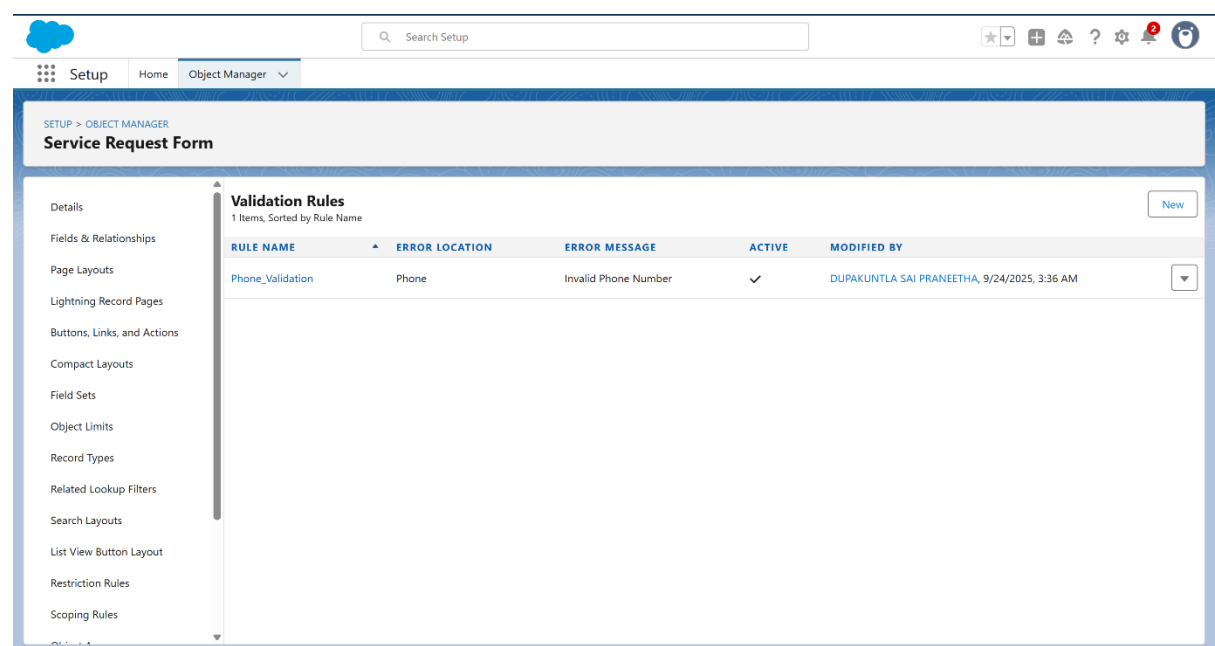


CG Sales & Service : Salesforce Implementation for Retail & Supply Chain Optimization

Phase 4: Process Automation (Admin)

1. Validation Rule – Phone Number

- **Objective:** Prevent users from entering invalid phone numbers.
- **Implementation:**
 - **Formula:**
 - **NOT(REGEX(Phone, "[0-9]{10}\$"))**
 - **Error Message: “Enter a valid 10-digit phone number.”**
- **Result:** Ensures all customer and contact records contain clean, standardized phone numbers.
- **Impact:** Prevents poor-quality data, improving communication reliability with customers.

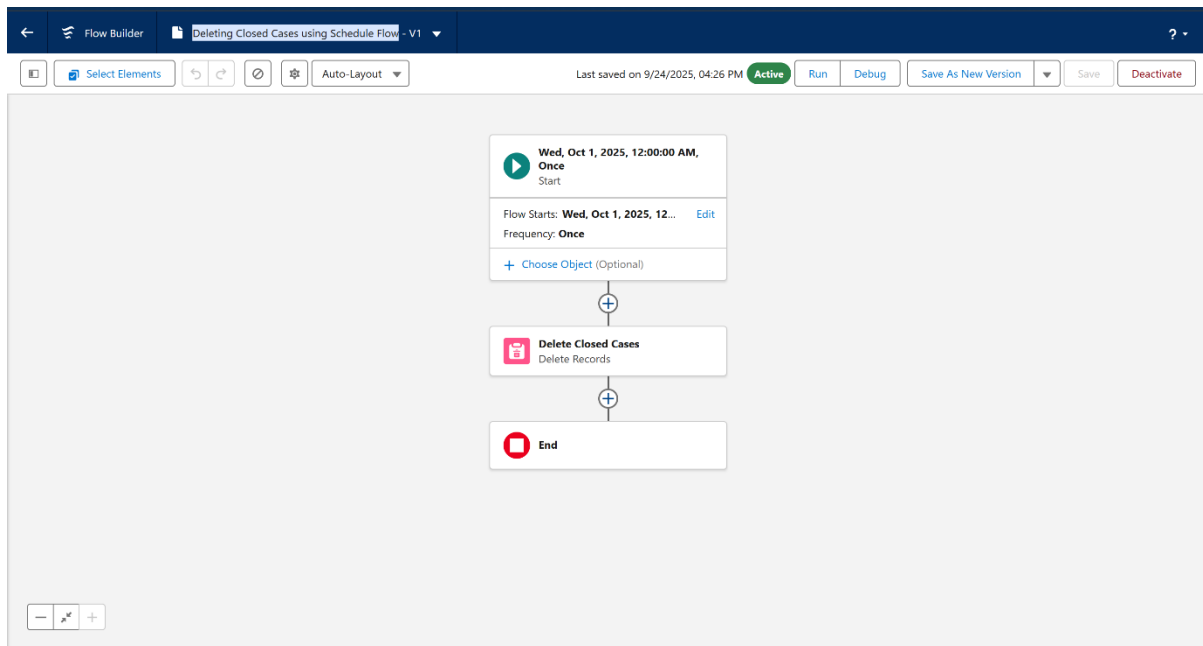


2. Flow Builder Automations

(a) Deleting Closed Cases – Scheduled Flow

- **Objective:** Automatically delete resolved cases that are no longer needed.
- **Flow Details:**
 - **Type:** Scheduled Flow.
 - **Schedule:** Runs weekly at midnight.

- **Logic:** Finds Cases where Status = Closed and ClosedDate is more than 60 days old.
- **Action:** Deletes these records.
- **Business Impact:**
 - Reduces database clutter.
 - Improves system performance by keeping only active/supportive cases.
 - Ensures reports focus only on relevant, recent cases.



(b) Payment Success Flow

- **Objective:** Automate actions after successful payment.
- **Flow Details:**
 - **Type:** Record-Triggered Flow.
 - **Trigger:** When a Payment__c record updates to Status = Success.
 - **Actions:**
 - **Field Update:** Update Order__c.Status → *Paid*.
 - **Email Alert:** Send confirmation to the customer using a predefined template.
 - **Task Creation:** Generate a verification task for Finance.
- **Business Impact:**
 - Provides instant confirmation to customers.
 - Keeps Finance aligned with real-time transactions.

- Eliminates manual intervention, reducing delays and errors.

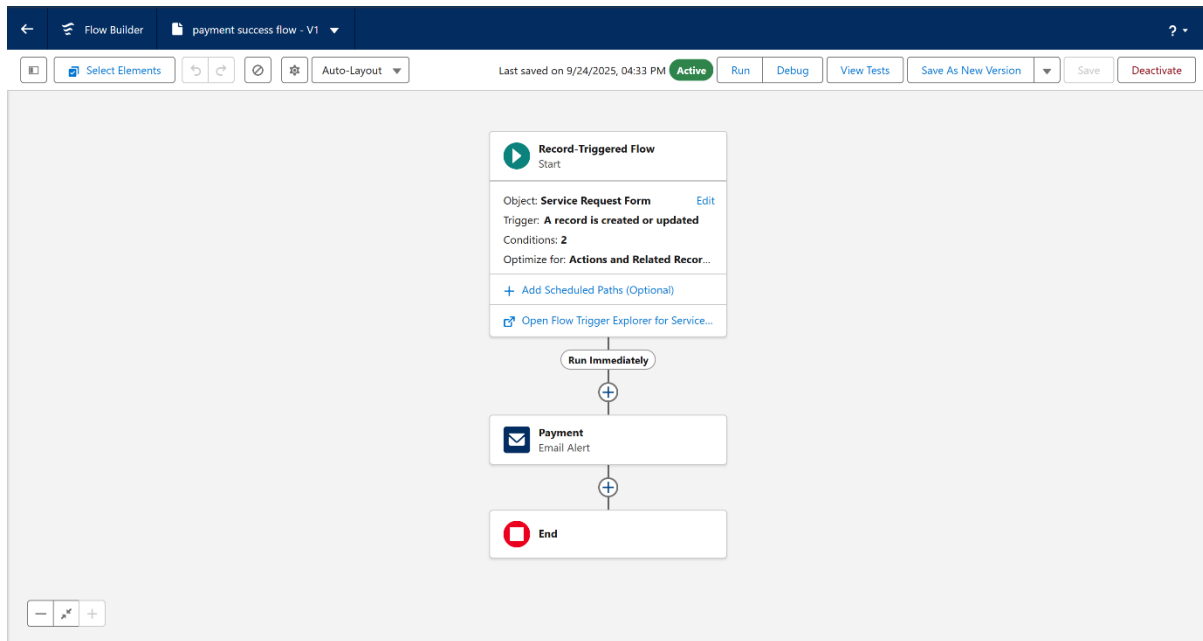


Fig 2.b.1 Payemnt success flow

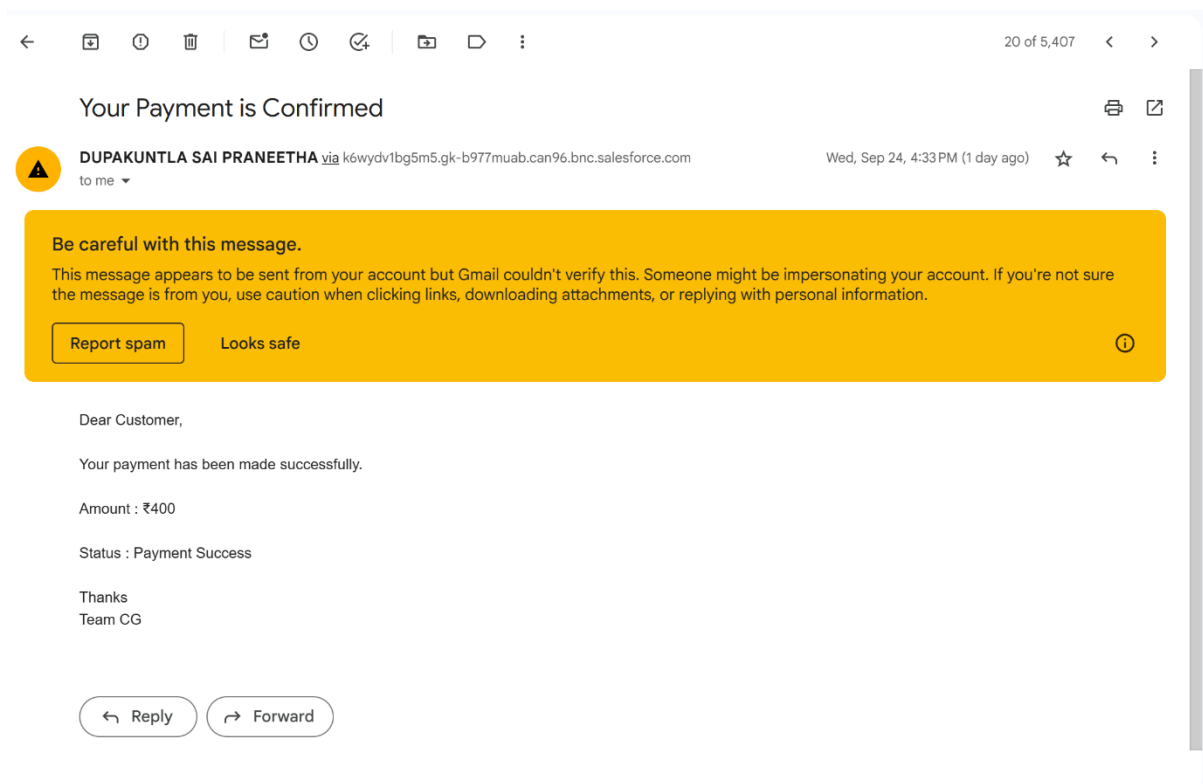


Fig 2.b.2 Payemnt success mail

(c) Field Update & Task Flow

- **Objective:** Automate field updates and ensure follow-up activities are assigned.
- **Flow Details:**
 - **Type:** Record-Triggered Flow.
 - **Trigger:** Executes when specific fields are updated (e.g., Opportunity → Stage = *Closed Won*).
 - **Actions:**
 - **Field Update:** Update related records (e.g., Account.Status → Active).
 - **Task Creation:** Assign a follow-up task (e.g., “Contact customer for onboarding”).
- **Business Impact:**
 - Keeps related data consistent and accurate across objects.
 - Ensures no follow-up action is missed.
 - Improves collaboration between sales and support teams.

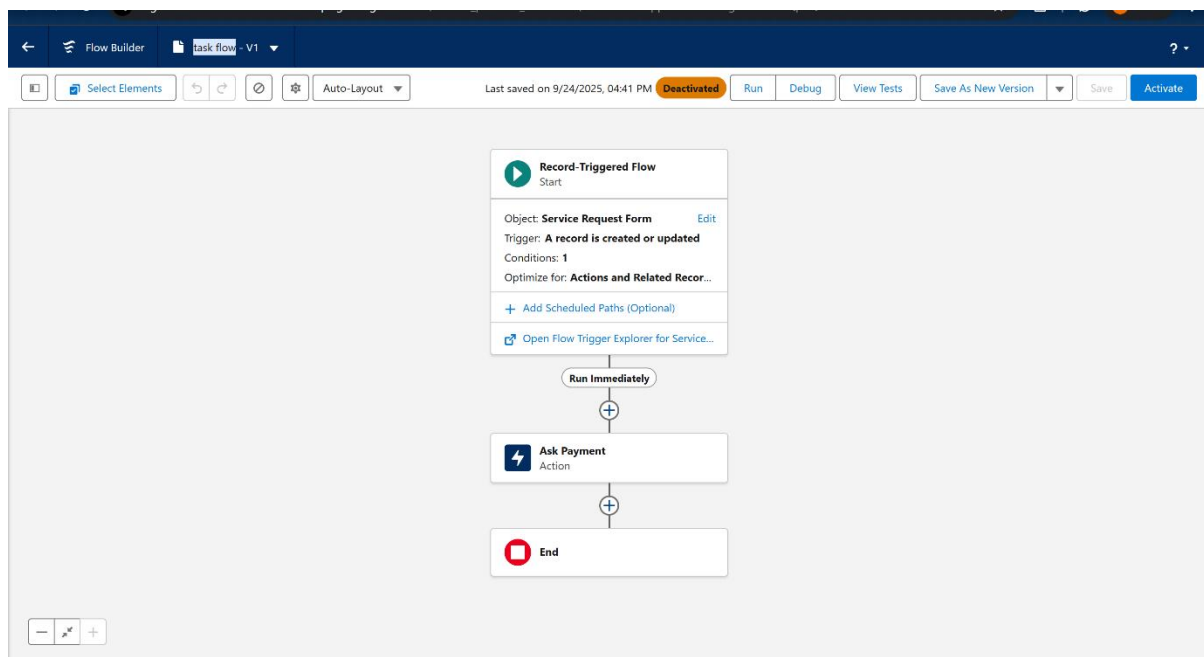


Fig 2.c.1 Task flow

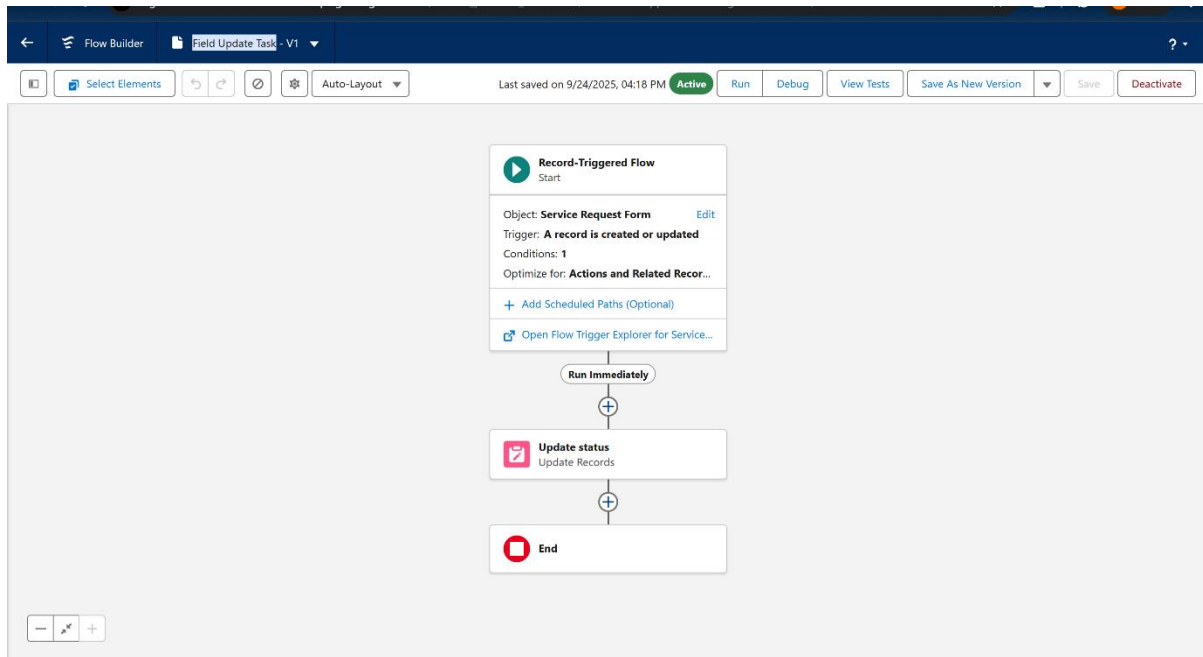
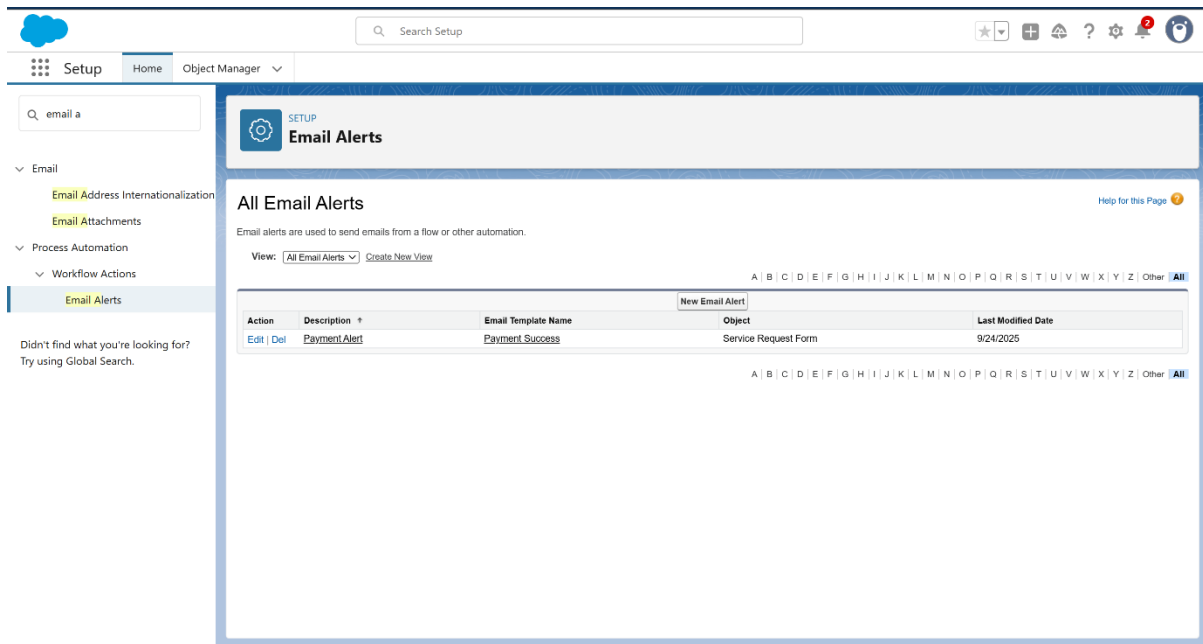


Fig 2.c.2 Field Update

3.Email Alert – Payment Success

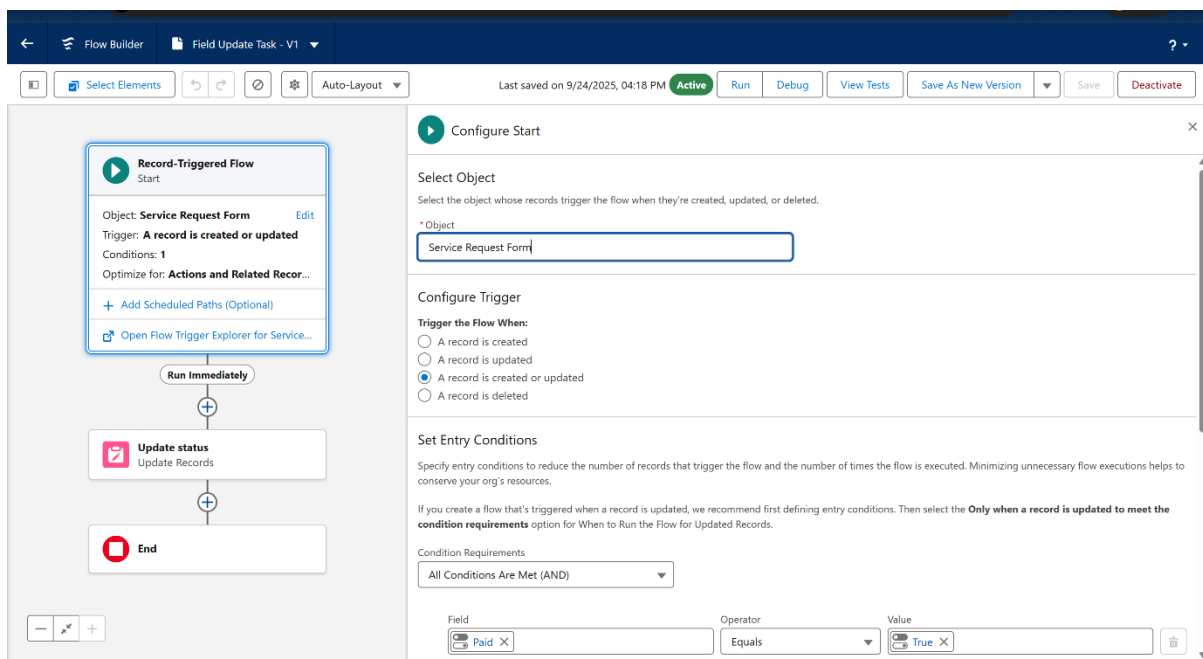
An automated **Email Alert** was configured to notify customers immediately after a successful payment. This was achieved using **Flow Builder** integrated with a predefined email template.

- **Trigger:** When a Payment__c record is updated with Status = Success.
- **Action:** The Flow sends a confirmation email to the customer, including payment and order details.
- **Business Impact:**
 - Provides customers with **instant confirmation**, building trust and transparency.
 - Reduces the need for manual follow-up by the Finance Team.
 - Ensures smooth communication and enhances the overall **customer experience**.



4.Field Updates

- Field updates are used to automatically change the value of a field on a record when certain criteria are met.
- In my project, instead of using the standard **Workflow Field Update**, I created a **Flow** to handle the update.
- The flow updates the **Status** of a **Service_Request__c** record to **“Paid”** automatically when the **Paid** field equals **“Paid”**, indicating that the payment was successful.
- This ensures that service requests are automatically marked as completed or paid without manual intervention, improving accuracy and efficiency.



5.Tasks

- **Tasks** are actionable items or “to-do” records assigned to users in Salesforce.
- In my project, instead of creating tasks manually, I used a **Flow** named “**Task Flow**” to automate task creation.
- The flow creates a task automatically whenever the **Paid** field on a **Service_Request__c** record is **false**, indicating that payment is pending.
- This ensures that the assigned user is notified to follow up on unpaid service requests, improving efficiency and timely collection.

The screenshot displays the Salesforce Flow Builder interface for a flow named "task flow - V1". The flow is configured as a Record-Triggered Flow. The configuration panel on the right shows the following settings:

- Select Object:** Service Request Form
- Configure Trigger:**
 - Trigger the Flow When:** A record is created or updated (selected)
- Set Entry Conditions:**
 - Condition Requirements:** All Conditions Are Met (AND)
 - Field:** Paid
 - Operator:** Equals
 - Value:** False

The flow canvas on the left shows the following steps:

- Record-Triggered Flow Start** (Triggered by a record being created or updated on the Service Request Form object, with 1 condition and optimized for Actions and Related Records).
- Run Immediately** (Intermediate step).
- Ask Payment** (Action step).
- End** (Final step).