Phase 4: Process Automation (Admin)

1. Validation Rules

Explanation:
Validation Rules ensure that only clean, logical, and consistent data enters the system.
They prevent users from saving invalid or incomplete records, which helps maintain high data integrity.

Use Case 1: Prevent Negative Stock in Product

Scenario:
While updating the available stock for a product, the system should not allow a negative value.

Rule Name: Check_Negative_Stock
Formula:

Available_Stock__c < 0

Error Message: “Available stock cannot be negative.”
Error Location: Field → Available Stock

Use Case 2: Quantity Validation in Order Item

Scenario:
When creating or editing an order item, the quantity must always be greater than zero.

Rule Name: Check_Quantity
Formula:

Quantity__c <= 0

Error Message: “Quantity must be greater than zero.”

Use Case 3: Mandatory Supplier in Purchase Order

Scenario:
Every purchase order must have a supplier assigned.

Rule Name: Check_Supplier
Formula:

```
ISBLANK(Supplier__c)
```

Error Message: “Supplier cannot be blank.”

2. Workflow Rules

Explanation:
Workflow Rules were used for simple field updates and email notifications.
They help automate repetitive tasks like sending confirmation emails when an order is marked “Delivered”.

Use Case: Order Delivery Notification

Scenario:
When the *Status* of a Purchase Order changes to “Delivered”, an automated email should go to the Inventory Manager.

Action: Email Alert
Template: “Order Delivery Confirmation – PO Notification”

3. Flow Builder (Record-Triggered Flow)

Explanation:
Flows are powerful no-code automations that execute logic dynamically.
Here, they’re used to auto-populate fields and update stock data.

Use Case 1: Auto-Set Expected Delivery Date

Scenario:
When a new Purchase Order is created, automatically set the *Expected Delivery Date = Order Date + 3 days*.

Flow Type: Record-Triggered
Object: Purchase Order
Trigger: When record is created or updated
Action: Update the same record

Formula Used:

```
Order_Date__c + 3
```

Use Case 2: Auto-Update Product Stock

Scenario:
When an Inventory Transaction is created:

- If type = “Stock In”, increase available stock.
- If type = “Stock Out”, decrease available stock.

Flow Type: Record-Triggered
Object: Inventory Transaction
Logic:

```
IF(Transaction_Type__c = "Stock In",
Product.Available_Stock__c + Quantity__c,
Product.Available_Stock__c - Quantity__c)
```

4. Approval Process

Explanation:
An Approval Process was added to simulate real-world procurement approval — ensuring accountability before large orders are finalized.

Use Case: Approve High-Value Purchase Orders

Scenario:
If the *Total Order Cost* > ₹10,000, it must be approved by the *Inventory Head* before processing.

Steps:

1. Entry Criteria: `Total_Order_Cost__c > 10000`
2. Approver: Inventory Head
3. Email Notification: Sent to approver on submission.
4. Final Action: Status updated to "Approved".

5. Email Alerts

Explanation:
Email alerts notify team members of critical changes.

Use Case: Notify on Low Stock

Scenario:
When Available Stock < Minimum Stock Level → Send email to Inventory Manager.

Tool Used: Workflow + Email Alert
Email Template: "Low Stock Alert – Product Reorder Required"

6. Field Updates

Explanation:
Field updates automatically modify values based on conditions.

Use Case: Update Status on Approval

When the Purchase Order is approved, automatically set the *Status* field to "Delivered".

7. Tasks and Custom Notifications

Explanation:
Tasks remind users to perform follow-ups or checks, and custom notifications pop up for real-time alerts.

Use Case: Follow-up Reminder

When a Purchase Order's *Expected Delivery Date* is reached but the status isn't "Delivered", create a Task:

> Follow up with the supplier for delayed delivery.