

LEASE MANAGEMENT

College Name: Nandha Arts And Science College

College Code: bru4j

TEAM ID: NM2025TMID23195

TEAM MEMBERS:

Team LeaderName: VIKRAMAN D

Email: vikramrajchetty30@gmail.com

Team Member: THIRUMALAI V

Email: thirumalaivasudhavan@gmail.com

Team Member: VIMALRAJ T

Email: sktvimalrajtamilselvan2005@gmail.com

Team Member: VINISHKUMAR G

Email: vinish2006kumar@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.



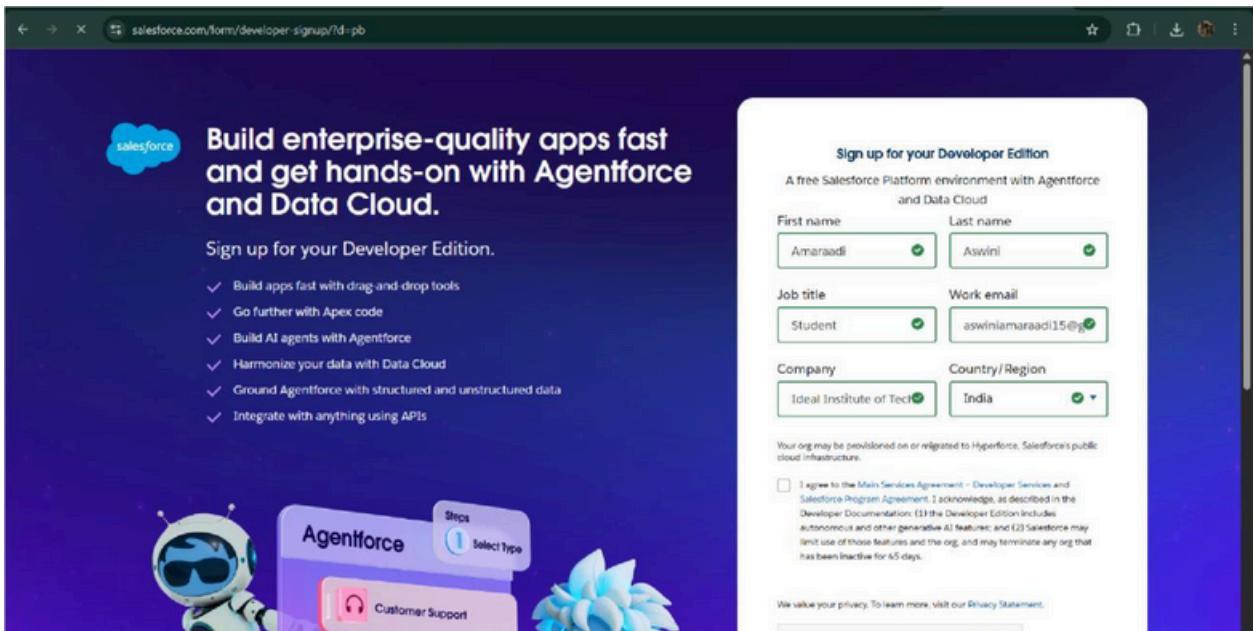
1.2 Purpose

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. The left sidebar lists various object settings like 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 Accessible Rules. The main area displays the details for the 'property' object. The 'Details' tab is selected, showing the API Name (property__c), Custom checkbox checked, Singular Label (property), and Plural Label (property). On the right, there are sections for Description, Enable Reports (checkbox checked), Track Activities (checkbox checked), Track Field History (checkbox checked), Deployment Status (Deployed), Help Settings, and Standard salesforce.com Help Window. At the bottom right are Edit and Delete buttons.

orgfarm-Sdf1e805f2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various configuration options under 'Details'. The main pane displays the 'Details' tab for the 'Tenant' object. The API Name is set to 'Tenant__c'. The singular label is 'Tenant' and the plural label is 'Tenants'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window').

orgfarm-Sdf1e805f2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various configuration options under 'Details'. The main pane displays the 'Details' tab for the 'lease' object. The API Name is set to 'lease__c'. The singular label is 'lease' and the plural label is 'lease'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window').

Object Manager

Payment for tenant

Details

Description

API Name: Payment_for_tenant__c
Custom
Singular Label: Payment for tenant
Plural Label: Payments

Buttons, Links, and Actions

Available Reports

- ✓ Trunk Activities
- ✓ Trunk Field History
- ✓ Deployment Status
- ✓ Help Settings
- Standard Salesforce.com Help Window

- Configured fields and relationships

Fields & Relationships

| 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

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

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: Placeholder image of a person in a suit.

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

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

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

App Settings

Navigation Items

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

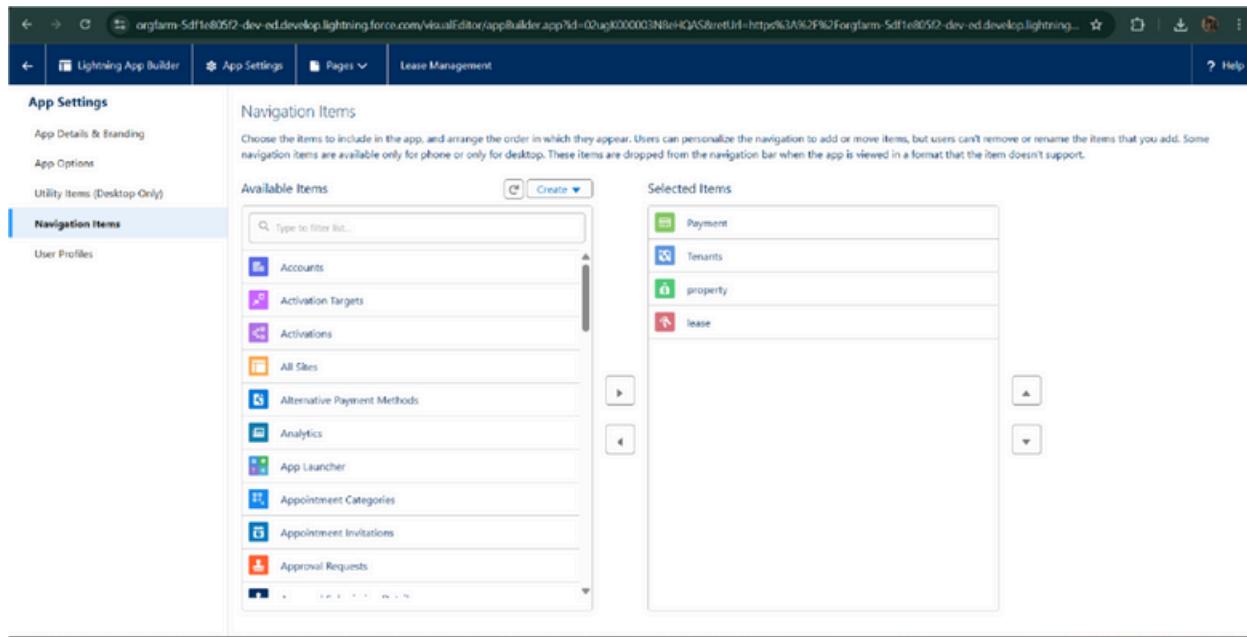
Available Items

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

User Profiles

Choose the user profiles that can access this app.

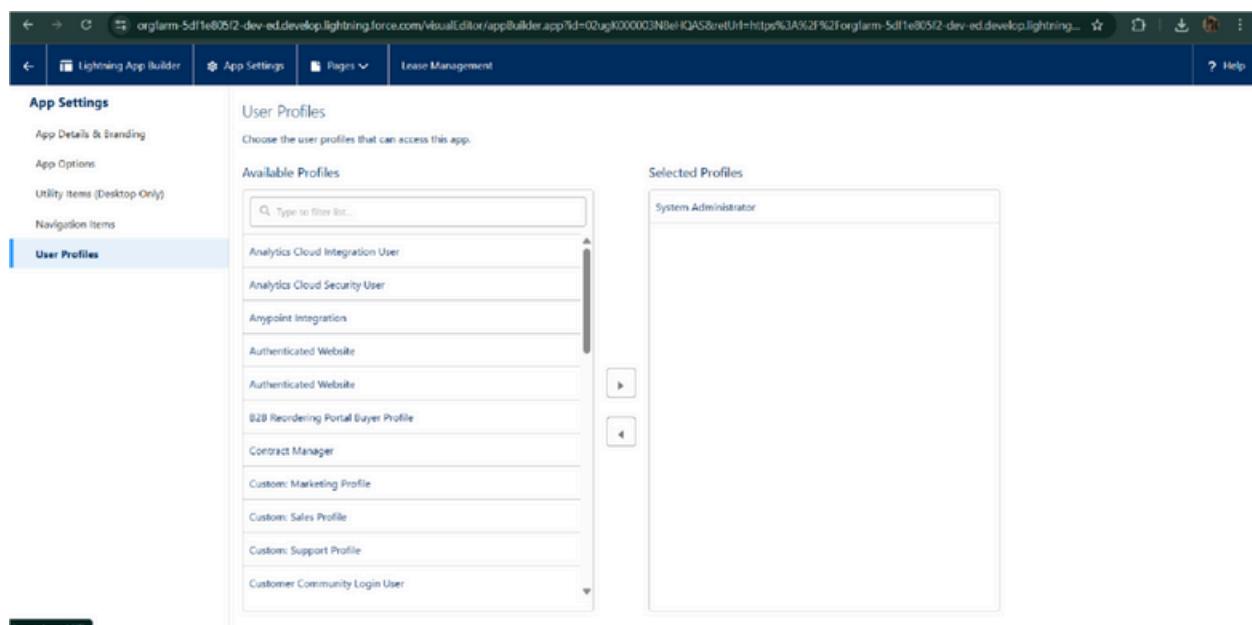
Available Profiles

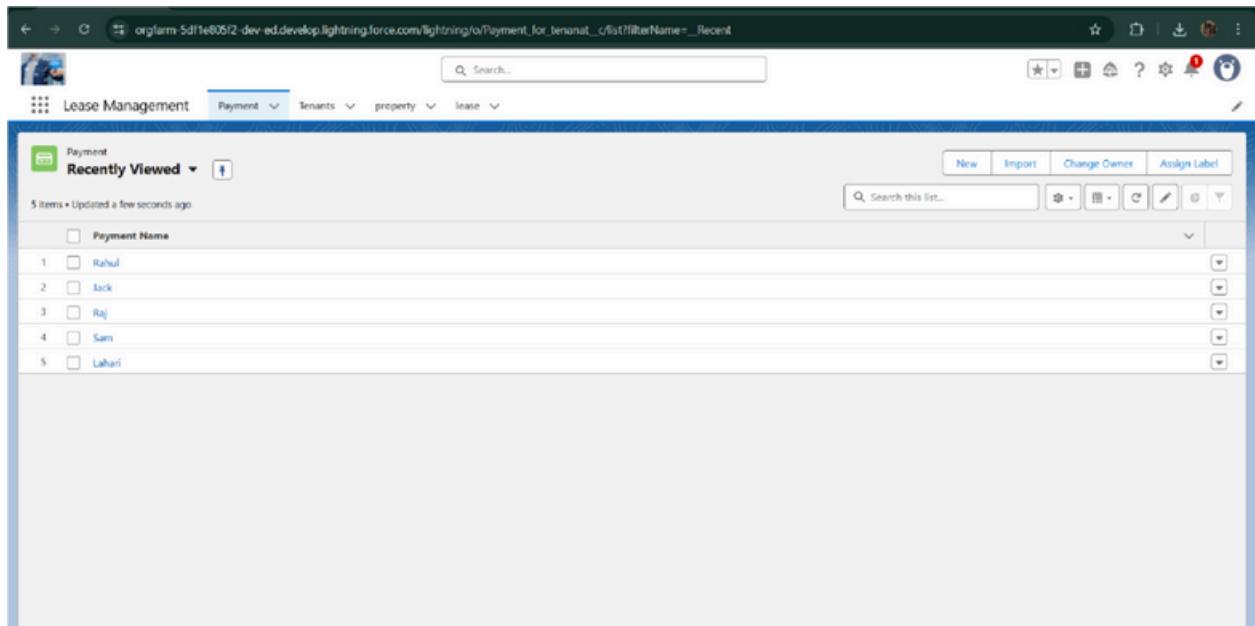
Type to filter list...

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

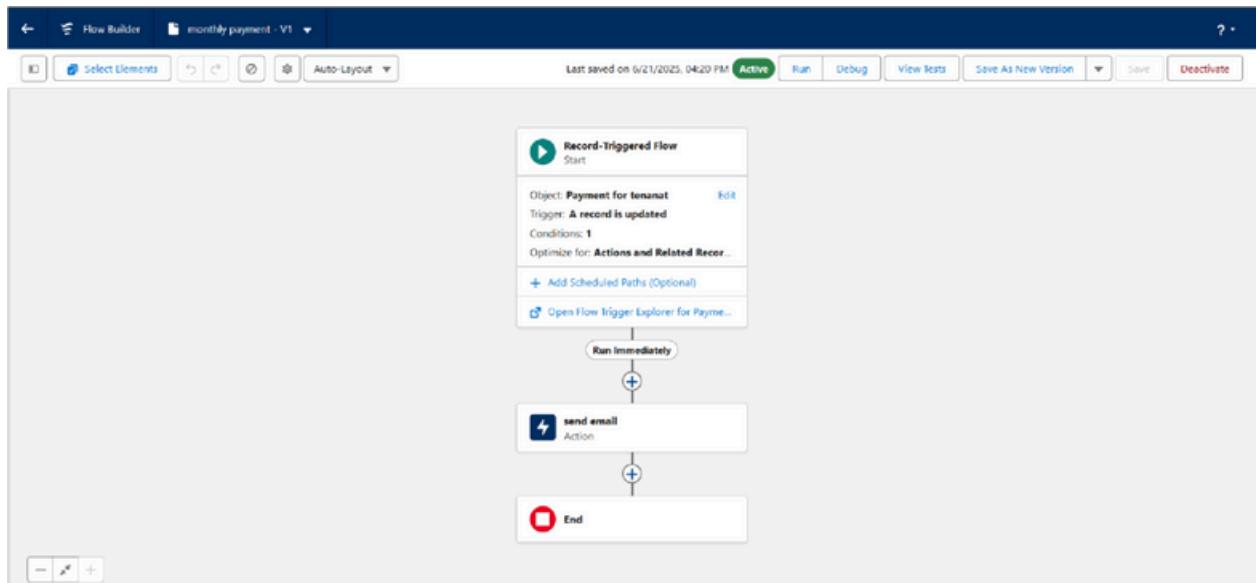
Selected Profiles

- System Administrator





- Implemented Flows for monthly rent and payment success

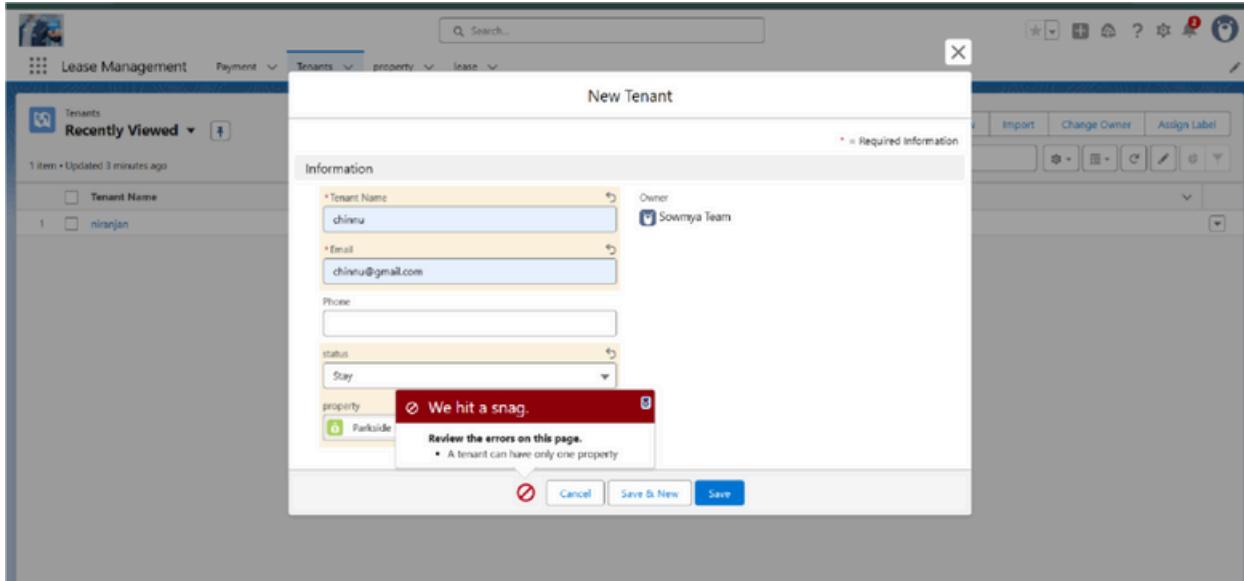


- To create a validation rule to a Lease Object

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 'Error Message' field is empty.

The screenshot shows the 'Validation Rule Detail' screen for the 'lease' object. The validation rule is named '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 'Active' status is checked. The 'Created By' field shows 'Soamva Team' and the 'Modified By' field shows 'Soamva Team'. Both were modified on '6/19/2025, 5:37 AM'.

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using 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 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     }
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 for 'Leave approved'. The main content area displays the 'Leave approved' email template detail page. The template name is 'Leave approved', and it is described as a 'Text Email Template' under 'Leave approved'. The preview pane shows the email 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.

short leave is imminent. Your kind leave note.

```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item for 'tenant leaving'. The main content area displays the 'tenant leaving' email template detail page. The template name is 'tenant leaving', and it is described as a 'Text Email Template' under 'tenant leaving'. The preview pane shows the email 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 'Classic Email Templates' page open. The template name is 'Leave rejected'. The 'Email Template Detail' section includes fields for Email Templates from Salesforce (Unified Public: Classic Email Templates), Email Template Name (Leave rejected), Template Unique Name (Leave_rejected), Encoding (Unicode (UTF-8)), Author (S00mya_Itam [Change]), Description (None), Created By (S00mya_Itam), Last Used Date (6/20/2025, 1:11 AM), Available For Use (checked), Times Used (0), and Modified By (S00mya_Itam). 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 'Classic Email Templates' page open. The template name is 'Tenant Email'. The 'Email Template Detail' section includes fields for Email Templates from Salesforce (Unified Public: Classic Email Templates), Email Template Name (Tenant_Email), Template Unique Name (Tenant_Email), Encoding (Unicode (UTF-8)), Author (S00mya_Itam [Change]), Description (None), Created By (S00mya_Itam), Last Used Date (6/20/2025, 1:12 AM), Available For Use (checked), Times Used (0), and Modified By (S00mya_Itam). 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.'

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. Under the 'Email' section, 'Classic Email Templates' is selected. A specific template, 'tenant payment', is highlighted. The 'Email Template Detail' section shows the following information:

- Email Templates from Salesforce
- Email Template Name: tenant payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Soma Team (Change)
- Created By: Soma Team
- Last Used Date: 6/20/2025, 1:13 AM
- Available For Use: checked
- Times Used: 1
- Modified By: Soma Team
- Modified Date: 6/20/2025, 1:13 AM

The 'Email Template' section below contains the template's content:

```

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:

The screenshot shows the Salesforce Setup interface with the search bar set to 'approval'. Under the 'Data' section, 'Approval Processes' is selected. A specific process, 'Tenant: TenantApproval', is highlighted. The 'Process Definition Detail' section shows the following information:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description:
- Entry Criteria: tenant__r.status equals Stay
- Record Editability: Administrator ONLY
- Action: Active (checked)
- Next Automated Approver Determined By:
- Allow Submitters to Recall Approval Requests: unchecked
- Created By: Soma Team
- Created Date: 6/23/2025, 3:41 AM
- Modified By: Soma Team
- Modified Date: 6/26/2025, 11:57 PM

The 'Initial Submission Actions' section shows:

| Action Type | Description |
|-------------|-----------------------------------|
| Record Lock | Lock the record from being edited |

The 'Approval Steps' section shows:

| Action | Step Number | Name | Description | Criteria | Assigned Approver | Reject Behavior |
|--------------|-------------|--------|-------------|----------|-------------------|-----------------|
| Show Actions | 1 | Step 1 | | | User:Soma Team | Final Rejection |

For Check for Vacant:

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

- Page Title:** Approval Processes
- Process Name:** check for vacant
- Unique Name:** Check_for_vacant
- Description:** Tenant: check for vacant
- Entry Criteria:** Tenant__c.status equals Leaving
- Direct Eligibility:** Administrator ONLY
- Next Automated Approver Determined By:** Allowing Submitters to Recall Approval Requests
- Approval Assignment Email Template:** Leave 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: SOWMYA.BABU@SYNTELINITY.COM
- 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 Name:** testHandler
- Code:**

```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```
- Modal Dialog (Entity Type Selection):**

| Entity Type | Edition | Related |
|-----------------|---------|------------------|
| Classes | None | Namespace |
| Triggers | test | Name |
| Pages | | Object |
| Page Components | | Static Resources |
| Objects | | Packages |

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

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

testHandler.apc [MonthlyBatchScheduler.apc]

Code Coverage: None • API Version: 64 • 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 }
```

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/ApexCSIPage

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

testHandler.apc [MonthlyBatchScheduler.apc]

Code Coverage: None • API Version: 64 • 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(existingTenant.Id);
10
11         }
12
13         for (Tenant__c newTenant : newList) {
14
15             if (newTenant.Property__c != null) {
16
17                 newTenant.addError('A');
18
19             }
20
21         }
22     }
23 }
```

Entity Type Entity Name Namespace Related

| Entity Type | Entity | Related |
|------------------|-----------------------|-------------|
| Entity Set | testHandler | test |
| Classes | | ApexTrigger |
| Triggers | MonthlyBatchScheduler | property |
| Pages | | Customer |
| Page Components | | References |
| Objects | | Tenant__c |
| Static Resources | | Subject |
| Package | | References |

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-Sdf1t0805f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

testHandler.apc [testHandler.apc] HostedDynamicScheduler.apc []

Code Coverage Name: API Version: 54 Go To

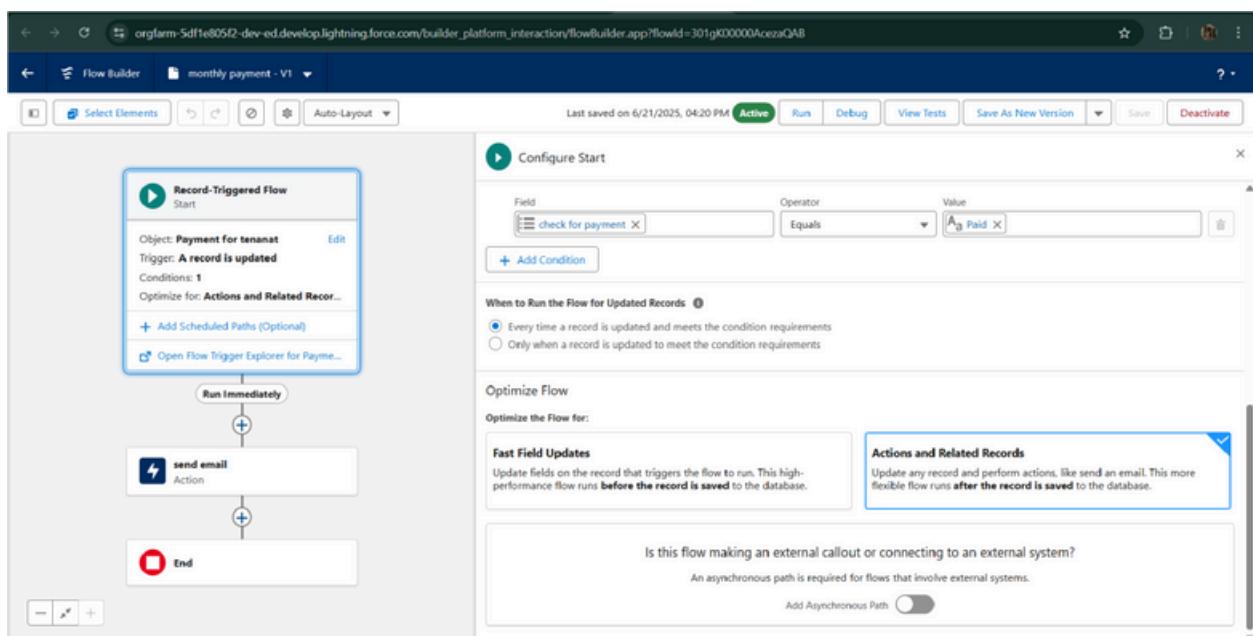
```

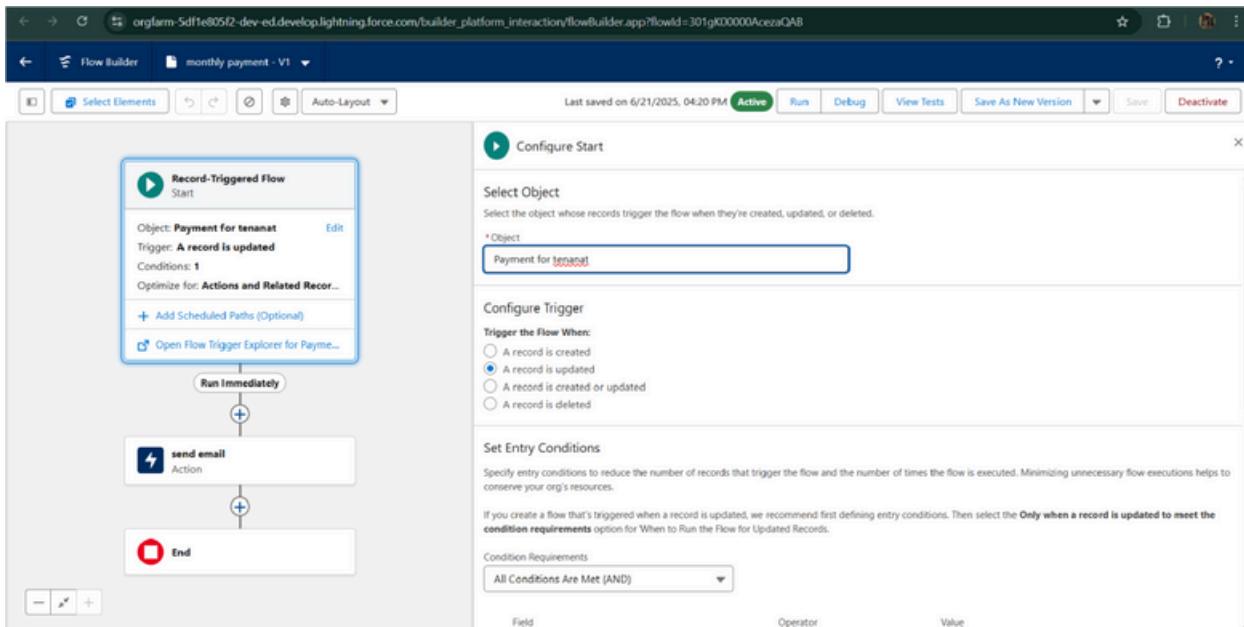
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
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     }
26
27 }
```

Logs Tests Checkpoints Query Editor View State 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
22             String recipientEmail = tenant.Email__c;
```

Open

| Entity Type | Entities | Related |
|------------------|-----------------------|-------------------------------|
| Entity Type | Name Namespace | Name Parent Direction |
| Classes | testHandler | Email CustomField References |
| Triggers | MonthlyEmailScheduler | < Tenant__c Object References |
| Pages | | < Tenant__c Object References |
| Page Components | | |
| Objects | | |
| Static Resources | | |
| Packages | | |

```
Developer Console - Google Chrome
http://orgfarm-5df1e05f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage
50% Reset

global class monthlyRentalScheduler implements schedulable {
    global void execute(Database.BatchableContext BC) {
        Integer currentDay = Date.Today().day();
        if (currentDay == 1) {
            sendEmail();
        }
    }

    public static void sendEmail() {
        List<tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
        for (tenant__c tenant : tenants) {
            String recipientEmail = tenant.Email__c;
            String rentalMessage = "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.";
            String emailSubject = "Rental Reminders";
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToRecipients(String.valueOf(recipientEmail));
            email.setSubject(emailSubject);
            email.setPlainTextEmailBody(emailContent);
            email.setHtmlEmailBody(emailContent);
            Messaging.SingleEmailMessage[] emails = {email};
        }
    }
}
```

Schedule Apex class

Lease Management

Search...

New Contact | Edit | New Opportunity

New Case | New Lead | Delete | Close | Change Owner | Refresh | Printable View | Submit for Approval | Edit Labels

Related **Details**

* = Required Information

Tenant Name: Aswini

Email: aswinimaraadi15@gmail.com

Phone: (905) 223-5567

Status: Leaving

Property: Imran

Created By: Soomya Team, 6/26/2025, 6:05 AM

Last Modified By: Soomya Team, 6/26/2025, 11:06 PM

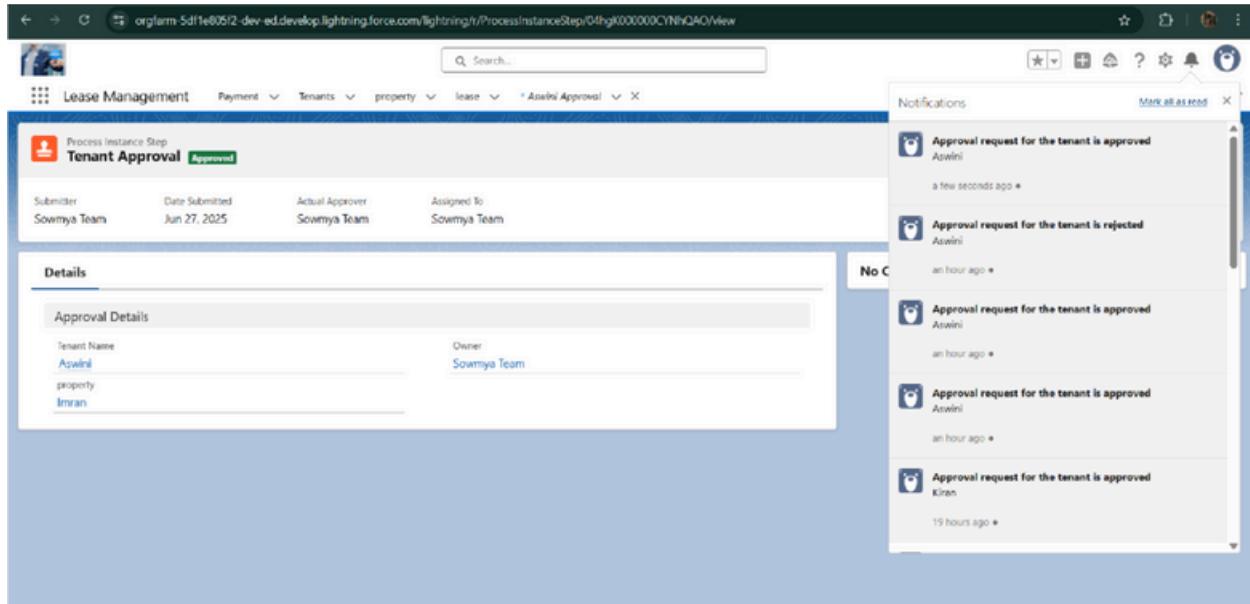
Activity

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.

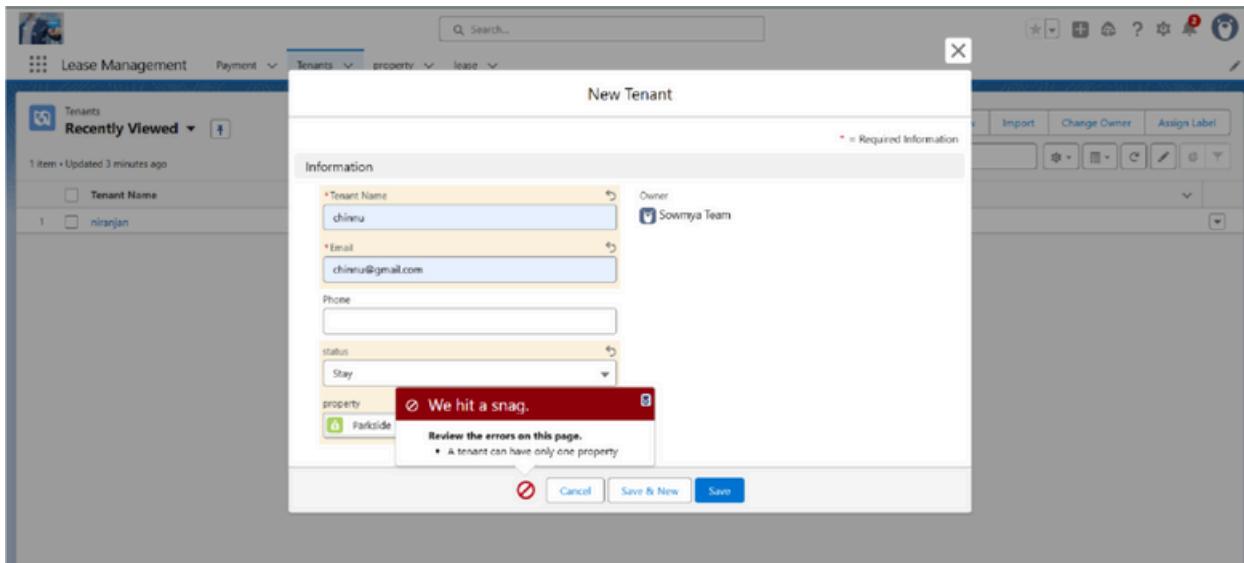
The screenshot shows the Salesforce Lightning interface for the 'Lease Management' application. The top navigation bar includes tabs for 'Payment', 'Tenants', 'property', and 'lease'. A search bar is located at the top right. The main content area displays a tenant record for 'Aswini'. The 'Details' tab is selected, showing fields for 'Tenant Name' (Aswini), 'Email' (aswiniamaraadi15@gmail.com), 'Phone' ((905) 223-5567), 'status' (Leaving), and 'property' (Imran). The 'Owner' field is listed as 'Sowmya Team'. A green success message at the top right states 'Tenant was submitted for approval.' On the right side, there is an 'Activity' sidebar with sections for 'Upcoming & Overdue' (empty) and 'Past activity' (empty). The bottom of the screen shows the Windows taskbar with various pinned icons.



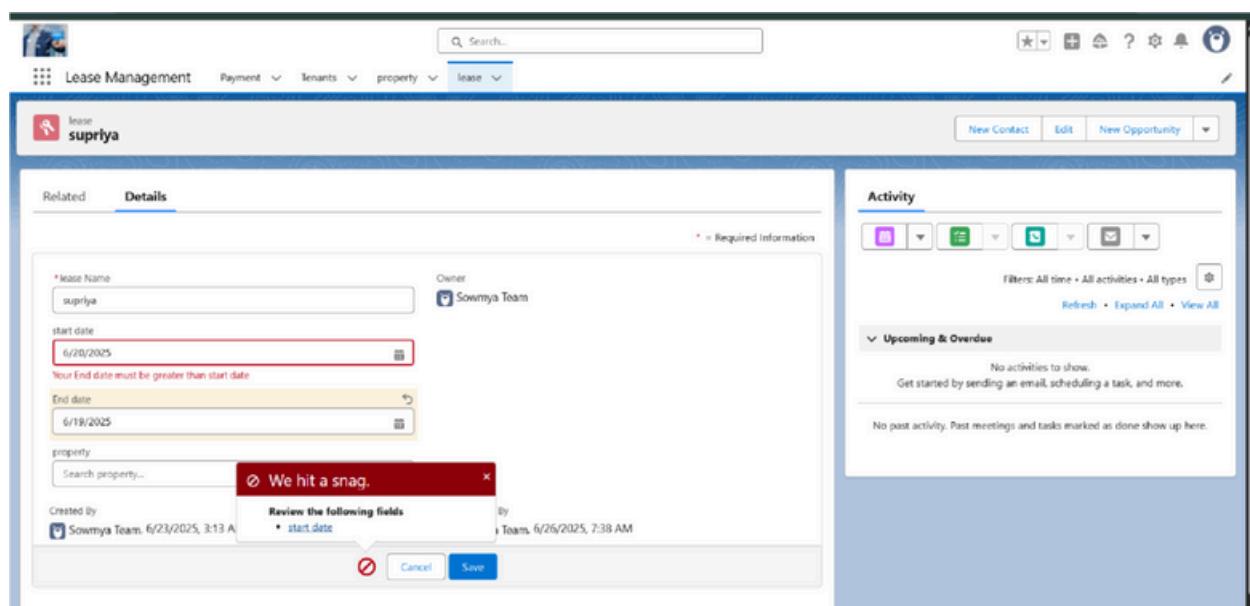
FUNCTIONAL AND PERFORMANCE TESTING

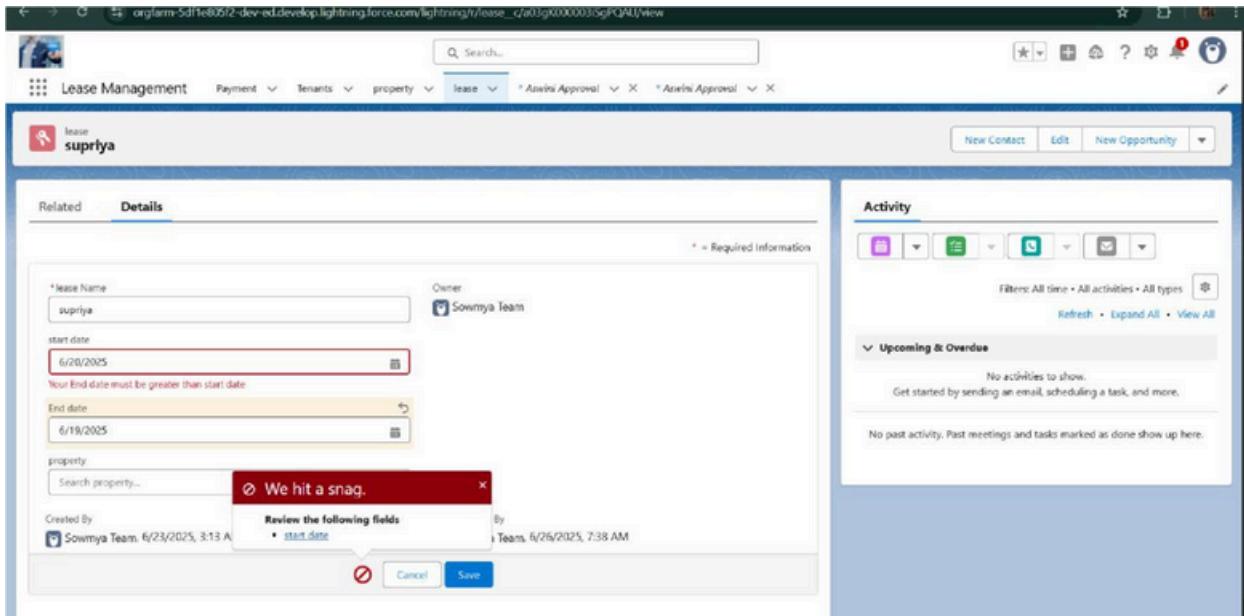
Performance Testing

- Trigger validation by entering duplicate tenant-property records

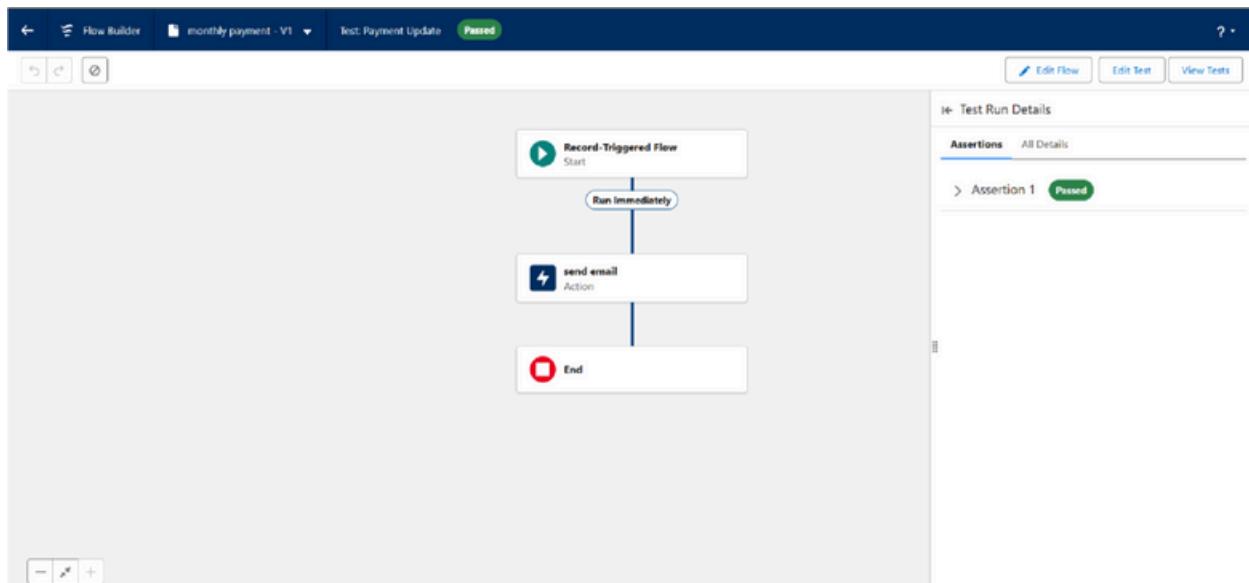


- Validation Rule checking





- 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, 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 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

RESULTS

Output Screenshots

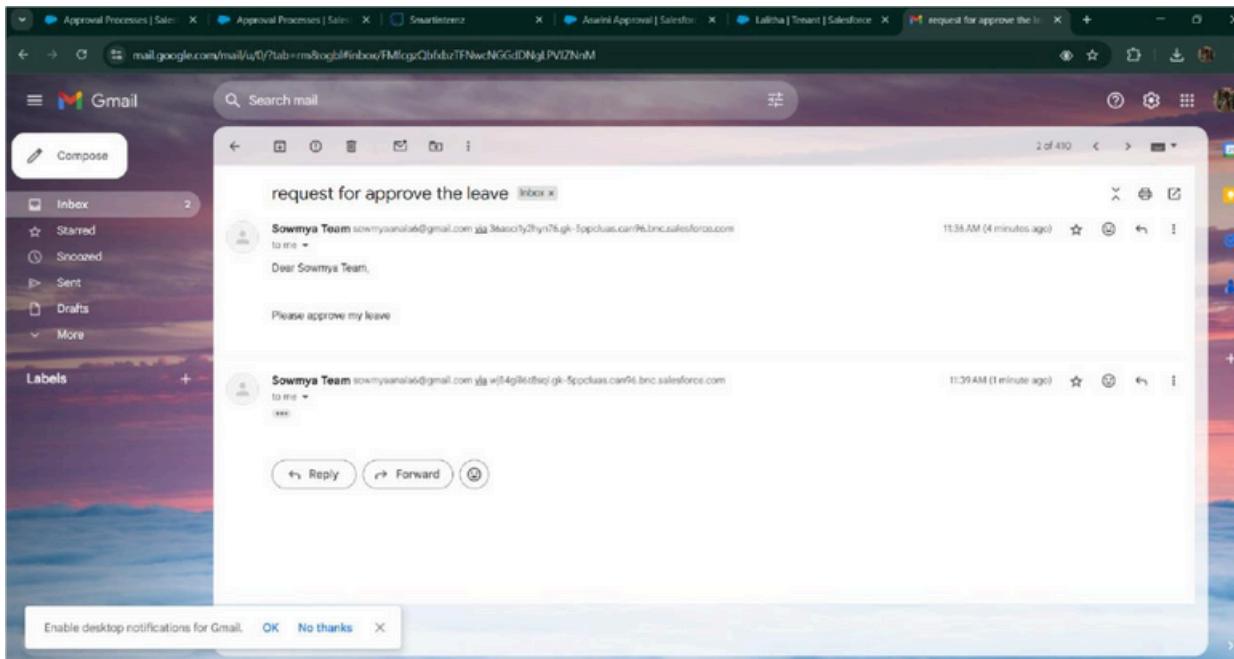
- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It displays three main sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Issue', 'Payment', 'Record', and 'Records', each with a 'Tab Style' icon (e.g., Keys, Credit card, Back, Map). The 'Web Tabs' and 'Visualforce Tabs' sections both indicate 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

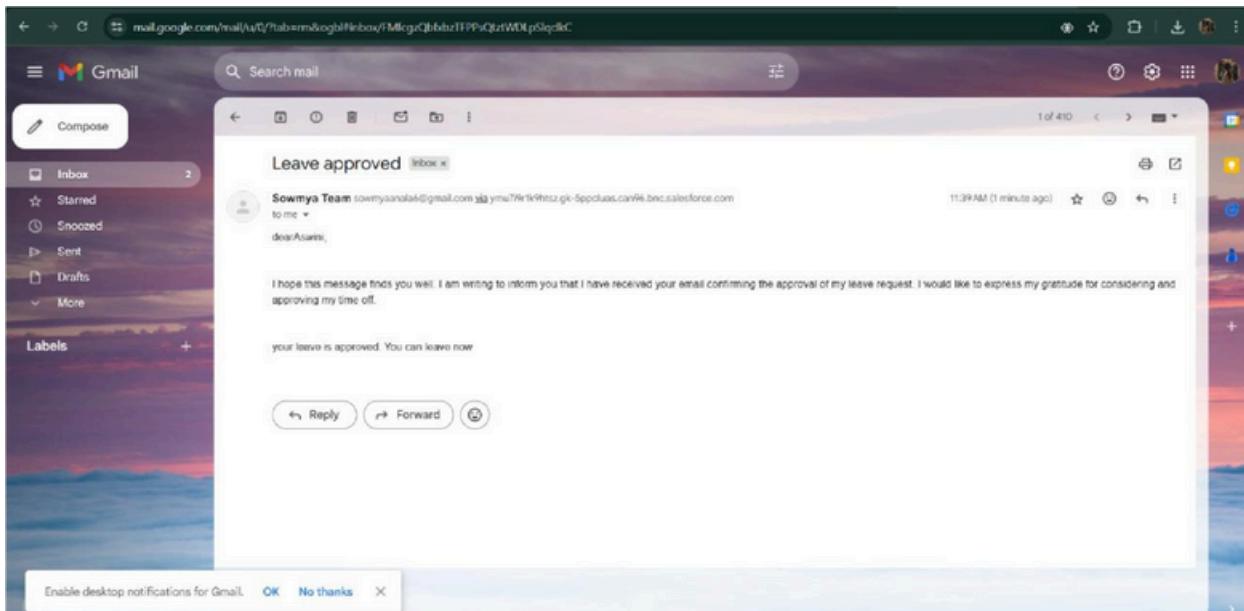
- Email alerts

The screenshot shows the 'Approval History' page within the 'Lease Management' application. The page title is 'Approval History' and it shows 8 items sorted by 'Is Pending' (updated a few seconds ago). The table columns include Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The steps listed are: Step 1 (Approved, Sowmya Team, Sowmya Team, approved), Approval Request Submitted (Submitted, Sowmya Team, Sowmya Team, leaving), Step 1 (Rejected, Sowmya Team, Sowmya Team, Rejected), Approval Request Submitted (Submitted, Sowmya Team, Sowmya Team, Leaving), Step 1 (Approved, Sowmya Team, Sowmya Team, Approved), Approval Request Submitted (Submitted, Sowmya Team, Sowmya Team, leaving), Step 1 (Approved, Sowmya Team, Sowmya Team, Approval Approved), and Approval Request Submitted (Submitted, Sowmya Team, Sowmya Team, Leaving).

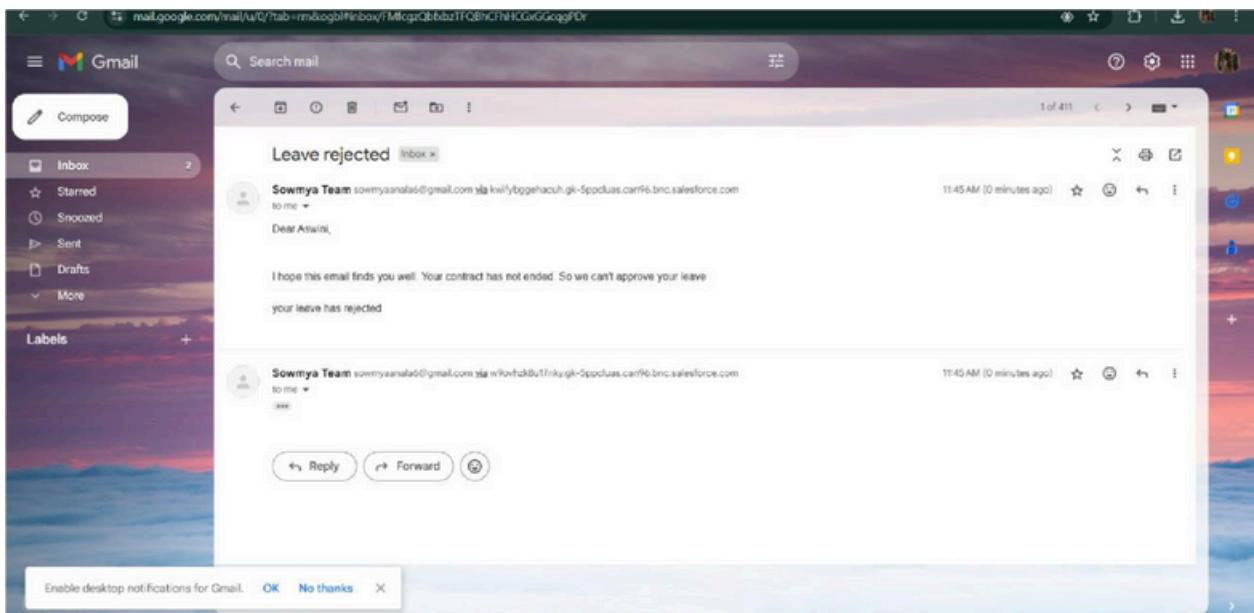
- Request for approve the leave



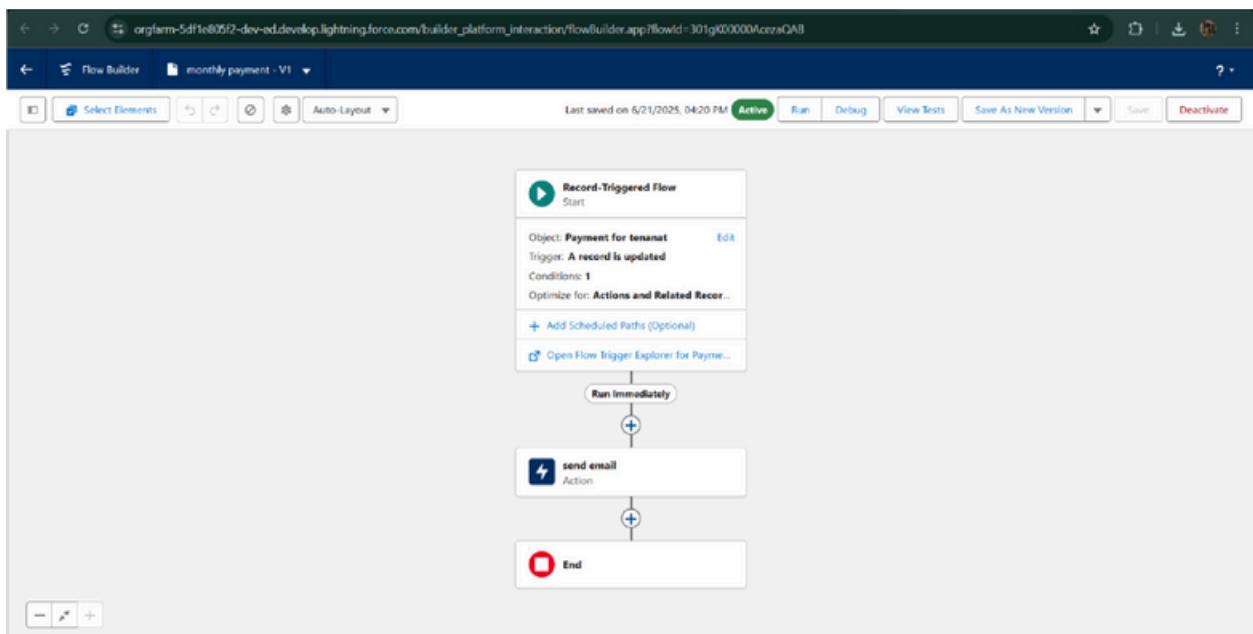
- Leave approved



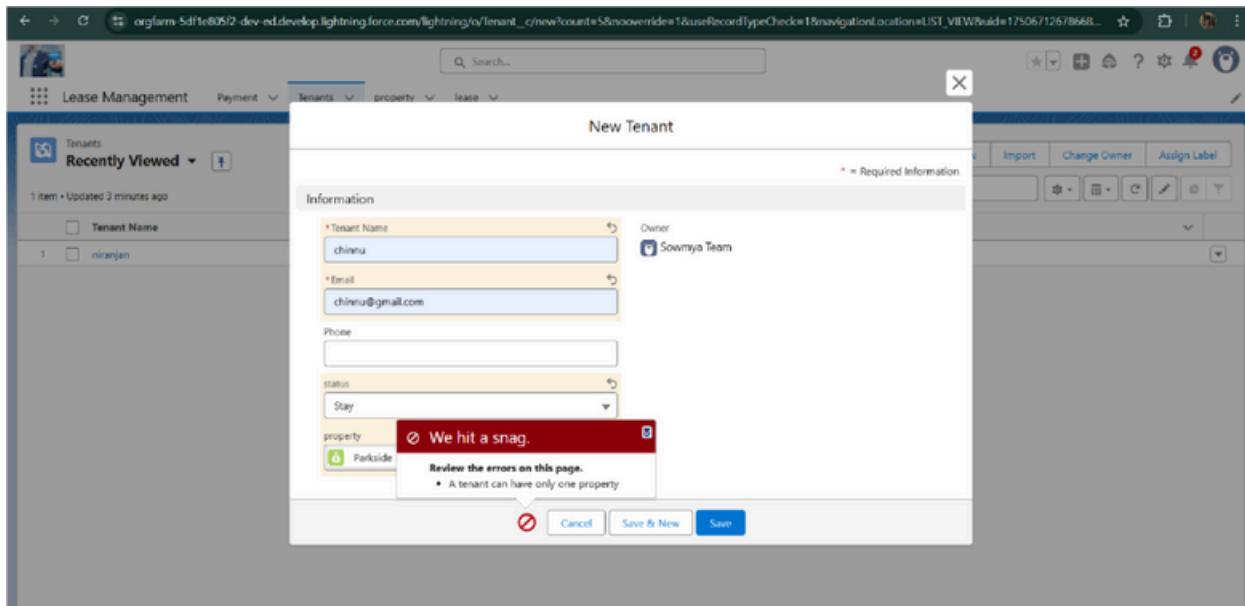
- Leave rejected



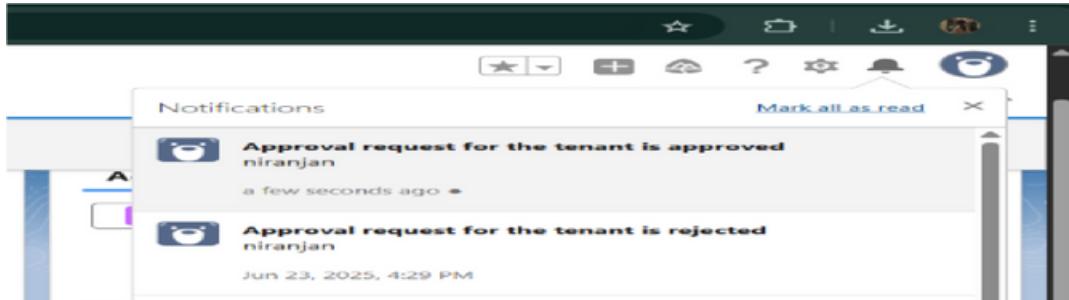
- Flow runs



- Trigger error messages



- Approval process notifications



ADVANTAGES & DISADVANTAGES

CONCLUSION

TheLease Management System successfully streamlines the operations of leasing through a structured, automatedSalesforce 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 teston 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>  
{  
existingPropertyId  
s = 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});  
}  
}
```