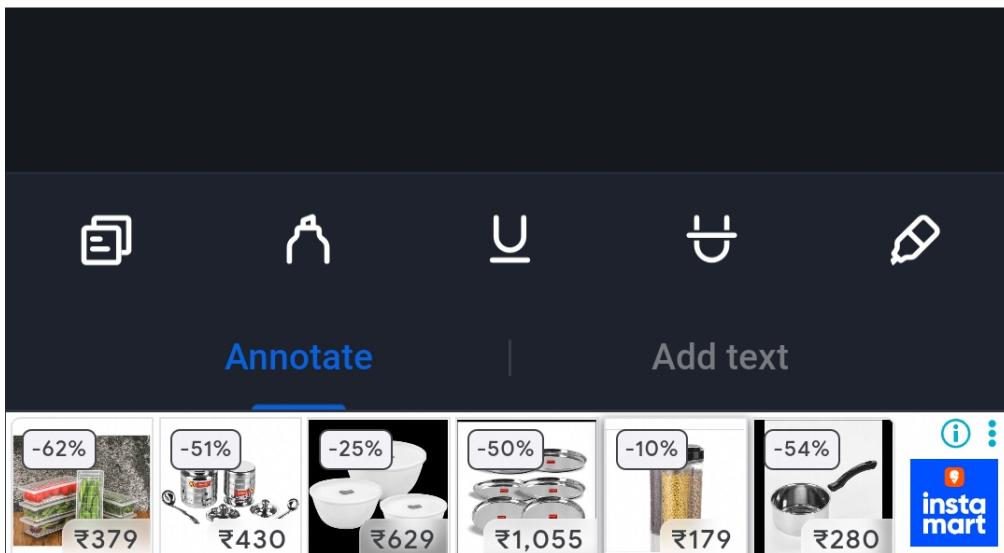
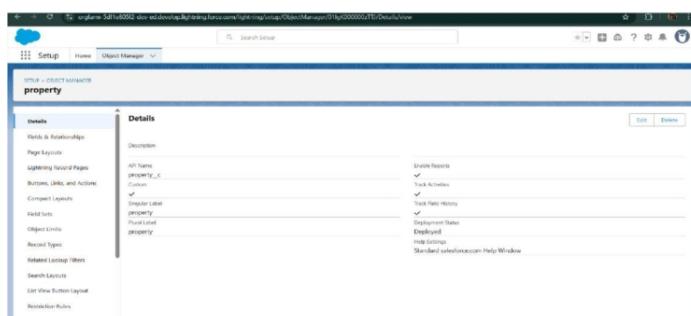


- Created objects: Property, Tenant, Lease, Payment



[SETUP](#) > [OBJECT MANAGER](#)

## Tenant

[Edit](#) [Delete](#)

**Details**

Description

API Name: Tenant\_c  
Custom: ✓  
Singular Label: Tenant  
Plural Label: Tenants

Enable Reports:

- ✓ Track Activities
- ✓ Track Field History
- ✓ Deployment Status: Deployed
- Help Settings
- Standard salesforce.com Help Window

[Fields & Relationships](#)

[Page Layouts](#)

[Lightning Record Pages](#)

[Buttons, Links, and Actions](#)

[Compact Layouts](#)

[Field Sets](#)

[Object Limits](#)

[Record Types](#)

[Related Lookup Filters](#)

[Search Layouts](#)

[List View Button Layout](#)

[Restriction Rules](#)

[Scoping Rules](#)

[SETUP](#) > [OBJECT MANAGER](#)

## lease

[Edit](#) [Delete](#)

**Details**

Description

API Name: Lease\_c  
Custom: ✓  
Singular Label: Lease  
Plural Label: Lease

Enable Reports:

- ✓ Track Activities
- ✓ Track Field History
- ✓ Deployment Status: Deployed
- Help Settings
- Standard salesforce.com Help Window

[Fields & Relationships](#)

[Page Layouts](#)

[Lightning Record Pages](#)

[Buttons, Links, and Actions](#)

[Compact Layouts](#)

[Field Sets](#)

[Object Limits](#)

[Record Types](#)

[Related Lookup Filters](#)

[Search Layouts](#)

[List View Button Layout](#)

[Restriction Rules](#)

- Configured fields and relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(52768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User/Group)		<input checked="" type="checkbox"/>
property	property__c	Lookup(property)		<input checked="" type="checkbox"/>
property Name	Name	Text(60)		<input checked="" type="checkbox"/>
slug	slug__c	Text(18)		
Type	Type__c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER  
Payment for tenantat

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules

**Fields & Relationships**  
7 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check for payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)		✓

Setup Home Object Manager

SETUP > OBJECT MANAGER  
lease

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules Scoping Rules

**Fields & Relationships**  
7 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
property	property__c	Lookup(property)		✓
start date	start_date__c	Date		

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup[User]		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup[User]		
Owner	OwnerId	Lookup[User,Group]		✓
Phone	Phone_c	Phone		
status	status_c	Picklist		
Tenant Name	Name	Text(80)		✓

- Developed Lightning App with relevant tabs

**App Settings**

**App Details & Branding**

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

**App Details**

\* App Name: Lease Management

\* Developer Name: Lease\_Management

**App Branding**

Image:

Primary Color Hex Value: #00700D

Description: Application to efficiently handle the processes related to leasing real estate properties.

Org Theme Options:  Use the app's image and color instead of the org's custom theme

**App Launcher Preview**

**Lease Management**  
Application to efficiently handle the processes relate...

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

### App Settings

#### Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items	Selected Items
Accounts	Payment
Activation Targets	Tenants
Activations	property
All Sites	lease
Alternative Payment Methods	
Analytics	
App Launcher	
Appointment Categories	
Appointment Invitations	
Approval Requests	
Custom	

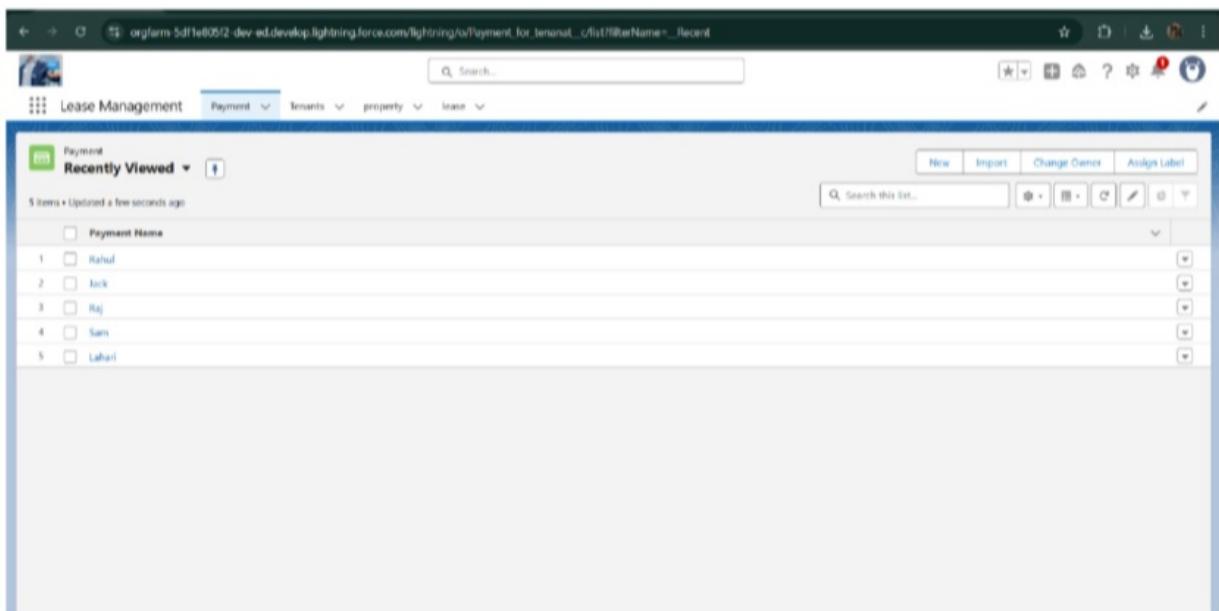
[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

### App Settings

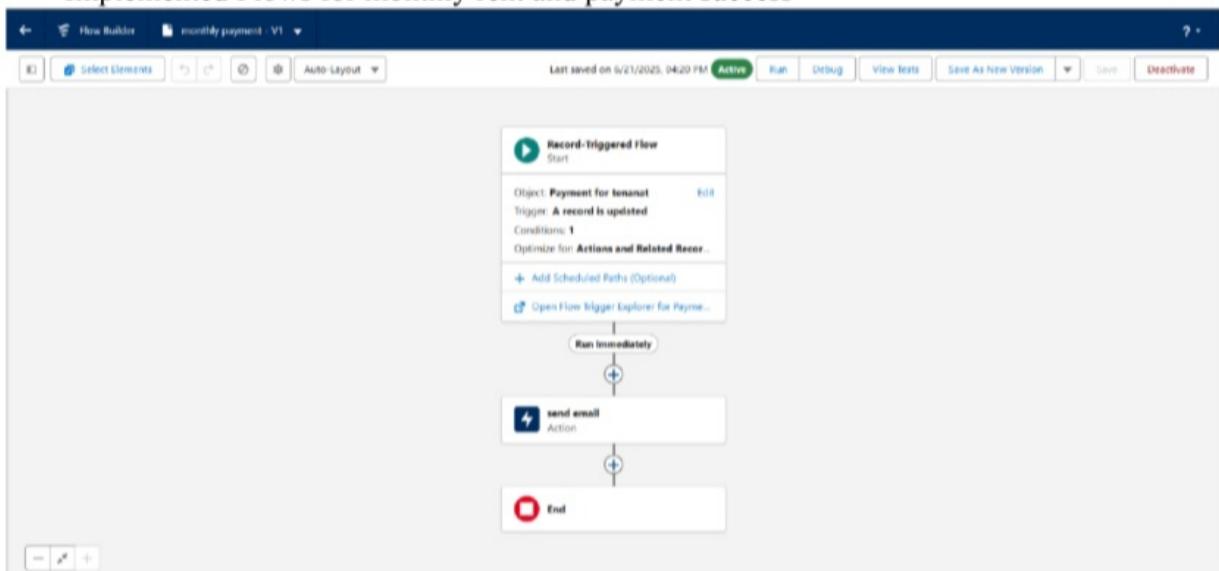
#### User Profiles

Choose the user profiles that can access this app.

Available Profiles	Selected Profiles
Analytics Cloud Integration User	System Administrator
Analytics Cloud Security User	
Anypoint Integration	
Authenticated Website	
Authenticated Website	
B2B Rendering Portal Buyer Profile	
Contract Manager	
Custom: Marketing Profile	
Custom: Sales Profile	
Custom: Support Profile	
Customer Community Login User	



- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

Validation Rule Edit

Rule Name: lease\_end\_date

Active:

Description:

Error Condition Formula:

```
Example: Discount_Percent_c>0.30 More Examples...
Display an error if Discount is more than 30%
If this formula expression is true, display the text defined in the Error Message area
```

Functions: All Function Categories ->

Insert Field: Insert Operator: End\_date\_\_c <= start\_date\_\_c

Check Syntax: Help on this function

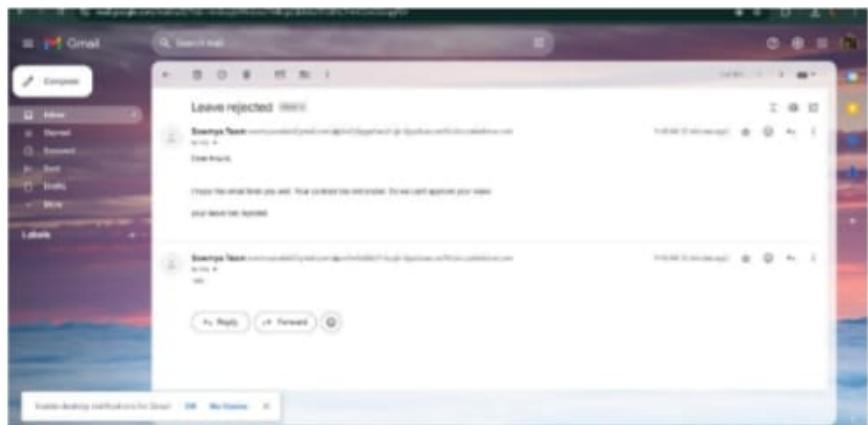
lease Validation Rule

Validation Rule Detail

Rule Name: lease_end_date	Active: <input checked="" type="checkbox"/>
Error Condition Formula: End_date__c <= start_date__c	Error Location: start date
Error Message: Your End date must be greater than start date	
Description:	Created By: ScamvaTeam 6/19/2025, 8:37 AM
	Modified By: ScamvaTeam 6/26/2025, 7:47 AM

- Added Apex trigger to restrict multiple tenants per property

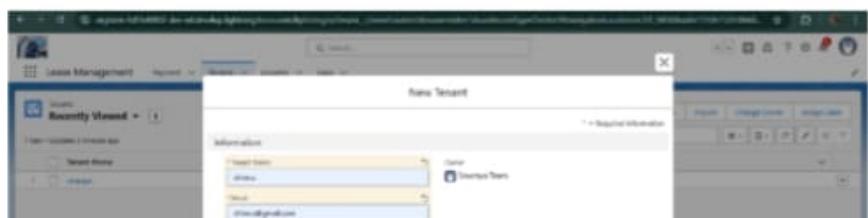
- Leave rejected



- Flow runs



- Trigger error messages



**Classic Email Templates**

**tenant payment**

Email Template Details

Email Template Name	tenant payment
Template Status	Active
Description	Success message for payment.
Created By	System User, 1111111111111111

**Email Template**

**Subject:** Confirmation of Recent Monthly Payment

**From:** RentMaster

**To:** [Email]

We have received your email. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment!

- Approval Process creation

For Tenant Leaving:

**Approval Processes**

**Tenant: TenantApproval**

**Process Definition Detail**

Process Name	Unlocked
Owner Name	Unlocked
Description	Process to approve tenant flag
Entry Criteria	Process is active AND flag = Unlocked
Decommissioned	Administrators ONLY
Approved Requests Count	0
Initial Submission Actions	Record Lock
Owner Type	Record Lock

**Initial Submission Actions**

Action Type	Record Lock
-------------	-------------

**Approval Steps**

Order	Step Number	Name	Description	Owner	Assigned Receiver	Next Receiver
1	1	Step 1		System User, 1111111111111111	System User, 1111111111111111	Final Approval

For Check for Vacant:

**Approval Processes**

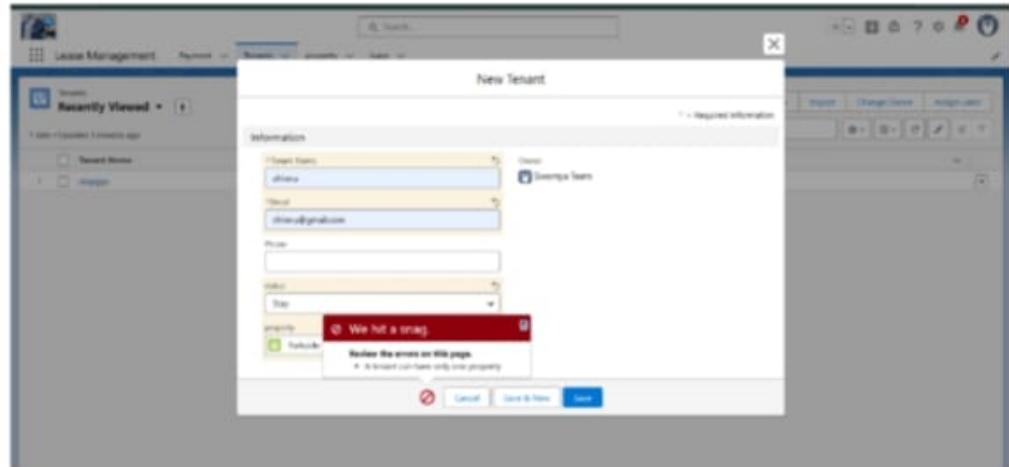
**Tenant: check for vacant**

**Process Definition Detail**

Process Name	Check for vacant
Owner Name	Check for vacant
Description	

**Initial Submission Actions**

Action Type	Record Lock
-------------	-------------

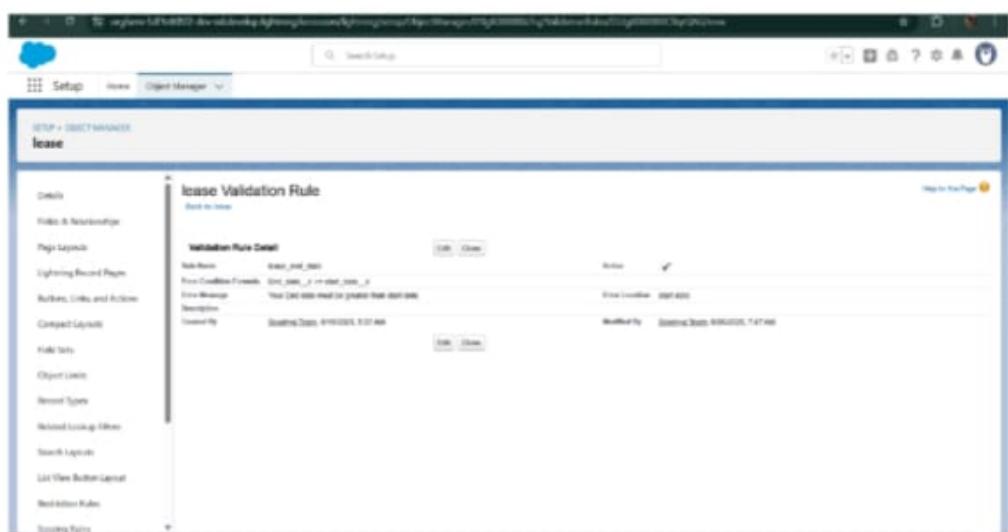
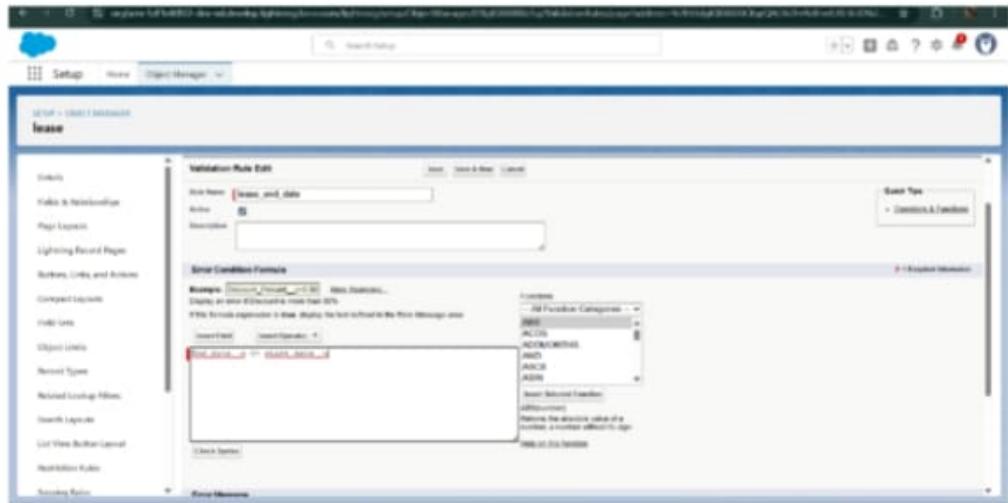


- Scheduled monthly reminder emails using Apex class

```
1 -> global class MonthlyReminder implements Schedulable {
2   global void execute(SchedulableContext sc) {
3     String queryBy = 'SELECT Id, Tenant__c FROM Tenant__c';
4     if (queryBy == null) {
5       selectById();
6     }
7
8     addAllStatusAndAvailability();
9
10    List<Tenant__c> tenants = [SELECT Id, Tenant__c FROM Tenant__c];
11
12    for (Tenant__c tenant : tenants) {
13
14      String recipientEmail = tenant.Tenant__c;
15
16      String emailContent = 'Dear Tenant ' + tenant.Tenant__c + ', I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain';
17
18      String emailSubject = 'Tenant Monthly Rent Payment Due';
19
20      Messaging.SingleEmailMessage mesg = new Messaging.SingleEmailMessage();
21
22      mesg.setToAddresses(new String[]{recipientEmail});
23
24      mesg.setPlainTextBody(emailContent);
25
26      mesg.setSubject(emailSubject);
27
28    }
29  }
30}
```

- Built and tested email templates for leave request, approval, rejection, payment, and reminders

- To create a validation rule to a Lease Object



- Added Apex trigger to restrict multiple tenants per property

- Built and tested email templates for leave request, approval, rejection, payment, and reminders

**Leave approved**

Preview your email template below.

**Email Template Detail**

Email Template Name: Leave approved	Contact Public Classico Email Templates
Template Owner Name: Leave Approved	Leave Approved
Template Owner Role: Leave Requester	Leave Approved
Template Status: Enabled	Leave Approved
Author: System User	Leave Approved
Description: System User	Leave Approved
Created By: System User	Leave Approved
Modified By: System User	Leave Approved

**Email Template**

**Subject:** Leave approved

**From:** [Leave Approved]

**Text:**

I have this message from you and I am writing to let you know I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my leave.

Leave Approved is a document. Click here to view.

**Leave Pending**

Preview your leave template below.

**Email Template Detail**

Email Template Name: Leave Pending	Contact Public Classico Email Templates
Template Owner Name: Leave Pending	Leave Pending
Template Owner Role: Leave Pending	Leave Pending
Template Status: Enabled	Leave Pending
Author: System User	Leave Pending
Description: System User	Leave Pending
Created By: System User	Leave Pending
Modified By: System User	Leave Pending

**Email Template**

**Subject:** Import leave approval

**From:** [Leave Pending]

**Text:**

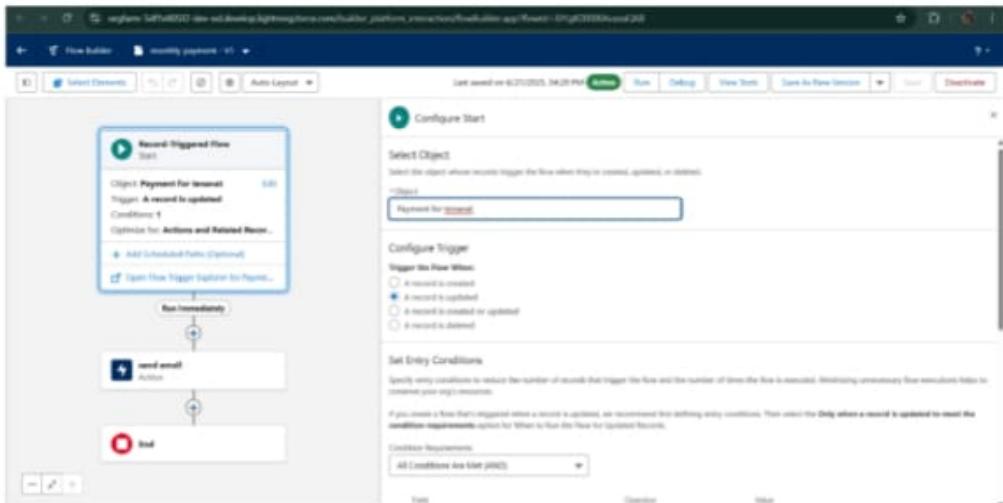
Please approve my leave.

**Leave rejected**

Preview your leave template below.

**Email Template Detail**

Email Template Name: Leave rejected	Contact Public Classico Email Templates
Template Owner Name: Leave rejected	Leave rejected
Template Owner Role: Leave Rejected	Leave rejected
Template Status: Enabled	Leave rejected
Author: System User	Leave rejected
Description: System User	Leave rejected
Created By: System User	Leave rejected
Modified By: System User	Leave rejected



- Schedule class:  
Create an Apex Class

```

1 // global class MonthlyEmailScheduler implements Scheduleable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8             sendMonthlyEmails();
9         }
10    }
11
12    public static void sendMonthlyEmails() {
13        List<Tenant__c> tenants = [SELECT
14            For (Tenant__c tenant : tenants
15            String recipientEmail = tenant.Email__c;
16
17        ]

```

```

1 // global class MonthlyEmailScheduler implements Scheduleable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8             sendMonthlyEmails();
9         }
10    }
11
12    public static void sendMonthlyEmails() {
13        List<Tenant__c> tenants = [SELECT
14            For (Tenant__c tenant : tenants
15            String recipientEmail = tenant.Email__c;
16
17        ]

```

```
1 trigger test on Tenant__c (before insert)
2 {
3     if(trigger.isInsert && trigger.isBefore){
4         testHandler.preventInsert(trigger.new);
5     }
6 }
```

## Create an Apex Handler class

```
1 public class TestHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for(Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8             existingPropertyIds.add(existingTenant.Property__c);
9         }
10
11         for(Tenant__c newTenant : newList) {
12
13             if(newTenant.Property__c != null) {
14                 newTenant.addError('A');
15             }
16         }
17     }
18 }
```

