

LEASE MANAGEMENT

**College Name: NANDHA ARTS AND SCIENCE
COLLEGE**

College Code:bru4j

TEAM ID: NM2025TMID22885

TEAM MEMBERS:

Team Leader Name: Dharanish S

Email: dharan018@gmail.com

Team Member 1:Dhanush P

Email:dhanush807282@gmail.com

Team Member 2:Darsha Sabarish R

Email:rdarshasabarish@gmail.com

Team Member 3:Gowtham S

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.



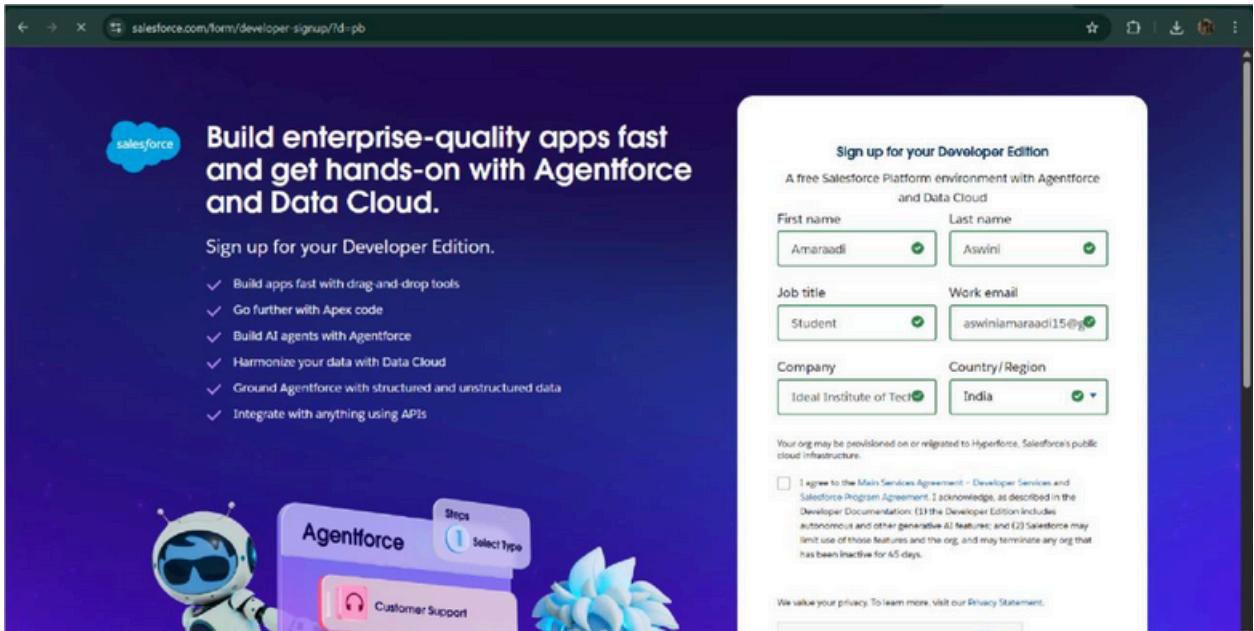
1.2 Purpose

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.

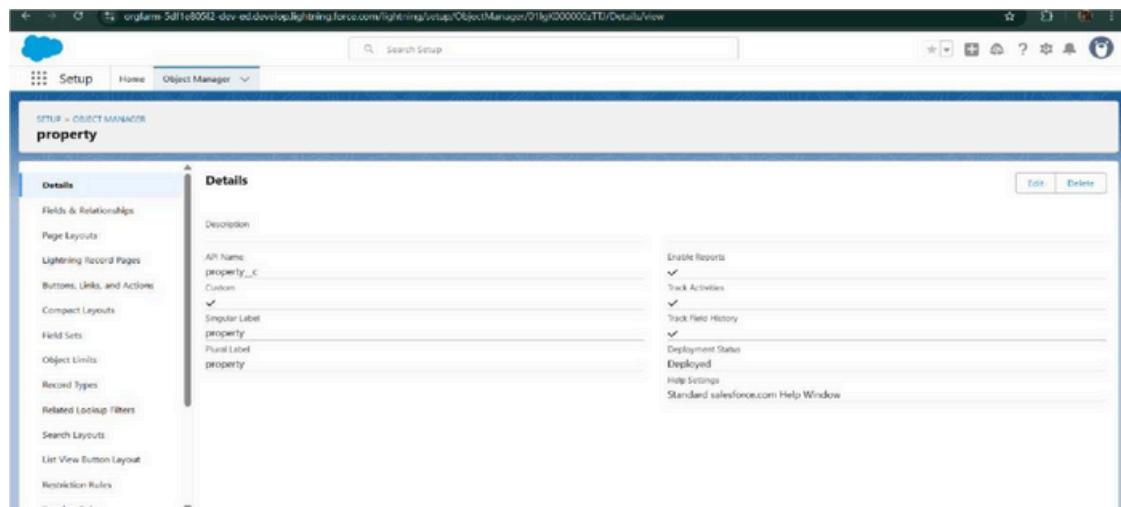
DEVELOPMENT PHASE

Creating Developer Account:

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



- Created objects: Property, Tenant, Lease, Payment



SETUP > OBJECT MANAGER

Tenant

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

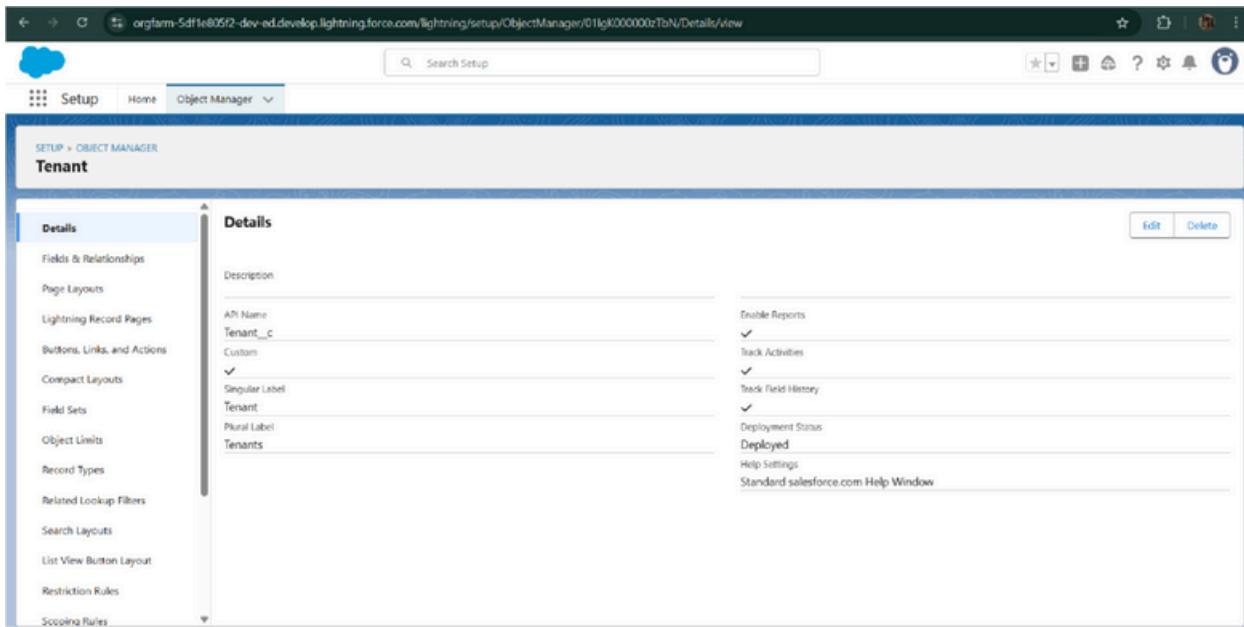
Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Edit Delete



SETUP > OBJECT MANAGER

lease

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

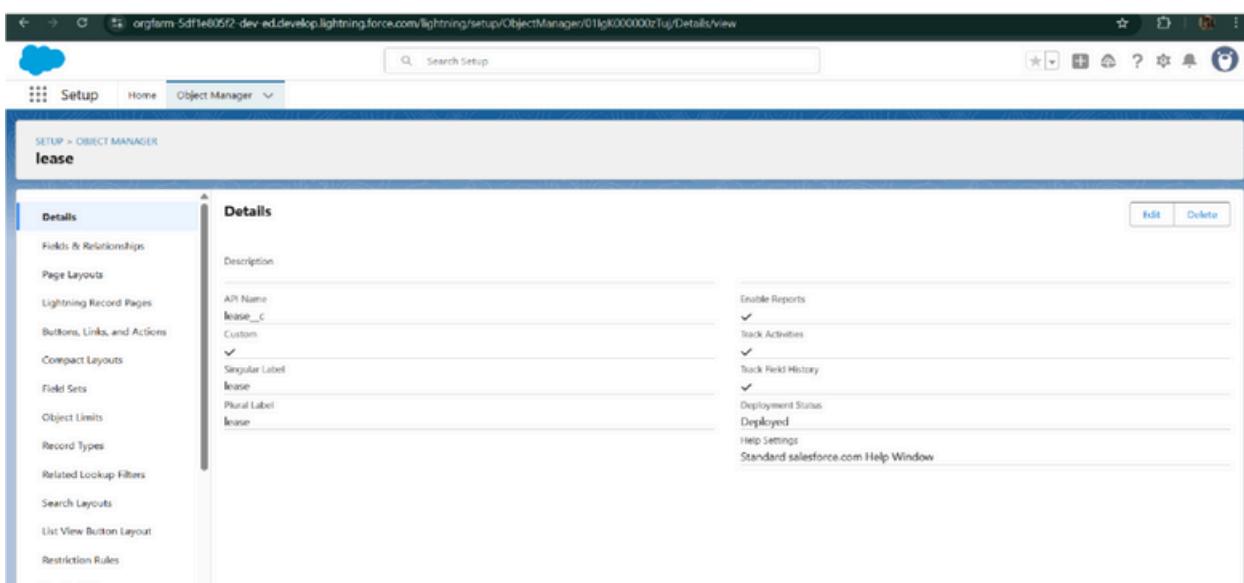
Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Edit Delete



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details' and 'Fields & Relationships'. The main content area displays the 'Details' tab for the 'Payment for tenantat' object. It shows fields like 'API Name' (Payment_for_tenantat_c), 'Singular Label' (Payment for tenantat), and 'Plural Label' (Payment). To the right, there are sections for 'Unstable Reports', 'Trunk Activities', 'Trunk Field History', 'Deployment Status', and 'Help Settings'. Buttons for 'Edit' and 'Delete' are at the bottom right.

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'property'. On the left, a sidebar lists various configuration options under 'Details' and 'Fields & Relationships'. The main content area displays the 'Fields & Relationships' tab, which lists 9 items sorted by field label. The table includes columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The table rows are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Text(25)		
Owner	OwnerId	Lookup(User,Group)		
property	property_c	Lookup(property)		
property Name	Name	Text(30)		
sft	sft_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	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		

Fields & Relationships
7 items. Sorted by Field Label

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

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

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

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

Type to filter list... [Create](#)

- 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

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

User Profiles

Choose the user profiles that can access this app.

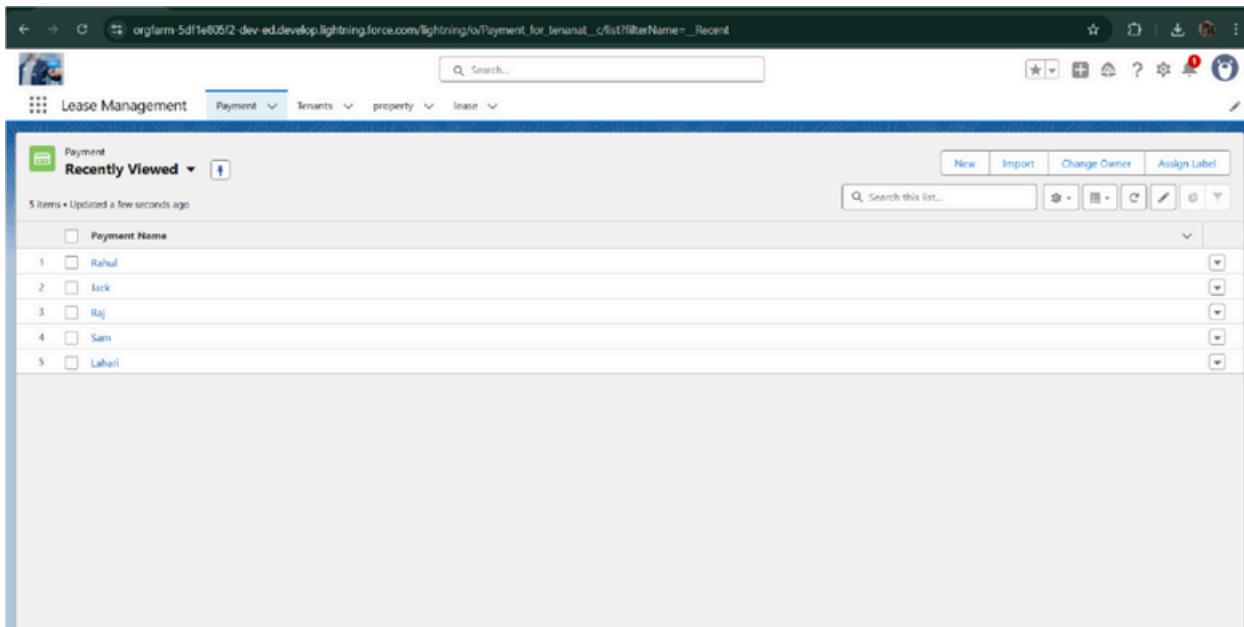
Available Profiles

Type to filter list...

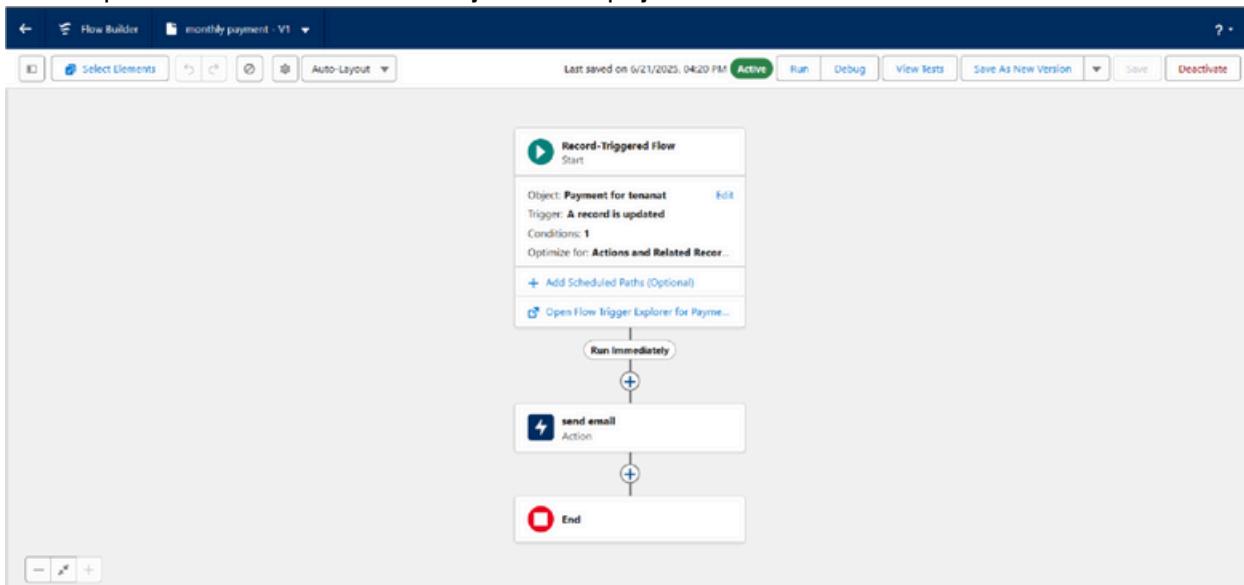
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator



- Implemented Flows for monthly rent and payment success

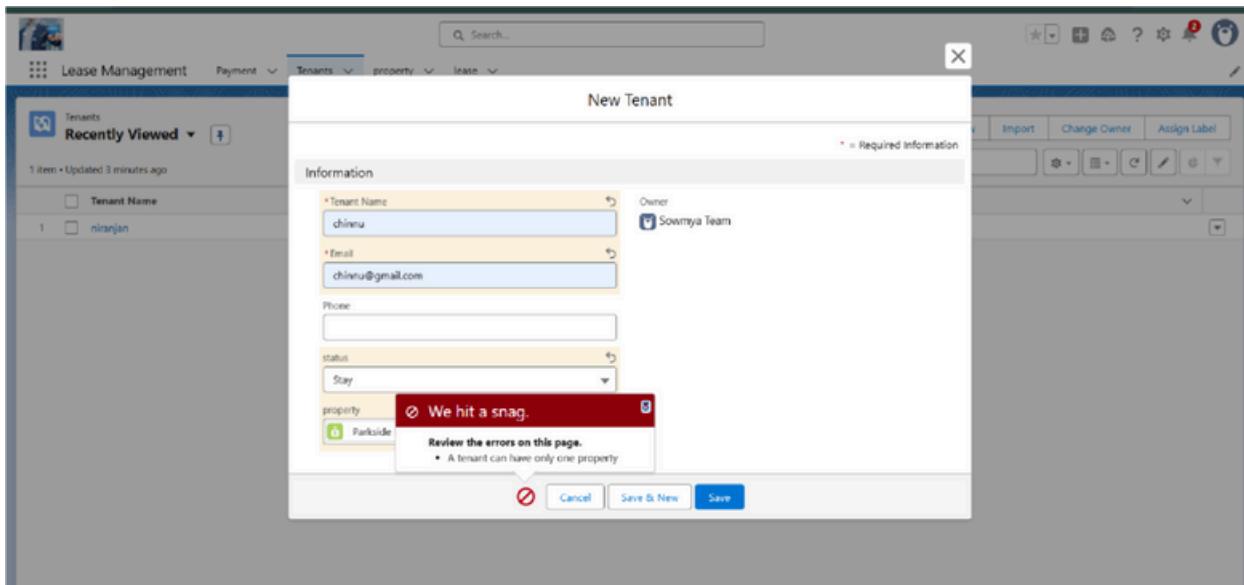


- To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' screen for the 'lease' object. The 'Role Name' field contains 'lease_end_date'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'End_date_c <= start_date_c'. A tooltip for the 'ABS' function is displayed, stating: 'Returns the absolute value of a number, a number without its sign'. The 'Functions' dropdown menu lists various mathematical functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc.

The screenshot shows the 'Validation Rule Detail' screen for the 'lease' validation rule. The rule name is 'lease_end_date'. The error condition formula is 'End_date_c <= start_date_c'. The error message is 'Your End date must be greater than start date'. The error location is 'start date'. The rule was created by 'Sowmya Team' on 6/19/2025, 5:37 AM, and modified by 'Sowmya Team' on 6/26/2025, 7:47 AM.

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```
1 *global class MonthyEmailScheduler implements Schedulable {
2 *
3 *    global void execute(SchedulableContext sc) {
4 *
5 *        Integer currentDay = Date.today().day();
6 *
7 *        if (currentDay == 1) {
8 *
9 *            sendMonthyEmails();
10 *
11     }
12 }
13
14
15 *public static void sendMonthyEmails() {
16 *
17     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19     for (Tenant__c tenant : tenants) {
20
21         String recipientEmail = tenant.Email__c;
22
23         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";
24
25         String emailSubject = "Reminder: Monthly Rent Payment Due";
26
27         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28
29         email.setToAddresses(new String[]{recipientEmail});
30
31         email.setSubject(emailSubject);
32
33         email.setPlainTextBody(emailContent);
34
35     }
36 }
```

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

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item 'Leave approved'. The main content area displays the 'Leave approved' email template detail. The template name is 'Leave approved' and it is available for use. The subject is 'Leave approved' and the body contains the following text:

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.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item 'tenant leaving'. The main content area displays the 'tenant leaving' email template detail. The template name is 'tenant_leaving' and it is available for use. The subject is 'request for approve the leave' and the body contains the following text:

Plain Text Preview

Dear {Tenant__c.CreatedBy},

Please approve my leave.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Email', 'Classic Email Templates' is highlighted. A search bar at the top right contains the text 'Search Setup'. The main content area displays a 'Text Email Template' named 'Leave rejected'. The 'Email Template Detail' section includes fields for Email Template Name ('Leave rejected'), Template Unique Name ('Leave_rejected'), Encoding ('Unicode (UTF-8)'), Author ('Sohma Team (Changed)'), Description (''), Created By ('Sohma Team, 6/20/2025, 1:11 AM'), and Modified By ('Sohma Team, 6/20/2025, 1:11 AM'). Buttons for 'Edit', 'Delete', and 'Clone' are available. Below this, 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 'Email', 'Classic Email Templates' is highlighted. A search bar at the top right contains the text 'Search Setup'. The main content area displays a 'Text Email Template' named 'Tenant Email'. The 'Email Template Detail' section includes fields for Email Template Name ('Tenant Email'), Template Unique Name ('Tenant_Email'), Encoding ('Unicode (UTF-8)'), Author ('Sohma Team (Changed)'), Description (''), Created By ('Sohma Team, 6/20/2025, 1:12 AM'), and Modified By ('Sohma Team, 6/20/2025, 1:12 AM'). Buttons for 'Edit', 'Delete', and 'Clone' are available. Below this, 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.'

Email Template Detail

- Email Template Name: tenant payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Screema Team (Change)
- Description: Created By: Screema Team, 6/20/2025, 1:13 AM
- Modified By: Screema Team, 6/20/2025, 1:13 AM

Email Template Preview

Subject: Confirmation of Successful Monthly Payment

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:

Process Definition Detail

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Next Automated Approver Determined by
- Entry Criteria: tenant__status equals Stay
- Record Editability: Administrator ONLY
- Allow Submitters to Recall Approval Requests:

Initial Submission Actions

Action Type	Description
Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	1	Step 1			User:Screema Team	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:
 - Entry Criteria: Tenant__c.Status equals Leaving
 - Direct Entitlement: Administrator ONLY
 - Next Automated Approver Determined By: Allowing Submitters to Recall Approval Requests
 - Approval Assignment Email Template: Lesser approved
 - Initial Submitters: Tenant Owner
 - Created By: Sowmya_Team, 6/20/2025, 3:18 AM
 - Modified By: Sowmya_Team, 6/20/2025, 11:02 PM
- Initial Submission Actions:**
 - Action: Record Lock, Type: Record Lock, Description: Lock the record from being edited.
 - Action: Email Alert, Type: Email Alert, Description: SISISI_BOERNSV.MY.RAWG
- Approval Steps:**

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

● Apex Trigger

Create an Apex Trigger

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

- Trigger:** test on Tenant__c (before insert)
- Code:**

```
1 trigger test on Tenant__c (before insert)
2
3 {
4
5     if(trigger.isInsert && trigger.isBefore){
6
7         testHandler.preventInsert(trigger.new);
8
9     }
10 }
```
- Modal Window (Open):**

Entity Type	Edition	Related
Classes	test	
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • Go To

```
1 trigger test on Tenant__c (before insert)
2
3 {
4     if(trigger.isInsert && trigger.isBefore){
5         testHandler.preventInsert(trigger.new);
6     }
7 }
8
9 }
10
11 }
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Create an Apex Handler class

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • Go To

```
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(existin
10     }
11
12
13     for (Tenant__c newTenant : newList) {
14
15         if (newTenant.Property__c != null) {
16
17             newTenantaddError('A t
18
19
20
21
22
23 }
```

Entity Type Entity Name Namespace Related

Entity Type	Entity Name	Namespace	Related
Empty List			
Classes	testHandler	MonthlyTaskScheduler	+< test ApexTrigger References
Triggers			+< property CustomerId References
Pages			+< Tenant__c Subject References
Page Components			+< Tenant__c Subject References
Objects			
Item Resources			
Packages			

Open Filter: Filter the repository (* = any string) Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-Sd1t0e05f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

Test.Javascript testHandler.apc [] HostedDynamicScheduler.apc []

Code Coverage Class Name API Version 54 Go To

```

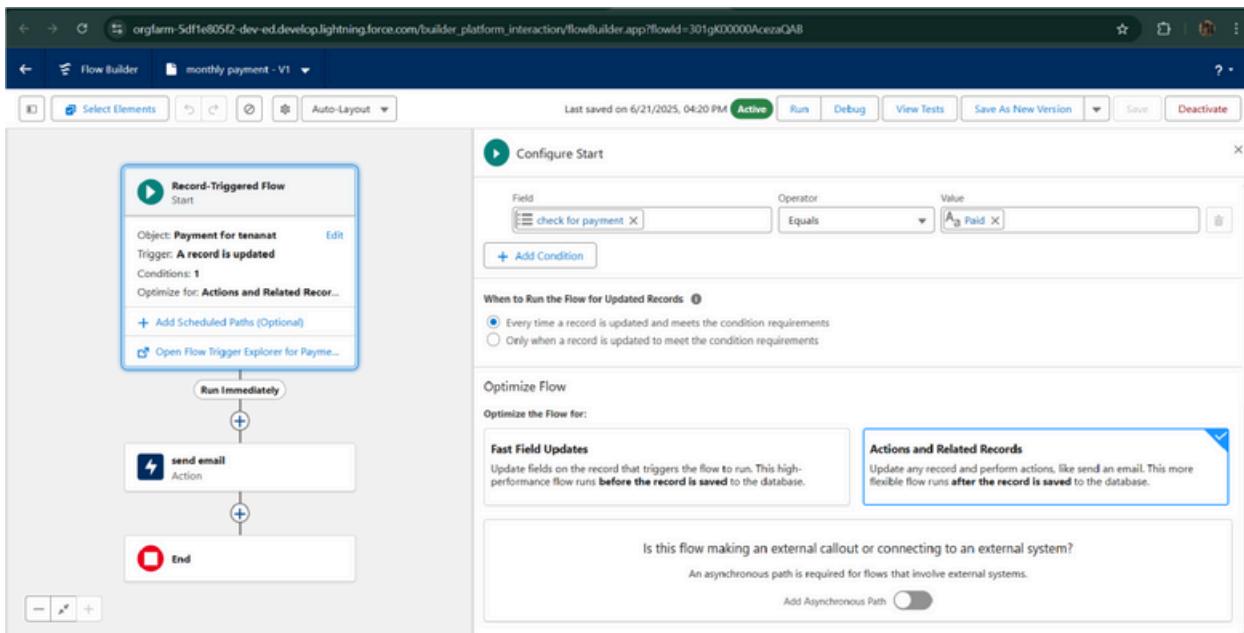
1 • public class testHandler {
2
3 •     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8             existingPropertyIds.add(existingTenant.Property__c);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

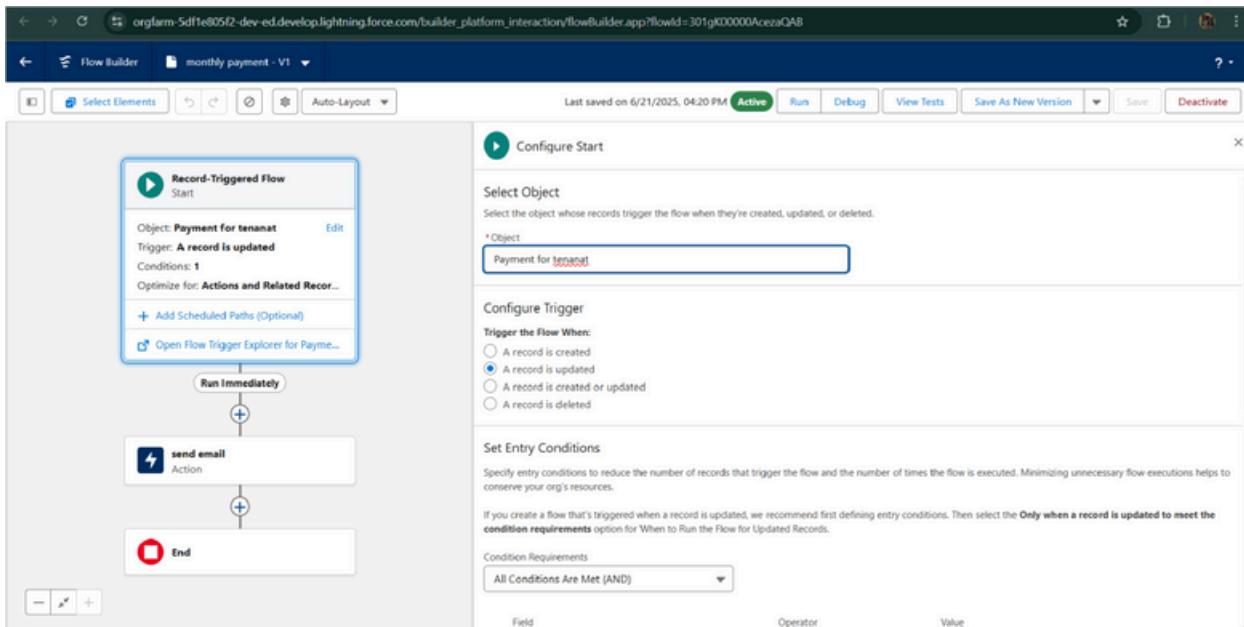
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

● FLOWS





- Schedule class:
Create an Apex Class

```

1 * global class MonthlyEmailScheduler implements Schedulable {
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 sendMonthlyEmail
17
18     List<Tenant__c> tenants = [SELECT
19
20         for (Tenant__c tenant : tenants)
21             String recipientEmail = tenant.Email__c;
  
```

Entity Type: Classes
Name: MonthlyEmailScheduler Namespace:
Related: Email, CustomField..., Tenant__c, Tenant__c

```

1  global class MonthlyEmailScheduler implements schedulable {
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     public static void sendMonthlyEmails() {
16
17         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19         for (Tenant__c tenant : tenants) {
20
21             String recipientEmail = tenant.Email__c;
22
23             String emailSubject = 'Reminder: monthly rent payment due';
24
25             String emailContent = 'Dear Sir/Madam, 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.' ;
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailBody(emailContent);
33
34             messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
35         }
36     }
37 }
38

```

Schedule Apex class

The screenshot shows the Salesforce Setup interface with the search bar set to "apex". The left sidebar is expanded to show categories like Email, Custom Code, and Environments. Under "Apex Classes", the "MonthlyEmailScheduler" class is selected. The main pane displays the Apex Class Detail for "MonthlyEmailScheduler". The "Name" field is "MonthlyEmailScheduler", and the "Status" is "Active". The "Created By" field shows "Sonoya Team" and the "Last Modified By" field also shows "Sonoya Team". Below the details, there are tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags". The "Class Body" tab is active, showing the Apex code for the class.

Apex Class Detail	
Name	MonthlyEmailScheduler
Namespace Prefix	
Created By	Sonoya Team
Last Modified By	Sonoya Team
Status	Active
Code Coverage	0% (0/15)

Class Body

```

1  global class MonthlyEmailScheduler implements Schedulable {
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     public static void sendMonthlyEmails() {
16
17         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19         for (Tenant__c tenant : tenants) {
20
21             String recipientEmail = tenant.Email__c;
22
23             String emailSubject = 'Reminder: monthly rent payment due';
24
25             String emailContent = 'Dear Sir/Madam, 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.' ;
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailBody(emailContent);
33
34             messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
35         }
36     }
37 }
38

```

The screenshot shows a Salesforce Lightning page for a 'Tenant' record. The record details are as follows:

- Tenant Name:** Aswini
- Email:** aswinimaraadi15@gmail.com
- Phone:** (905) 223-5567
- Status:** Leaving
- Property:** Imran

The record was created by Sowmya Team on 6/26/2025, 6:05 AM, and last modified by Sowmya Team on 6/26/2025, 11:06 PM.

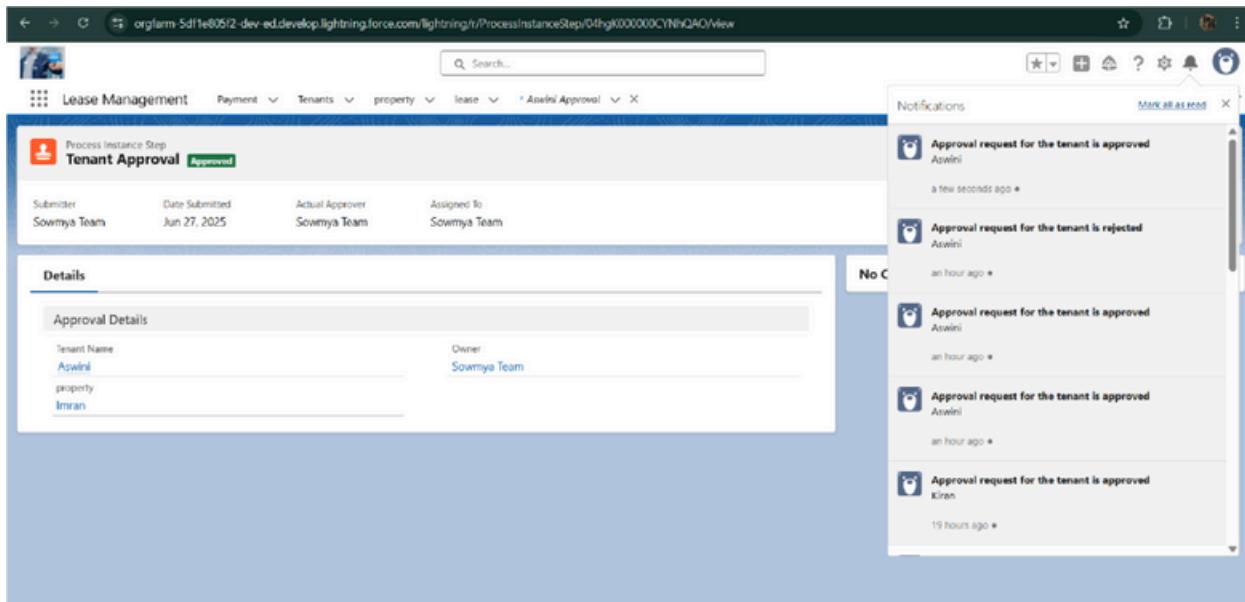
In the top right corner, there is a context menu with options like New Case, New Lead, Delete, Close, Change Owner, Printable View, Submit for Approval, and Edit Labels.

The screenshot shows the same Salesforce Lightning page for the 'Tenant' record for 'Aswini'. A green success message at the top indicates that the record has been submitted for approval.

The record details remain the same as in the previous screenshot.

In the top right corner, the context menu now includes 'Submit for Approval' and 'Edit Labels' under the 'Submit for Approval' section.

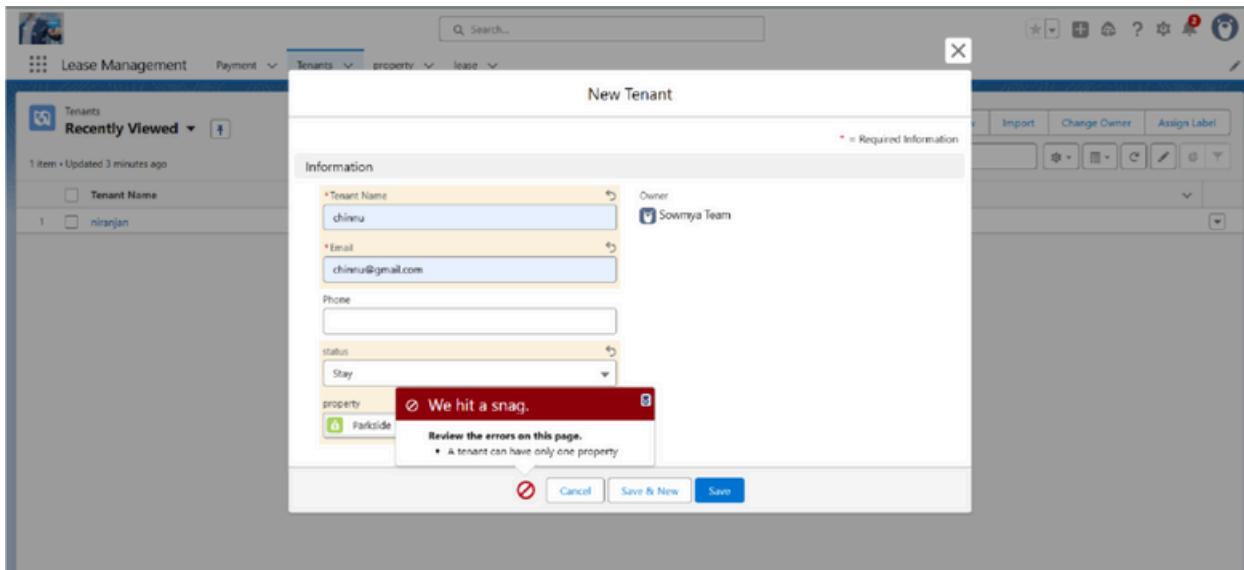
The system status bar at the bottom shows the date as 27-06-2025 and the time as 12:46.



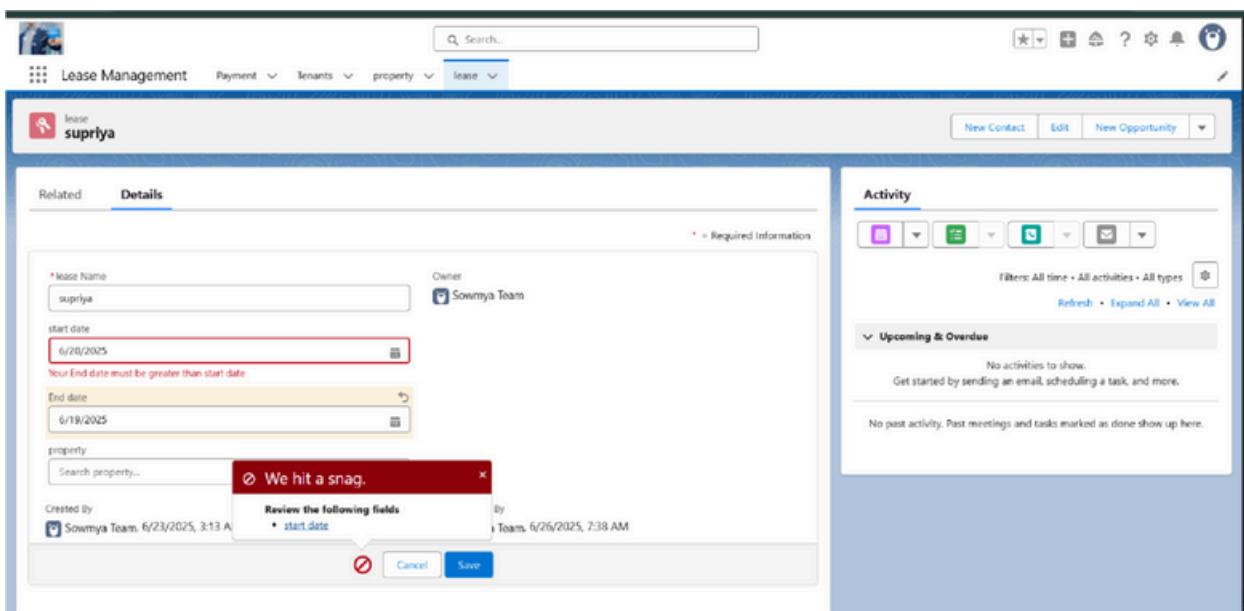
FUNCTIONAL AND PERFORMANCE TESTING

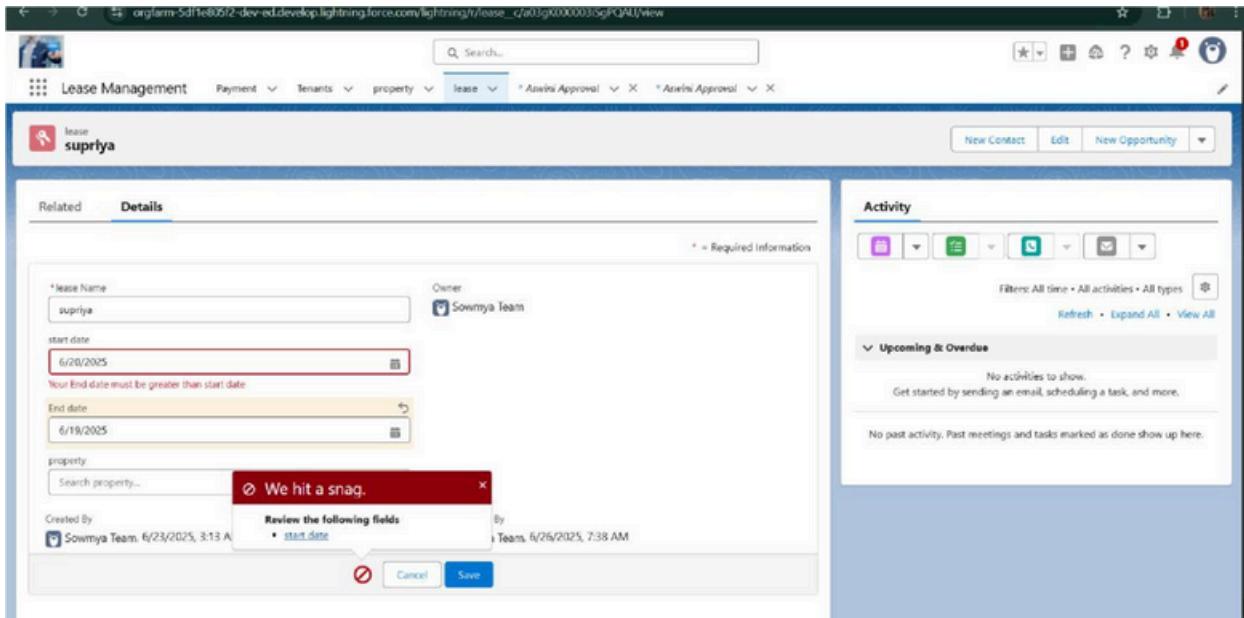
Performance Testing

- Trigger validation by entering duplicate tenant-property records

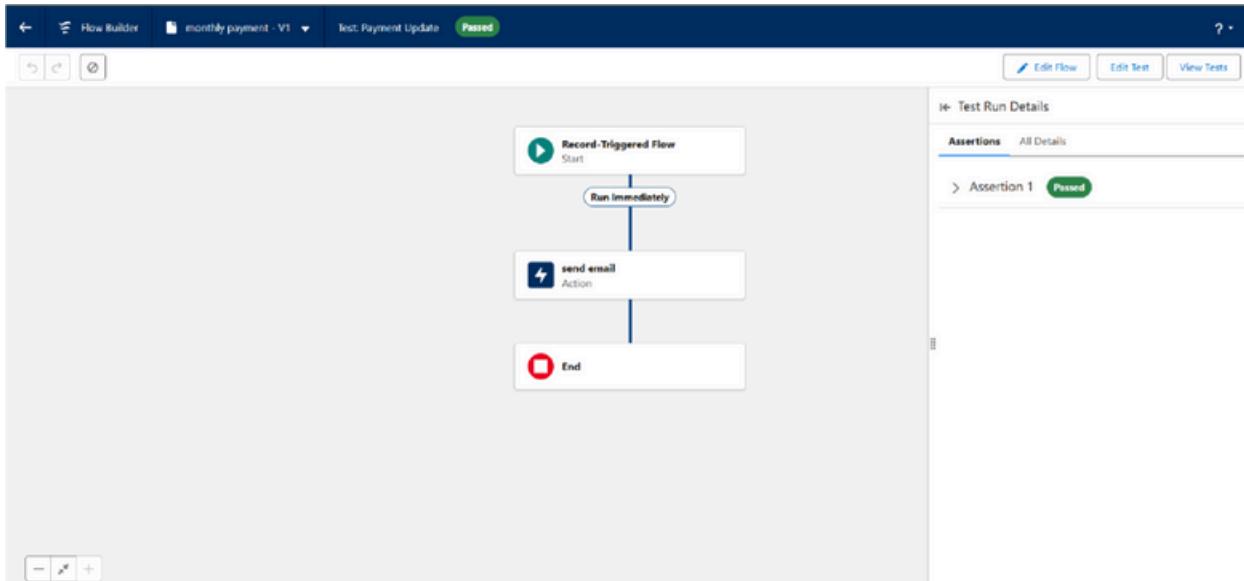


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Tenant: niranjan

Details

* = Required Information

* Tenant Name	niranjan	Owner	Sowmya Team
* Email	niranjan1506@gmail.com		
Phone			
Status	Stay		
Property	Parkside Lofts		
Created By	Sowmya Team	Last Modified By	Sowmya Team
Created: 6/23/2025, 2:33 AM Last Modified: 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 reps through the sales process.
Jun 20, 2025, 1:28 PM *

Tenant: niranjan

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	New
Jark	
Rahul	

View All

New Contact **Edit** **New Opportunity**

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 displays a table for 'Custom Object Tabs' with four entries:

Action	Label	Tab Style	Description
Edit Del	Lease	Keys	
Edit Del	Payment	Credit card	
Edit Del	Receipts	Bank	
Edit Del	Records	Map	

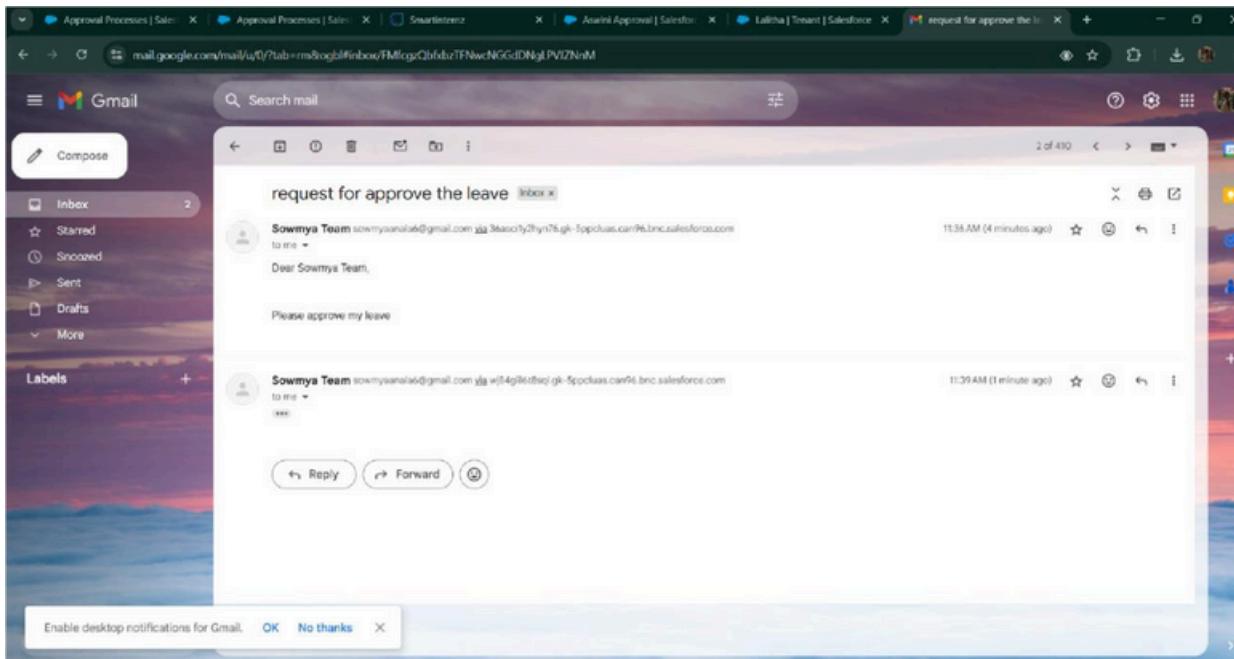
Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both currently empty.

- Email alerts

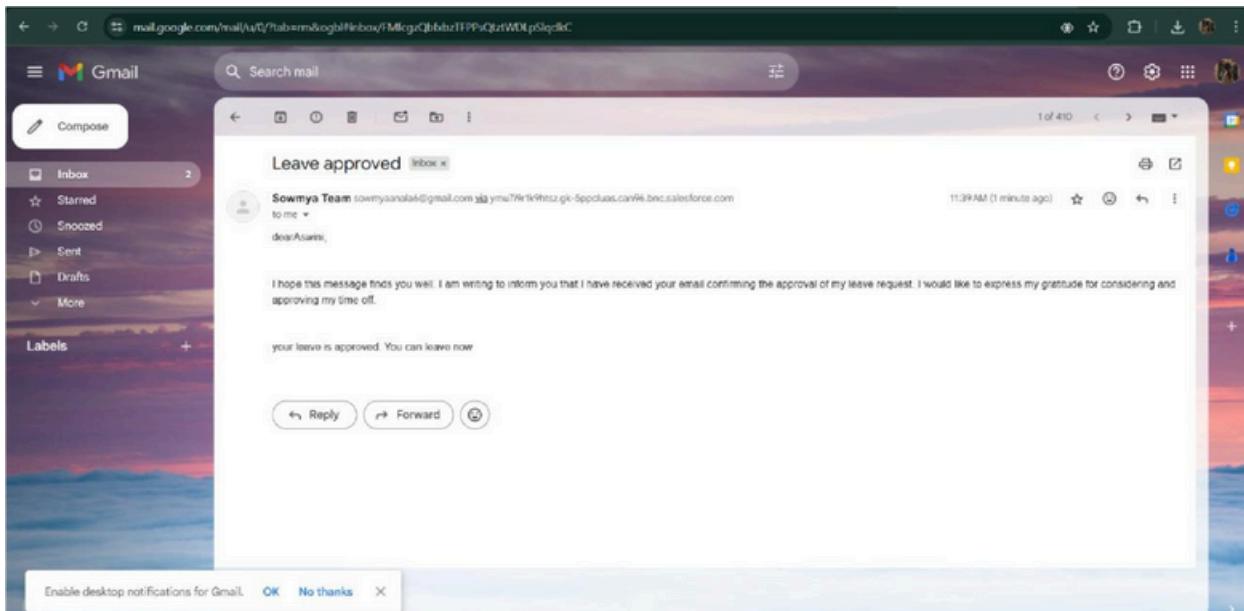
The screenshot shows the 'Approval History' table within the 'Lease Management' application. The table has the following columns:

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/23/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/23/2025, 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

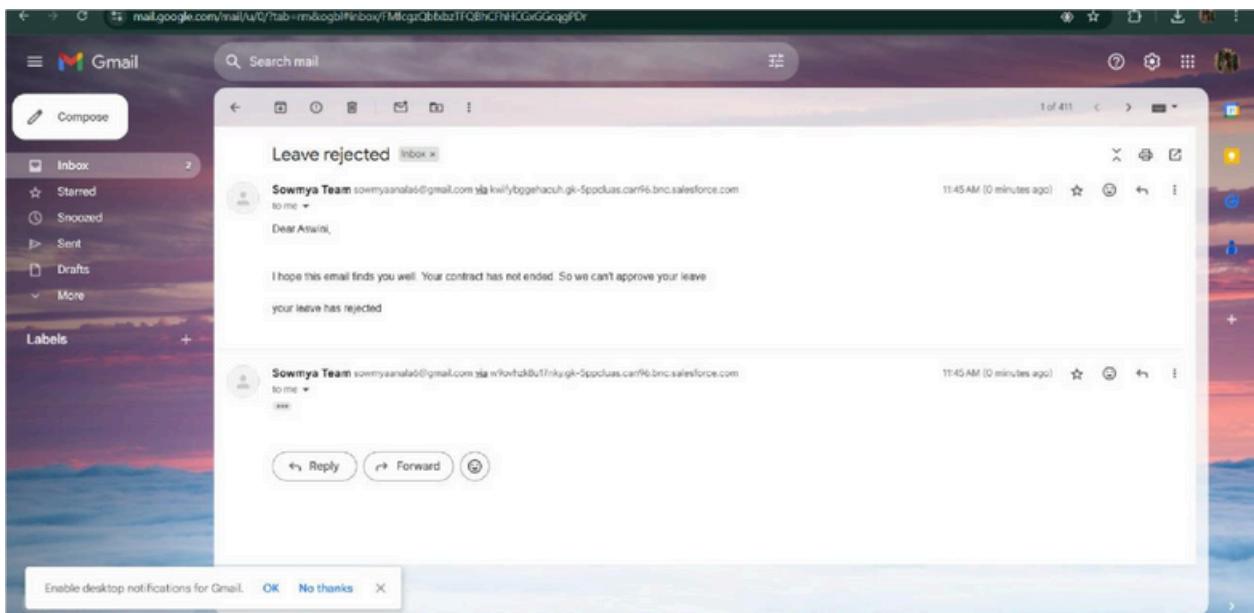
- Request for approve the leave



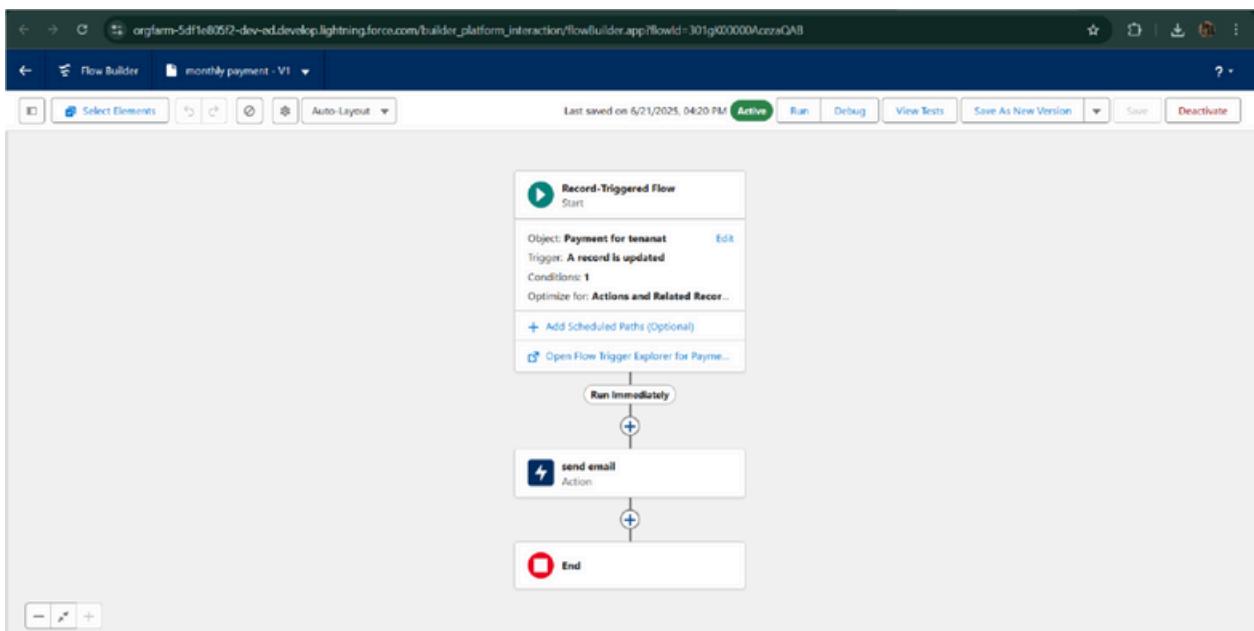
- Leave approved



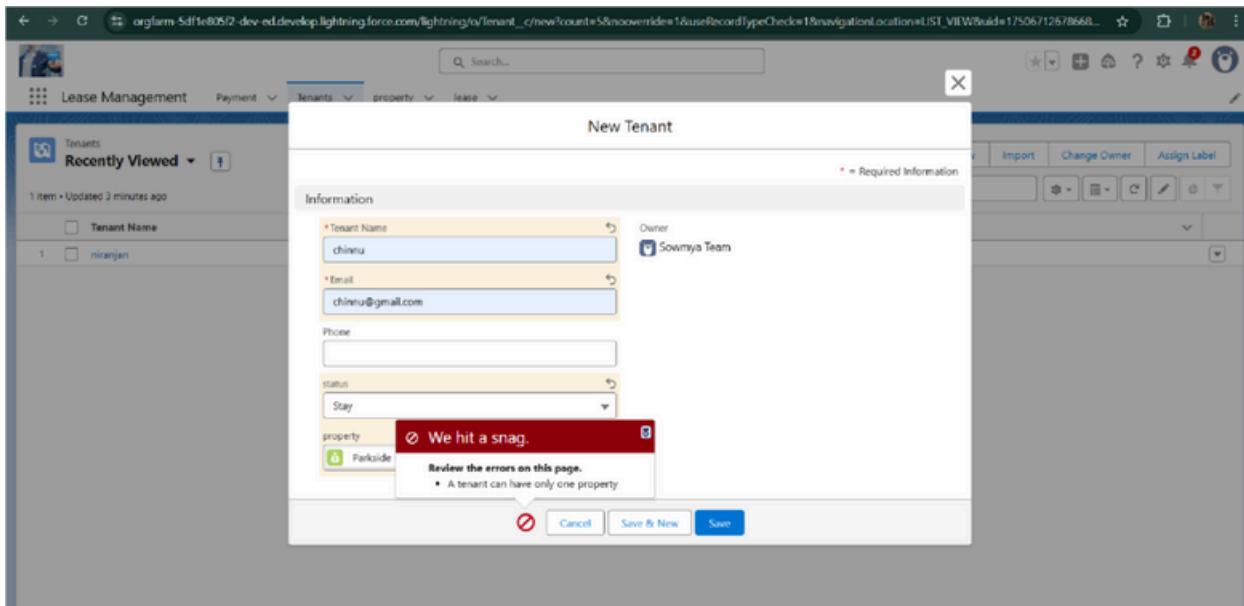
- Leave rejected



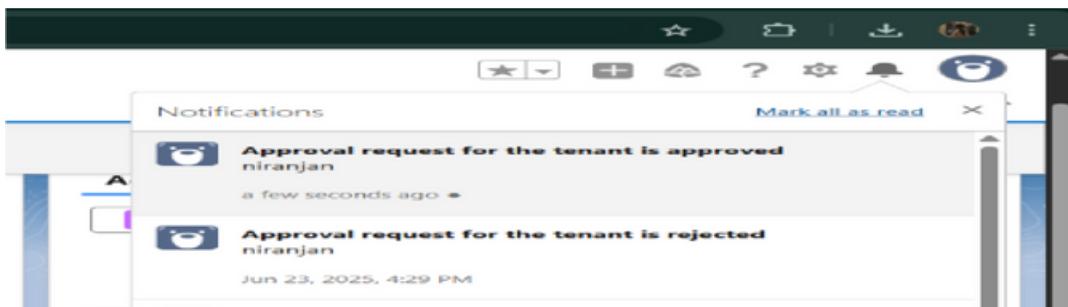
- Flow runs



- 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<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)) { newTenant.addError('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});  
    }  
}  
}
```