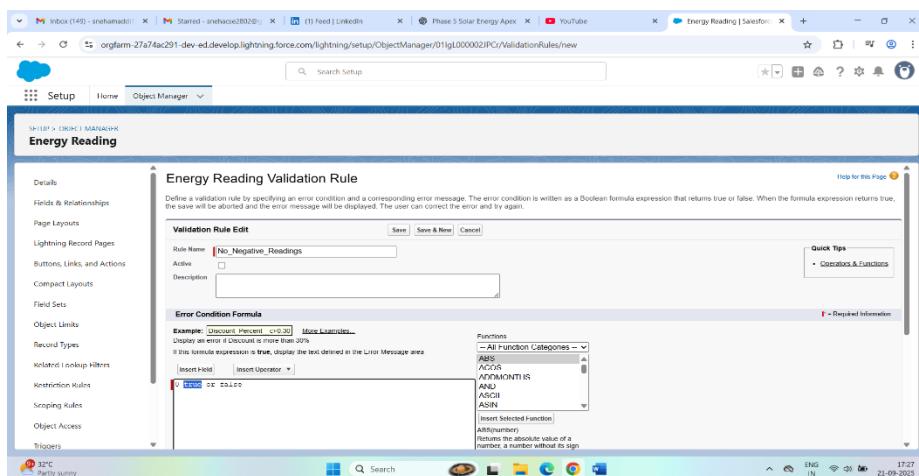


Phase 5: Apex Programming – Solar Energy Management System

◆ Step 1: Validation Rule (Prevent Wrong Data)

1. Go to **Setup** → **Object Manager** → choose **Energy Reading** object.
2. On the left, click **Validation Rules** → then **New**.
3. Rule Name: **No_Negative_Readings**.
4. Enter Formula:
5. **Energy_Produced__c < 0**
6. Error Message: “*Energy produced cannot be negative.*”
7. Error Location: Field → Energy Produced.
8. Save and **Activate**.



9.

◆ Step 2: Record-Triggered Flow (Auto Update Total Energy)

1. Go to **Setup** → **Flows** → **New Flow**.
2. Choose **Record-Triggered Flow** → Click **Create**.
3. Configure Start:
 - Object: **Energy Reading**
 - Trigger: **When a record is created**
 - Run Flow: **After Save** (because we’re updating another record).
4. Add Element → **Get Records**:
 - Label: Get Related Project

- Object: **Solar Project**
- Condition: $\text{Id} = \{!\$Record.\text{Solar_Project_c}\}$
- Store: First Record Only.

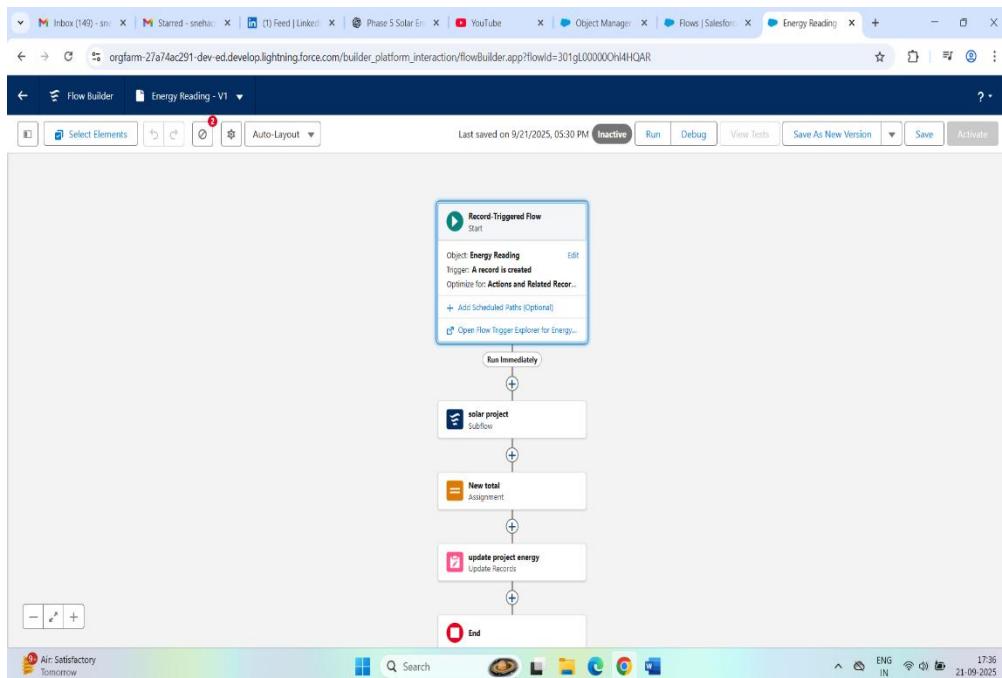
5. Add Element → Assignment:

- Label: Increase Total Energy
- Formula: $\text{NewTotal} = \text{Project.Total_Energy_c} + \$Record.\text{Energy_Produced_c}.$

6. Add Element → Update Records:

- Record: Project from Get Records.
- Field: $\text{Total_Energy_c} = \text{NewTotal}.$

7. Save as: Update Project Energy → Activate.

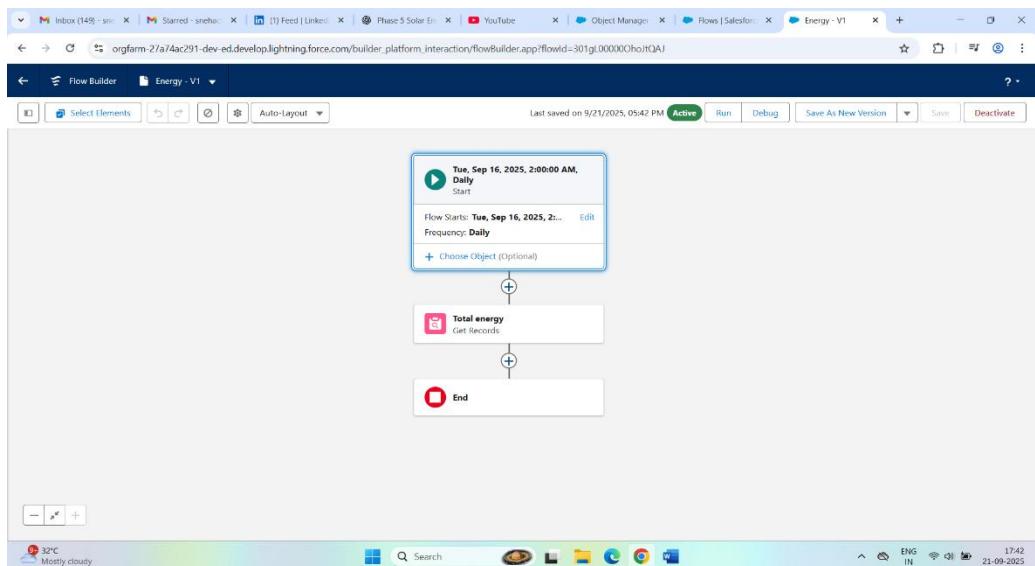


8.

◆ Step 3: Scheduled Flow (Daily Recalculation for Accuracy)

1. Setup → Flows → New Flow.
2. Choose **Scheduled-Triggered Flow**.
3. Set Frequency: **Daily** → Run Time: **2:00 AM**.
4. Add Element → **Get Records**:

- Label: Get All Projects
 - Object: Solar Project
 - Store All Records.
5. Add Element → **Loop** → Loop through Projects.
6. Inside Loop:
- Add **Get Records**: All Energy Readings for that project.
 - Add **Assignment**: Sum all readings into a variable.
 - Update the project's Total_Energy__c.
7. Save as: Nightly Energy Totals → **Activate**.



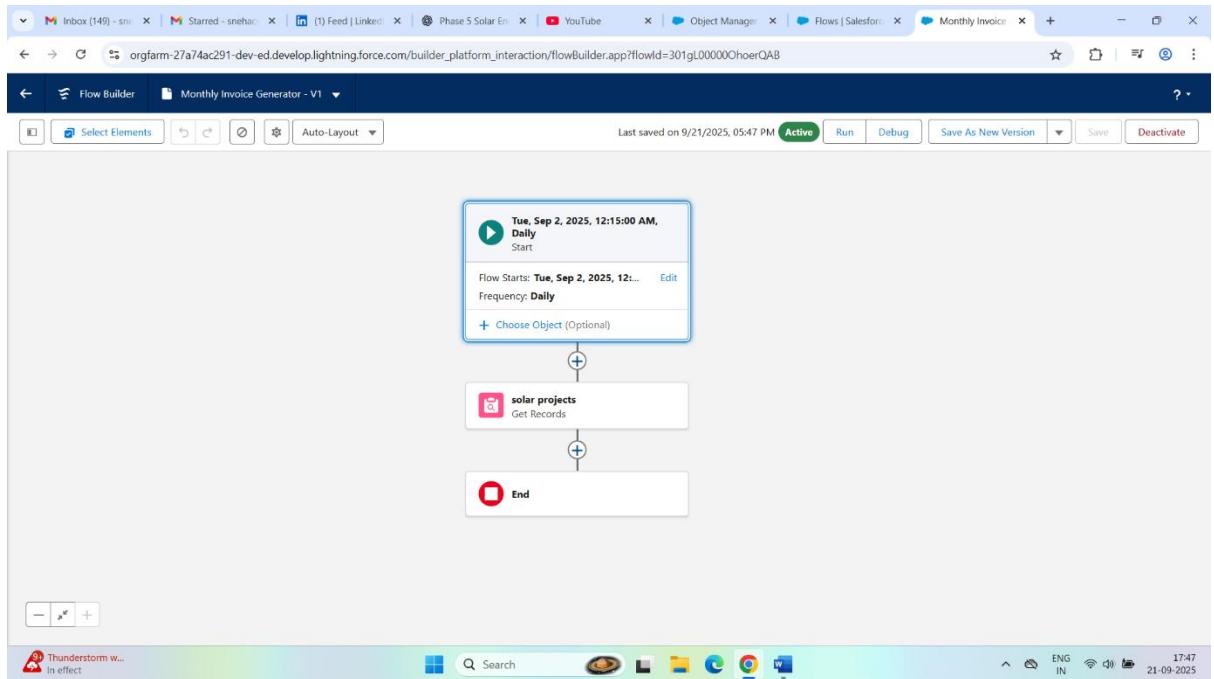
8.

◆ Step 4: Monthly Invoice Automation

1. Setup → **Flows** → **New Flow** → **Scheduled-Triggered Flow**.
2. Frequency: **Monthly**, Day = 1, Time = 12:05 AM.
3. Add Element → **Get Records**:
 - Object: Solar Project
 - Filter: Active = TRUE.
4. Add **Loop** → for each Project:
 - Create **Invoice** record:
 - Solar_Project__c = {!Loop_Project.Id}

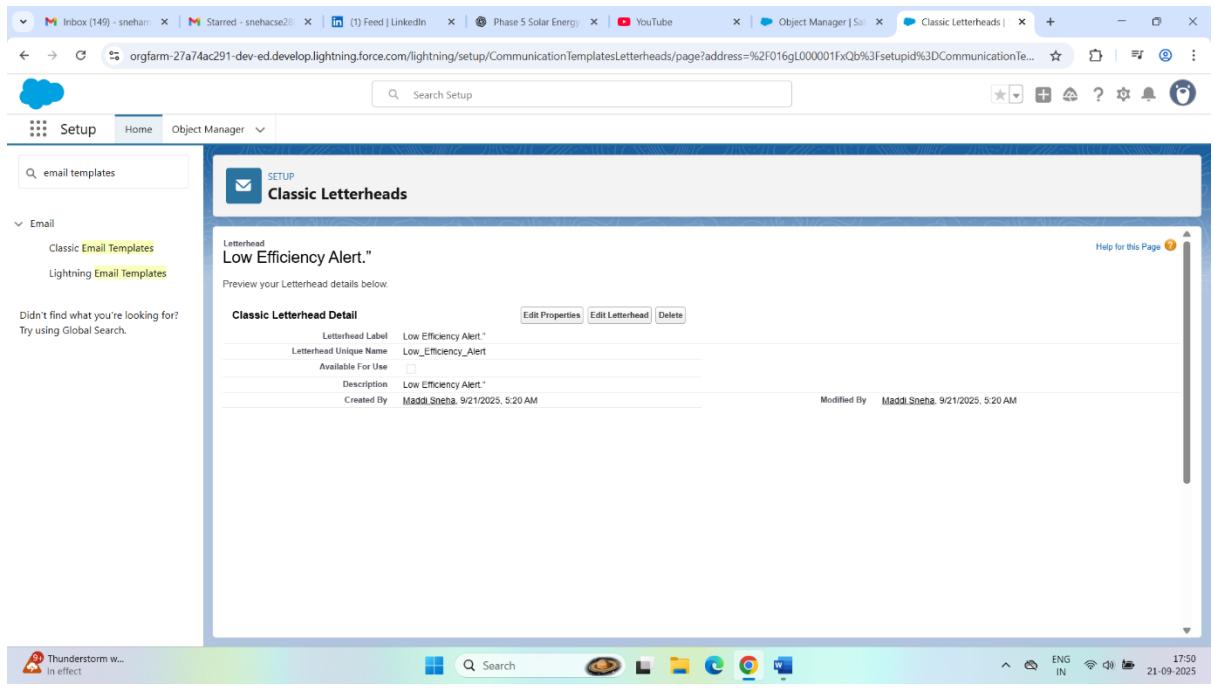
- `Period_Start_c` = First day of last month
- `Period_End_c` = Last day of last month
- `Amount_c` = `Loop_Project.Total_Energy_c * Tariff_c` (tariff = price/kWh).
- `Status_c` = Draft.

5. Save as: Monthly Invoice Generator → Activate.



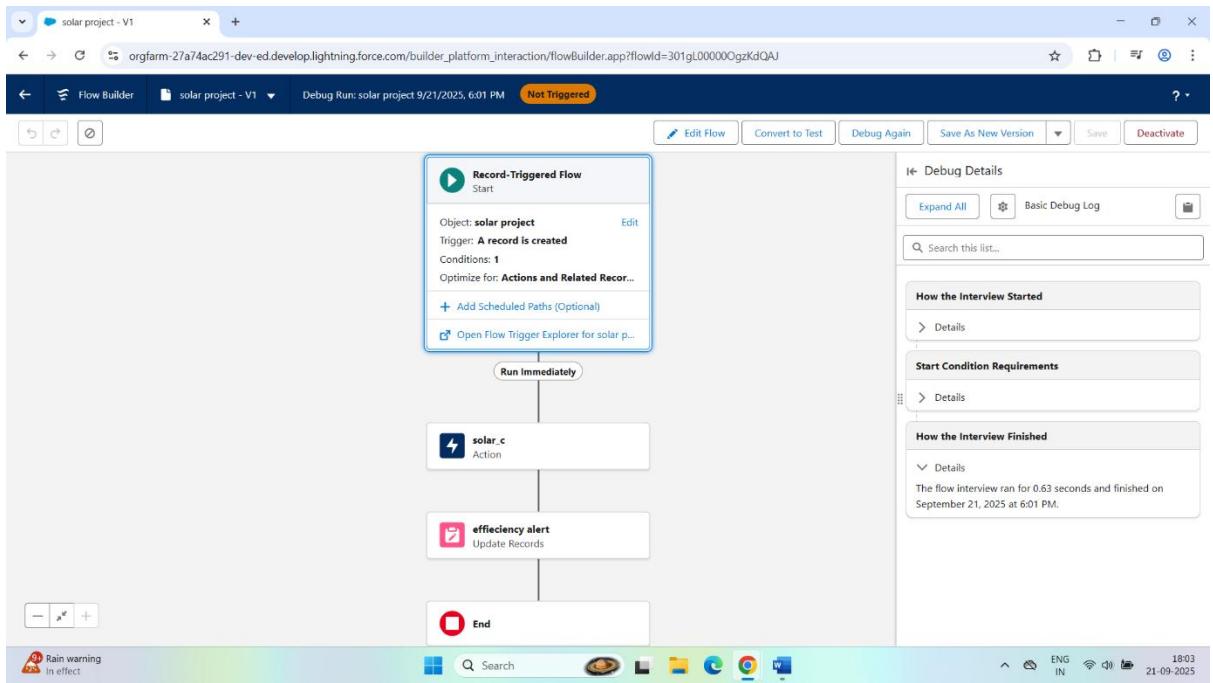
◆ Step 5: Email Alerts for Low Efficiency

1. Setup → **Email Templates** → Create a template: “*Low Efficiency Alert*”.



2.

3. Setup → Flows → New Flow → Record-Triggered Flow.
4. Object: Solar Project.
5. Trigger: When record is **created or updated**.
6. Condition: Efficiency_c < 60 AND Maintenance_Flag_c = FALSE.
7. Add Action → **Send Email**: To Maintenance Manager.
8. Add Action → **Update Records**: Set Maintenance_Flag_c = TRUE.
9. Save as: Low Efficiency Alert → **Activate**.



Step 6: Reports & Dashboards

1. App Launcher → Reports → **New Report**.
 - Type: **Solar Project with Energy Readings**.
 - Add Columns: Project Name, Total Energy, Efficiency, Status.
 - Save: Solar Project Performance.
2. App Launcher → Dashboards → **New Dashboard**.
3. Add Components:
 - **Bar Chart**: Top 5 projects by total energy.
 - **Gauge Chart**: Average efficiency across all projects.
 - **Table**: Projects below 60% efficiency.
 - **Line Chart**: Monthly revenue trend (from invoices).