **Label** | **API Name** | **Required** |
| --- | --- | --- | --- |
| Text | Donor Full Name | DonorName | ✅ |
| Email | Donor Email | DonorEmail | ✅ |
| Currency | Donation Amount | DonationAmount | ✅ |
| Date | Donation Date | DonationDate | Optional |

* + Screenshot placeholder: [Insert Screen Flow Screenshot Here]

1. **Create Records**
   * Element: Create Donation Record
   * Object: Donation\_\_c
   * Field mapping:

| **Field** | **Value / Resource** |
| --- | --- |
| Name | {!DonorName} |
| Donor\_Email\_\_c | {!DonorEmail} |
| Amount\_\_c | {!DonationAmount} |
| Donation\_Date\_\_c | {!DonationDate} |

1. **Connect Elements**
   * Start → Screen → Create Records → End
2. **Save & Activate**
   * Save → Activate

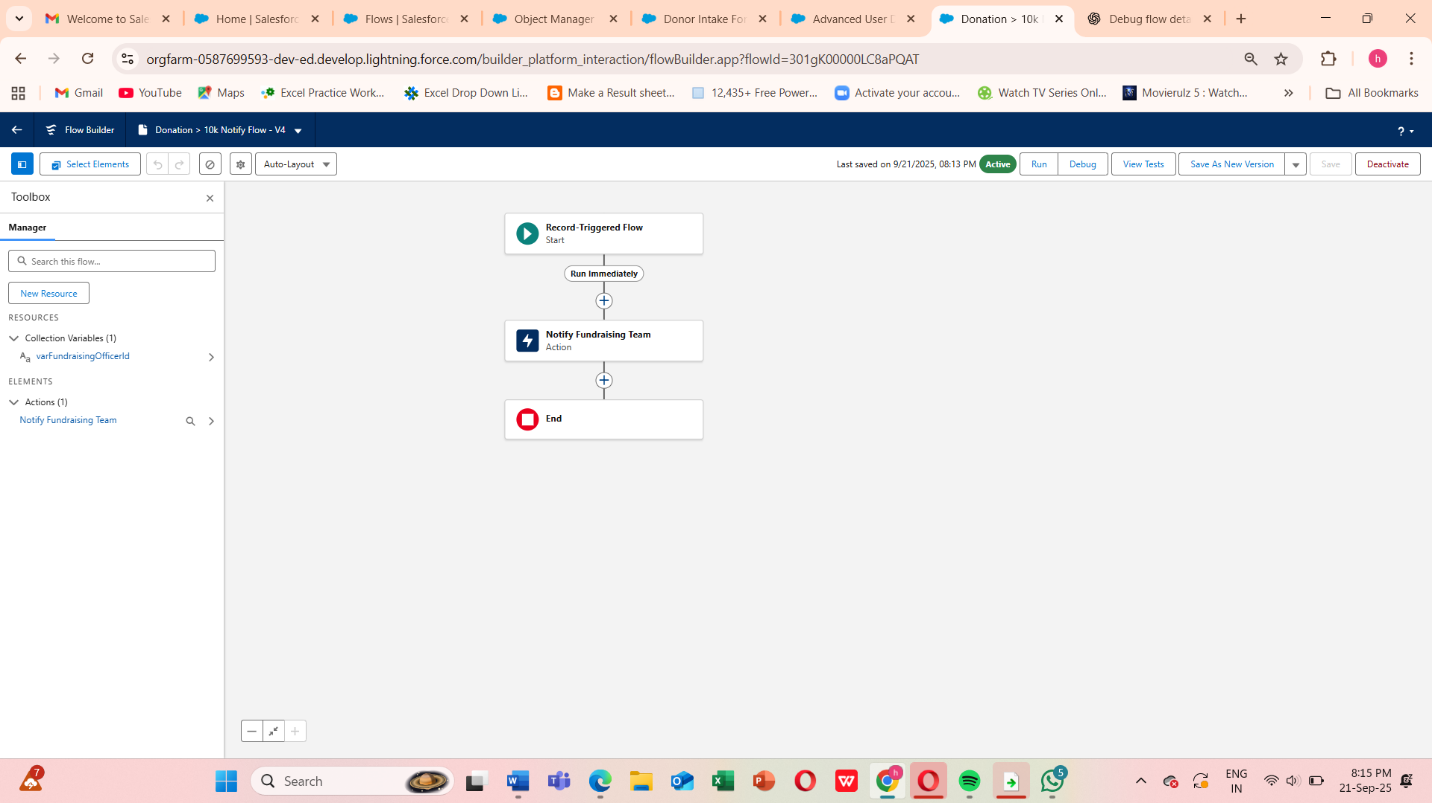


**Record-Triggered Flow — High Value Donation Notification**

**Goal:** Notify Fundraising Team when donation > ₹10,000.

**Steps:**

1. **Create Flow**
   * Setup → Flows → New Flow → Record-Triggered Flow → Create
   * **Flow Name:** Donation > 10k Notify Flow
2. **Configure Trigger**
   * Object: Donation\_\_c
   * Trigger: Created (or Created/Updated)
   * Entry Condition: Amount\_\_c > 10000
   * Optimize for: Actions and Related Records
3. **Add Action — Send Custom Notification**
   * Label: Notify Fundraising Team
   * Notification Type: Fundraising\_Alert
   * Title: New major donation: ₹{!$Record.Amount\_\_c}
   * Body: A donation of ₹{!$Record.Amount\_\_c} received from {!$Record.Donor\_\_c}
   * Recipient: Single User Id (admin for testing) or collection variable
   * Screenshot placeholder: [Insert Custom Notification Action Screenshot Here]
4. **Save & Activate**
   * Save → Activate



**Scheduled-Triggered Flow — Weekly Receipt Reminder**

**Goal:** Email donors with pending receipts every Monday 09:00 IST

**Steps:**

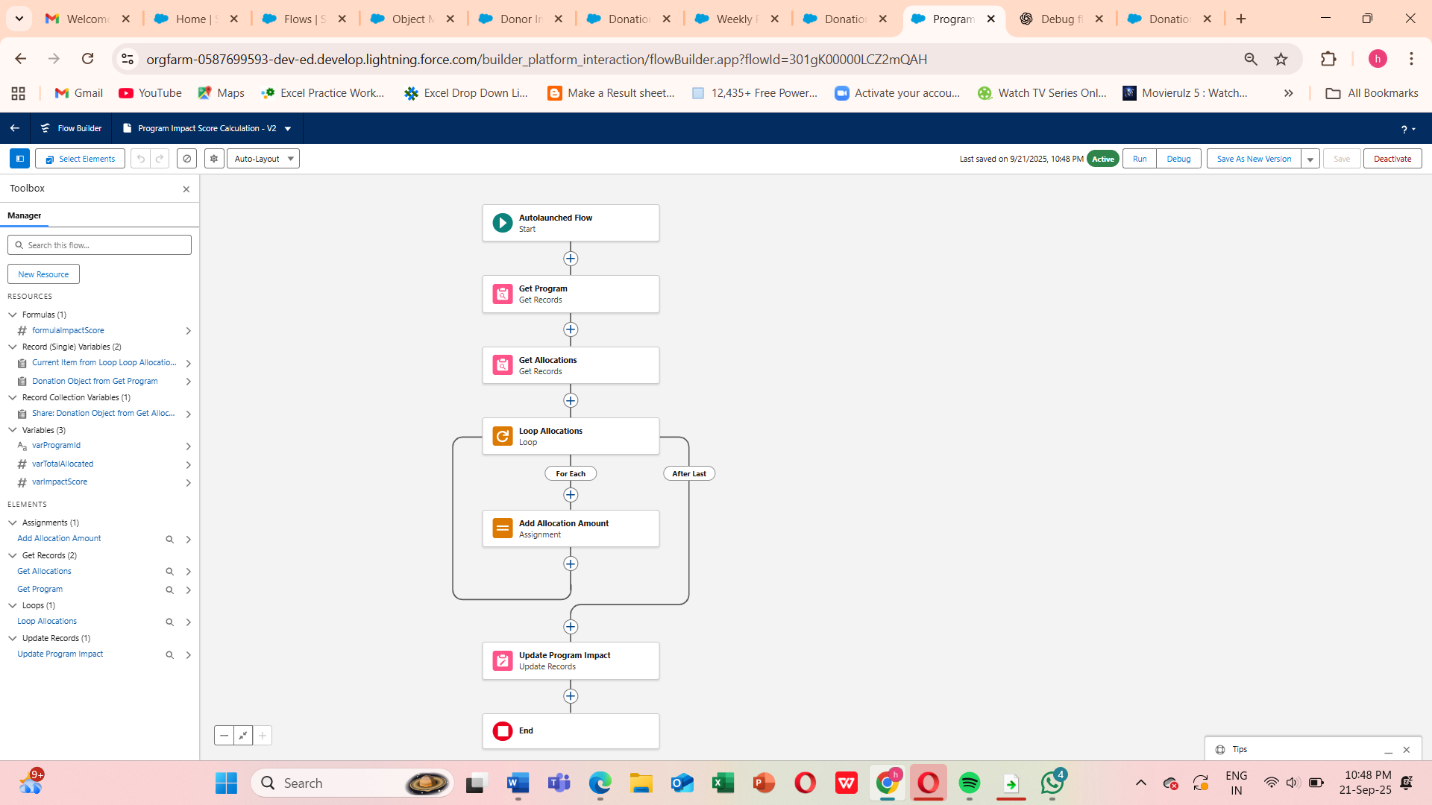
1. **Create Flow**
   * Setup → Flows → New Flow → Schedule-Triggered Flow → Create
   * **Flow Name:** Weekly Pending Receipt Reminder
2. **Set Schedule**
   * Start: Next Monday 09:00 IST
   * Frequency: Weekly → Monday
3. **Get Records**
   * Label: Get Pending Donations
   * Object: Donation\_\_c
   * Filter: Receipt\_Sent\_\_c = FALSE
   * Store all fields, All records
4. **Loop Through Records**
   * Label: Loop Pending Donations
   * Collection: Get Pending Donations
5. **Send Email**
   * Inside loop → Action → Send Email
   * To: {!Loop\_Donation.Donor\_Email\_\_c}
   * Subject: Reminder: Your donation receipt
   * Body: Dear {!Loop\_Donation.Donor\_Name\_\_c}, please find your donation receipt...
   * Screenshot placeholder: [Insert Send Email Action Screenshot Here]
6. **Connect Elements**
   * Start → Get Records → Loop → Send Email → Loop End → End
7. **Save & Activate**
   * Save → Activate

**Auto-Launched Flow — Program Impact Score Calculation**

**Goal:** Calculate Impact\_Score\_\_c for a program based on related donations

**Steps:**

1. **Create Flow**
   * Setup → Flows → New Flow → Auto-Launched → Create
   * **Flow Name:** Calculate Program Impact Flow
2. **Create Input Variable**
   * Name: varProgramId
   * Data Type: Text
   * Available for input: ✅
3. **Get Program Record**
   * Label: Get Program
   * Object: Program\_\_c
   * Filter: Id = {!varProgramId}
4. **Get Related Donation Allocations**
   * Label: Get Allocations
   * Object: Donation\_Allocation\_\_c
   * Filter: Program\_\_c = {!varProgramId}
5. **Loop & Assignment**
   * Loop over allocations → Add Assignment element to sum allocation amounts
6. **Update Program**
   * Element: Update Program Impact
   * Field: Impact\_Score\_\_c = {!varImpactScore}
7. **Save & Activate**
   * Save → Activate



**Custom Notifications Process**

**Create Custom Notification Type**

1. Go to **Setup → Quick Find → Notification Builder → Custom Notifications**
2. Click **New**
3. Enter:
   * **Name / Label**: Fundraising\_Alert
   * **API Name**: auto-fills as Fundraising\_Alert
   * **Supported Channels**: Desktop & Mobile (check both)
4. Click **Save**

This defines the type of notification you can send in Flows or Apex.

**Create Record-Triggered Flow**

1. Go to **Setup → Flows → New Flow → Record-Triggered Flow → Create**
2. **Object**: Donation\_\_c
3. **Trigger**: A record is **created** (or “created or updated”)
4. **Entry Conditions**: Amount\_\_c > 10000 (or your threshold)
5. **Optimize for**: Actions and Related Records
6. Click **Done**

**Add Action → Send Custom Notification**

1. Click **+ Add Element → Action**
2. Search **Send Custom Notification** → select it
3. Configure:
   * **Label**: Notify Fundraising Team
   * **Notification Type**: Fundraising\_Alert (the one you created)
   * **Title**: New Major Donation: ₹{!$Record.Amount\_\_c}
   * **Body**: Donation received from {!$Record.Donor\_\_c}
   * **Recipient Record IDs**:
     + You can use **a single User Id** (e.g., your admin) for testing
     + Or **query a Public Group** (Fundraising Team) using **Get Records → Group Members → collection of User Ids**
4. Click **Done**

**Connect & Activate Flow**

1. Connect **Start → Send Custom Notification → End**
2. Click **Save** → Name: High Value Donation Notification
3. Click **Activate**

