

# LEASE MANAGEMENT

**College Name:** HINDUSTHAN COLLEGE OF SCIENCE AND COMMERCE

**College Code:** CA

**Team ID :** NM2025TMID28221

**TEAM MEMBERS:** 5

**Team LeaderName:** ALFEYA.K

**Email:** [alfeyak366@gmail.com](mailto:alfeyak366@gmail.com)

**Team Member:** POORVISHA.V

**Email:** [vellingiriv313@gmail.com](mailto:vellingiriv313@gmail.com)

**Team Member:** MADHUMITHA.M

**Email:** [mmadhu83444@gmail.com](mailto:mmadhu83444@gmail.com)

**Team Member:** PRADEEP.R

**Email:** [pradeep12232510@gmail.com](mailto:pradeep12232510@gmail.com)

**Team Member:** AKASH.K

**Email:** [akashakash894039@gmail.com](mailto:akashakash894039@gmail.com)

## 1.INTRODUCTION

### 1.1 Project Overview

The Lease Management System is a Salesforce-based application designed to streamline the

processes associated with leasing real estate properties. It handles tenant management, lease contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



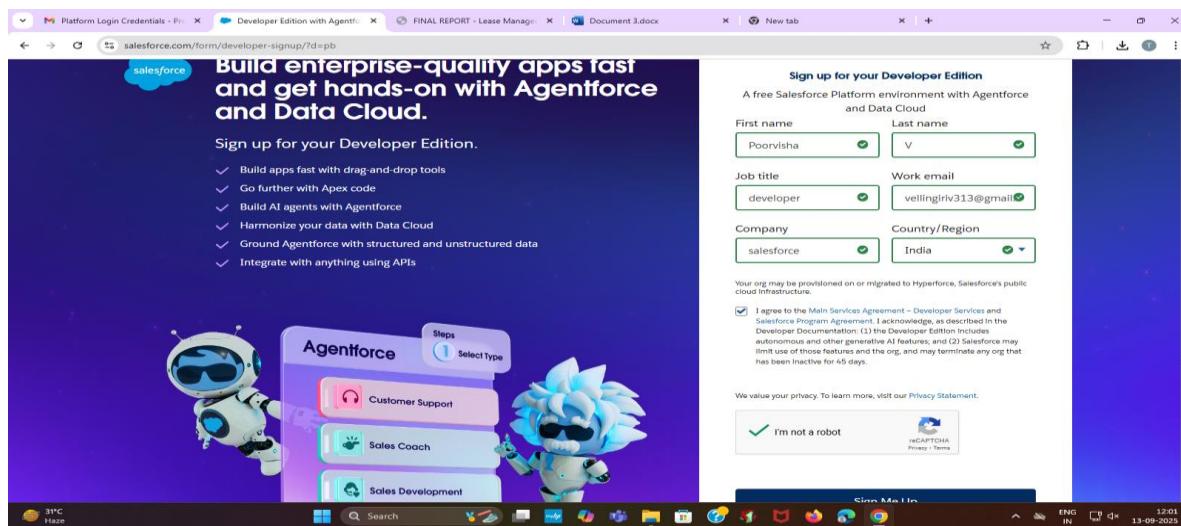
## 1.2 Purpose

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

The screenshot shows the Salesforce Object Manager interface for the 'property' 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 main 'Details' tab displays the following information:

- Description: property
- API Name: property\_c
- Custom: ✓
- Singular Label: property
- Plural Label: property
- Enable Reports: ✓
- Track Activities: ✓
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

At the bottom of the window, there are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists the same configuration options as the previous screenshot. The main 'Details' tab displays the following information:

- Description: Tenant
- 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

At the bottom of the window, there are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The left sidebar lists various configuration options for the 'lease' object, including 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 main panel displays the 'Details' tab for the 'lease' object. The 'Description' section contains the API Name 'lease\_c' (Custom), Singular Label 'lease', and Plural Label 'lease'. The 'Enable Reports' section includes checkboxes for Track Activities, Track Field History, and Deployment Status (set to 'Deployed'). A 'Help Settings' link points to the Standard salesforce.com Help Window.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The left sidebar lists various configuration options for the 'Payment' object, including Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Object Access, Validation Rules, and Conditional Field Formatting. The main panel displays the 'Details' tab for the 'Payment' object. The 'Description' section contains the API Name 'Payment' (Custom), Singular Label 'Payment', and Plural Label 'Payments'. The 'Enable Reports' section includes checkboxes for Track Activities, Track Field History, and Deployment Status. A 'Help Settings' link points to the Standard salesforce.com Help Window.

- Configured fields and relationships

Recently Viewed | Payment for | property | Salesforce | Inbox (351) - mmadhu83444@... | +

orgfarm-f9e740c205-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK000001x9Q5/FieldsAndRelationships/view

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(50)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User,Group)	✓	
property Name	Name	Text(80)	✓	
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

29°C Haze

Setup Home Object Manager

SETUP > OBJECT MANAGER  
property

Recently Viewed | Payment for | lease | Salesforce | Inbox (351) - mmadhu83444@... | +

orgfarm-f9e740c205-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK000001x4zu/FieldsAndRelationships/view

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
check for payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)	✓	
Payment	Payment__c	Lookup(Payment)	✓	
Payment date	Payment_date__c	Date		
property	property__c	Lookup(property)	✓	
property	properties__c	Master-Detail(property)	✓	
start date	start_date__c	Date		

29°C Haze

Setup Home Object Manager

SETUP > OBJECT MANAGER  
lease

- Developed Lightning App with relevant tabs

**App Details & Branding**

**App Details**

- \* App Name: Lease ManagementDeveloper
- \* Developer Name: Lease\_ManagementDeveloper
- Description: Enter a description...

**App Branding**

- Image: Placeholder image
- Primary Color Hex Value: #0070D2
- Upload button

**Org Theme Options**

Use the app's image and color instead of the org's custom theme

**App Launcher Preview**

LM Lease ManagementDevelo...

**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**

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

**Selected Items**

- Payment for tenant
- Tenants
- property
- lease
- Tenant Approval

The screenshot shows the 'User Profiles' section of the Lightning App Builder. On the left, a sidebar lists 'App Settings' with 'User Profiles' selected. The main area is titled 'User Profiles' and contains a sub-section 'Available Profiles' with a list of profiles. A 'Selected Profiles' panel on the right shows one profile, 'System Administrator', which has been moved from the available list. Navigation icons for back, forward, and search are visible at the bottom.

Available Profiles

- 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
- Customer Community Plus Login User

Selected Profiles

- System Administrator

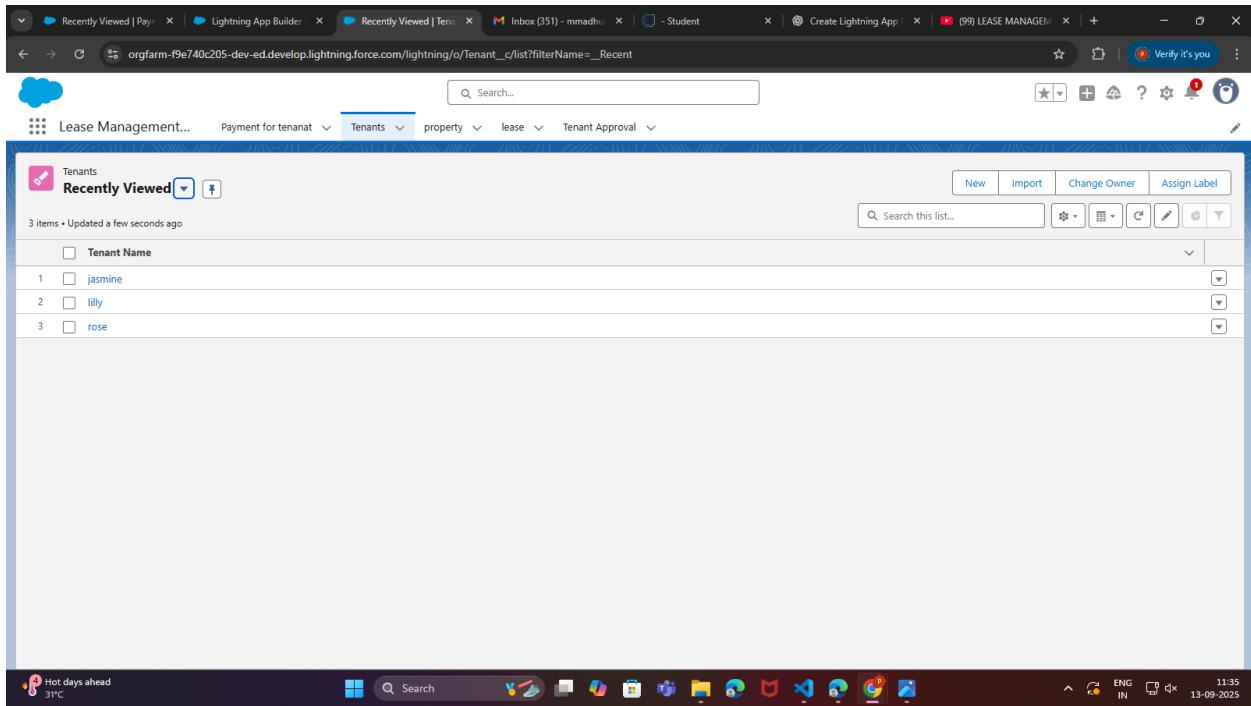
This screenshot is identical to the one above, showing the 'User Profiles' configuration screen in the Lightning App Builder. The sidebar, main content, and navigation icons are all the same, indicating no changes have been made to the user profiles since the previous screenshot.

Available Profiles

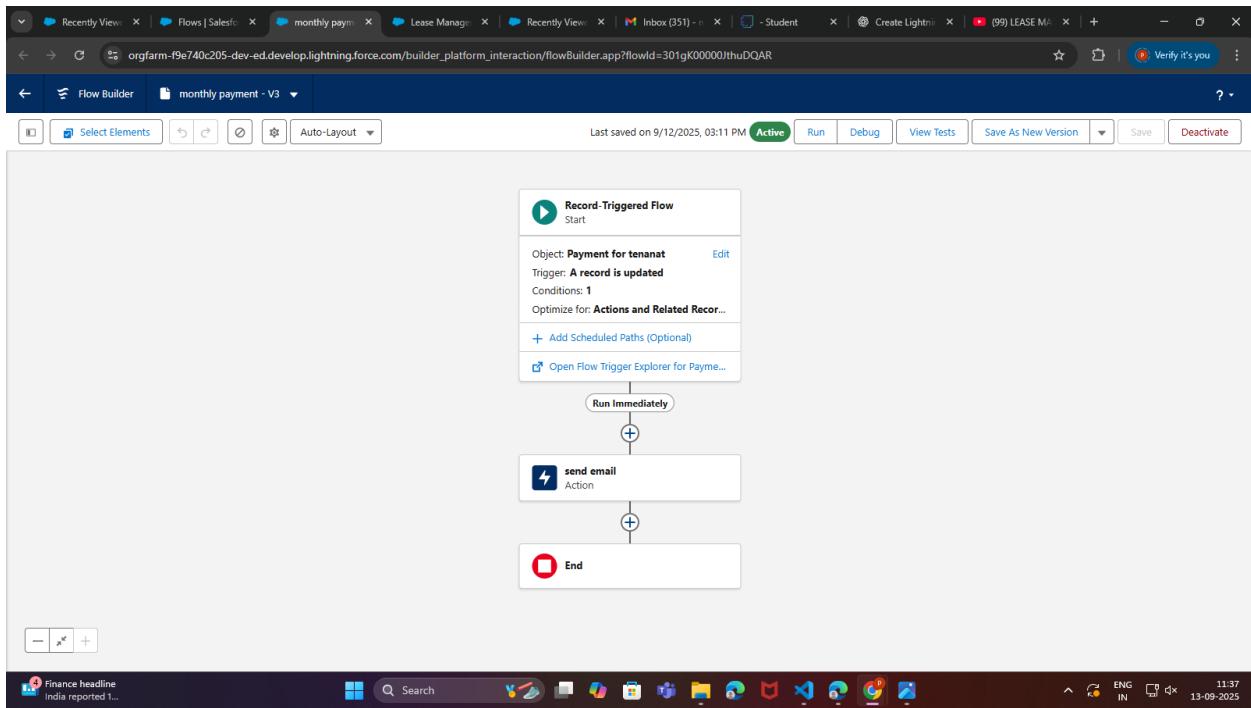
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User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

Selected Profiles

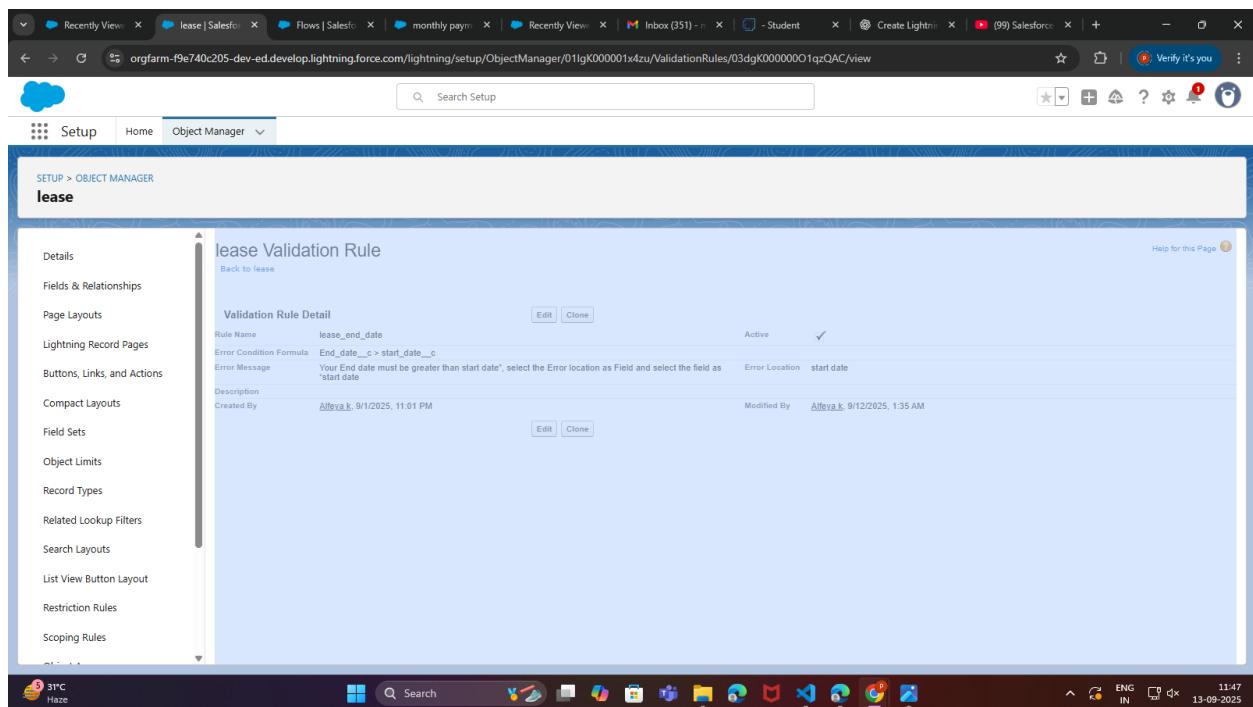
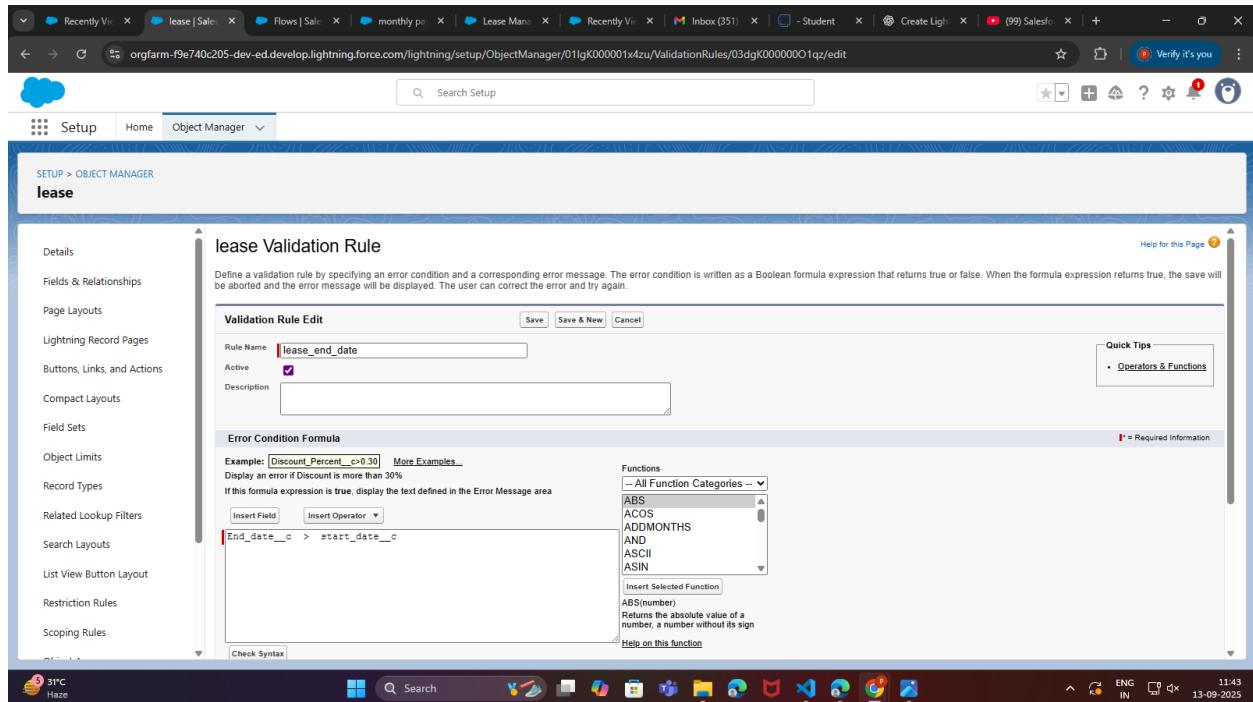
- System Administrator



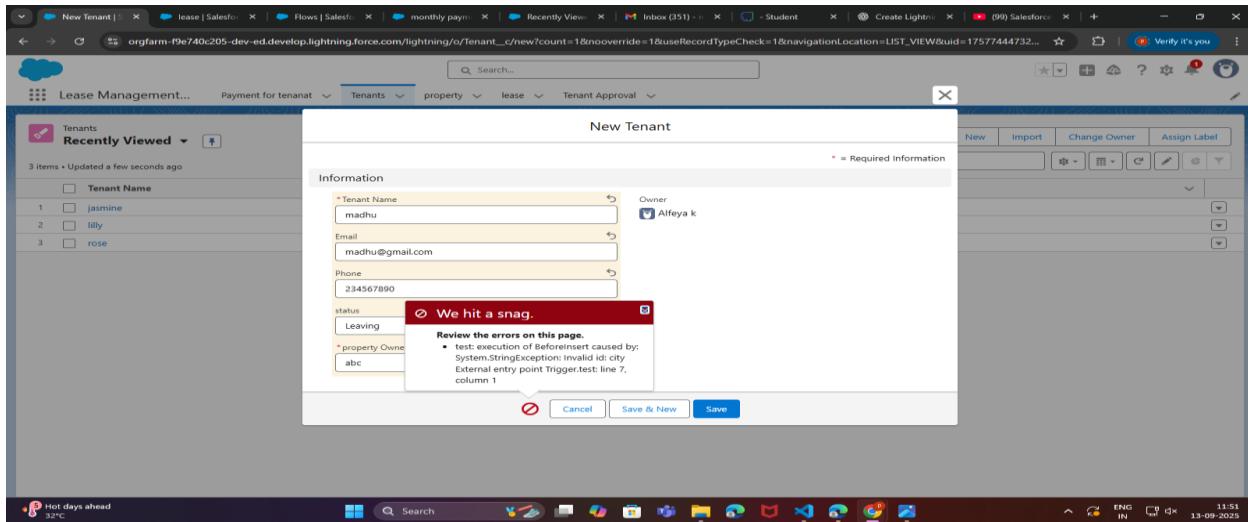
- Implemented Flows for monthly rent and payment success



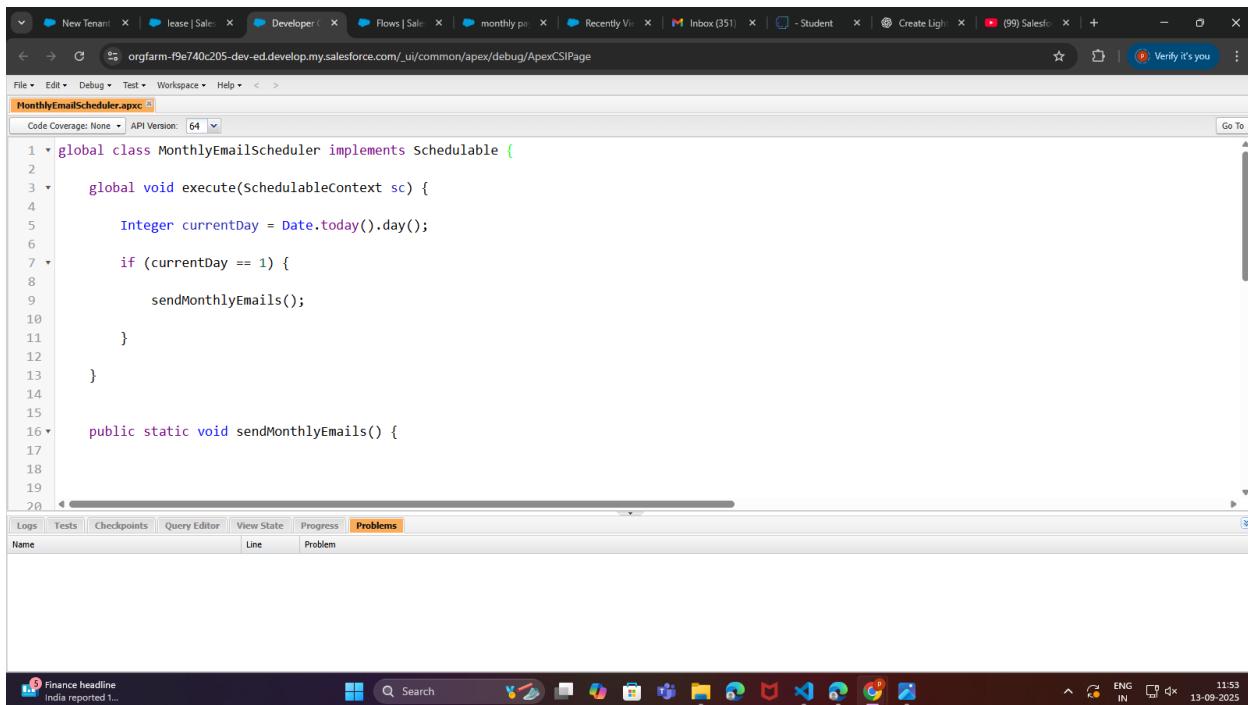
- To create a validation rule to a Lease Object



- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class



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

The screenshot shows the Salesforce Setup interface with the search bar set to 'class'. The left sidebar is expanded, showing sections for Email (Classic Email Templates, Classic Letterheads), Custom Code (Apex Classes), and Data Classification (Data Classification Download, Data Classification Settings, Data Classification Upload). A message at the bottom says 'Didn't find what you're looking for? Try using Global Search.' The main content area displays the 'Classic Email Templates' page for a 'Text Email Template' named '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
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Alleya K [Change]
Description	
Created By	Alleya K, 9/10/2025, 11:10 PM
Modified By	Alleya K, 9/10/2025, 11:54 PM

The 'Email Template' section contains the following content:

**Subject:** Leave approved  
**Plain Text Preview:**  
dear(!@10gK00000xAll.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 leave is approved. You can leave now

The screenshot shows the Salesforce Setup interface with the search bar set to 'class'. The left sidebar is expanded, showing sections for Email (Classic Email Templates, Classic Letterheads), Custom Code (Apex Classes), and Data Classification (Data Classification Download, Data Classification Settings, Data Classification Upload). A message at the bottom says 'Didn't find what you're looking for? Try using Global Search.' The main content area displays the 'Classic Email Templates' page for a 'Text Email Template' named '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
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Alleya K [Change]
Description	
Created By	Alleya K, 9/10/2025, 11:50 PM
Modified By	Alleya K, 9/10/2025, 11:54 PM

The 'Email Template' section contains 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 search bar set to 'class'. The left sidebar is expanded, showing sections for Email (Classic Email Templates, Classic Letterheads), Custom Code (Apex Classes), and Data Classification (Data Classification Download, Data Classification Settings, Data Classification Upload). A message at the bottom says 'Didn't find what you're looking for? Try using Global Search.' The main content area displays the 'Leave rejected' email template. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave_rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Allevya K [Chango]
Description	
Created By	Allevya K   9/1/2025, 11:11 PM
Modified By	Allevya K   9/10/2025, 11:54 PM

The 'Available For Use' checkbox is checked. Below this is the 'Email Template' preview section, which includes a subject line 'Leave rejected' and a plain text preview:

Subject : Leave rejected  
Plain Text Preview :  
Dear {!01eK000001xAljName},  
  
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 search bar set to 'class'. The left sidebar is expanded, showing sections for Email (Classic Email Templates, Classic Letterheads), Custom Code (Apex Classes), and Data Classification (Data Classification Download, Data Classification Settings, Data Classification Upload). A message at the bottom says 'Didn't find what you're looking for? Try using Global Search.' The main content area displays the 'Tenant Email' email template. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Tenant_Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Allevya K [Chango]
Description	Text Email Template: Tenant Email - Salesforce - Developer Edition
Created By	Allevya K   9/1/2025, 11:12 PM
Modified By	Allevya K   9/10/2025, 11:54 PM

The 'Available For Use' checkbox is checked. Below this is the 'Email Template' preview section, which includes a subject line 'Urgent: Monthly Rent Payment Reminder' and a plain text preview:

Subject : Urgent: Monthly Rent Payment Reminder  
Plain Text Preview :  
Dear {!01eK000001xAljName},  
  
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.  
  
This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due. To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

**Classic Email Templates**

**tenant payment**

**Email Template Detail**

Email Template Name	Unified Public Classic Email Templates
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	Alleya_k [Change]
Description	
Created By	Alleya_k, 9/1/2025, 11:13 PM
Modified By	Alleya_k, 9/10/2025, 11:54 PM

**Email Template**

Subject	Confirmation of Successful Monthly Payment
Plain Text Preview	<p>Dear (001lgK000001xAj00NgK00001ZKyhk),</p> <p>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.</p>

## ● Approval Process creation

For Tenant Leaving:

**Approval Processes**

**Tenant: check for vacant**

**Process Definition Detail**

Process Name	check for vacant
Unique Name	check_for_vacant
Description	
Entry Criteria	Tenant: status NOT EQUAL TO Leaving
Record Editability	Administrator ONLY
Approval Assignment Email Template	
Initial Submitters	Tenant Owner
Created By	Alleya_k, 9/12/2025, 1:45 AM
Modified By	Alleya_k, 9/12/2025, 1:49 AM

**Initial Submission Actions**

Action	Type	Description
Record Lock		Lock the record from being edited please_approve_my_leave

**Approval Steps**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	check for vacant		Manually Chosen		Final Rejection

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The process 'check for vacant' is selected. The 'Process Definition Detail' section includes fields for Process Name (check for vacant), Unique Name (check\_for\_vacant), Description (Tenant: check for vacant), Entry Criteria (Tenant: status NOT EQUAL TO Leaving), Record Editability (Administrator ONLY), and various other settings like 'Active' (checked) and 'Allow Submitters to Recall Approval Requests' (checked). The 'Initial Submission Actions' section lists a 'Record Lock' action with a description 'Lock the record from being edited' and a note 'please approve my leave'. The 'Approval Steps' section shows one step named 'check for vacant' with an 'Assigned Approver' set to 'Manually Chosen'.

## ● 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 an Apex trigger named 'test' in the 'testApex' class. The trigger is defined to run before insert on the 'Tenant\_\_c' object. It contains logic to prevent the insert if the trigger is an insert and is before the record is saved. The code uses a static method 'preventInsert' from a class named 'testHandler'.

### 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                 newTenantaddError('A tenant can have only one property');
19             }
20         }
21     }
22 }

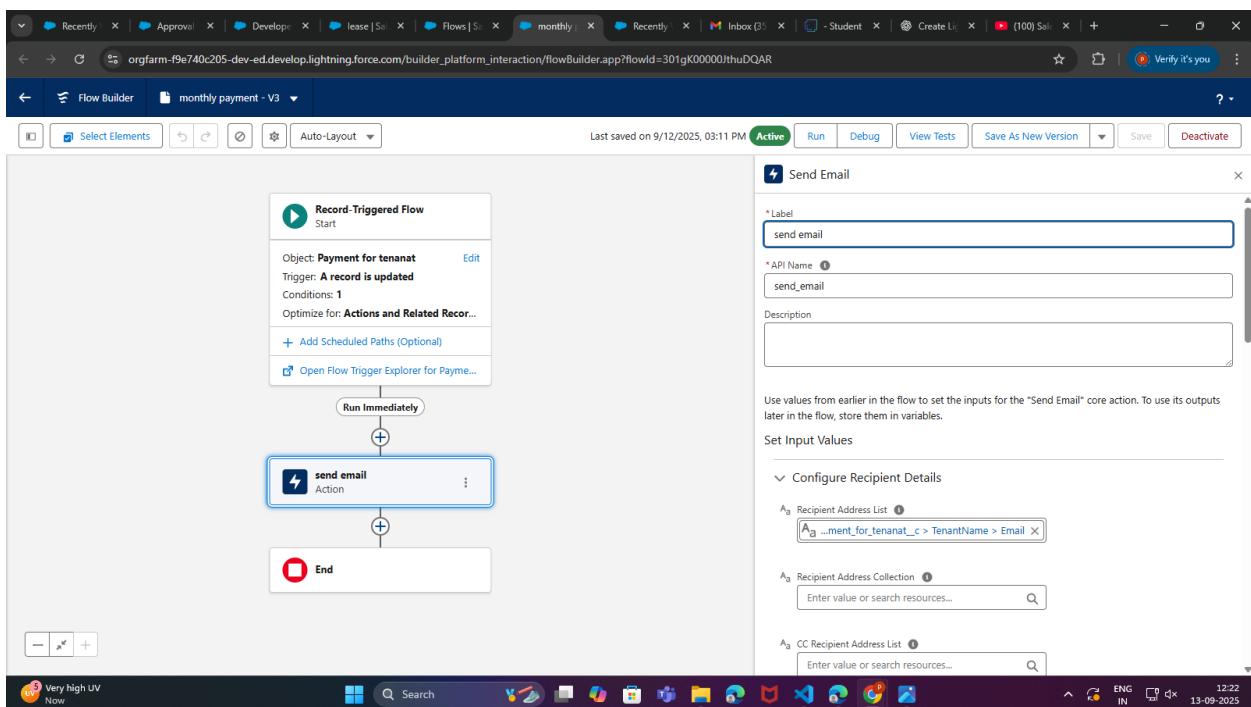
```

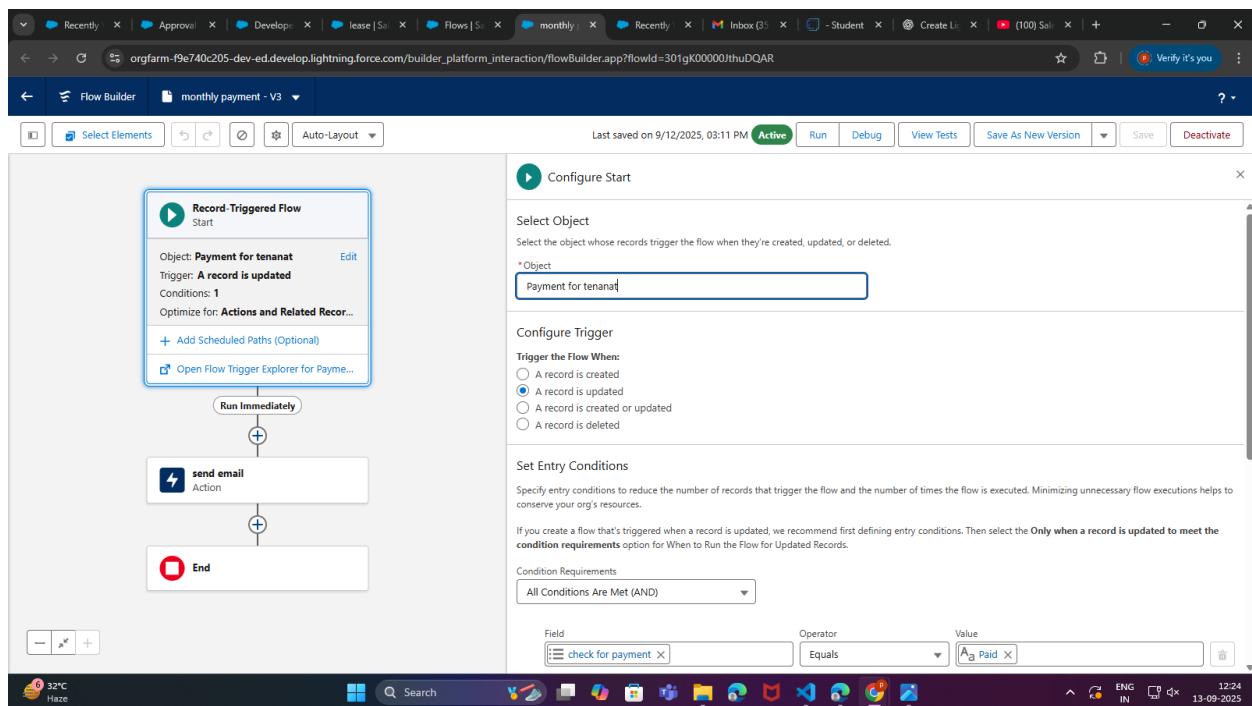
Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Finance headline India reported 1... 12:19 13-09-2025

## ● FLOWS





- Schedule class:

## Create an Apex Class

```

1  global class MonthlyRentalScheduler implements Schedulable {
2    global void execute(SchedulableContext sc) {
3      Integer currentDay = System.Today().getday();
4      If (currentDay == 15) {
5        sendEmailIfDue();
6      }
7    }
8
9
10   public static void sendEmailIfDue() {
11     List<Account> tenants = [SELECT Id, Email__c FROM Tenant__c];
12     For (Account__c tenant : tenants) {
13       String reminderEmail = tenant.Email__c;
14       String emailContent = 'I trust this small finds you well. I am writing to remind you that the monthly rent is due now timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
15       String emailSubject = 'Reminder: monthly rent payment due';
16       Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
17       email.setToAddresses(String.valueOf(tenant.Email__c));
18       email.setSubject(emailSubject);
19       email.setPlainTextBody(emailContent);
20       Messaging.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[] {email};
21       Messaging.sendEmail(emails);
22     }
23   }
24 }

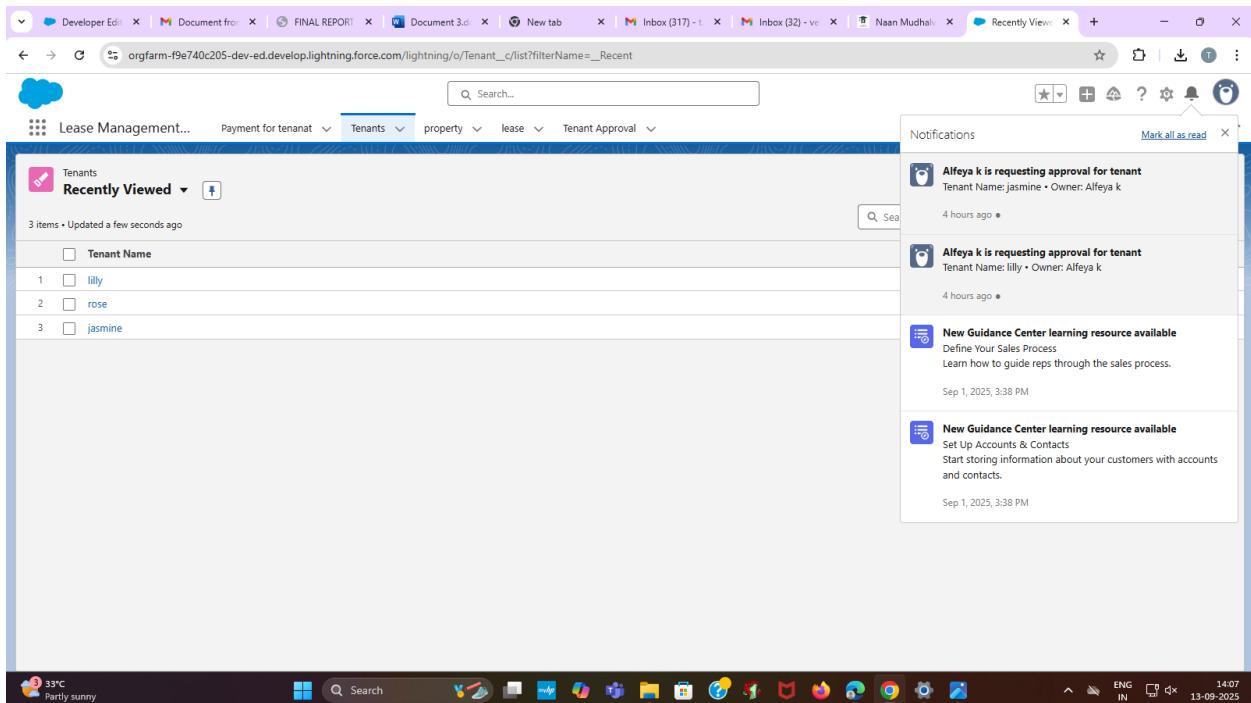
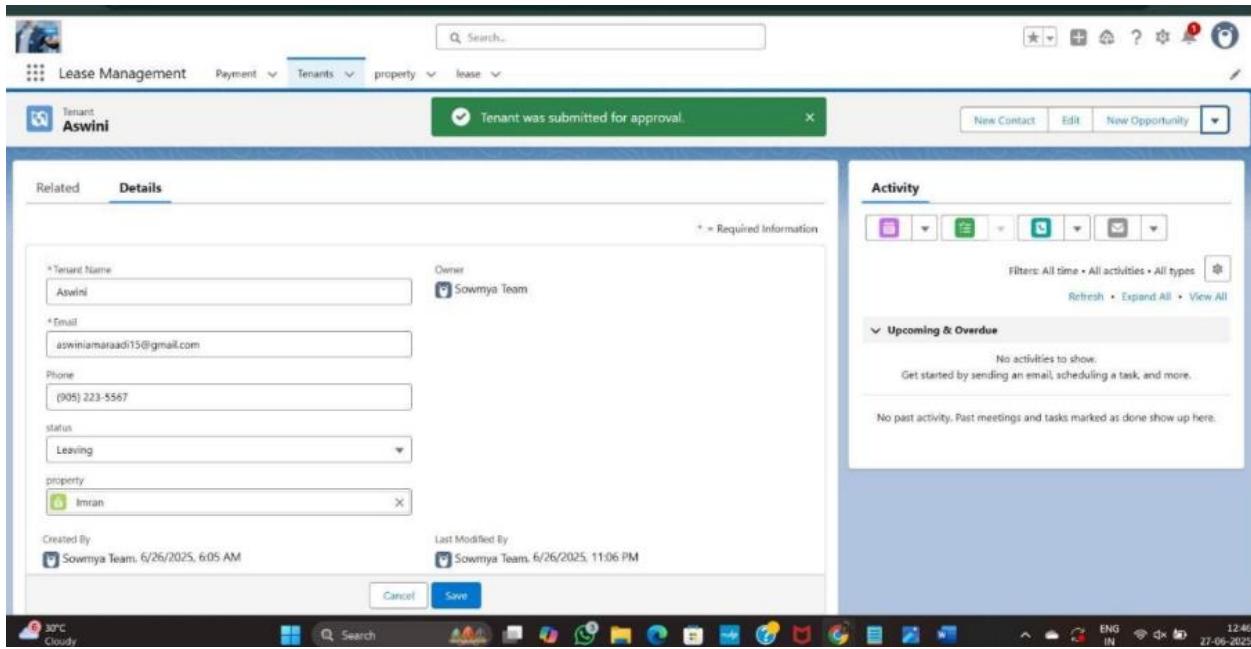
```

## Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left is collapsed, showing sections like Email, Custom Code, Environments, and Jobs. The main area displays the Apex Class MonthlyEmailScheduler. The code is as follows:

```
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) {  
    }
```

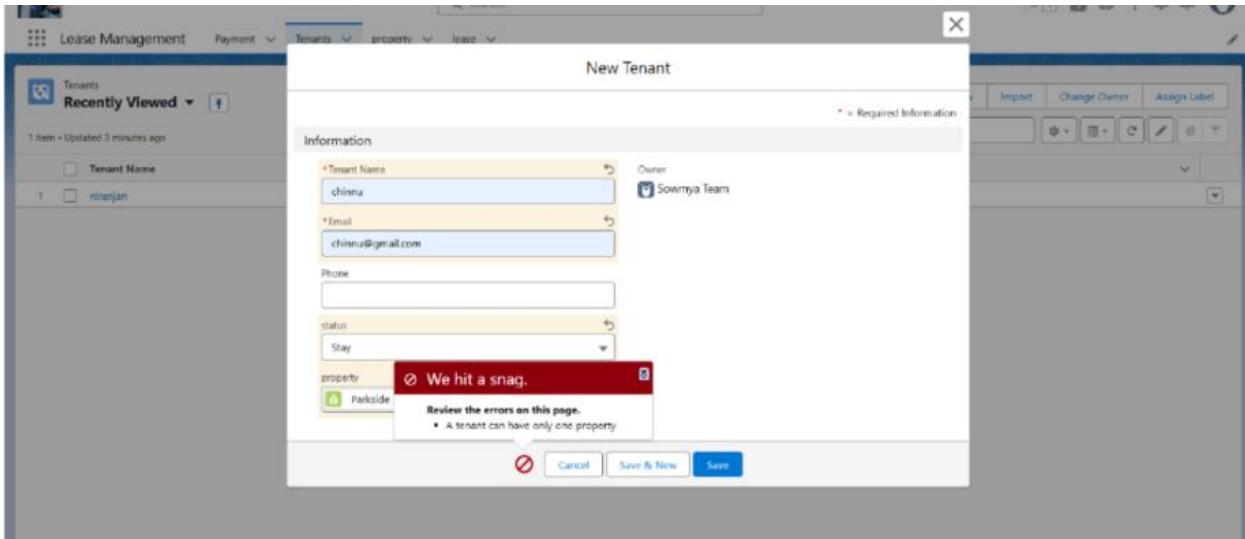
The screenshot shows the Salesforce Lease Management... page for a tenant named 'rose'. The sidebar on the right is open, showing the Activity section. The 'New Case' button is highlighted. Other options in the sidebar include 'Submit for Approval', 'Sharing Hierarchy', 'Edit', 'Clone', 'Edit Labels', 'Change Owner', 'Delete', 'Sharing', and 'Printable View'. The activity list is currently empty.



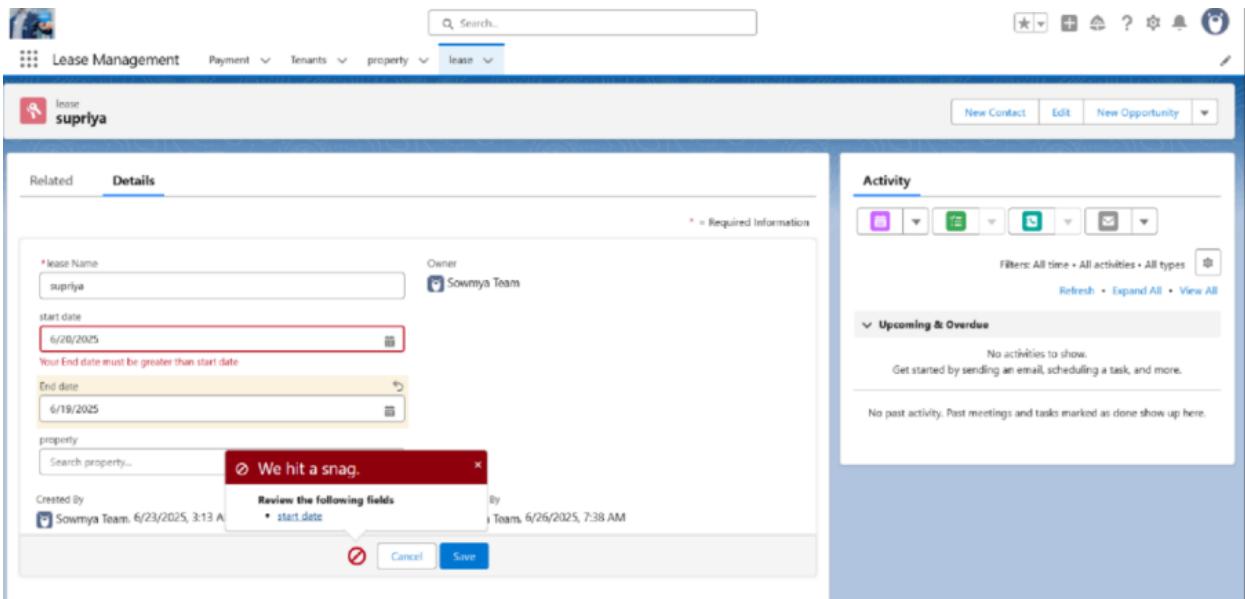
## FUNCTIONAL AND PERFORMANCE TESTING

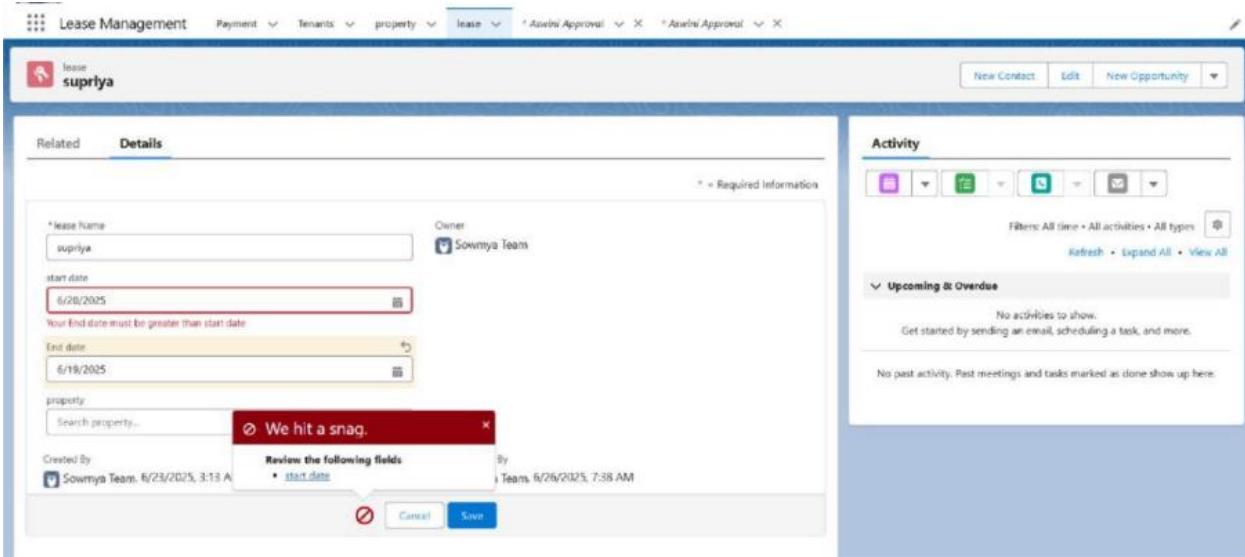
### Performance Testing

- Trigger validation by entering duplicate tenant-property records

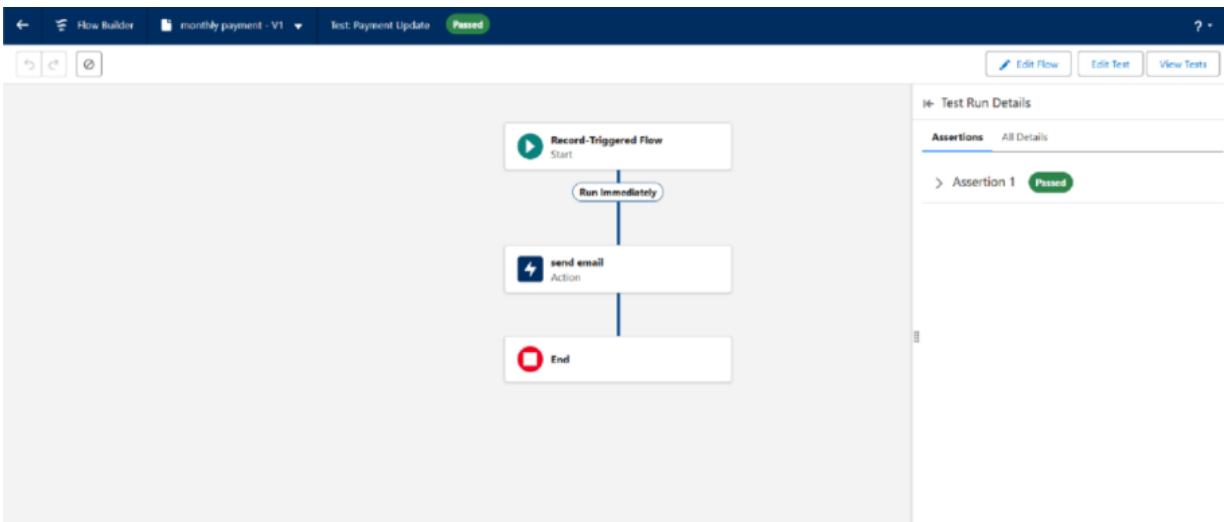


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Tenant: niranjan

Related Details

\* = Required Information

Tenant Name	niranjan
Email	niranjan1506@gmail.com
Phone	
Status	Stay
property	Parkside Lofts

Created By: Sowmya Team, 6/23/2025, 2:33 AM      Last Modified By: Sowmya Team, 6/23/2025, 3:58 AM

[Cancel](#) [Save](#)

Notifications

- A Approval request for the tenant is approved niranjan a few seconds ago \*
- B Approval request for the tenant is rejected niranjan Jun 23, 2025, 4:29 PM
- C Approval request for the tenant is approved niranjan Jun 23, 2025, 4:25 PM
- D Approval request for the tenant is approved niranjan Jun 23, 2025, 4:14 PM
- E New Guidance Center learning resource available Define Your Sales Process Learn how to guide rep through the sales process. Jun 20, 2025, 1:28 PM \*

Tenant: niranjan

[New Contact](#) [Edit](#) [New Opportunity](#)

Approval History (6+)

Step Name	Date	Status	Assigned To
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team
Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team
Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team

[View All](#)

Payment (2)

Payment Name
Jack
Rahul

[New](#)

No past activity. Past meetings and tasks marked as done show up here.

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It includes sections for 'Custom Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Tabs' section displays a table with columns for Action, Label, Tab Style, and Description. The 'Web Tabs' and 'Visualforce Tabs' sections show that no tabs have been defined.

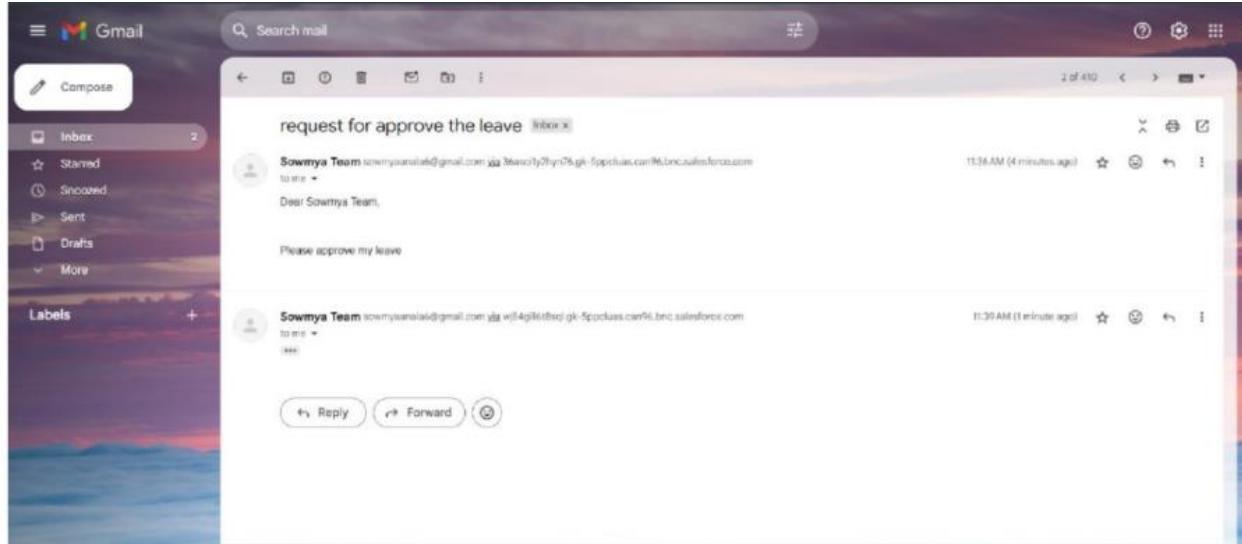
Action	Label	Tab Style	Description
Edit   Del	Issue	Keyns	
Edit   Del	Payment	Credit Card	
Edit   Del	Records	Back	
Edit   Del	Tickets	Map	

- Email alerts

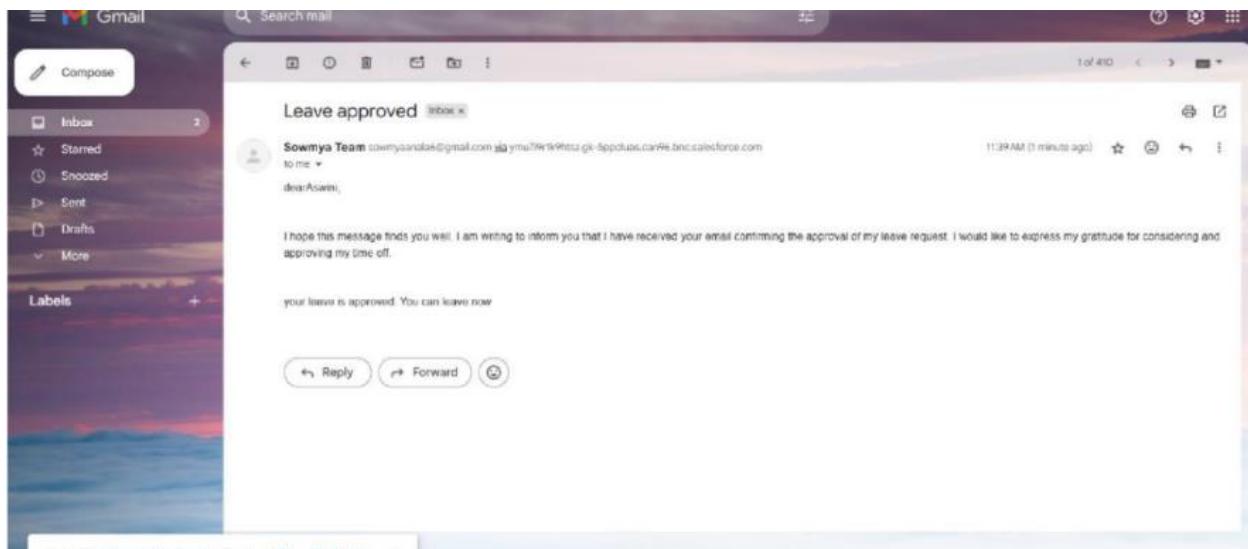
The screenshot shows the 'Lease Management' application interface. It displays a grid titled 'Approval History' with columns for Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The grid contains 8 items, all of which are pending approval.

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
3 Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
4 Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team	Sowmya Team	Leaving
5 Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team	Sowmya Team	Approved
6 Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team	Sowmya Team	leaving
7 Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
8 Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	Leaving

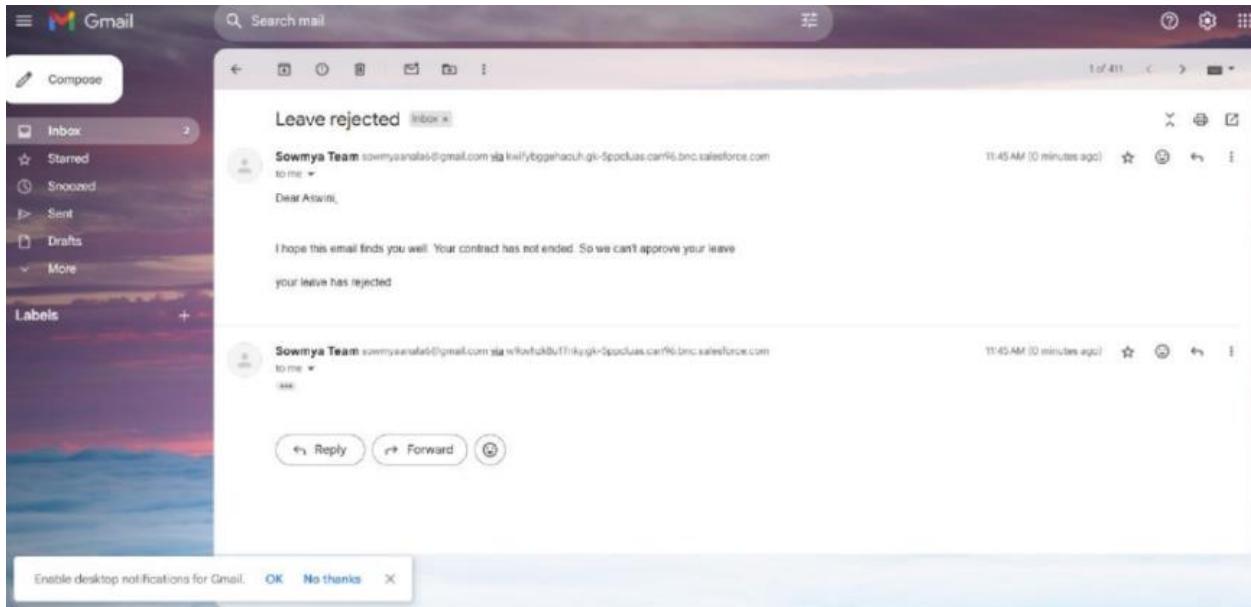
- Request for approve the leave



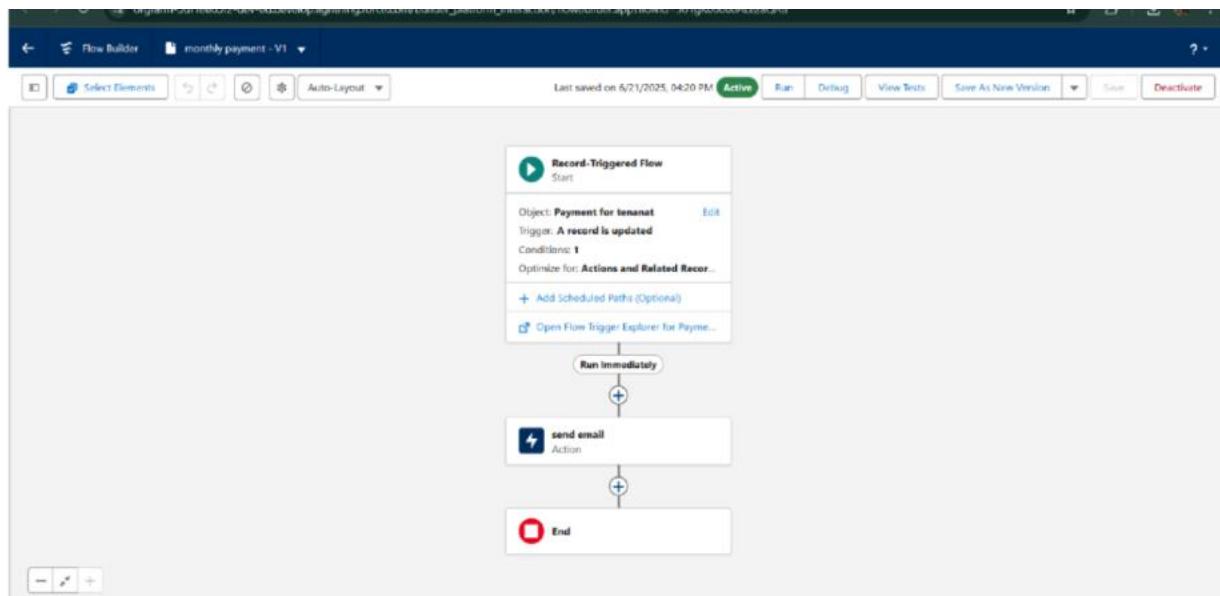
- Leave approved



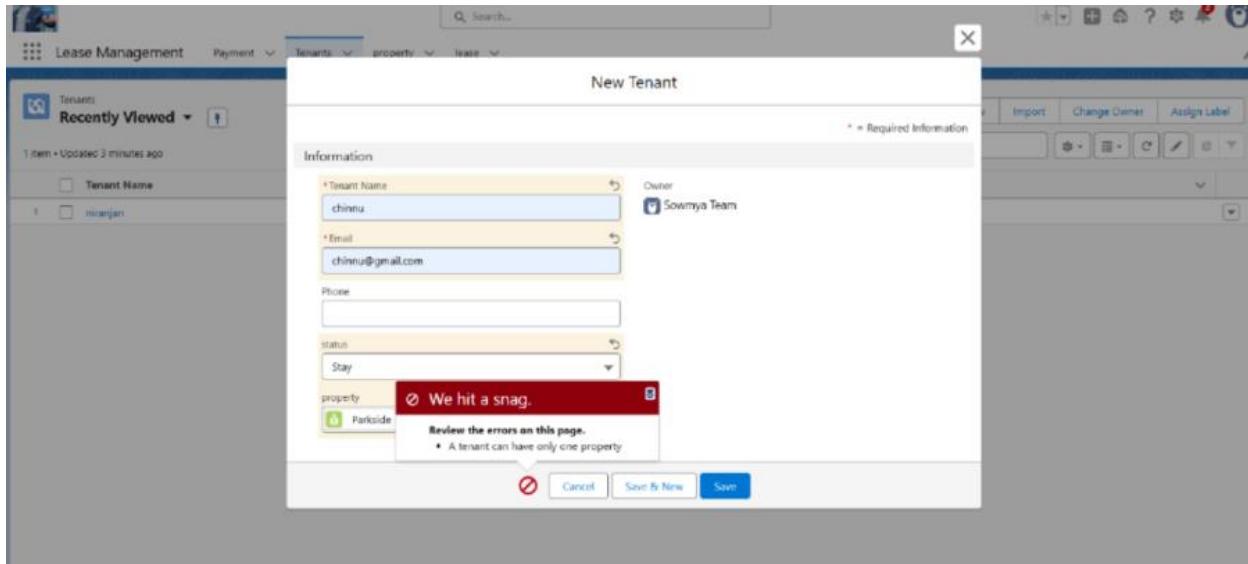
- Leave rejected



- Flow runs



- Trigger error messages



- Approval process notifications



# ADVANTAGES & DISADVANTAGES

## CONCLUSION

The Lease Management System successfully streamlines the operations of leasing through a

structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

## 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 existingPropertyIds
= new Set()

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 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});

}

}
```