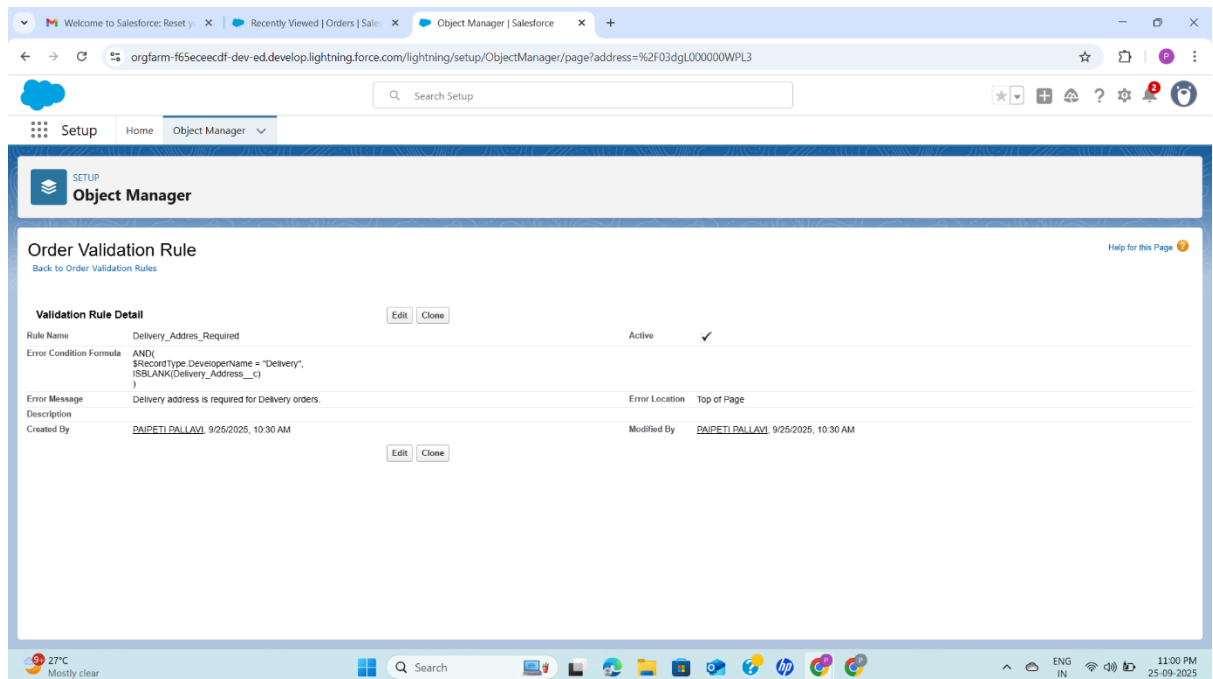


PHASE 4

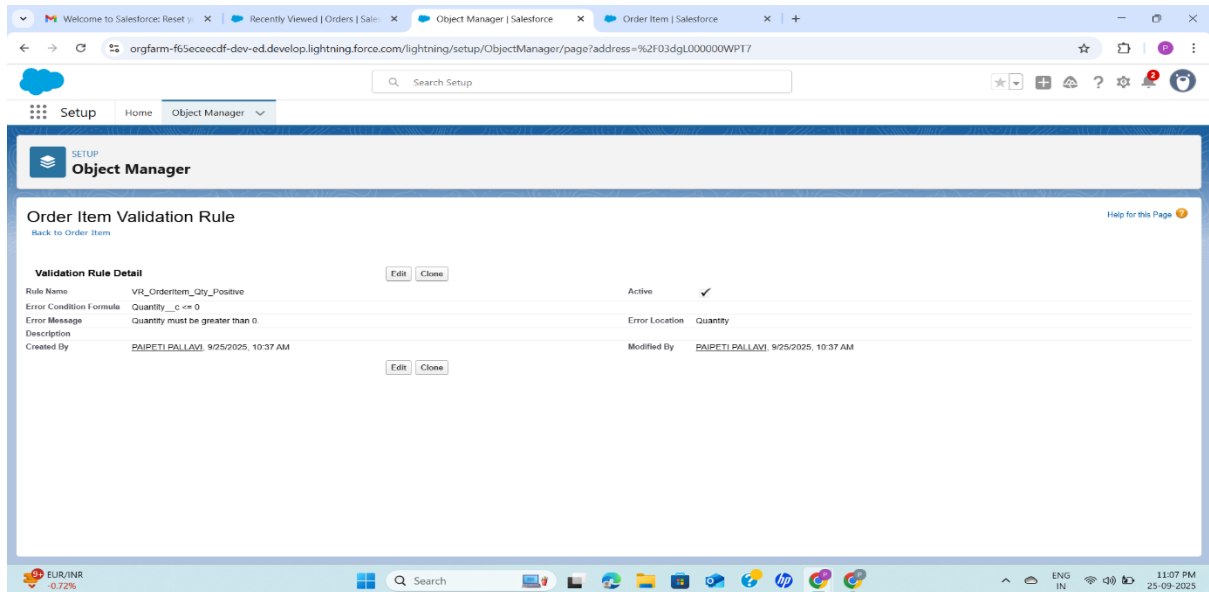
A. Delivery address required

1. Setup → **Object Manager** → **Order__c** → **Validation Rules** → **New**.
2. Fill:
 - **Rule Name:** Delivery_Address_Required
 - **Error Condition Formula:**
 - AND(
 - \$RecordType.DeveloperName = "Delivery",
 - ISBLANK(Delivery_Address__c)
 -)
 - **Error Message:** Delivery address is required for Delivery orders.
 - **Error Location:** Field → Delivery_Address__c
3. Click **Save**, then **Activate**.
4. **Test:** Create/update an Order record with Record Type = Delivery and no address → expect error.



B. Quantity must be > 0

1. Setup → **Object Manager** → **Order_Item__c** → **Validation Rules** → **New**.
2. Fill:
 - **Rule Name:** Quantity_Positive
 - **Error Condition Formula:**
 - Quantity__c <= 0
 - **Error Message:** Quantity must be greater than 0.
 - **Error Location:** Field → Quantity__c
3. Save & Activate.
4. **Test:** Create Order Item with Quantity = 0 or negative → expect error.



Screen Flow (Host Booking) — step-by-step

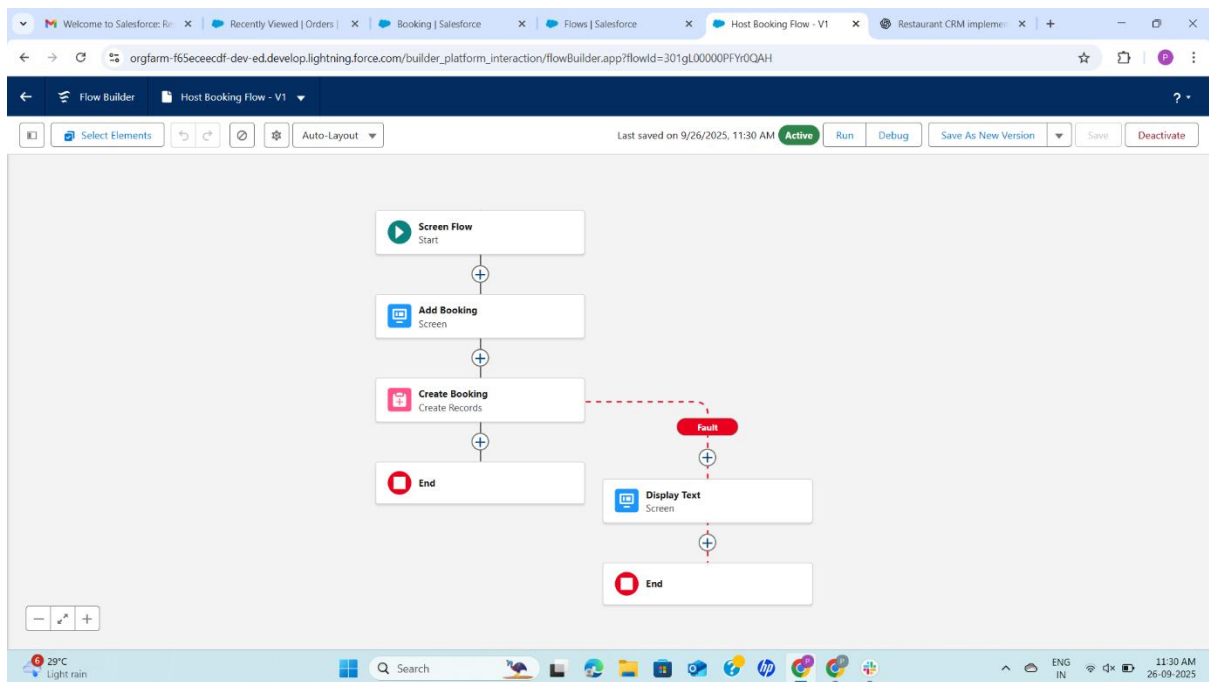
Goal: tablet-host enters booking, flow creates Booking__c, assigns table (auto or manual), notifies manager.

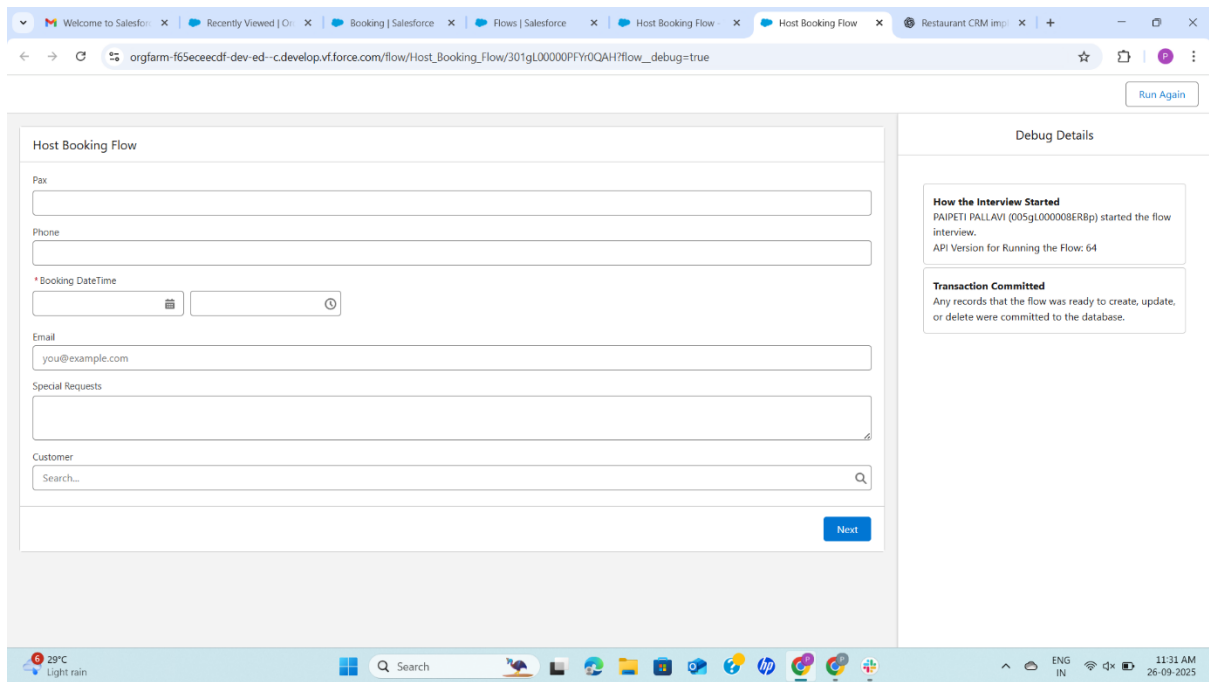
1. Setup → **Flow** → **New Flow** → choose **Screen Flow** → **Create**.
2. On canvas: + → **Screen**.
 - Label: Add Booking
 - Add components:
 - **Lookup** (Customer) → API/field: Customer__c (set Required)
 - **Number** (Pax) → Pax__c (Required)
 - **Date/Time** → Booking_DateTime__c (Required)
 - **Text Area** → Special_Requests__c
 - **Phone & Email** fields if needed
 - Save screen.
3. **Create Records** element:
 - Label: Create Booking Record
 - Create one Booking__c. Map screen fields to Booking__c fields (Customer__c ← {!Customer}, etc.).
4. **Decision** element: Table Available?
 - Option A: If you have a Table__c object, add a **Get Records** to find a table with Capacity__c >= {!Pax} and Is_Available__c = TRUE. If one found → route to Auto Assign.
 - Option B: If complex logic required, call an **Apex Invocable** (add **Action** → the invocable class) which returns Table Id.
5. **Screen** Confirm:
 - Show assigned table or a record choice to pick table manually. (Use Record Choice Set or Choice component bound to Table__c results.)

6. **Action(s):**

- Add **Create Task** or **Send Custom Notification** (Action → Send Custom Notification) for manager.
 - Or use **Action** → **Email Alert** (if you created an Email Alert) to send email.
7. Save → **Debug** with sample values → fix mapping.
8. Save → **Activate**.
9. Expose Flow:
- Add to Lightning App page (App Builder) or create a Quick Action that launches the flow (Object → Buttons, Links, Actions → New Action → Flow). For tablet, add to the app page or make it full-screen.

Testing: on a device, run flow, confirm Booking__c created, table assigned, notifications received.





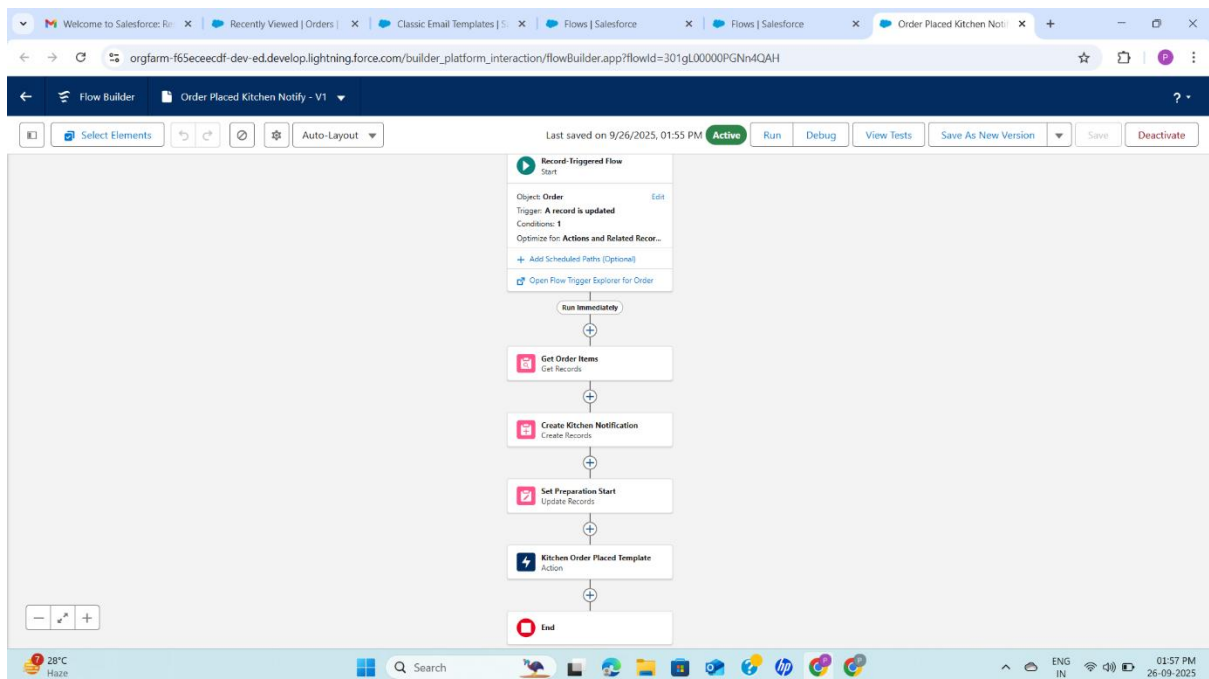
Record-Triggered Flow

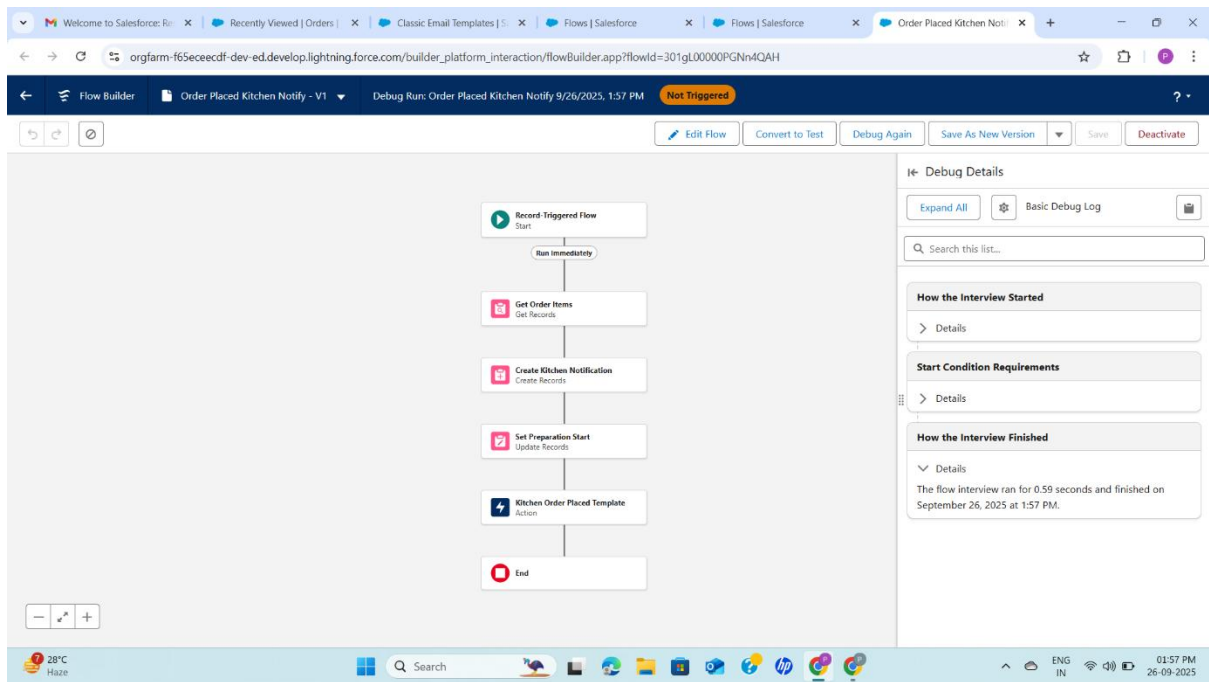
Goal: When Order__c.Status__c becomes **Placed**, notify kitchen, set Preparation_Start__c.

1. Setup → **Flows** → **New Flow** → **Record-Triggered Flow** → **Create**.
2. Configure start:
 - **Object:** Order__c
 - **Trigger:** A record is **updated**
 - **Condition Requirements:** Status__c EQUALS Placed (use picklist equals).
 - **When to Run:** After the record is saved (we will create other records & notifications).
 - **Optimize the Flow for:** Actions and Related Records.
 - Click **Done**.
3. First **Get Records**:
 - Label: Get Order Items
 - Object: Order_Item__c
 - Filter: Order__c = \$Record.Id
 - Get all records, store as collection.
4. (Optional) **Decision**: If no order items → end (avoid empty operations).
5. **Create Records**: create Kitchen_Order__c (or your custom notification record) — you can create one record per Order or one per item.
 - For large sets, create one summary Kitchen_Order and attach items via related list.
6. **Update Records**:
 - Label: Set Prep Start
 - Update record: choose the triggering record (Order__c where Id = \$Record.Id) → set Preparation_Start__c = \$Flow.CurrentDateTime (or use formula NOW()).
7. **Action**: Send Notification

- Option A: **Send Custom Notification** action → choose Notification Type (create one first) and set Recipient IDs (Queue/Users).
 - Option B: Use **Action** → **Email Alert** (pre-created Email Alert).
 - Option C: **Post to Chatter** action: Post to a group or user.
8. **Add Scheduled Path** (if you want follow-up/reminder or auto-cancel):
- In the Start element, click + **Add Scheduled Paths**, set label Auto Cancel If Not Confirmed, offset 24 Hours After Trigger (or X hours), then on that scheduled path add a Decision: if Order__c.Confirmed__c = FALSE → **Update Records** to set Status__c = 'Cancelled' and send email.
9. Save → **Debug**: run with a test order using “Run” debug in flow (use record Id).
Review Flow Interviews in Setup → Paused Flow Interviews (if scheduled).
10. Activate.

Testing: Update an Order to Placed → check Kitchen_Order created, Preparation_Start updated, notifications emailed.





Scheduled Flow

Goal: nightly job to process Loyalty_Account__c and update balances.

1. Setup → **Flows** → **New Flow** → **Scheduled-Triggered Flow** → **Create**.
2. Configure schedule:
 - **Start Date & Time**: pick next run time (e.g., tomorrow 02:00 AM)
 - **Frequency**: Daily
 - **Time Zone**: Org default (confirm timezone)
3. Canvas:
 - **Get Records**: Loyalty_Account__c → filter criteria as needed (e.g., Active = TRUE). **Store all records** into a collection.
 - **Loop** over the collection (Loop variable: loopLoyaltyAccounts).
 - In loop: **Decision** → has expiry? has activity?
 - If points to add: Add a record to a collection variable transactionsToCreate (build a record variable using Assignment).
 - Also use Assignment to compute new Points_Balance__c into a accountsToUpdate collection (create collection of records to update).
 - After Loop: **Create Records** (create many Loyalty_Transaction__c using transactionsToCreate) — this uses bulk create.
 - **Update Records** — use accountsToUpdate collection to bulk update balances.
4. Save → Debug (use a small test dataset) → Activate.
5. **Scale note**: If your org has thousands+ accounts, consider Batch Apex or break the scheduled flow to process smaller slices (e.g., criteria by last modified ranges) to avoid platform limits.

Create Notification Type (for in-app)

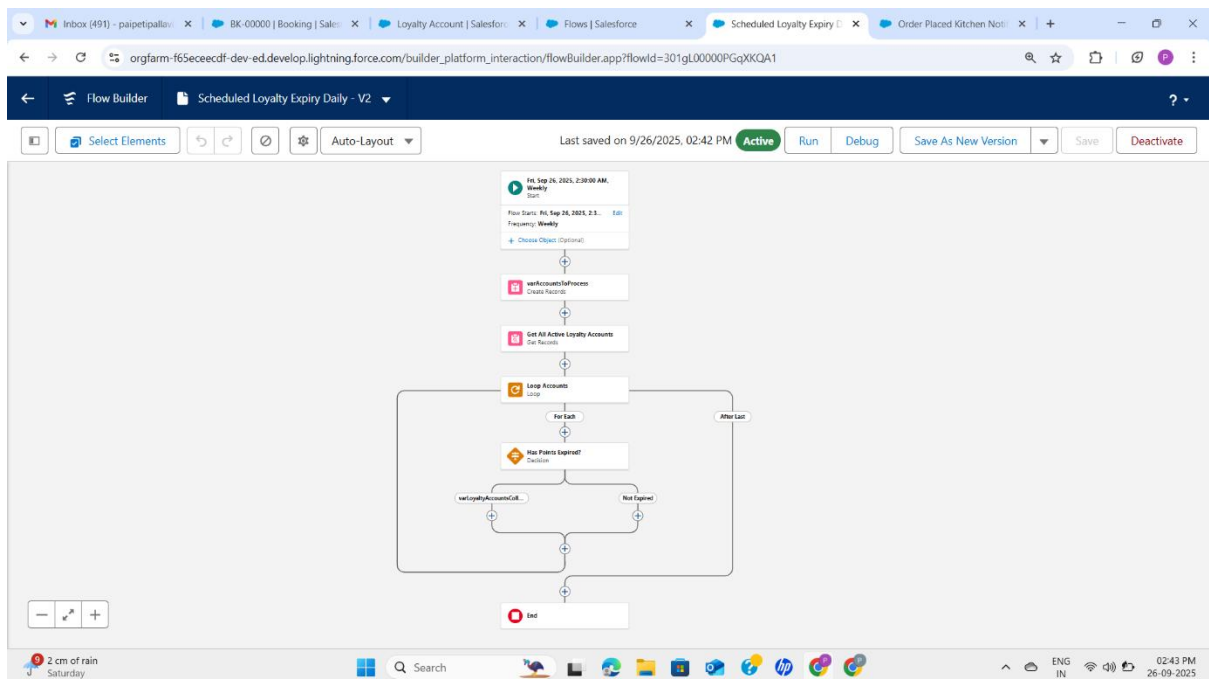
1. Setup → **Notification Builder** → **Notification Type** → **New**.
2. Label: Kitchen_Order_Alert, Channels: In-App and Desktop if needed. Save.

Create Email Template & Email Alert

1. Setup → **Email Templates** → **New Email Template** (Lightning) → create template using merge fields.
2. Setup → **Email Alerts** → **New Email Alert**:
 - Object: Order__c
 - Email Template: choose above
 - Recipients: User, Queue, Related Users (e.g., Order.Owner)
3. Use this Email Alert in Flow:
 - In Flow canvas: **Action** → search for **Email Alert** → choose your alert.

Send Custom Notification in Flow

1. In Flow canvas: **Action** → Send Custom Notification → choose Notification Type created earlier.
2. Supply Title, Body, Recipient Id(s) (use {!User.Id} or collection).



☐ Created a Booking record (Booking__c object)

- Record Name auto-generated: **BK-00000** (probably an Auto-Number field).

☐ Filled core Booking fields:

- **Contact** → Selected *Maria Clifton* (lookup to Contact).
- **Booking DateTime** → 9/27/2025, 12:00 PM.
- **Restaurant Table** → Linked to a Table record (lookup).
- **Status** → Set as *Pending* (picklist).
- **Account** → Left empty (lookup to Account).
- **Pax** → Empty (should store number of guests).
- **Phone, Email, Special Requests** → Empty (but available for input).

❑ **Record Owner** → You (Paipeti Pallavi).

- This means you are responsible for the booking.

❑ **Activity Panel** → No tasks/emails/events logged yet.

The screenshot displays a Salesforce Lightning interface for a booking record. The browser tabs at the top include 'Inbox (491) - paipeti.pallavi', 'BK-00000 | Booking | Sales', 'Classic Email Templates | S...', 'Flows | Salesforce', and 'Order Placed Kitchen Not...'. The address bar shows the URL: 'orgfarm-f65ececdf-dev-ed.develop.lightning.force.com/lightning/r/Booking_c/a00gl00000KSuqAQAT/view'. The page header features a search bar and navigation tabs: 'Navss resto', 'Orders', 'Bookings', 'Dashboards', 'Reports', 'Loyalty Accounts', and 'Order Items'. The main content area is titled 'Booking BK-00000' and includes buttons for 'New Contact', 'Edit', and 'New Opportunity'. The 'Details' tab is active, showing fields for 'Booking Name' (BK-00000), 'Contact' (Maria Clifton), 'Booking DateTime' (9/27/2025, 12:00 PM), 'Restaurant Table' (Table), 'Status' (Pending), 'Account', 'Pax', and 'Phone'. The 'Owner' field is set to 'PAIPETI PALLAVI'. The 'Activity' panel on the right shows filters for 'All time', 'All activities', and 'All types', with options to 'Refresh', 'Expand All', and 'View All'. It also displays a section for 'Upcoming & Overdue' with a message: 'No activities to show. Get started by sending an email, scheduling a task, and more. No past activity. Past meetings and tasks marked as done show up here.' The bottom of the screen shows a Windows taskbar with a search bar, application icons, and system status indicators including 'Rainy days ahead 28°C', 'ENG IN', and the time '02:09 PM 26-09-2025'.