**Property Management Application Using Salesforce - (Developer) – ( Long-Term)**

ID:  LTVIP2024TMID11592

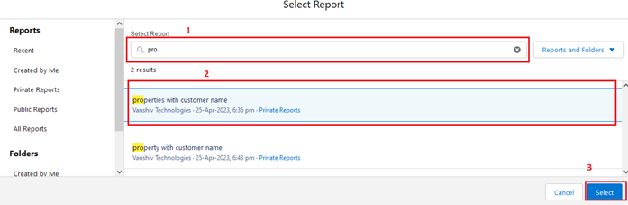
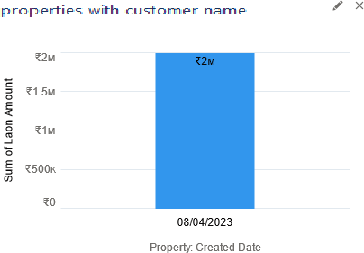
**Team leader:** N.INDHUMATHI

**Team member:**  S.SHREEDEVI

**Dashboards**

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you’ve gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

**Create Dashboards**

1.Click the Dashboards tab.  
2.Click New Dashboard.  
3.Name the Properties with Customer Name Report and click Create.  
4.Click +Component.  
5.Select the Properties with Customer Name Report and click Select  
  
6.Select the Vertical Bar Chart component(select in which format you want display chart and click Add.  
7.Click Save and then Done.  


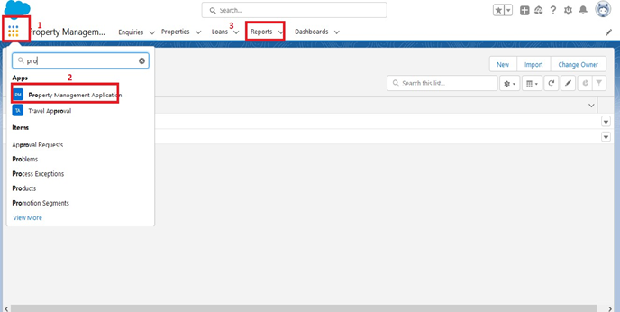
**Create Dashboards**

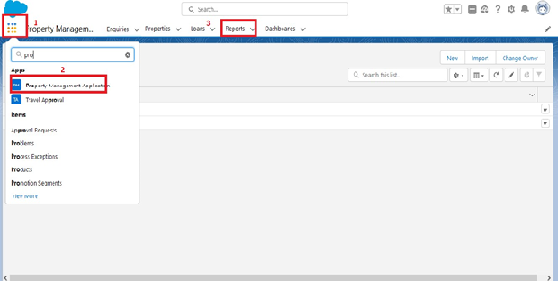
Follow the above steps  
1.Create the Dashboard for the Same Take Any Type of Dashboard(Chart) And Display Iton The App Home Page

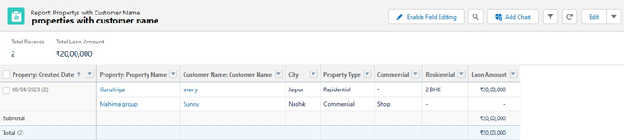
### View Report And Dashboard

**Report**

1. Click on App Launcher on left side of screen
2. Search property management & click on it.
3. Click on Reports Tab.
4. Click on Properties with Customer Name & see records



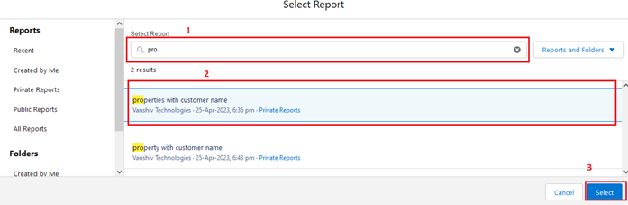
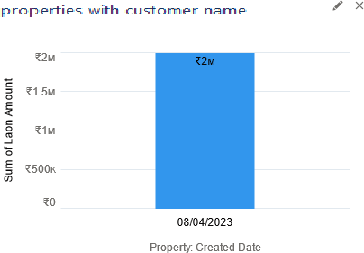




**Dashboard**

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you’ve gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

**View Dashboard**

1.Click the Dashboards tab.  
2.Click New Dashboard.  
3.Name the Properties with Customer Name Report and click Create.  
4.Click +Component.  
5.Select the Properties with Customer Name Report and click Select  
  
6.Select the Vertical Bar Chart component(select in which format you want display chart and click Add.  
7.Click Save and then Done.  


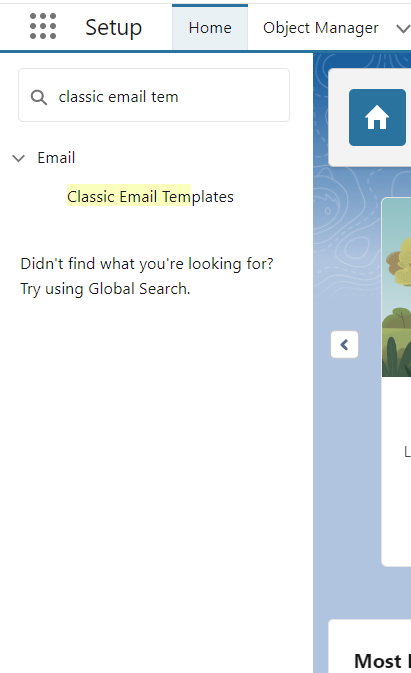
**Flow Builder**

Flows In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data.

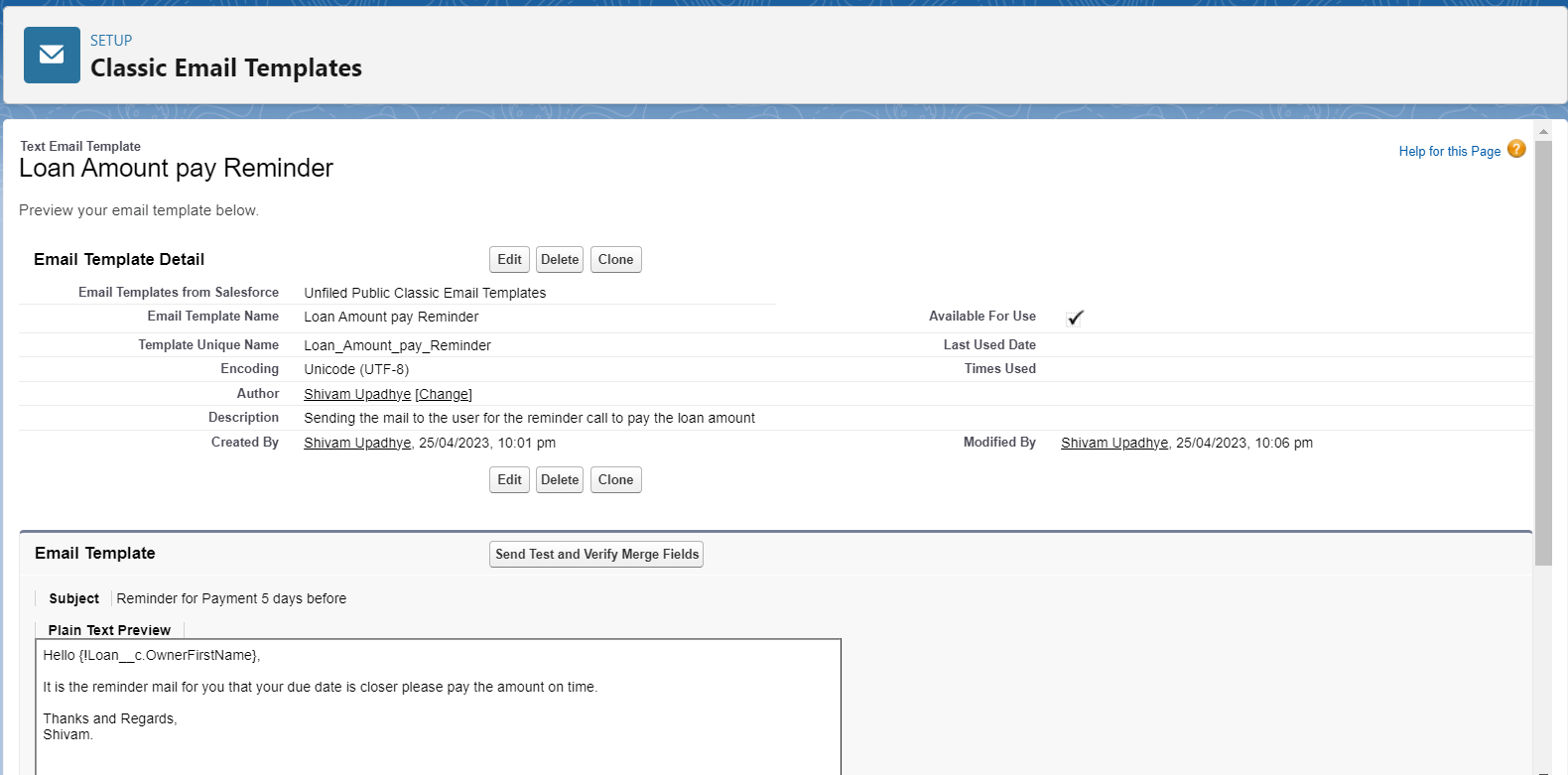
Flow Builder is the declarative interface used to build individual flows.

Flow Builder can be used to build code-like logic without using a programming language. Flows fall into five categories:

* Screen Flows
* Schedule-Triggered Flows
* Autolaunched Flows
* Record-Triggered Flows
* Platform Event-Triggered Flows
* **Classic Email Template**
* **Classic Email Template**
* **Note**: For creating this flow you have to create the Due date for Loan Payment
* field in the loan object with date and time field data type
* Go to the Gear Icon?Click on the home button and Search for the Classic Email Template
* ?Click on the New Template? name?oan Amount Pay Reminder



* Author as the System admin user?give Description?Reminder Calls through emails



Clone that Email Template and follow the steps which is mentioned above and give the following names you have to clone it multiple times that template

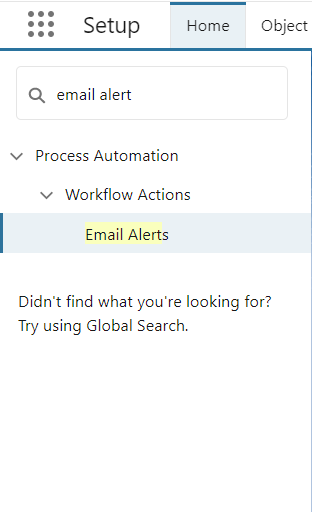
1. Loan Amount pay Reminder for 24 hrs

2. Overdue by one day

3. Overdue

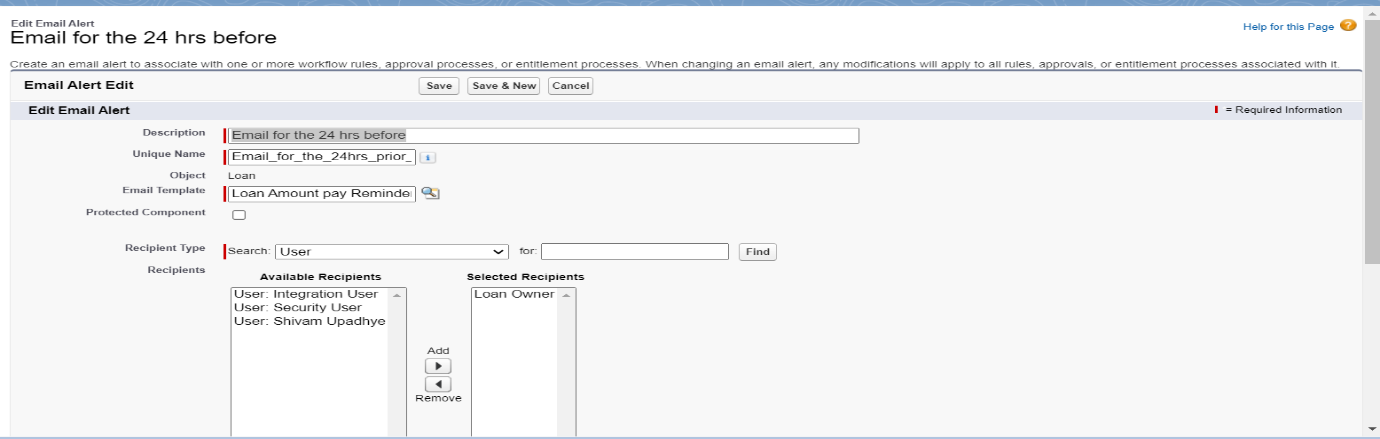
**Create The Email Alerts**

* Click on the home button and search for the Email Alerts  ---> There Click on the New Email alerts
* ---->and give the name as Email For the 24 hrs before---> select the email template which you have
* created for the 24 hrs before



There Click on the New Email alerts---->  and give the name as Email For the 24 hrs before

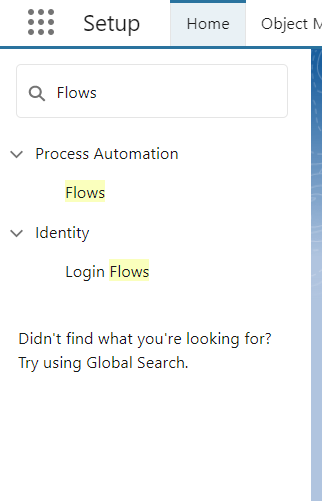
---->select the email template which you have created for the 24 hrs before and recipient

* for all condition is owner
* 
* Follow the Above steps and create the Following Email Alert by cloning with the Similar steps
* 1. Loan Amount Pay Reminder(Cloned Email Alert)
* 2. Overdue by one day(Cloned Email Alert)
* 3. Overdue(Cloned Email Alert)

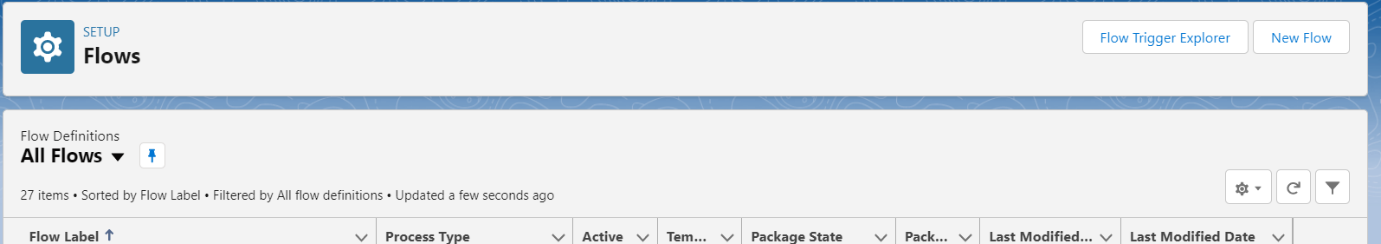
### 

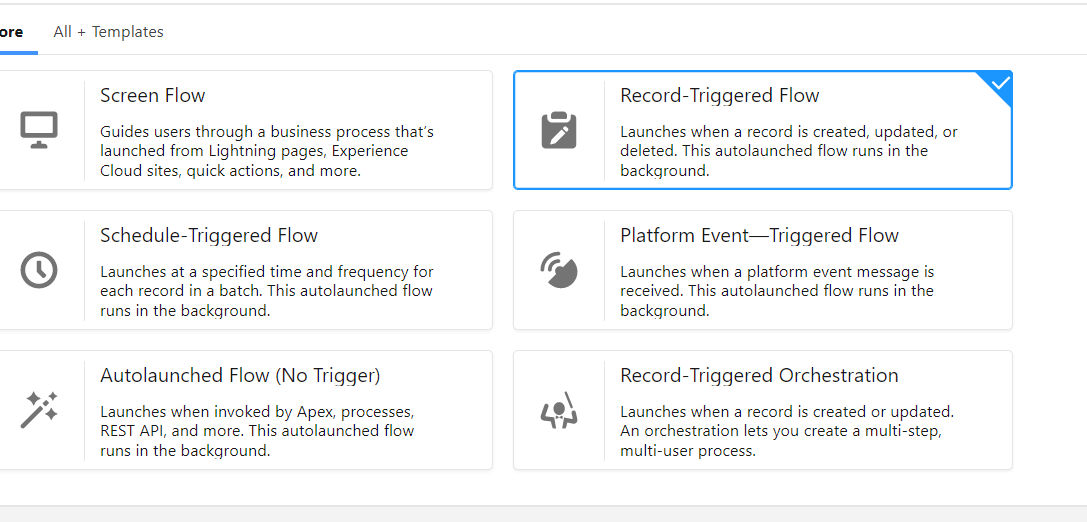
### Create The Flows

* Go to the Home Button and search for the flow

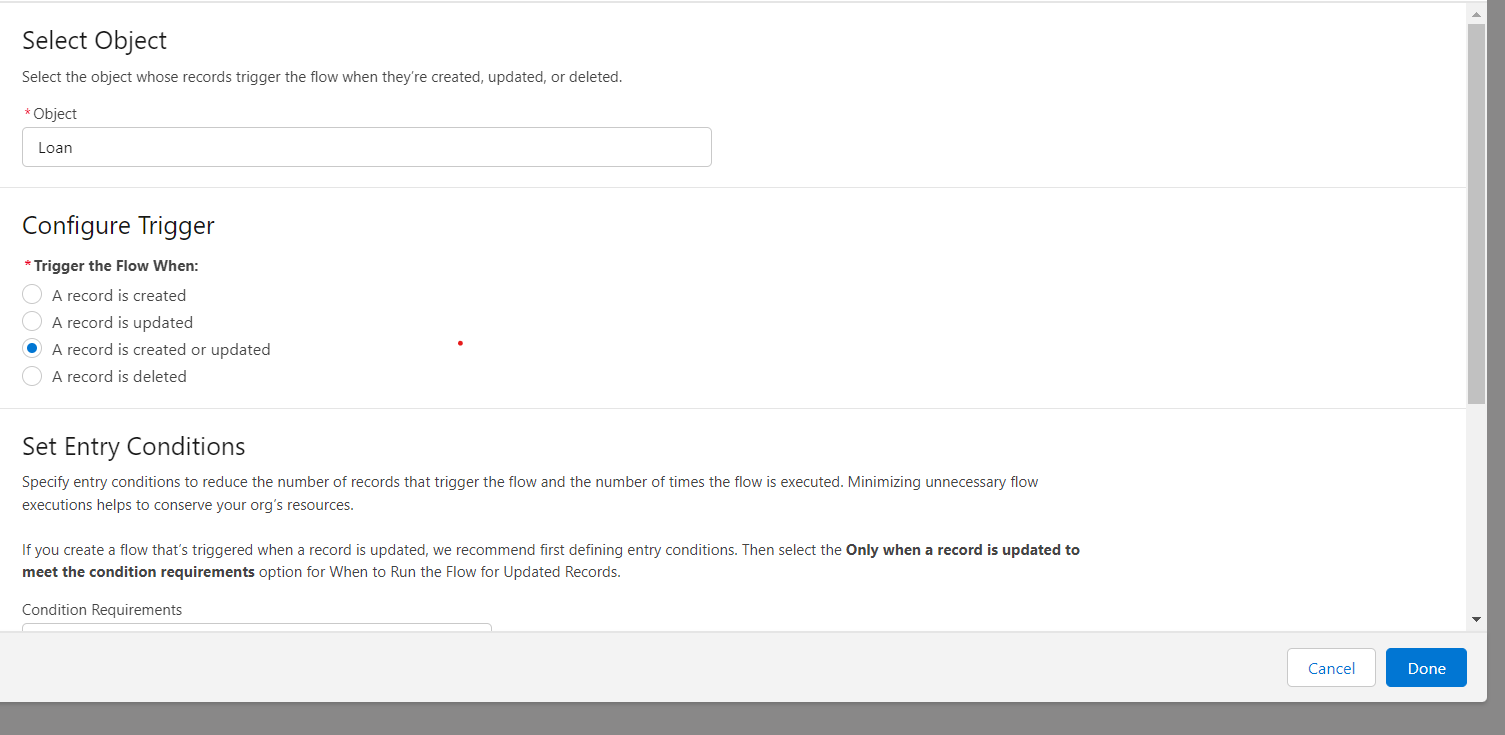


* Click on the New Flow---->Click on the Record Trigger Flow

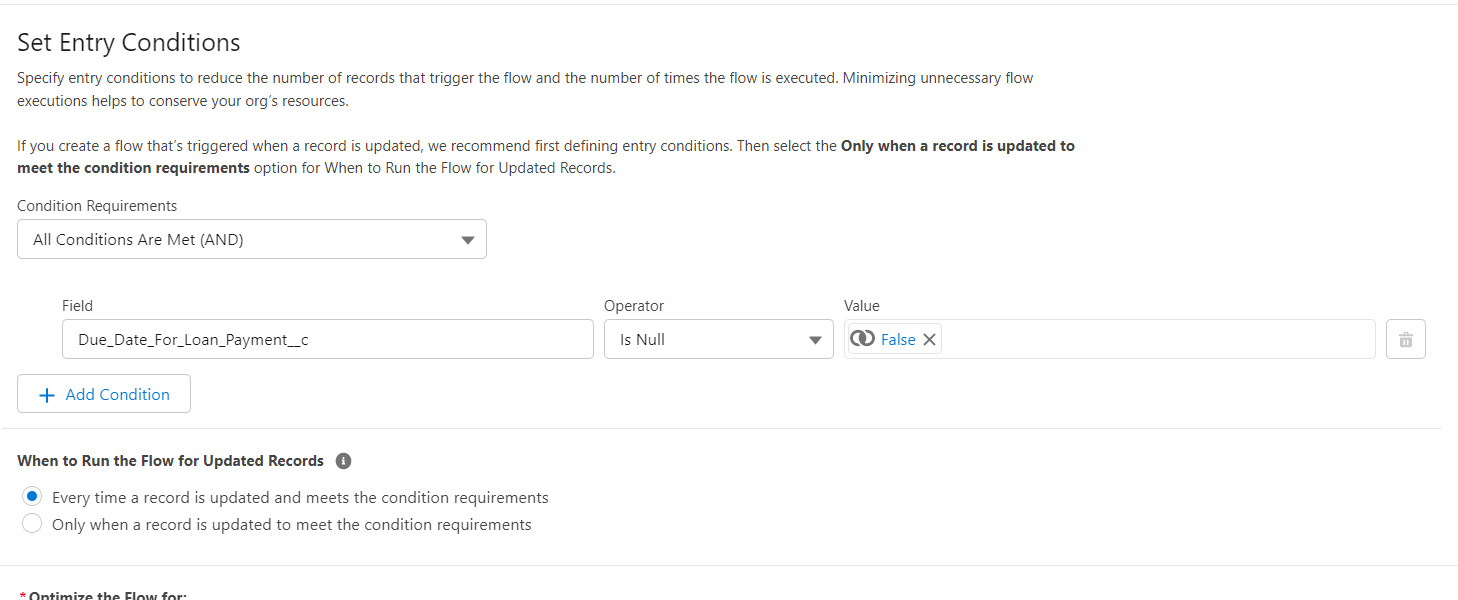




* Select the loan\_\_c object----> Trigger the Flow When----> A Record is created or updated



* Condition Requirement?All Conditions are met (AND) ----> Field---->Due Date For the Loan PaymentOperator as --->is Nullvalue False

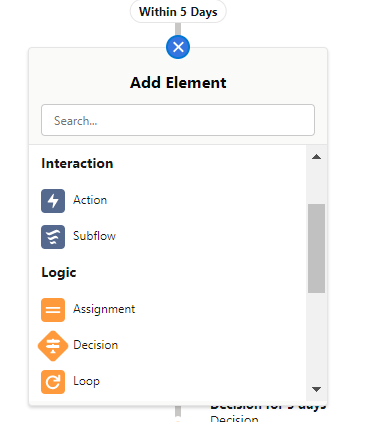


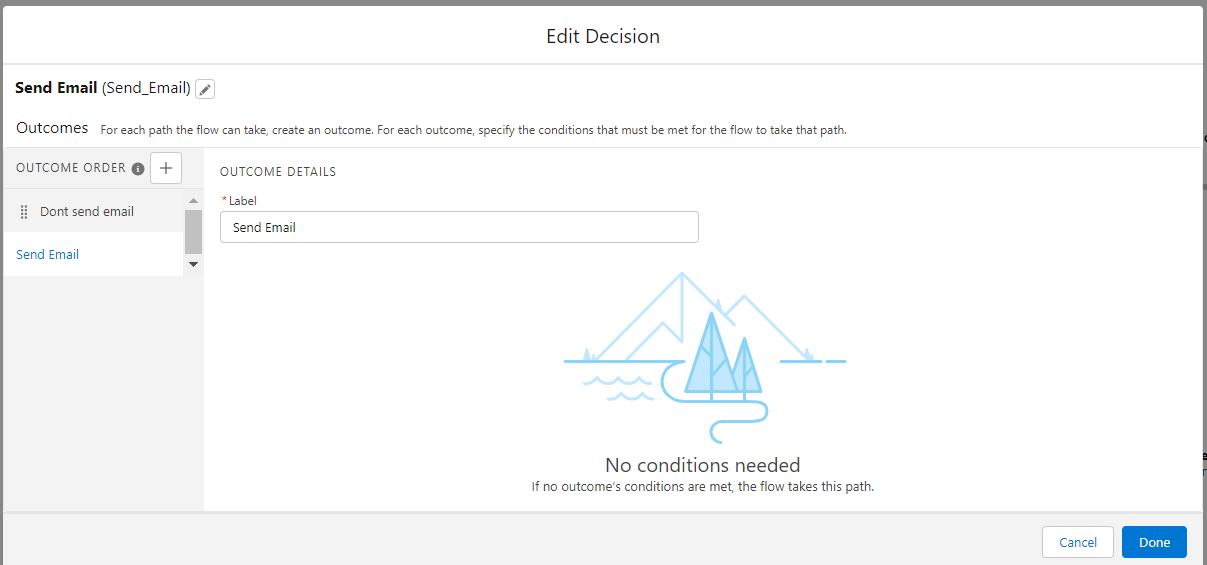
* Click on the Schedule path----> Path Label as---->within 1 Day----> API Name within\_1\_day?

               Time Source? Loan\_\_c:Due Date Loan payment ---> Offset number 24----> offset option

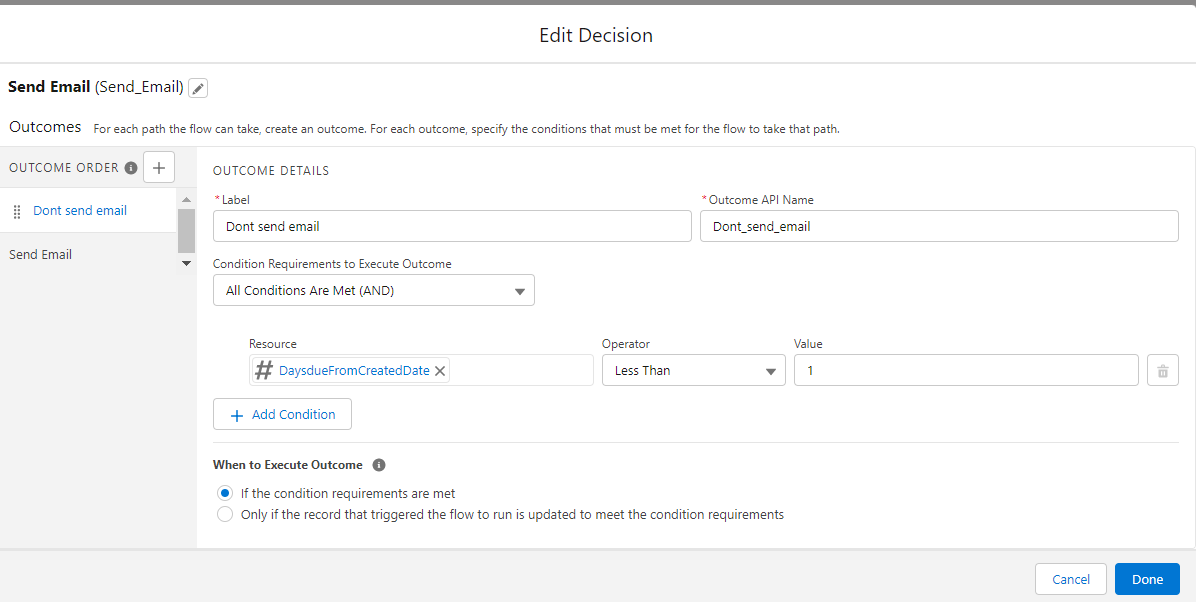
### Decision Element

* Than There is Decision for the just click on add----> select the Decision--->order outcomet--->here are two outcome 1. Send Email 2. don’t send the email



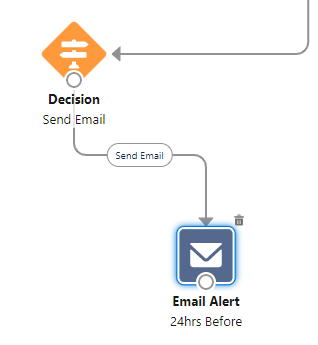
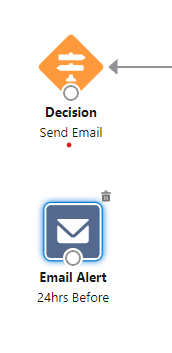


* On the outcome is there is the condition for that label don the send email ----> condition Requirement All Conditions are met (AND) ? Operator less than 1



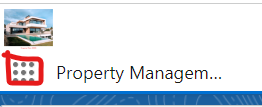
* Follow the same steps for the Following
* On the outcome is there is the condition for that label done the send the email---> condition Requirement All Conditions are met (AND) ----> Operator less than 5
* On the outcome is there is the condition for that label Dont send email for this ----> condition Requirement All Conditions are met (AND) ----> Operator Greater than 0
* On the outcome is there is the condition for that label Dont Send the Email for this ----> condition Requirement All Conditions are met (AND) ----> Operator Greater than 1

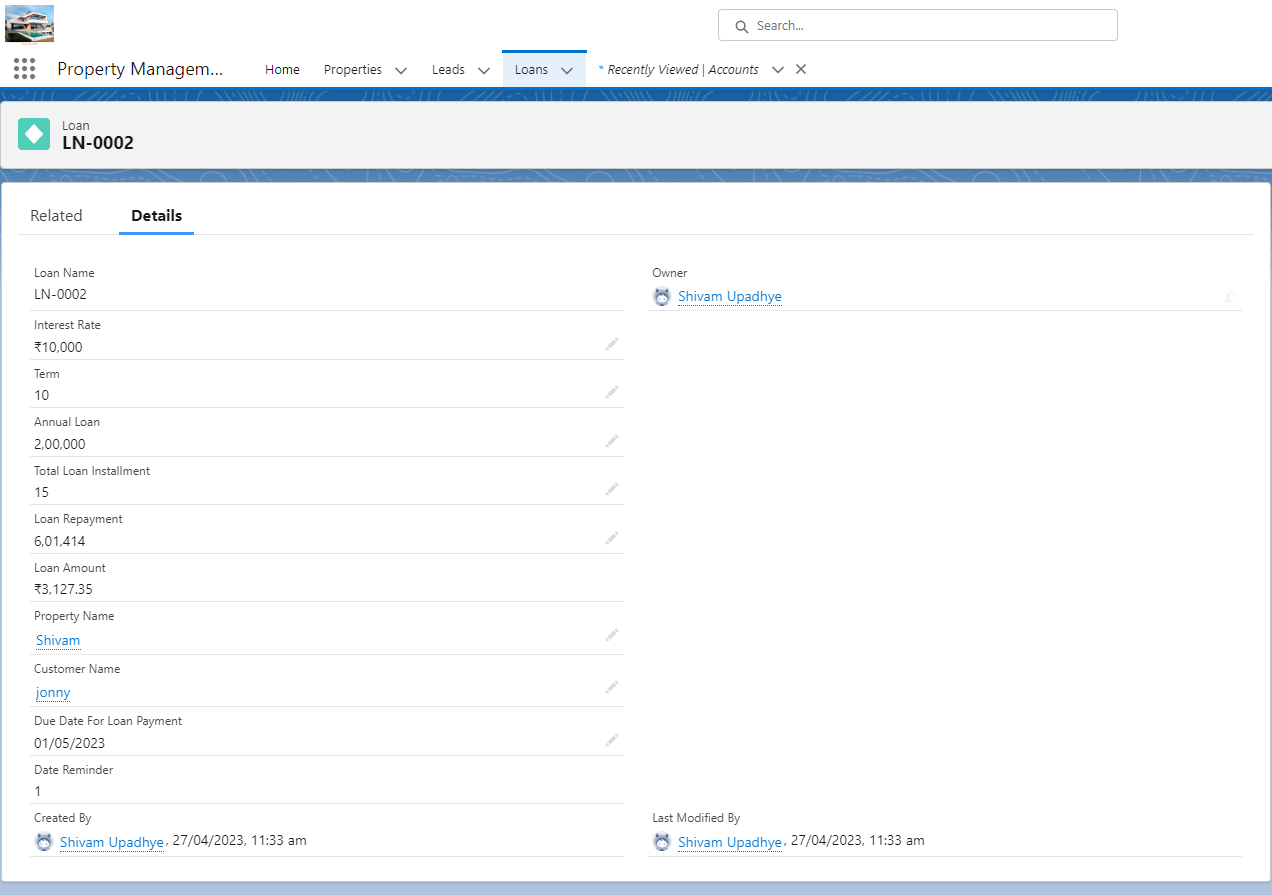
1. Go to the layouts change the auto layout with the free for delete both the thread for each decision and create one new connection thread similarly do for all threads and come back to auto layout

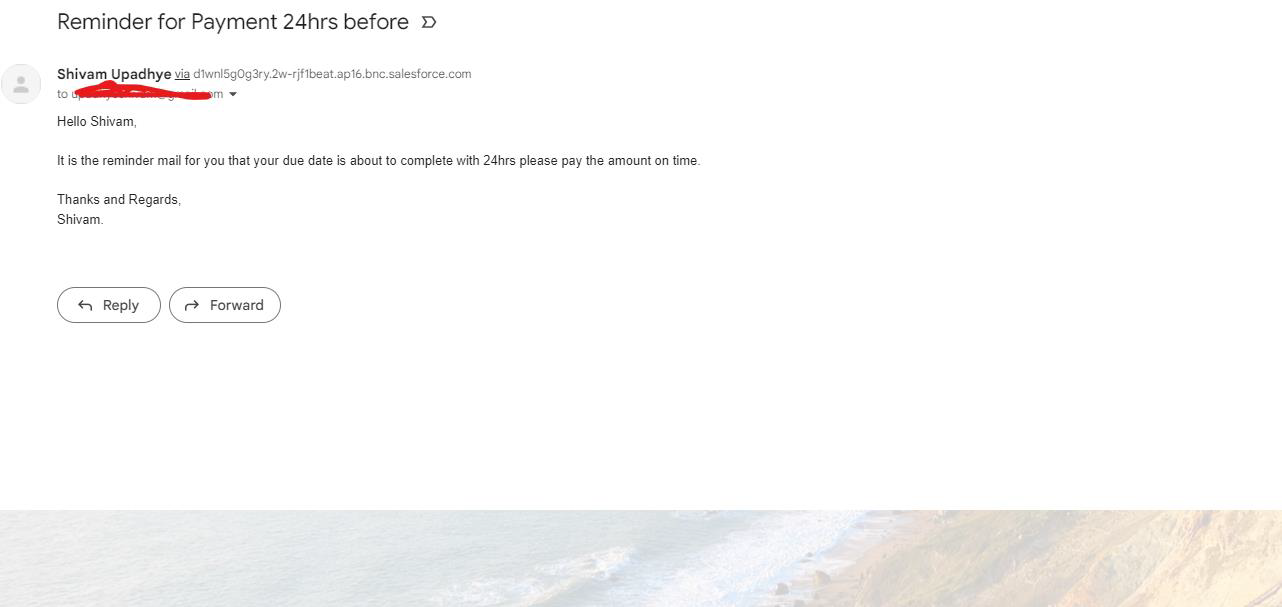


**Create The Record To Test The Flow**

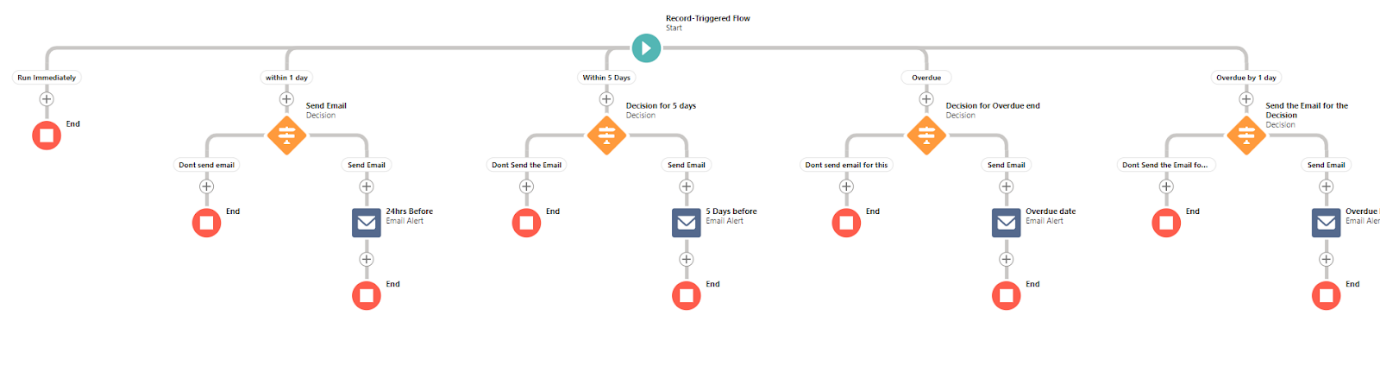
Go to the App Launcher and select the property management application than Go to the Loan Object create one record with the following values Go to you personal email you will get the mail for the selected date







Here you can see the mail for the 24 hrs before the condition is like duedate – created for the remaining days so it will trigger the email accordingly



### Apex Triggers

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

insert

update

delete

merge

upsert

undelete

For example, you can have a trigger run before an object's records are inserted into the database,

after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact

or an Account, some standard child objects, such as a CaseComment, and custom objects. To

define a trigger, from the object management settings for the object whose triggers you want

to access, go to Triggers.

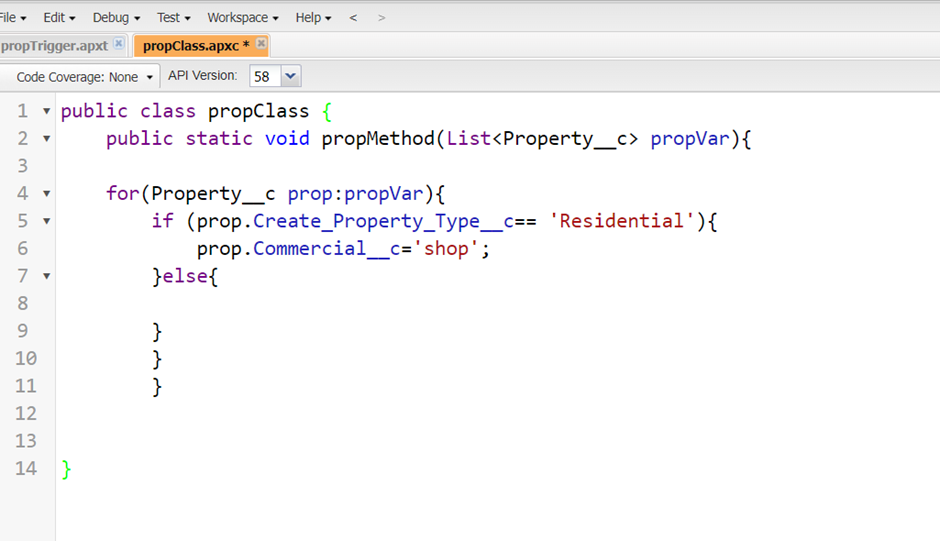
Use Case: When We Are Changing the property type form “Commercial” to “Residential” the

Than commercial field should with “Shop”

### Trigger

**Use Case:** Apex Trigger is Related  to Property Object in that there is the field “Create Property Type” which is having the picklist values in that field(Residential, Commercial, Industrial) the condition is like if we select the Create Property type as “Residential” than there is  Commercial field so it should get populated with “Shop”

**Apex Class:**

****

public static void propMethod(List<Property\_\_c> propVar){

public class propClass {

    for(Property\_\_c prop:propVar){

        if (prop.Create\_Property\_Type\_\_c== 'Residential'){

            prop.Commercial\_\_c='shop';

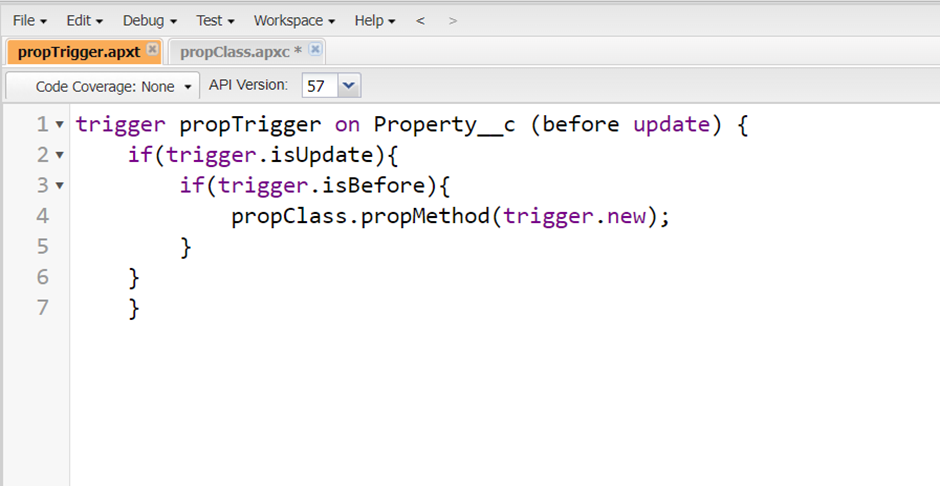
        }else{

        }

        }

        }

}



trigger propTrigger on Property\_\_c (before update) {

    if(trigger.isUpdate){

        if(trigger.isBefore){

propClass.propMethod(trigger.new);

    }

    }

    }

* Just Go to the Property Object and Check Whether Your Trigger Is Working or not as per the requirement