# Phase 4 – Process Automation (Admin) — Step-by-step Implementation Guide

# Validation Rules

Validation Rules are used to check conditions before saving a record.

They help in maintaining data quality and enforce business rules.

Example: Require Template field if Task Status is 'Submitted'.

Example: Feedback comments must be filled if Feedback\_Submitted = True.

Error messages should be clear and placed on the right field.  
  
  
**Why:** Prevent users from saving incomplete records.

How to add:

1. Setup → Object Manager → Task → **Validation Rules** → **New**.
2. Name it Require\_Template\_On\_Submit.
3. Paste this formula:

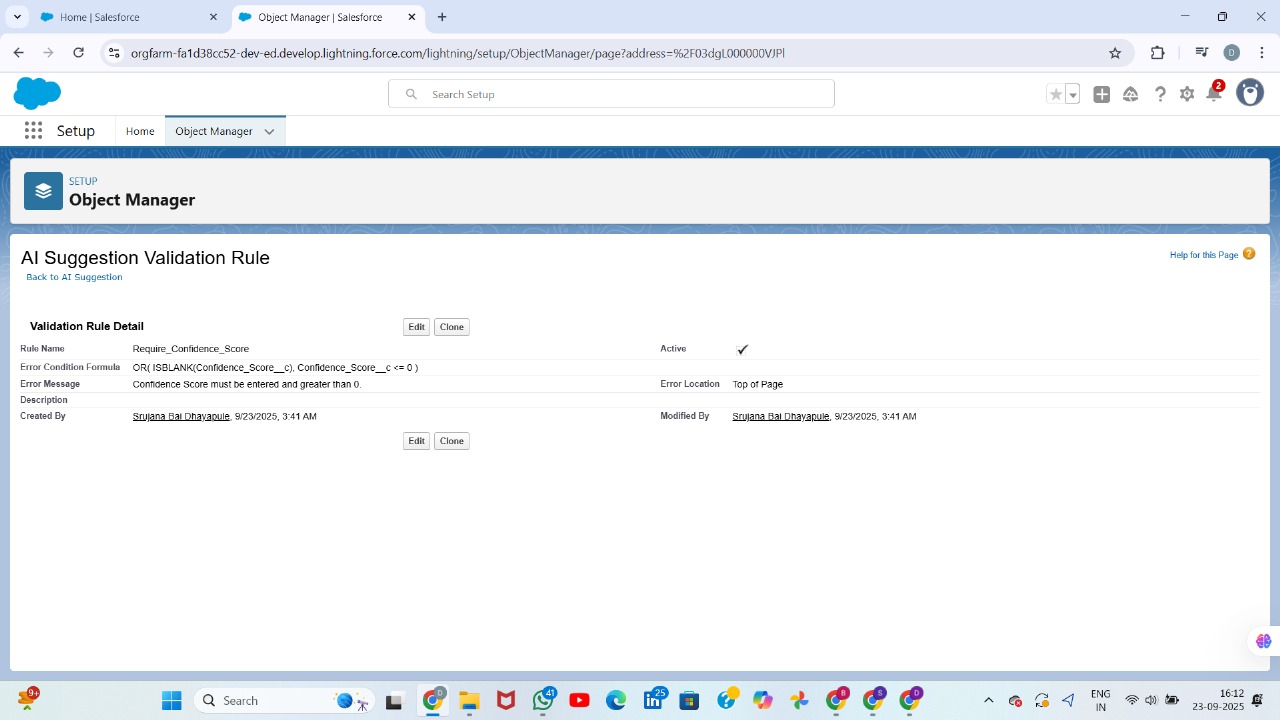
AND(ISPICKVAL(Status\_\_c, 'Submitted'), ISBLANK(TEXT(Template\_\_c)))

1. Error Message: “Please select a Template before submitting.” → set error location to Template field → Save.

Second rule (require comments when feedback submitted):

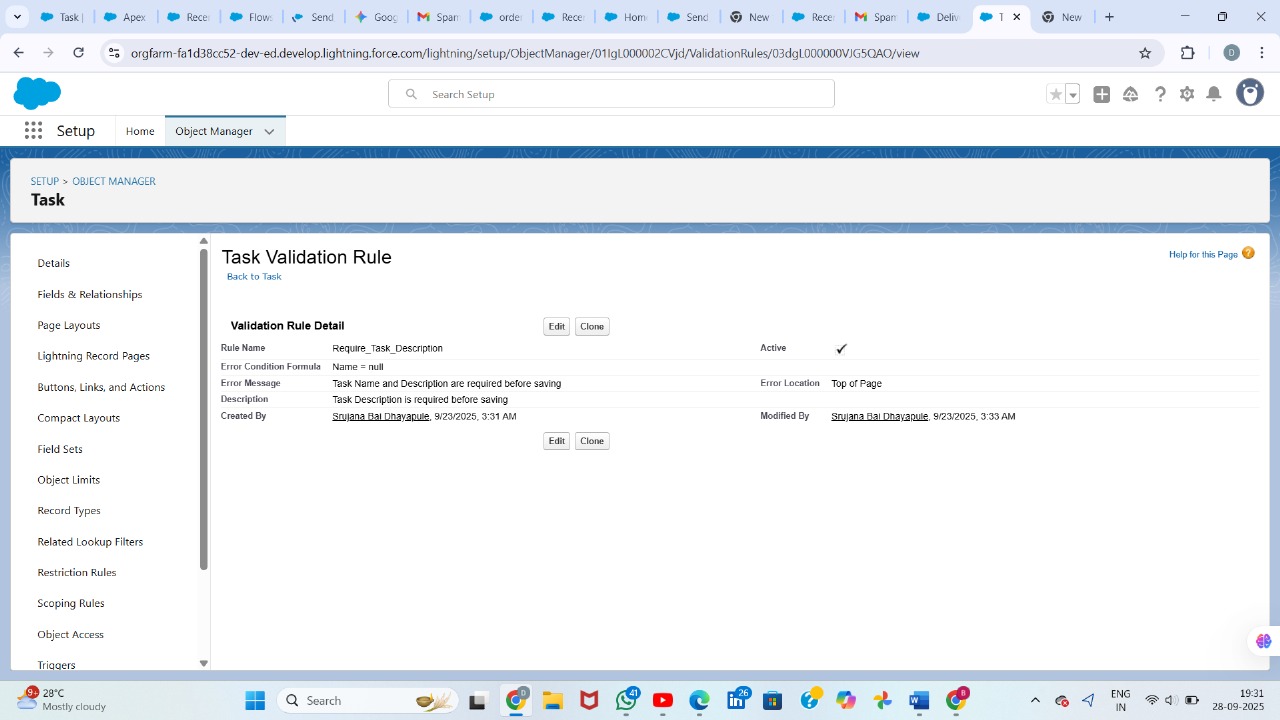
AND(Feedback\_Submitted\_\_c = TRUE, ISBLANK(Feedback\_Comments\_\_c))

Error message: “Please enter feedback comments before submitting feedback.”

**1. Data Quality Enforcement: Task Validation Rule**

This section details the implementation of a **Validation Rule** on the standard **Task** object. This rule prevents users from saving a Task record unless critical fields, specifically the **Name** and **Description**, contain data, thereby ensuring high data quality.

**Step-by-Step Implementation:**

1. **Navigate to the Task Object:**
   * From **Setup**, navigate to **Object Manager**.
   * Select the **Task** object.
   * In the sidebar, click **Validation Rules**.
2. **Create the New Rule:**
   * Click the **New** button.
3. **Define Rule Details:**
   * Set the **Rule Name** to Require\_Task\_Description.
   * Ensure the **Active** checkbox is checked to immediately enforce the rule.
4. **Write the Error Condition Formula:**
   * In the **Error Condition Formula** box, enter the formula: ISBLANK(Name)
   * *Note: While your image shows Name = null, ISBLANK(Name) is the preferred and more robust formula to check if the field is empty.*
5. **Configure the Error Message:**
   * Enter the **Error Message** that the user will see: Task Name and Description are required before saving.
6. **Set the Error Location:**
   * Select **Top of Page** for the **Error Location** so the message is highly visible to the user.
7. **Save the Rule:**
   * Click **Save**. The rule is now active and will fire whenever a user attempts to save a Task without a value in the Name field

**Flows**2. Automated Communication: Record-Triggered Flow

This section outlines the Record-Triggered Flow developed to automatically send a confirmation email containing Task details immediately after a new Task is successfully created.

Step-by-Step Implementation:  
1 Access Flow Builder:

* From Setup, search for and select Flows.
* Click New Flow.

2 Select the Flow Type:

* Choose Record-Triggered Flow and click Create.

3 Configure the Flow Trigger:

* Object: Select Task.
* Trigger the Flow When: Select A record is created.
* Optimize the Flow For: Select Actions and Related Records.

4 Add the Action Element (Send Email):

* Click the Add Element (+) icon below the Start element.
* Select the Action element.

5 Configure the Action Details:

* Action: In the search bar, type and select Send Email.
* Label: Enter Task submitted Email.
* API Name: This will auto-populate as Task\_submitted\_Email.

6 Set Recipient Details:

* Under Set Input Values, click Recipient Address List.
* For the value, select the appropriate resource for the recipient's email.
  + *If sending to the Task Creator:* Enter {!$Record.CreatedBy:User.Email}
  + *If sending to the Task Owner:* Enter {!$Record.Owner:User.Email}

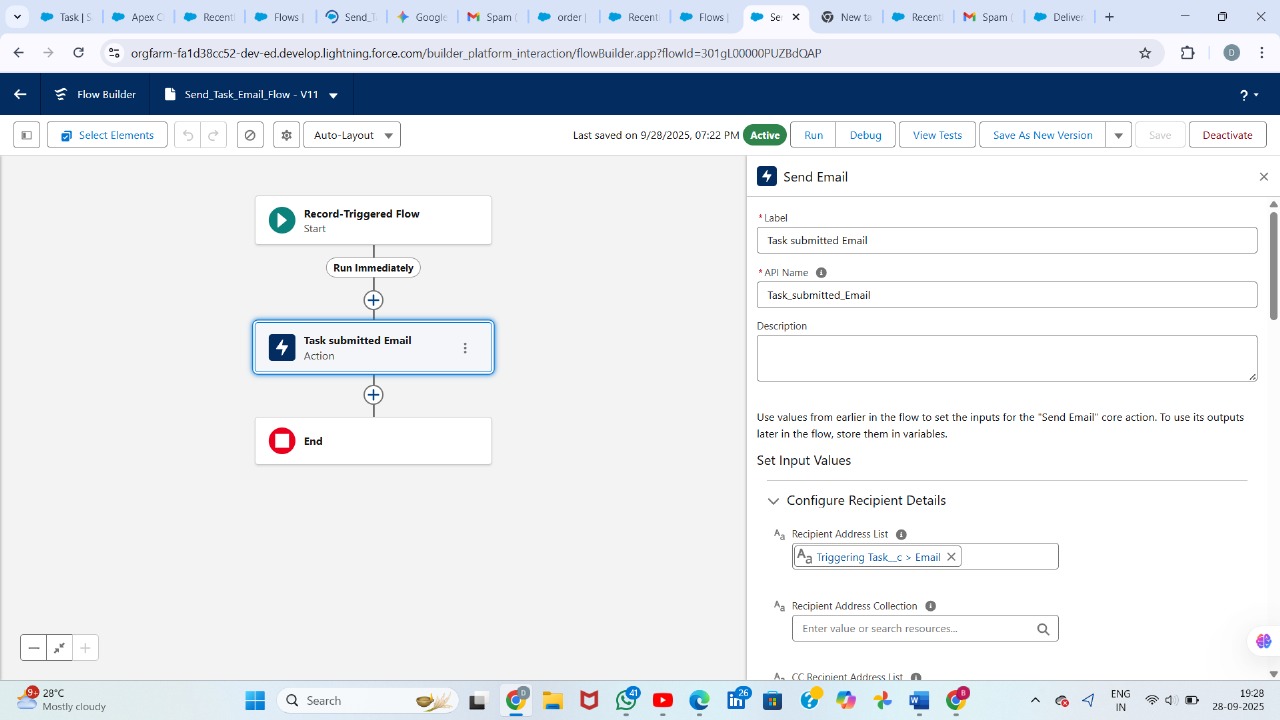
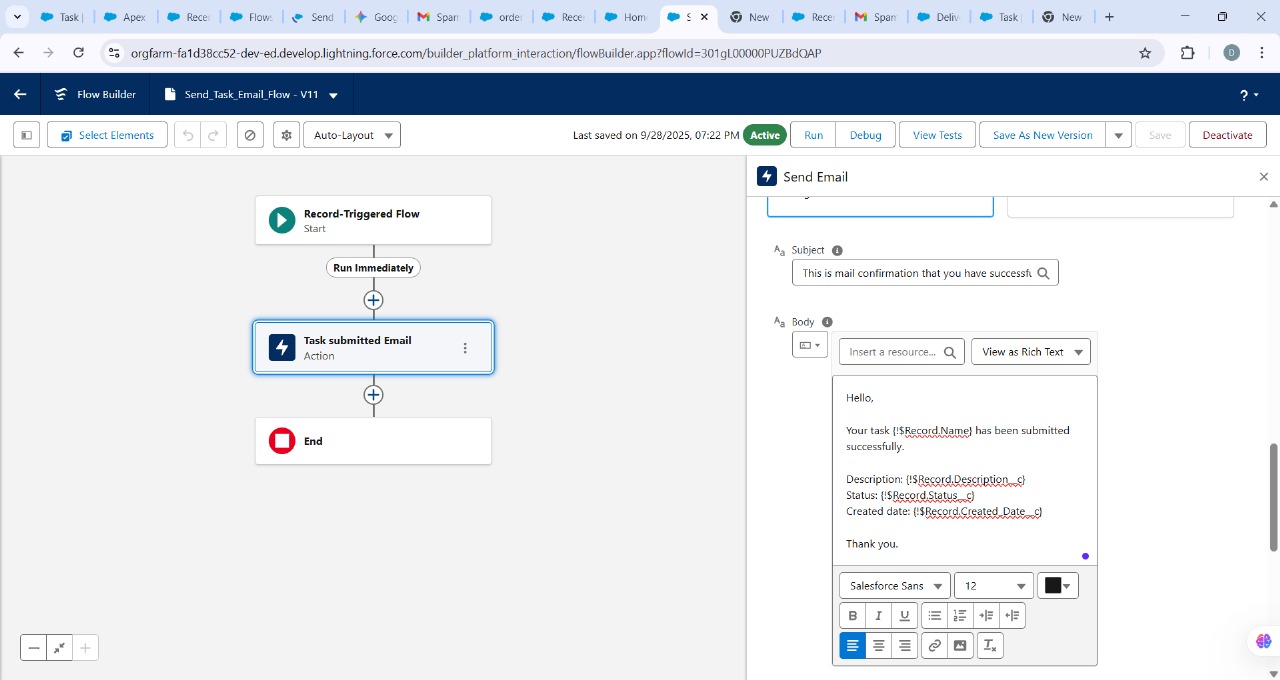
7 Configure Email Content:

* Add a Subject line: This is mail confirmation that you have successfully submitted a Task.
* Add the Body content, using the Rich Text editor to format the email and insert Record resources for dynamic fields:
  + Start with a greeting: Hello,
  + Insert a line confirming submission and the Task Name: Your task: {!$Record.Name} has been submitted successfully.
  + List the key details using merge fields:
    - Description: {!$Record.Description\_\_c}
    - Status: {!$Record.Status}
    - Created date: {!$Record.CreatedDate}
  + End with a closing: Thank you.

8 Connect and Save the Flow:

* Ensure the Action element is connected to the Start and End elements.
* Click Save.
* Flow Label: Enter Send Task Email Flow.
* Flow API Name: This will auto-populate.

9 Activate the Flow:

Click the Activate button in the top right corner to deploy the automation.  
  
  
Email Body:  
Mail Confirmation to gmail:  
