

# **LEASE MANAGEMENT**

**College Name:** Hindusthan College Of Science And Commerce

**College Code:** CA

**TEAM ID: NM2025TMID28247**

**TEAM MEMBERS:**

**Team LeaderName:** JANANI.P

**Email:** [aswinjanani2011@gmail.com](mailto:aswinjanani2011@gmail.com)

**Team Member1:** SWETHA.P

**Email:** [2006swethap@gmail.com](mailto:2006swethap@gmail.com)

**Team Member:** MEENA.P

**Email:** [meenapalani5040@gmail.com](mailto:meenapalani5040@gmail.com)

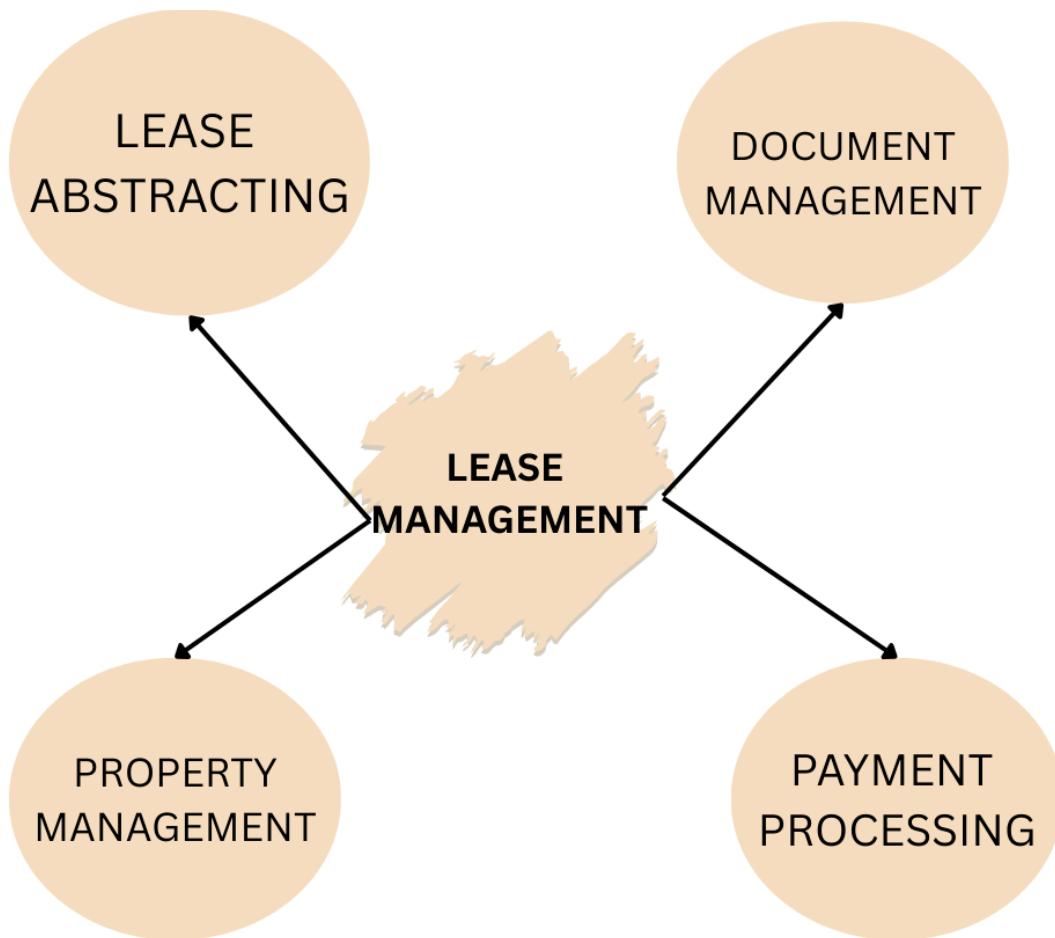
**Team Member:** KUZHANDHAIMERY.M

**Email:** [kuzhandhaimery@gmail.com](mailto:kuzhandhaimery@gmail.com)

# 1.INTRODUCTION

## 1.1 Project Overview

This project is focused on the development and implementation of a Lease Management System using Salesforce, designed to address the challenges of manual property lease tracking, tenant management, payment processing, and communication within the organization. The goal is to deliver a comprehensive solution that streamlines and automates various aspects of lease management, from contract creation to payment tracking, approval workflows, and reporting.



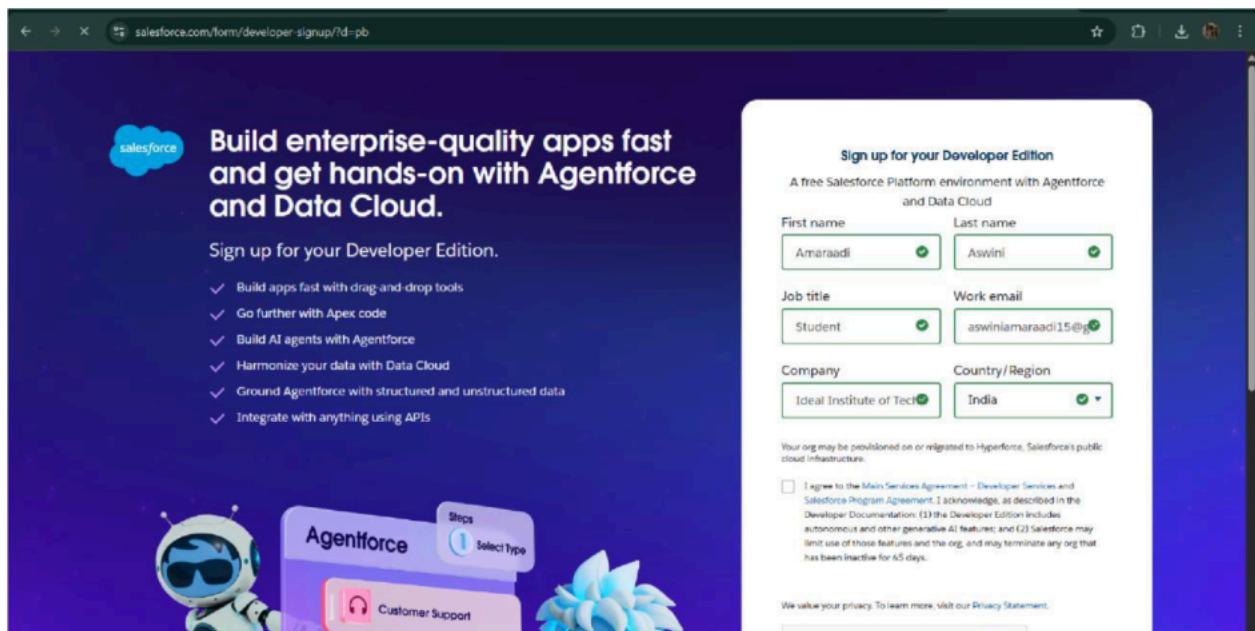
## 1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

# DEVELOPMENT PHASE

### Creating Developer Account:

By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>



●Created objects: Property, Tenant, Lease, Payment

**Tenant**

**Details**

Description	Enable Reports
API Name Tenant__c	✓
Custom	✓
Singular Label Tenant	✓
Plural Label Tenants	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

**Payment for tenant**

**Details**

Description	Enable Reports
API Name Payment_for_tenant__c	✓
Custom	✓
Singular Label Payment for tenant	✓
Plural Label Payment	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

**Property**

**Details**

Description	Enable Reports
API Name Property__c	✓
Custom	✓
Singular Label Property	✓
Plural Label Properties	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

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. The main area displays the configuration details for the 'lease' object. The left sidebar lists various configuration options: 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, and Scoping Rules. The right panel shows the 'Details' section for the 'lease' object, including its API name ('lease\_\_c'), singular label ('lease'), and plural label ('leases'). It also includes sections for 'Enable Reports', 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), and 'Help Settings'. The status bar at the bottom indicates 'Google Chrome'.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. The main area displays the configuration details for the 'Payment for tenantat' object. The left sidebar lists various configuration options: 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, and Restriction Rules. The right panel shows the 'Fields & Relationships' section, which lists seven items sorted by Field Label. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The listed fields are: 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), and Payment Name (Name, Text(80)). The 'Owner' field has an 'Indexed' checkbox checked.

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)	<input checked="" type="checkbox"/>	
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)	<input checked="" type="checkbox"/>	

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	CreatedById	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User/Group)		✓
property	property_c	Lookup(property)		✓
start date	start_date_c	Date		

Setup > OBJECT MANAGER

## Tenant

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)		
Email	Email_c	Email		
Last Modified By	LastModifiedBy	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 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

Description: Enter a description...

**App Branding**

Image:  Primary Color Hex Value: #0070D2

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

**App Launcher Preview**

LM Lease Management

**App Settings**

App Details & Branding

App Options

Utility Items (Desktop Only)

**Navigation Items**

User Profiles

**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**

Q: Type to filter list...

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- ...

**Selected Items**

- Payment
- Tenants
- property
- lease

Lightning App Builder | App Settings | Pages | Lease Management | ? Help

App Settings

User Profiles

Choose the user profiles that can access this app.

Available Profiles

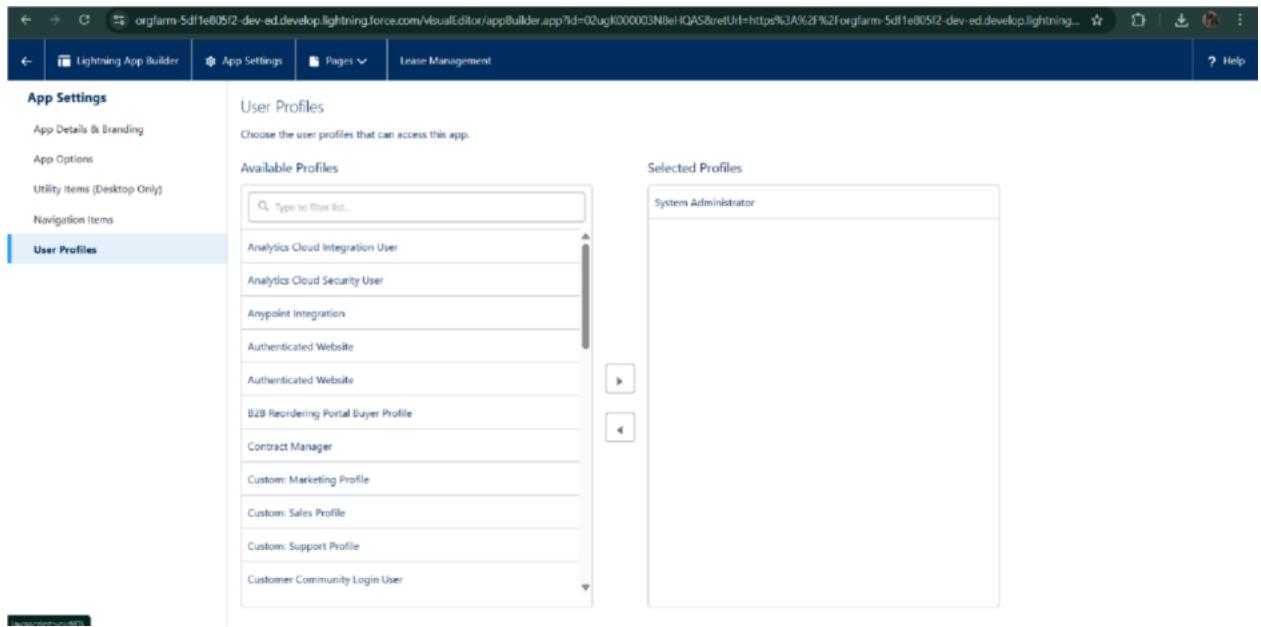
Type to filter list...

- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

System Administrator

▶◀



Lease Management | Payment | Tenants | Properties | Lease

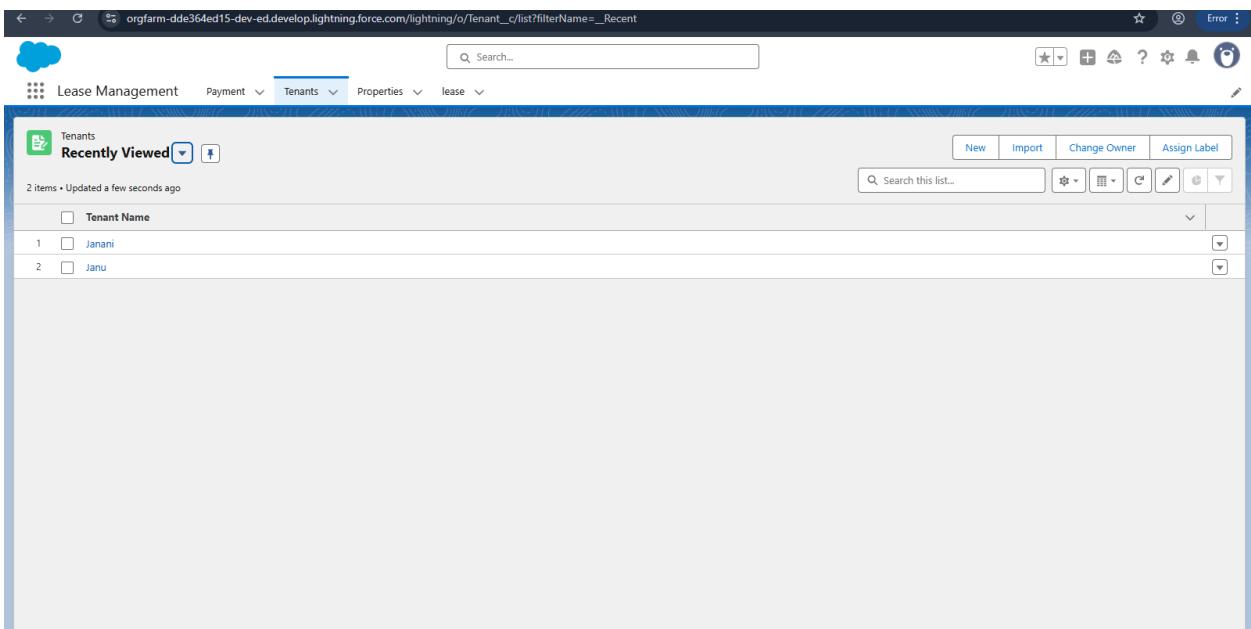
Recently Viewed

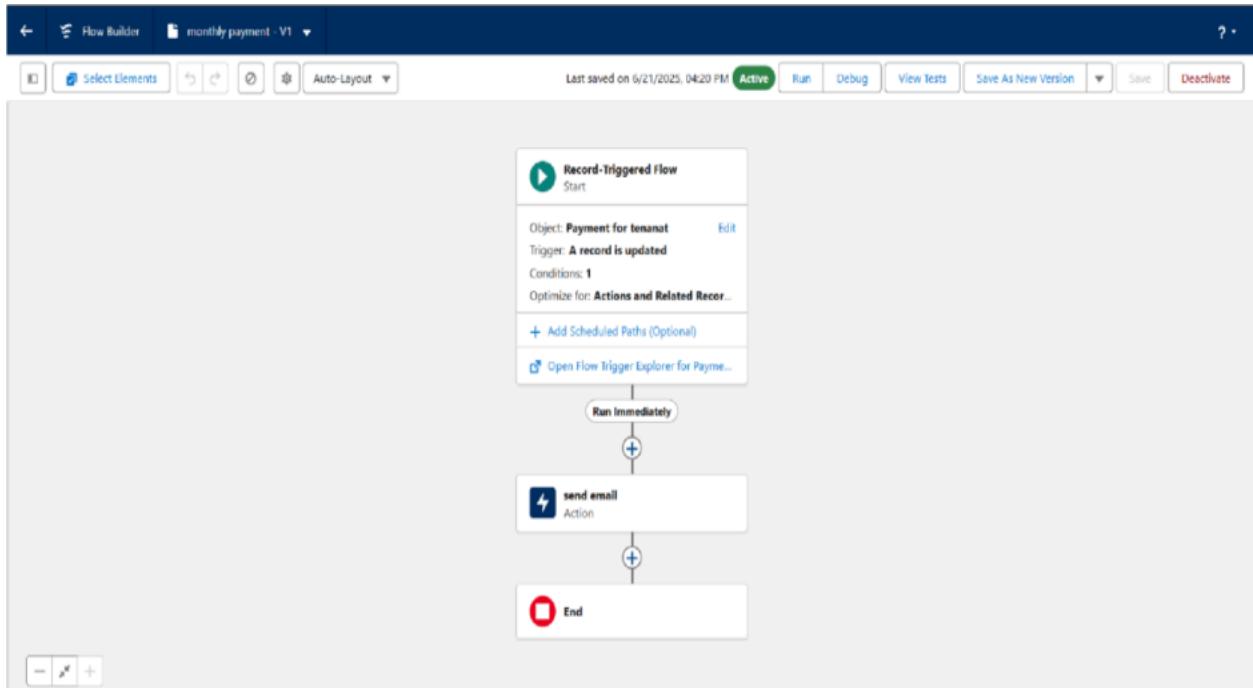
2 items • Updated a few seconds ago

	Tenant Name
1	Janani
2	Janu

New Import Change Owner Assign Label

Search this list... Filter View





- Implemented Flows for monthly rent and payment success
- To create a validation rule to a Lease Object

Setup > Object Manager

## lease

Validation Rule Edit

Role Name: lease\_end\_date

Active:

Description:

Error Condition Formula

Example: `Discount_Percent_C>0.30` More Examples...

If this formula expression is true, display the text defined in the Error Message area.

Functions

Insert Field Insert Operator

`End_date_c <= start_date_c`

Check Syntax

Quick Tips

Required Information

Operators & Functions

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

Setup > Object Manager

## lease

lease Validation Rule

Validation Rule Detail

Role Name: lease\_end\_date

Active:

Error Condition Formula: `End_date_c <= start_date_c`

Error Message: Your End date must be greater than start date

Description:

Created By: Sowmya\_Team, 6/19/2025, 5:37 AM

Modified By: Sowmya\_Team, 6/26/2025, 7:47 AM

Help for this Page

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

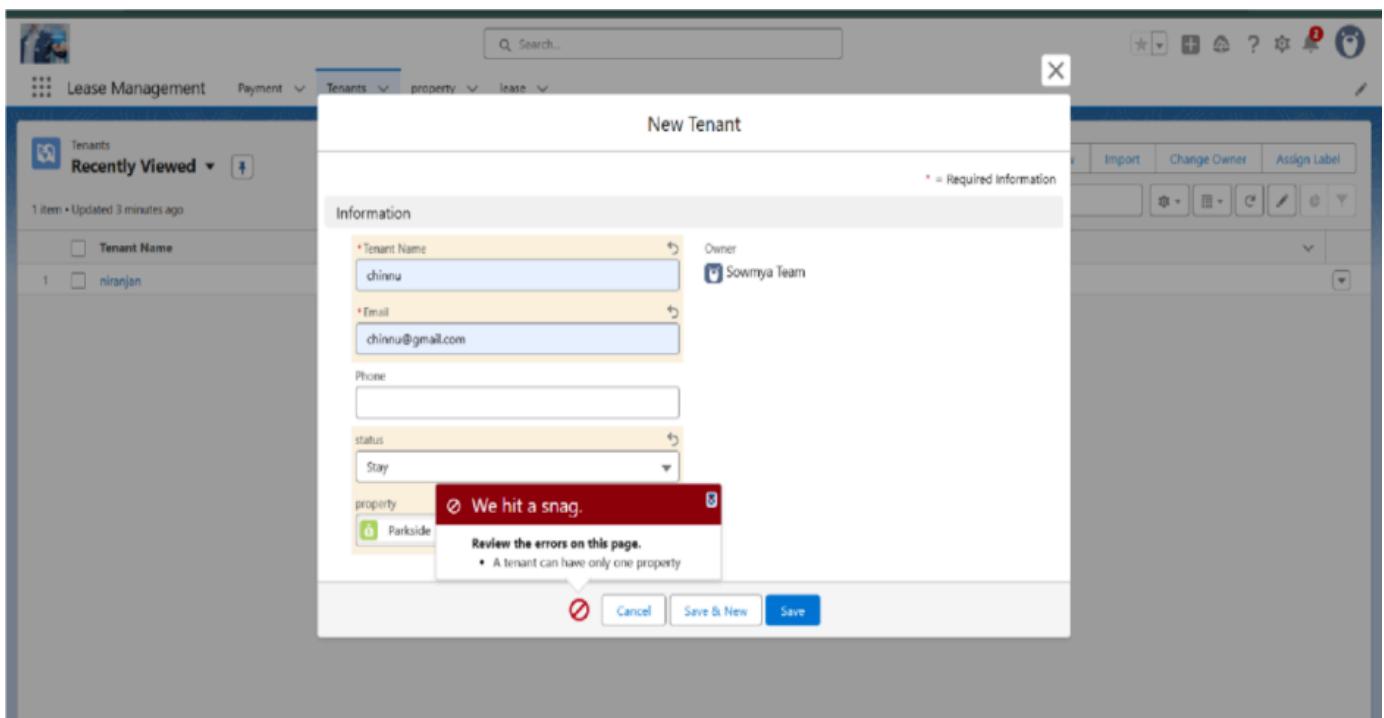
Search Layouts

List View Button Layout

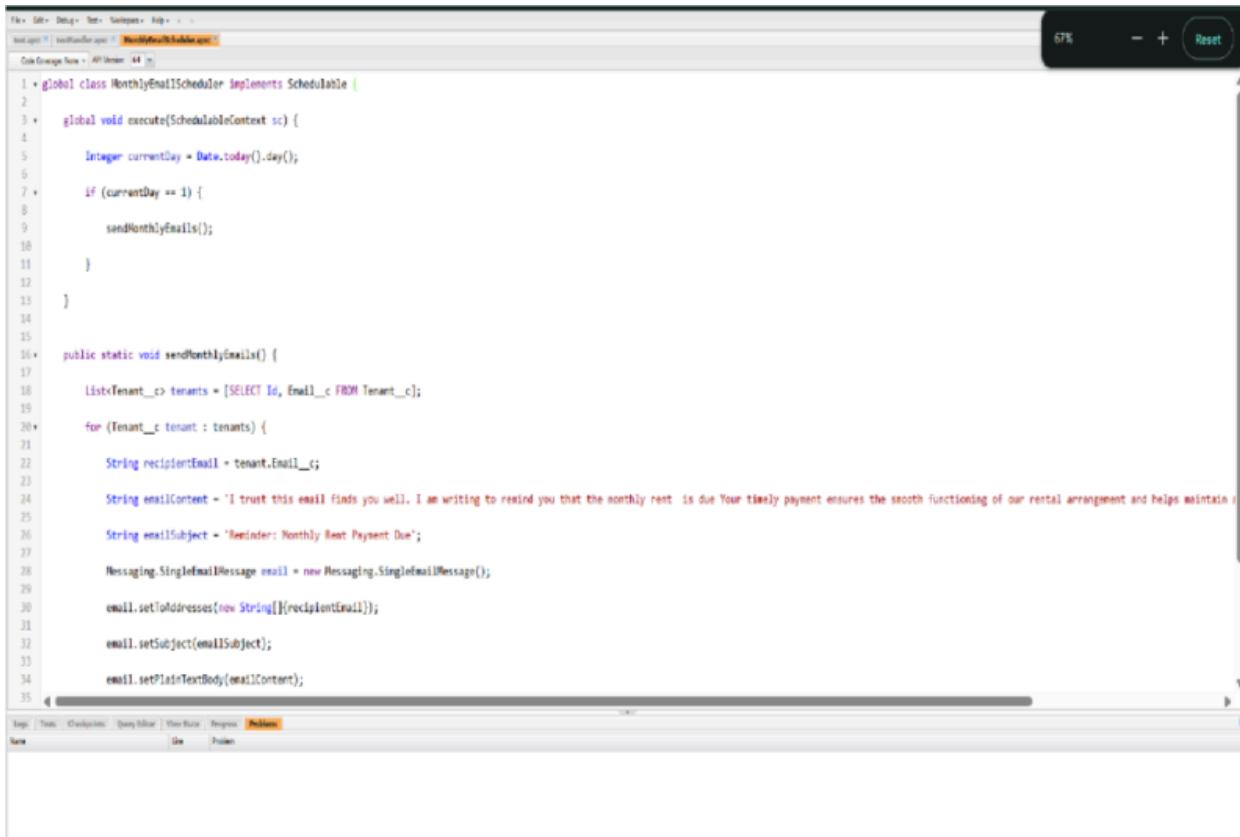
Restriction Rules

Scoping Rules

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class



The screenshot shows the Salesforce Developer Console with the following details:

- Page Title:** MonthlyEmailScheduler.apx
- Code Coverage Area:** API Version: 44
- Code:**

```

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
9             sendMonthlyEmails();
10        }
11    }
12
13 }
14
15
16 * public static void sendMonthlyEmails() {
17
18     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20     for (Tenant__c tenant : tenants) {
21
22         String recipientEmail = tenant.Email__c;
23
24         String emailContent = 'I trust this email finds you well. 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';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setToAddresses(new String[]{recipientEmail});
31
32         email.setSubject(emailSubject);
33
34         email.setPlainTextBody(emailContent);
35

```

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

The screenshot shows the Salesforce Setup interface with the 'Classic Email Templates' page open. The template name is 'Leave approved'. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name: Leave approved	Leave_approved
Template Unique Name: Leave_approved	Leave_approved
Encoding: Unicode (UTF-8)	Unicode (UTF-8)
Author: Scenova Team [Changed]	Scenova Team [Changed]
Description:	
Created By: Scenova Team	Scenova Team, 6/20/2025, 1:08 AM

On the right, there are status indicators: Available For Use (checked), Last Used Date, and Times Used.

The 'Email Template' preview section shows the following content:

**Subject:** Leave approved

**Plain Text Preview:**

```
dear{Tenant__c.Name}.

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

Your status is announced. You can review now.
```

The screenshot shows the Salesforce Setup interface with the 'Classic Email Templates' page open. The template name is 'tenant leaving'. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name: tenant leaving	tenant_leaving
Template Unique Name: tenant_leaving	tenant_leaving
Encoding: Unicode (UTF-8)	Unicode (UTF-8)
Author: Scenova Team [Changed]	Scenova Team [Changed]
Description:	
Created By: Scenova Team	Scenova Team, 6/20/2025, 1:06 AM

On the right, there are status indicators: Available For Use (checked), Last Used Date, and Times Used.

The 'Email Template' preview section shows the following content:

**Subject:** request for approve the leave

**Plain Text Preview:**

```
Dear {Tenant__c.CreatedBy}.

Please approve my leave.
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item 'Leave rejected'. The main content area displays the 'Leave rejected' email template details. The 'Email Template Detail' section includes fields like Email Template Name (Leave rejected), Encoding (Unicode (UTF-8)), and Author (Sohamya\_Team [Channe]). The 'Email Template' section shows the subject 'Leave rejected' and a plain text preview: 'Dear {[Tenant\_\_c.Name]}, I hope this email finds you well. Your contract has not ended. So we can't approve your leave. Your leave has rejected'.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item 'Tenant Email'. The main content area displays the 'Tenant Email' email template details. The 'Email Template Detail' section includes fields like Email Template Name (Tenant\_Email), Encoding (Unicode (UTF-8)), and Author (Sohamya\_Team [Channe]). The 'Email Template' section shows the subject 'Urgent: Monthly Rent Payment Reminder' and a plain text preview: 'Dear {[Tenant\_\_c.Name]}, I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.' Below the preview, there is a note: 'This message contains no or partially encoded non-ASCII characters which may need decoding before it can be displayed as intended.'

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. The left sidebar shows 'Email' categories: 'Classic Email Templates' (selected) and 'Lightning Email Templates'. A message at the top says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Classic Email Templates' and shows a 'Text Email Template' named 'tenant payment'. The 'Email Template Detail' section includes fields like 'Email Template Name' (tenant payment), 'Template Unique Name' (tenant\_payment), 'Encoding' (Unicode (UTF-8)), 'Author' (Sowmya\_Team [Change]), 'Description' (Created By Sowmya\_Team, 6/20/2025, 1:13 AM), and 'Modified By' (Sowmya\_Team, 6/20/2025, 1:13 AM). The 'Available For Use' checkbox is checked. Below this is a preview window showing the email template's subject ('Confirmation of Successful Monthly Payment') and plain text preview ('Dear {Tenant\_\_c.Email\_\_c}, We hope this email finds you well. 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:

The screenshot shows the Salesforce Setup interface with the search bar set to 'approval'. The left sidebar shows 'Data' categories: 'Mass Transfer Approval Requests', 'Feature Settings' (with 'Approval Settings' selected), and 'Process Automation' (with 'Approval Processes' selected). A message at the top says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Approval Processes' and shows an 'Approval Process' named 'Tenant: TenantApproval'. The 'Process Definition Detail' section includes fields like 'Process Name' (TenantApproval), 'Unique Name' (TenantApproval), 'Description' ( ), 'Entry Criteria' (Tenant\_\_c.status equals Stay), 'Record Editability' (Administrator ONLY), 'Next Automated Approver Determined By' (Active), 'Approval Assignment Email Template' ( ), 'Initial Submitters' (Tenant Owner), 'Created By' (Sowmya\_Team, 6/23/2025, 3:41 AM), and 'Modified By' (Sowmya\_Team, 6/26/2025, 11:57 PM). Below this are sections for 'Initial Submission Actions' (Record Lock: Lock the record from being edited) and 'Approval Steps' (Step 1: Step 1, Assigned Approver: User\_SowmyaTeam, Reject Behavior: Final Rejection).

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the following details:

- Approval Processes:** Tenant: check for vacant
- Process Definition Detail:**
  - Process Name: check for vacant
  - Unique Name: check\_for\_vacant
  - Description: Tenant: status EQUALS Leaving
  - Entry Criteria: Record Editability Administrator ONLY
  - Next Automated Approver Determined By: Allow Submitters to Recall Approval Requests
  - Approval Assignment Email Template: Leave approved
  - Initial Submitters: Tenant Owner
  - Created By: Sowmya\_Team
  - Modified By: Sowmya\_Team
- Initial Submission Actions:**
  - Action: Type Record Lock Description: Lock the record from being edited
  - Action: Type Email Alert Description: PLEASE APPROVE MY RATES
- Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	step1			User:Sowmya_Team	Final Rejection

- Apex Trigger

## Create an Apex Trigger

```

trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}

```

The screenshot shows the Salesforce Developer Console with the following details:

- Trigger:** ImtHandler.apex
- Code Coverage:** None
- API Version:** 54
- Trigger Details:**

```

trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}

```
- Open Modal:** A modal window titled "Open" is displayed, listing various entity types (Entity Type) such as Classes, Triggers, Pages, Objects, Static Resources, and Packages. The entry "test" is selected under the "Classes" category.
- Logs:** Logs, Tests, Checkpoints, Query Editor, View Status, Progress, Problems

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

testHandler.apc [ ] MonthlyEmailScheduler.apc [ ]

Code Coverage Noise API Version 64 Go

```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Use Problem

## Create an Apex Handler class

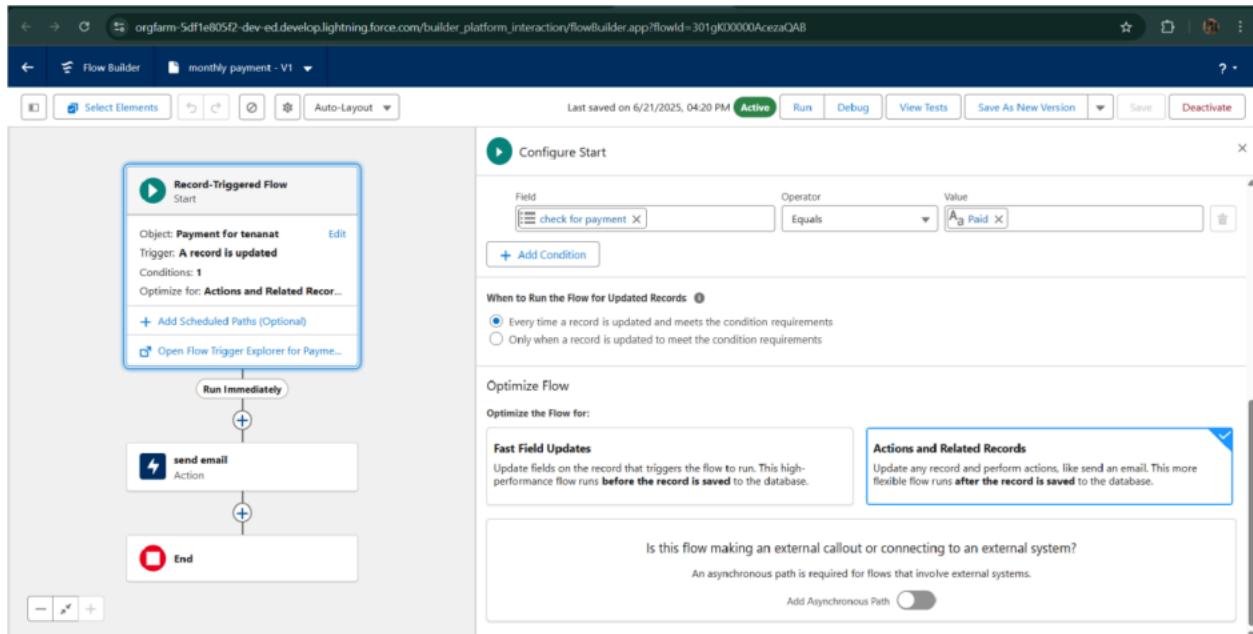
```

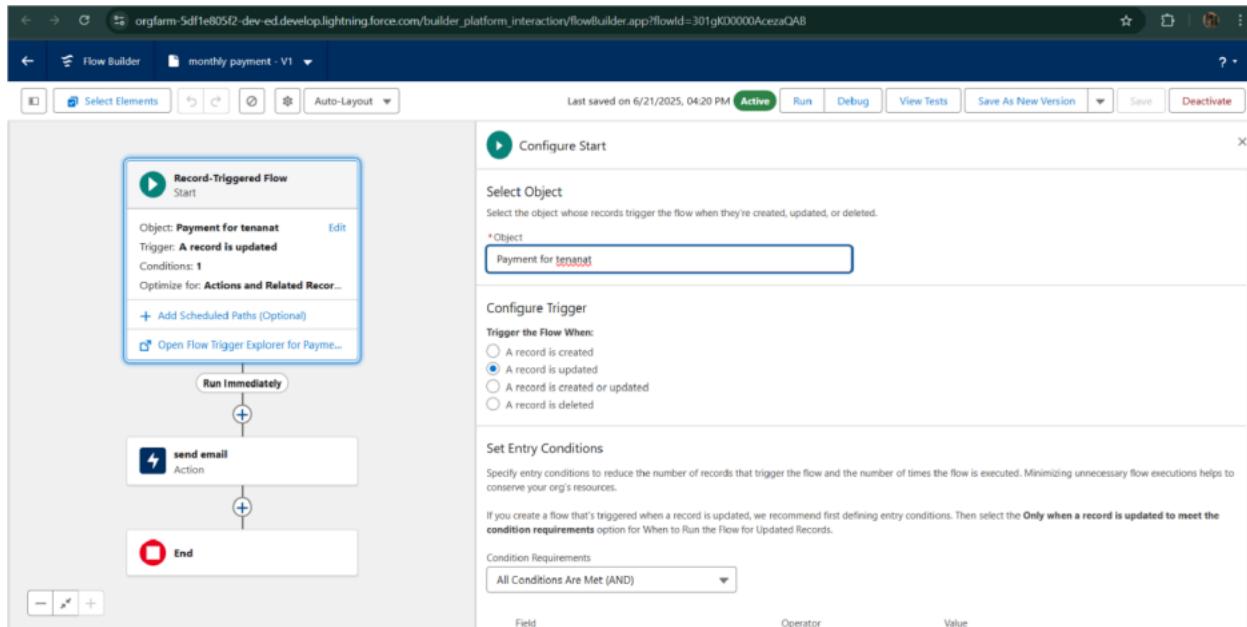
1 • public class testHandler {
2
3 •     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11     }
12
13
14     for (Tenant__c newTenant : newList) {
15
16
17         if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
18
19             newTenantaddError('A tenant can have only one property');
20
21         }
22
23     }
24
25 }

```

The screenshot shows the Salesforce Developer Console interface. The tab bar at the top includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and 'Go To'. Below the tabs, there are sections for 'Code Coverage', 'New', 'API Version', and '64'. The main area displays the Apex code for 'testHandler.apc'. The code defines a static method 'preventInsert' that takes a list of 'Tenant\_\_c' objects. It first creates a set of existing property IDs. Then, it iterates through the new tenants. If a new tenant has a property and its ID is already in the set, an error message is added to the tenant's record.

## ● FLOWS





- Schedule class:  
Create an Apex Class

```
global class MonthlyEmailScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Integer currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }
    public static void sendMonthlyEmails() {
        List<Tenant__c> tenants = [SELECT ...];
        for (Tenant__c tenant : tenants) {
            String recipientEmail = tenant.Email__c;
        }
    }
}
```

**Open**

Entity Type	Entity	Related
Classes	MonthlyEmailScheduler	Name, Extent, Directories, Email, CustomField..., References, <- Tenant__c Subject, References, <- Tenant__c Subject, References
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

```

1 *global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.Today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8
9     public static void sendMonthlyEmails() {
10        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            String recipientEmail = tenant.Email__c;
13            String emailSubject = 'Wentor: monthly rent payment due';
14            String emailContent = 'I trust this email finds you well. 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 a positive living environment for all.' ;
15            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
16            email.setToAddresses(new String[]{recipientEmail});
17            email.setSubject(emailSubject);
18            email.setPlainTextEmailBody(emailContent);
19            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
20        }
21    }
22}

```

## Schedule Apex class

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with navigation links like Home, Object Manager, and a search bar. The main area is titled "Apex Classes" under the "Apex Class" section. A table displays the details for the "MonthlyEmailScheduler" class, including its Name (MonthlyEmailScheduler), Namespace Prefix (empty), Created By (SOMMYA Team), Status (Active), Code Coverage (0% (0/15)), and Last Modified By (SOMMYA Team). Below the table, tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags" are visible. The "Class Body" tab is selected, showing the Apex code for the MonthlyEmailScheduler class.

```

1 *global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.Today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8
9     public static void sendMonthlyEmails() {
10        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            String recipientEmail = tenant.Email__c;
13            String emailSubject = 'Wentor: monthly rent payment due';
14            String emailContent = 'I trust this email finds you well. 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 a positive living environment for all.' ;
15            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
16            email.setToAddresses(new String[]{recipientEmail});
17            email.setSubject(emailSubject);
18            email.setPlainTextEmailBody(emailContent);
19            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
20        }
21    }
22}

```

The screenshot shows a Salesforce Lightning interface for a Tenant record. The top navigation bar includes icons for back, forward, search, and user profile. The main header displays the URL: orgfarm-dde364ed15-dev-ed.develop.lightning.force.com/lightning/r/Tenant\_c/a01gL00000KblYIQAN/view. Below the header, a blue ribbon menu lists 'Lease Management', 'Payment', 'Tenants' (selected), 'Properties', and 'lease'. A sidebar on the left shows a 'Tenant' icon and the name 'Janani'. The main content area has tabs for 'Related' and 'Details'. Under 'Details', fields include 'Tenant Name' (Janani), 'Email' (aswinjanani2011@gmail.com), 'Phone', 'status' (Stay), 'Property' (House), 'Created By' (JANANI P, 9/5/2025, 5:16 AM), and 'Last Modified By' (JANANI P, 9/8/2025, 7:54 AM). To the right is an 'Activity' sidebar with a toolbar, filters (All time, All activities, All types), and sections for 'Upcoming & Overdue' (empty) and 'Past activity' (empty).

This screenshot shows the same Salesforce record detail page for 'Janani' after an action has been taken. The 'Notifications' sidebar on the right is now populated with five entries, each showing a blue bell icon and the message 'Approval request for the tenant is approved' followed by 'Janani'. The dates next to the messages indicate they were sent on Sep 8, 2025, at various times between 8:24 PM and 8:29 PM, and one was sent on Sep 6, 2025, at 2:42 PM. The 'Mark all as read' link is visible at the top of the notifications list.

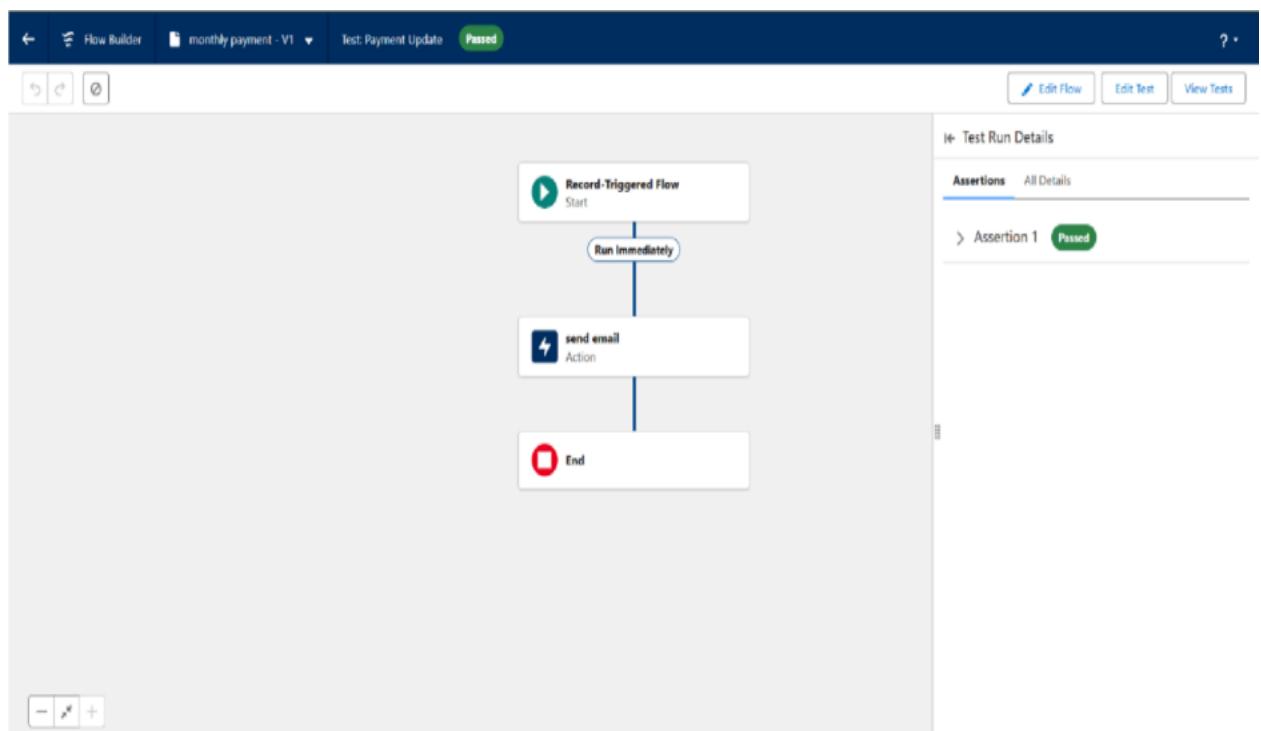
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

- Trigger validation by entering duplicate tenant-property records



- Validation Rule checking



- Test flows on payment update

The screenshot displays a software interface for 'Lease Management'. On the left, a 'Tenant' record for 'Janani' is shown with details like Name, Email, Phone, Status, and Property type. The 'Details' tab is selected. On the right, a 'Notifications' sidebar lists five recent approvals from 'JANANI P.'.

Notification Details	Date
Approval request for the tenant is approved Janani	3 hours ago
Approval request for the tenant is approved Janani	Sep 8, 2025, 8:24 PM
Approval request for the tenant is approved Janani	Sep 8, 2025, 8:21 PM
Approval request for the tenant is approved Janani	Sep 6, 2025, 2:42 PM
Approval request for the tenant is approved Janani	Sep 6, 2025, 2:39 PM

Inbox (326) - 2006swethap@g... Janani | Tenant | Salesforce lease | Salesforce Lease Management Project Do... Error

Lease Management Project Do... Error

orgfarm-dde364ed15-dev-ed.develop.lightning.force.com/lightning/r/Tenant\_c/a01g.00000KbIYQAN/view

Lease Management

Tenants Properties lease

Search...

New Contact Edit New Opportunity

Tenant Janani

Related Details

Payment (0) New

Approval History (6+)

Step Name	Date	Status	Assigned To
Submit for approval	9/11/2025, 9:29 PM	Approved	JANANI P
Approval Request Submitted	9/11/2025, 9:28 PM	Submitted	JANANI P
Submit for approval	9/8/2025, 7:54 AM	Approved	JANANI P
Approval Request Submitted	9/8/2025, 7:54 AM	Submitted	JANANI P
Submit for approval	9/8/2025, 7:51 AM	Approved	JANANI P
Approval Request Submitted	9/8/2025, 7:50 AM	Submitted	JANANI P

[View All](#)

Activity

Filters: All time • All activities • All types

Refresh • Expand All • View All

Upcoming & Overdue

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.

29°C Weather Search ENG 1014 13:00 2025

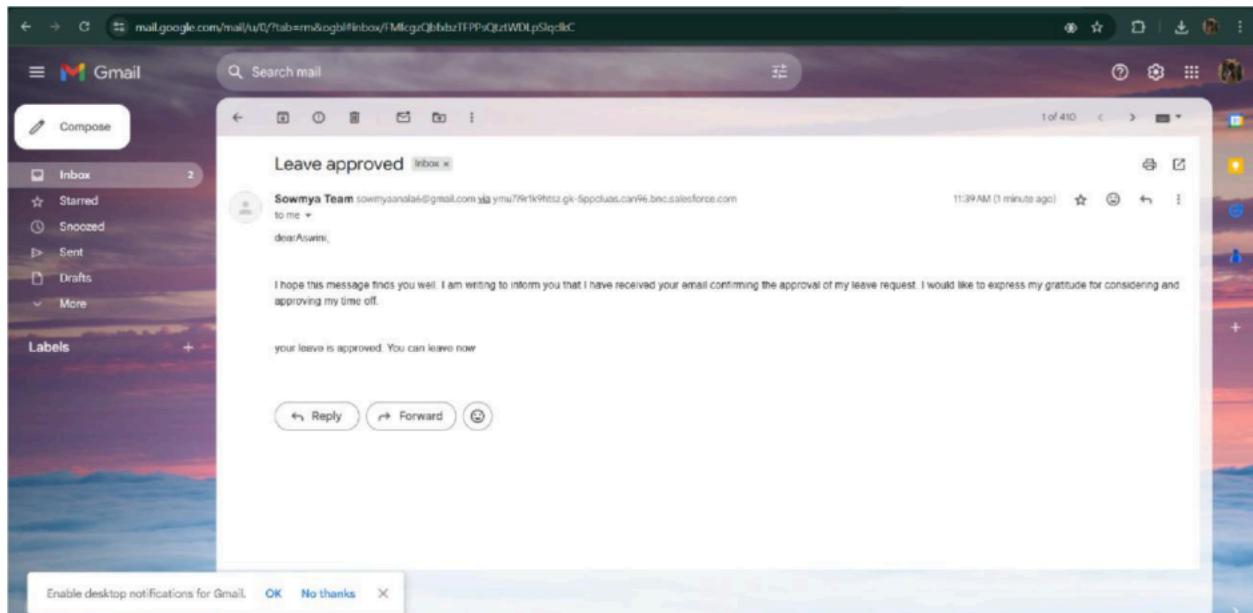
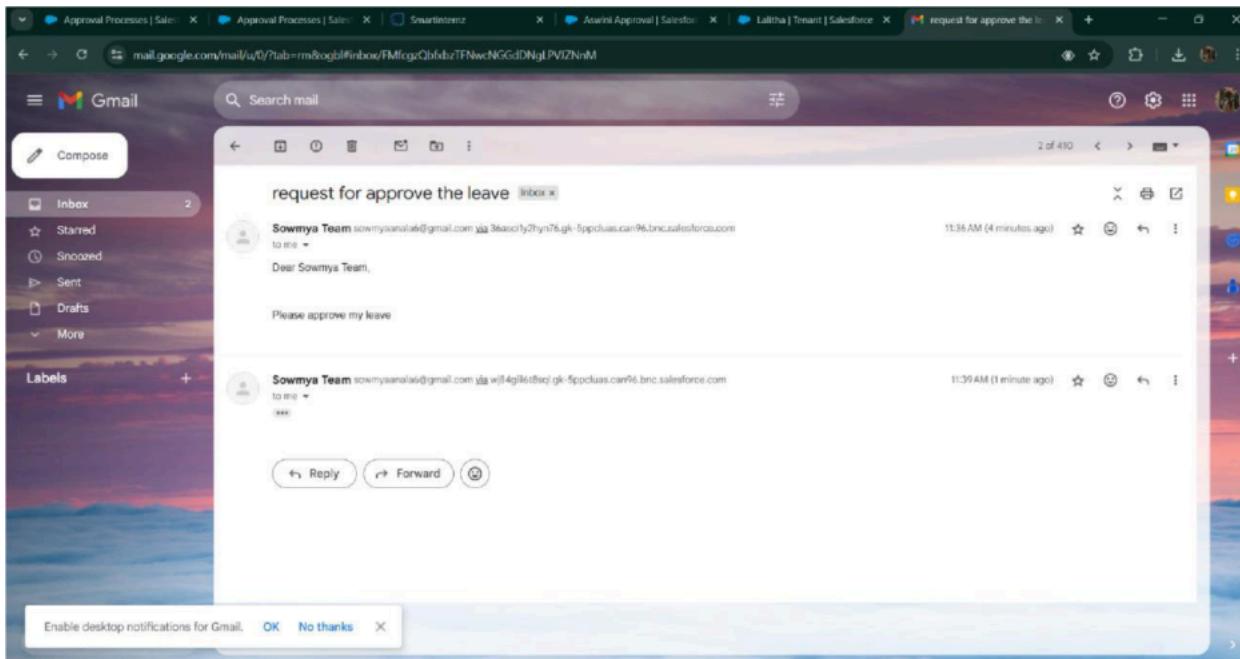
# RESULTS

## Output Screenshots

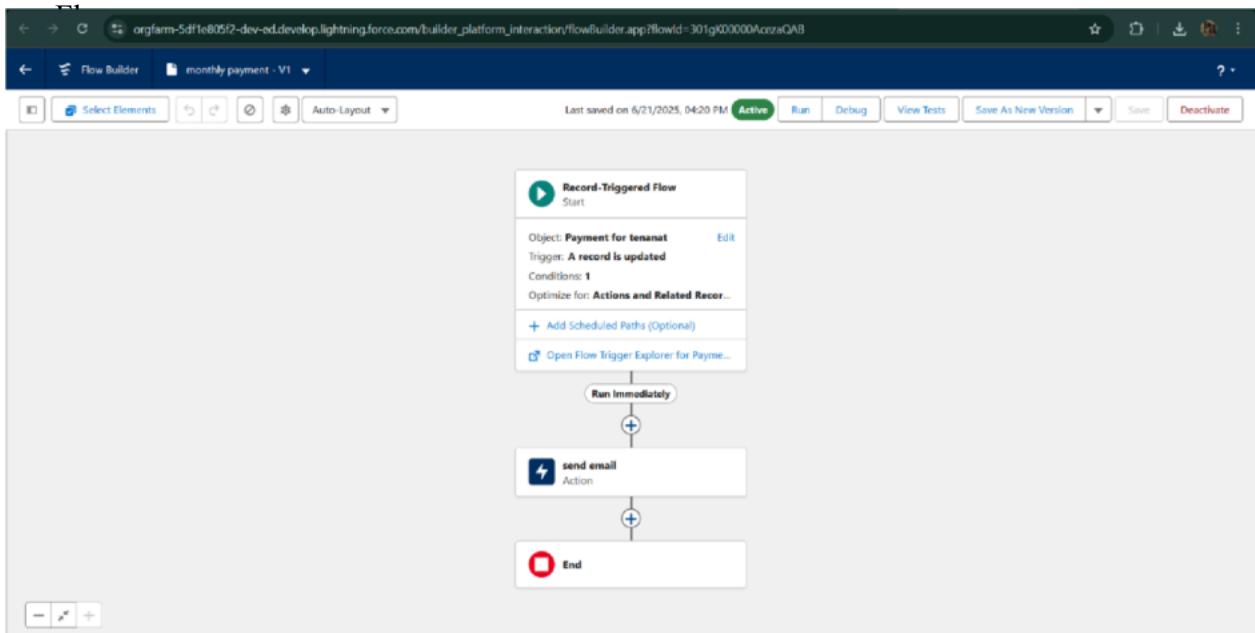
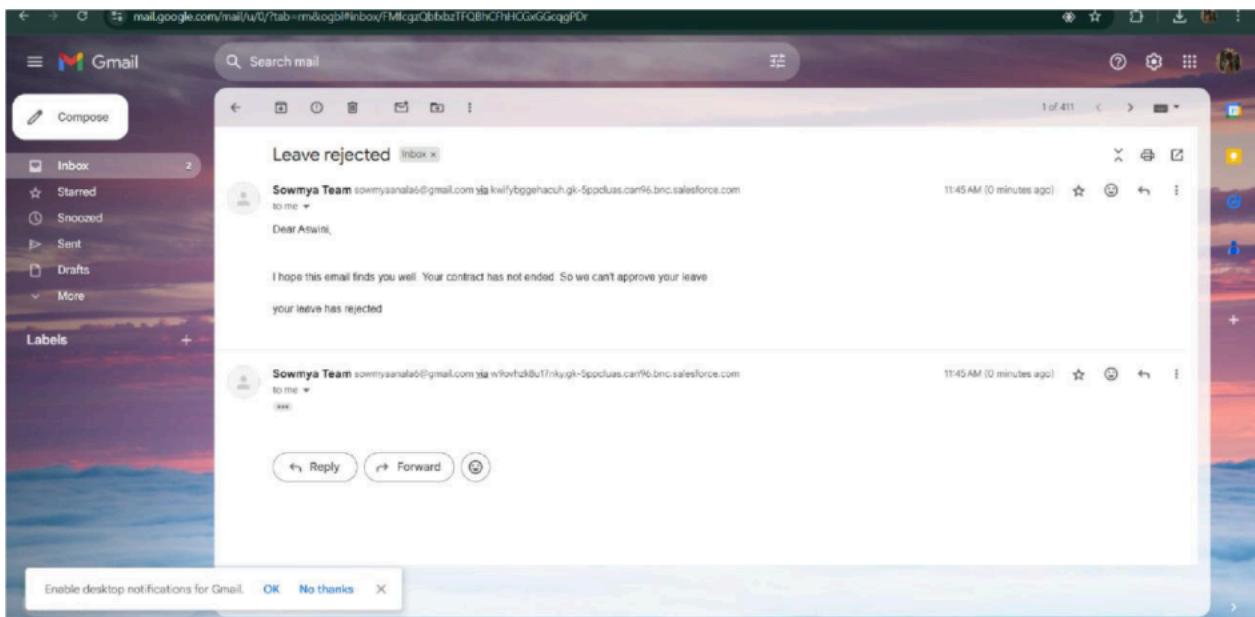
This screenshot shows the Salesforce Setup interface under the 'Tabs' section. The page title is 'Custom Tabs'. It includes a search bar and a help link. The main content area is divided into sections: 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', and 'Lightning Component Tabs'. The 'Custom Object Tabs' section lists four tabs: 'Lease' (Chip style), 'Payment' (Fan style), 'Properties' (Cell phone style), and 'Tenants' (Form style). Each tab entry has 'Edit | Del' links.

This screenshot shows the 'Lease Management' detail page for a tenant named 'Janani'. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'Properties', and 'lease'. The main content area displays the 'Payment' section (0 items) and the 'Approval History' section, which contains six entries. The 'Approval History' table has columns for 'Step Name', 'Date', 'Status', and 'Assigned To'. The 'Status' column shows 'Approved' or 'Submitted' for each entry. The 'Assigned To' column shows 'JANANI P'. Below the table is a 'View All' link. On the right side of the page, there is an 'Activity' sidebar with sections for 'Upcoming & Overdue' and 'Past activity'.

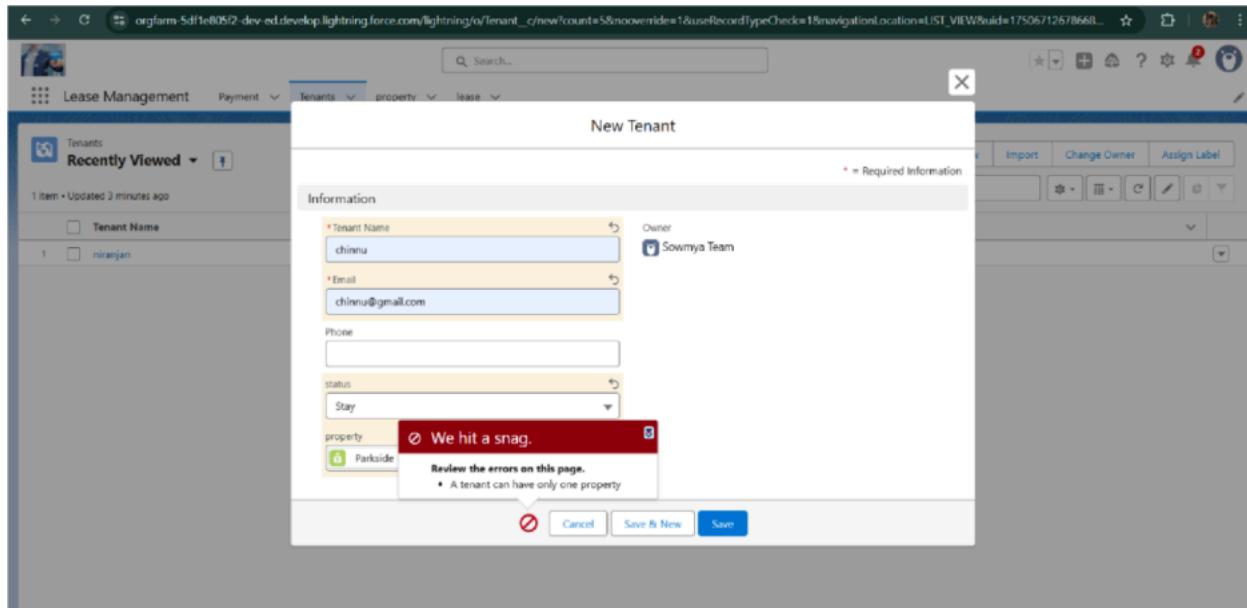
- Request for approve the leave



- Leave rejected



- Trigger error messages



- Approval process notifications

Step Name	Date	Status	Assigned To
Submit for approval	9/11/2025, 9:29 PM	Approved	JANANI P
Approval Request Submitted	9/11/2025, 9:28 PM	Submitted	JANANI P
Submit for approval	9/8/2025, 7:54 AM	Approved	JANANI P
Approval Request Submitted	9/8/2025, 7:54 AM	Submitted	JANANI P
Submit for approval	9/8/2025, 7:51 AM	Approved	JANANI P
Approval Request Submitted	9/8/2025, 7:50 AM	Submitted	JANANI P

# ADVANTAGE

**Lower Upfront Costs:** Leasing eliminates large capital expenditures, allowing businesses to preserve cash for other needs.

**Flexibility:** Businesses can easily upgrade assets at the end of a lease term helping to avoid obsolescence.

**Tax Benefits:** Lease payments are often deductible as operating expenses, providing a tax advantage for the lessee.

# DISADVANTAGE

**Higher Long-Term Costs:** While initial costs are lower, the total cost of leasing can be higher than buying over the asset's lifespan.

**No Ownership Equity:** At the end of the lease, lessees do not gain ownership of the asset, and their payments do not build equity.

**Restrictions:** Leased assets often come with limitations, such as mileage caps for vehicles or restrictions on customization.

# **CONCLUSION**

## **Summary of Achievements**

The Salesforce implementation for Lease Management has successfully streamlined and automated critical aspects of the leasing lifecycle. Key accomplishments include:

### **1. Comprehensive Lease Lifecycle Management**

- o Implemented a robust system to manage lease agreements, from creation and amendments to renewals and terminations.

### **2. Automated Payment Processes**

- o Enabled scheduled rent tracking, payment reminders, and overdue alerts, reducing manual oversight and improving cash flow management.

### **3. Efficient Expiration and Renewal Workflows**

- o Automated notifications and streamlined renewal processes, ensuring timely follow-ups and improved lease retention rates.

### **4. Enhanced Property and Tenant Management**

- o Centralized information for properties, units, and tenants, improving operational visibility and collaboration.

### **5. Actionable Insights and Reporting**

- o Delivered custom reports and dashboards to monitor revenue, lease performance, and occupancy metrics, supporting data-driven decision-making.

## **6. Improved User Experience**

- o Provided intuitive interfaces and mobile accessibility for leasing agents, property managers, and tenants, enhancing productivity and satisfaction.

## **7. Secure and Scalable Solution**

- o Ensured role-based access, field-level security, and compliance with legal requirements, enabling a reliable and scalable system for lease management.

## **8. Integration with External Systems**

- o Established seamless connections with payment gateways and accounting systems for end-to-end process automation.

# **REFERENCE**

1. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT):

<https://ijarsct.co.in/Paper7646.pdf>

2. Lease-Management-System-on-Salesforce:

<https://github.com/VedantRajGaur/Lease-Management-System-on-Salesforce>.

3. Github lease Management:

<https://github.com/mukunthan893/NMLEASE-MANAGEMENT>

4. Salesforce documentation:

Build an Integrated Real Estate CRM for Tenant Management.

# APPENDIX

- Source Code:** Provided in Apex Classes and Triggers

## Test.apxt:

```
trigger test on Tenant__c (before insert) { if  
(trigger.isInsert && trigger.isBefore){  
    testHandler.preventInsert(trigger.new);  
}  
}
```

## testHandler.apxc:

```
public class  
testHandler {  
  
    public static void  
    preventInsert(List<  
        Tenant__c> newlist)  
    {  
        Set<Id>  
        existingPropertyIds  
        = new Set<Id>()  
  
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c  
        WHERE Property__c != null]) {  
  
            existingPropertyIds.add(existingTenant.Property__c);  
        }  
    }  
}
```

```

    } for (Tenant__c newTenant :
newlist) {

    if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) { newTenantaddError('A
tenant can have only one property');

    }

}
}
}

```

### **MothlyEmailScheduler.apxc:**

```

global class MonthlyEmailScheduler implements Schedulable {

    global void execute(SchedulableContext sc) { Integer
currentDay = Date.today().day(); if (currentDay == 1) {
sendMonthlyEmails();

}

} public static void
sendMonthlyEmails() { List<Tenant__c>
tenants = [SELECT Id, Email__c FROM
Tenant__c]; for (Tenant__c tenant :
tenants) {

    String recipientEmail = tenant.Email__c;
    String emailContent = 'I trust this email finds you well. 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 a positive living environment for all.';

}
}
}
}

```

```
String emailSubject = 'Reminder: Monthly Rent Payment Due';
Messaging.SingleEmailMessage email = new

Messaging.SingleEmailMessage(); email.setToAddresses(new
String[] {recipientEmail}); email.setSubject(emailSubject);
email.setPlainTextBody(emailContent);

Messaging.sendEmail(new Messaging.SingleEmailMessage[] {email});

}

}

}
```

