

# LEASE MANAGEMENT

COLLEGENAME:KG College of Arts And Science

TEAM ID:NM2025TMID23633

TEAM SIZE:4

## TEAM MEMBERS:

Team Leader Name:Narmathashree T

Email:[2326kb29@kgcas.com](mailto:2326kb29@kgcas.com)

Team Member:Nishanth C

Email:[2326kb30@kgcas.com](mailto:2326kb30@kgcas.com)

Team Member:Nithish S

Email:[2326kb31@kgcas.com](mailto:2326kb31@kgcas.com)

Team Member:Nithya Shree S

Email:[2326kb32@kgcas.com](mailto:2326kb32@kgcas.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.



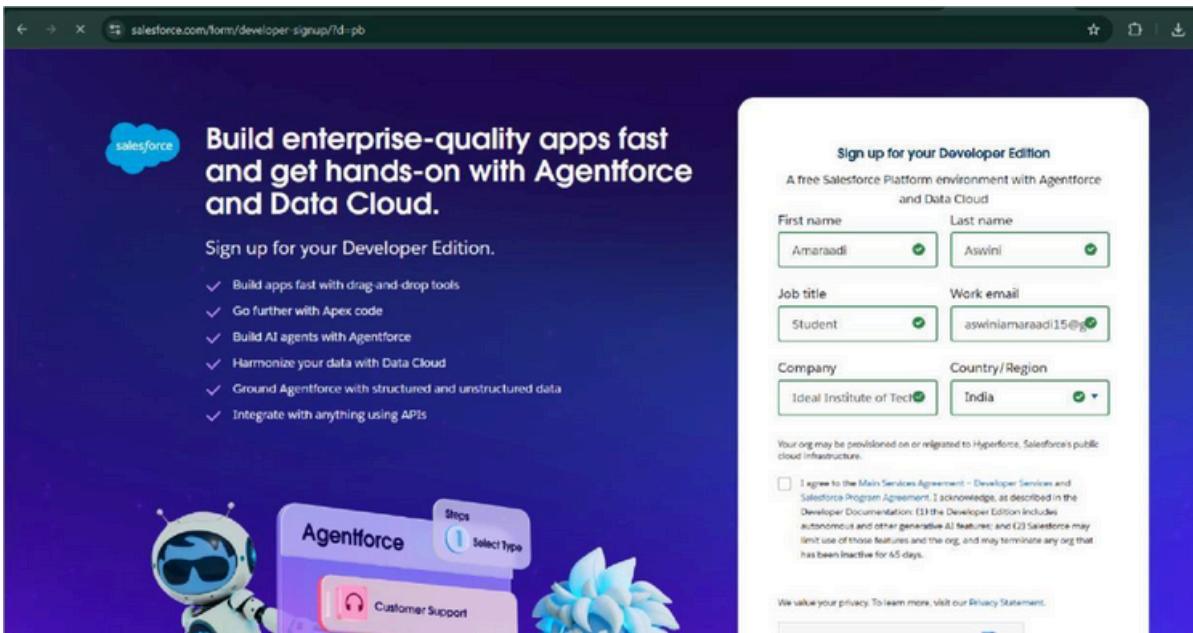
## 1.2 Purpose

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

- Here are 8 key points from the screenshot: Object
- Name: property
- API Name: property\_c Custom
- Object: Yes (✓) Singular
- Label: property Plural Label:
- property Enable Reports:
- Checked (✓)
- Track Activities and Field History: Enabled (✓)
- Deployment Status: Deployed

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTbN/Details/view

Search Setup

Object Manager

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

Relationships

outs

Record Pages

Links, and Actions

Layouts

Buttons

Types

Lookup Filters

Layouts

Button Layout

in Rules

Rules

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTuJ/Details/view

Search Setup

Object Manager

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

Relationships

outs

Record Pages

Links, and Actions

Layouts

Buttons

Types

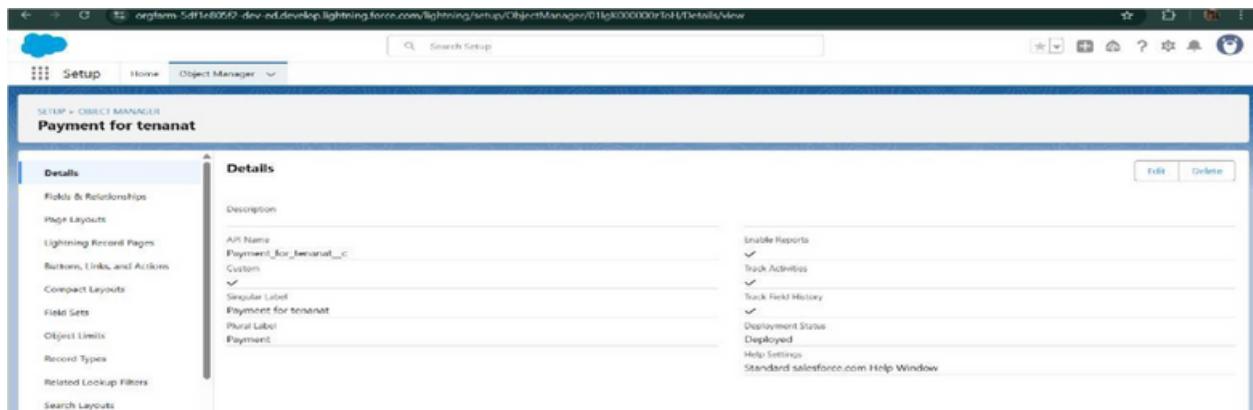
Lookup Filters

Layouts

Button Layout

in Rules

Rules



- Configured fields and relationships

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Relationships	Address	Address__c	Long Text Area(32768)		
Record Pages	Created By	CreatedById	Lookup(User)		
Buttons, Links, and Actions	Last Modified By	LastModifiedById	Lookup(User)		
Layouts	Name	Name__c	Text(25)		
Fields	Owner	OwnerId	Lookup(User,Group)		✓
Lookup Filters	property	property__c	Lookup(property)		✓
Buttons	property Name	Name	Text(50)		✓
Custom Layout	sfqt	sfqt__c	Text(18)		
Rules	Type	Type__c	Picklist		
Labels					

SETUP > OBJECT MANAGER  
**lease**

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

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

SETUP > OBJECT MANAGER  
**Tenant**

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

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone__c	Phone		
status	status__c	Picklist		
Tenant Name	Name	Text(80)		

- Developed Lightning App with relevant tabs

The screenshot shows the 'App Details & Branding' section of the Lightning App Builder. On the left, a sidebar lists 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main area has tabs for 'App Details & Branding' (selected), 'App Options', 'Pages', and 'Lease Management'. Under 'App Details & Branding', there are fields for 'App Name' (Lease Management), 'Developer Name' (Lease\_Management), 'Image' (a placeholder image of a person in a suit), 'Primary Color Hex Value' (#0070D2), and a 'Description' box (Application to efficiently handle the processes related to leasing real estate properties.). There is also an 'Org Theme Options' checkbox and an 'App Launcher Preview' section showing a preview card with the app name, image, and description.

- App Name is “Lease Management”. Developer
- Name is “Lease\_Management”.
- Description explains handling leasing real estate processes. Primary
- Color Hex Value is “#0070D2”.
- App Launcher Preview displays app name, image, and description.

The screenshot shows the 'Navigation Items' section of the Lightning App Builder. The sidebar includes 'App Options', 'Utility Items (Desktop Only)' (selected), and 'User Profiles'. The main area shows 'Available Items' on the left (Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, etc.) and 'Selected Items' on the right (Payment, Tenants, property, lease). Navigation arrows between the two lists allow items to be moved.

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

**User Profiles**

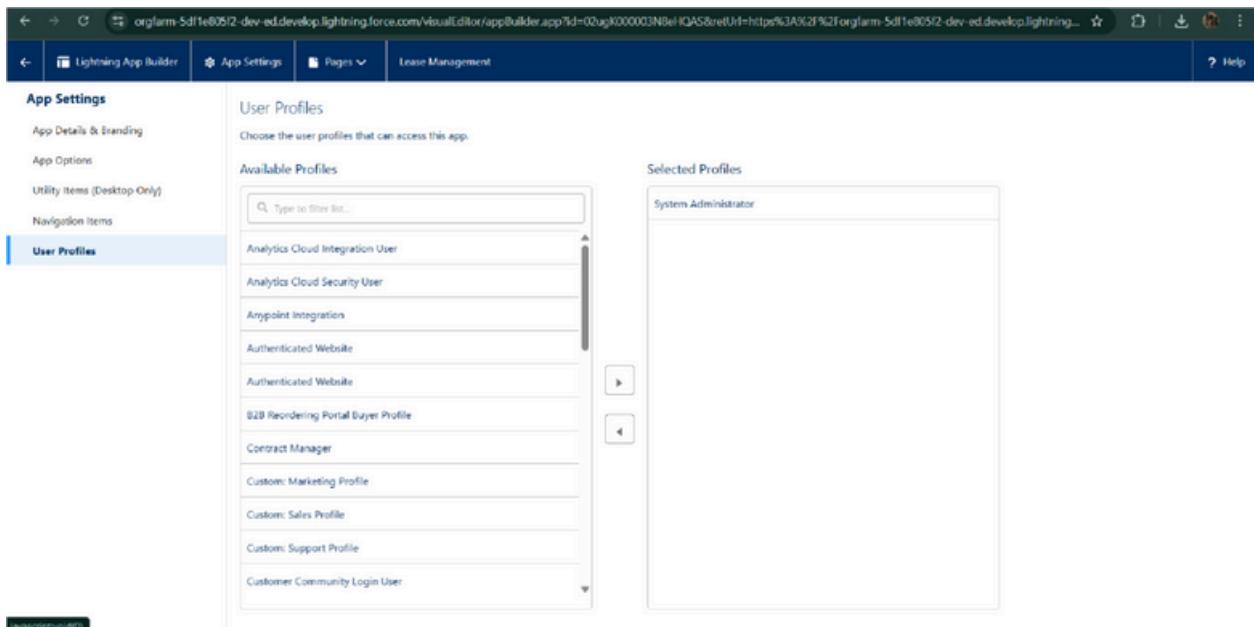
Choose the user profiles that can access this app.

**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

**Selected Profiles**

- System Administrator

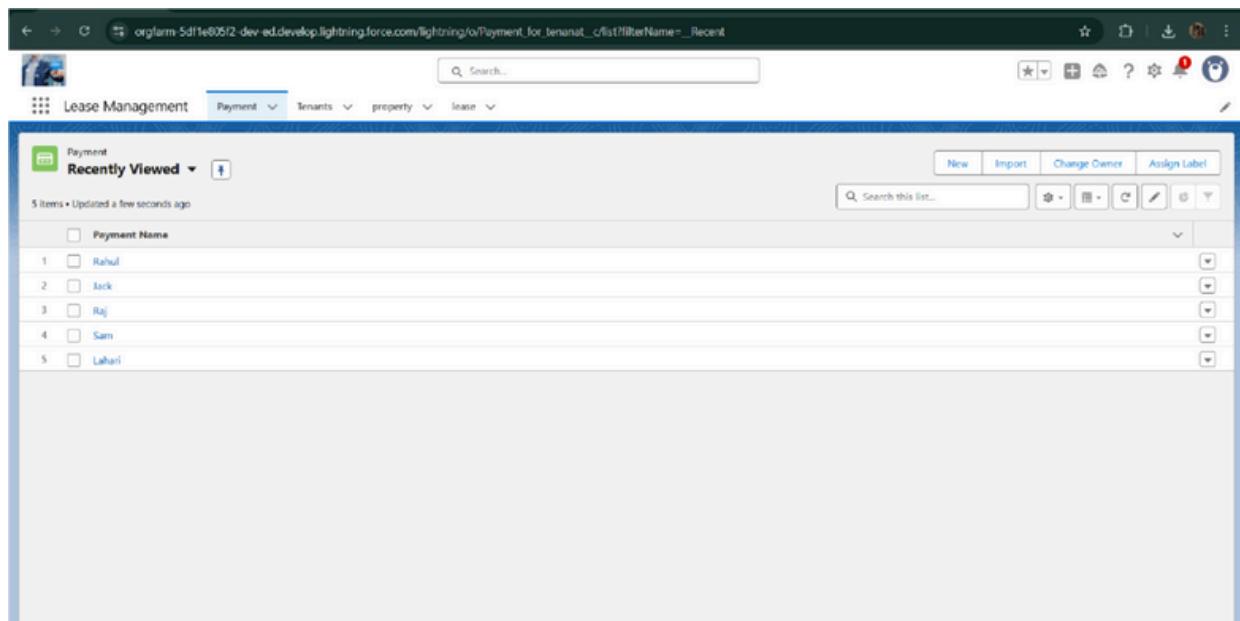


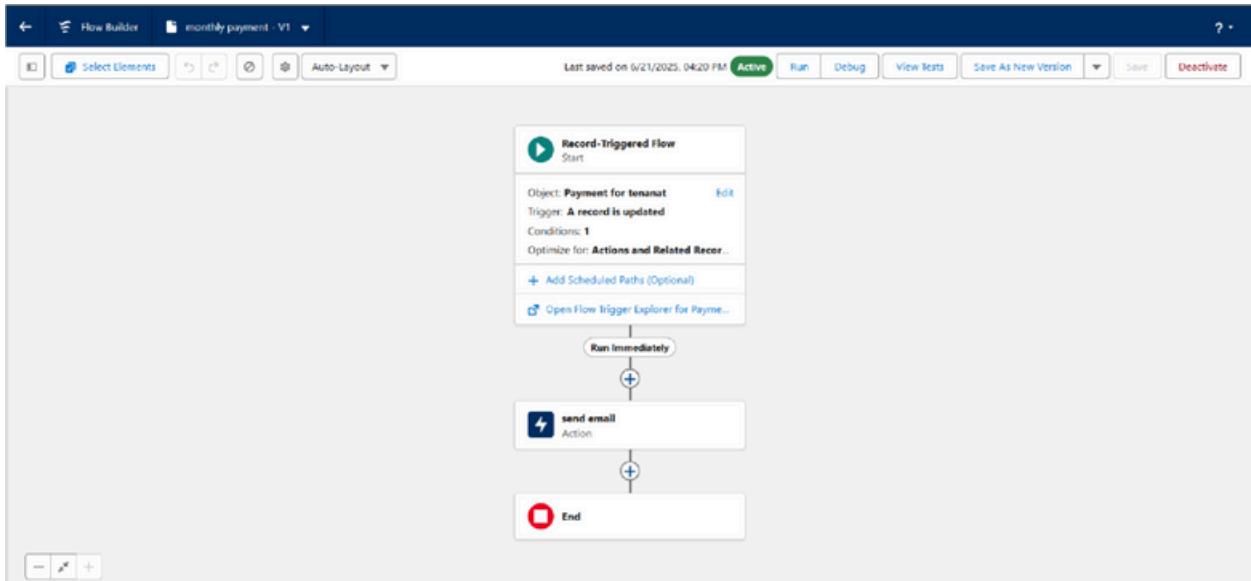
Lease Management | Payment | Tenants | property | lease | Search... | New | Import | Change Owner | Assign Label | Recently Viewed |

Recently Viewed

5 items • Updated a few seconds ago

Payment Name	Owner	Last Modified
Rahul	Rahul	1 day ago
Jack	Jack	1 day ago
Raj	Raj	1 day ago
Sam	Sam	1 day ago
Lahari	Lahari	1 day ago





- Implemented Flows for monthly rent and payment success

The screenshot shows the Salesforce Object Manager with the "lease" object selected. A validation rule is being edited with the name "lease\_end\_date". The formula entered is "End\_date\_\_c <= Start\_date\_\_c". The formula editor includes a dropdown for functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, and a help section for ABS(number).

- To create a validation rule to a Lease Object

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'lease' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area displays a 'Validation Rule Detail' for a rule named 'lease\_end\_date'. The rule's formula is 'End\_date\_c < start\_date\_c', with an error message stating 'Your End date must be greater than start date'. The rule is active and was created by the 'Sowmya Team' on June 19, 2025, at 5:37 AM. It was last modified by the same team on June 26, 2025, at 7:47 AM.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the 'Lease Management' application. A modal window titled 'New Tenant' is open, prompting for information such as Tenant Name ('chinu') and Email ('chinu@gmail.com'). An error message at the bottom states: 'We hit a snag. Review the errors on this page. \* A tenant can have only one property.' The background shows a list of tenants, including 'niranjan'.

- Scheduled monthly reminder emails using Apex class

The screenshot shows the Salesforce Apex code editor with the following code:

```

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

```

The code implements the Schedulable interface and defines a static method to send emails to all tenants whose day of the month is 1. It uses the Messaging.SingleEmailMessage class to prepare and send the emails.

- Built and tested email templates for leave request, approval, rejection, payment, and reminders
- Here are 5 key points (in 2 lines each) from the code in the screenshot: Apex
- class MonthlyEmailScheduler implements Schedulable.
- It runs scheduled jobs to send monthly reminder emails. execute
- method checks if the day is the 1st.
- If true, it calls the sendMonthlyEmails() method.
- sendMonthlyEmails() queries Tenant records.
- It fetches Id and Email\_c fields from Tenant\_c object. For each tenant, an email is prepared.
- Email includes subject "Reminder: Monthly Rent Payment Due". Uses
- Messaging.SingleEmailMessage to send mail.
- Sets recipient, subject, and email content dynamically.

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

**Classic Email Templates**

**tenant leaving**

**Email Template Detail**

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

**Email Template**

**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 following details:

**Classic Email Templates**

**tenant leaving**

**Email Template Detail**

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

**Email Template**

**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 following details:

**Page Header:** Search Setup, Help for this Page

**Left Navigation:** Setup, Home, Object Manager, Email (selected), Classic Email Templates (selected), Lightning Email Templates.

**Section Header:** Classic Email Templates

**Section Sub-Header:** Tenant Email

**Text:** Preview your email template below.

**Email Template Detail:**

Email Templates from Salesforce		Unified Public Classic Email Templates
Email Template Name	Tenant_Email	Available For Use <input checked="" type="checkbox"/>
Template Unique Name	Tenant_Email	Last Used Date
Encoding	Unicode (UTF-8)	Times Used
Author	Sowmya_Team [Changed]	
Description		
Created By	Sowmya_Team, 6/20/2025, 1:12 AM	Modified By Sowmya_Team, 6/20/2025, 1:12 AM

**Email Template:** Send Test and Verify Merge Fields

**Subject:** Urgent: Monthly Rent Payment Reminder

**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 following details:

**Page Header:** Search

**Left Navigation:** Templates (selected), Email Templates.

**Section Header:** Classic Email Templates

**Section Sub-Header:** tenant payment

**Text:** Preview your email template below.

**Email Template Detail:**

Email Templates from Salesforce		Unified Public Classic Email Templates
Email Template Name	tenant_payment	Available For Use <input checked="" type="checkbox"/>
Template Unique Name	tenant_payment	Last Used Date
Encoding	Unicode (UTF-8)	Times Used
Author	Sowmya_Team [Changed]	
Description		
Created By	Sowmya_Team, 6/20/2025, 1:13 AM	Modified By Sowmya_Team, 6/20/2025, 1:13 AM

**Email Template:** Send Test and Verify Merge Fields

**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.
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', a template named 'tenant payment' is selected. The 'Email Template Detail' section shows the following information:

- Email Templates from Salesforce: tenant payment
- Email Template Name: tenant payment
- Template Unique Name: tenant\_payment
- Encoding: Unicode (UTF-8)
- Author: Scenova Team [Channe]
- Description: Created By: Scenova Team, 6/20/2025, 1:13 AM
- Modified By: Scenova Team, 6/20/2025, 1:13 AM
- Available For Use: checked
- Last Used Date: N/A
- Times Used: N/A

The 'Email Template' section below shows the email content:

```

Subject: Confirmation of Successful Monthly Payment
Plain Text Preview
Dear {Recipient_email}.

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 'Data' section selected. Under 'Approval Processes', an approval process named 'Tenant: TenantApproval' is selected. The 'Process Definition Detail' section shows the following information:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Tenant's status equals Stay
- Entry Criteria: Administrator ONLY
- Record Editability: Allow Submitters to Recall Approval Requests
- Active: checked
- Next Automated Approval Determined by: N/A
- Created By: Scenova Team, 6/23/2025, 3:41 AM
- Modified By: Scenova Team, 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   Edit	1	Step 1			User:Scenova Team	Final Rejection

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page selected. The process 'check for vacant' is displayed with the following details:

- Process Name:** check for vacant
- Unique Name:** check\_Uv\_vacant
- Description:** Tenant: check for vacant
- Entry Criteria:** Tenant\_\_c status EQUALS LEAVING
- Recent Entitlement:** Administrator ONLY
- Next Automated Approver Determined By:** Active (checked)
- Approval Assignment Email Template:** Leave approved
- Initial Submitters:** Tenant Owner
- Created By:** Scouting Team
- Modified By:** Scouting Team
- Date Created:** 6/20/2025, 3:16 AM
- Date Modified:** 6/26/2025, 11:02 PM

**Initial Submission Actions:**

Action	Type	Description
Record Lock		Lock the record from being edited
Email Alert		2032280005.MY.ICBVS

**Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	1	step1			User.Scouting Team	Final Rejection

- Apex Trigger

### Create an Apex Trigger

```

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

```

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'test'. The trigger code prevents insertions for the 'Tenant\_\_c' object. A modal window titled 'Open' is displayed, showing a list of entities in the 'test' namespace.

Entity Type	Edition	Referred
Classes	None	
Triggers	test	
Pages		
Page Components		
Objects		
Static Resources		
Packages		

Developer Console - Google Chrome  
org:farm-5d1fe605f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

test.apexp testHandler.apexp MonthlyEmailScheduler.apexp

Code Coverage: None API Version: 64 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

None Use Problem

## Create an Apex Handler class

Developer Console - Google Chrome  
org:farm-5d1fe605f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

test.apexp testHandler.apexp MonthlyEmailScheduler.apexp

Code Coverage: None API Version: 64 Go To

```
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.Id);
        }
        for (Tenant__c newTenant : newList) {
            if (newTenant.Property__c != null) {
                newTenantaddError('A');
            }
        }
    }
}
```

Existing Tenant Class References

Entity Type	Entity	Related
Classes	testHandler	Name Extent Direction
Triggers	MonthlyEmailScheduler	← test ApexTrigger Referenced
Pages		property CustomField References
Page Components		← Tenant__c SObject References
Objects		← Tenant__c SObject References
Static Resources		
PDF Generators		

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

Logs Tests Checkpoints Query Editor View State Progress Problems

None Use Problem

Developer Console - Google Chrome

orgfarm-Sdf1e805f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

testHandler.apc [ testHandler.apc ] Hosted by salesforce.com

Code Coverage Name: API Version: 64 Go To

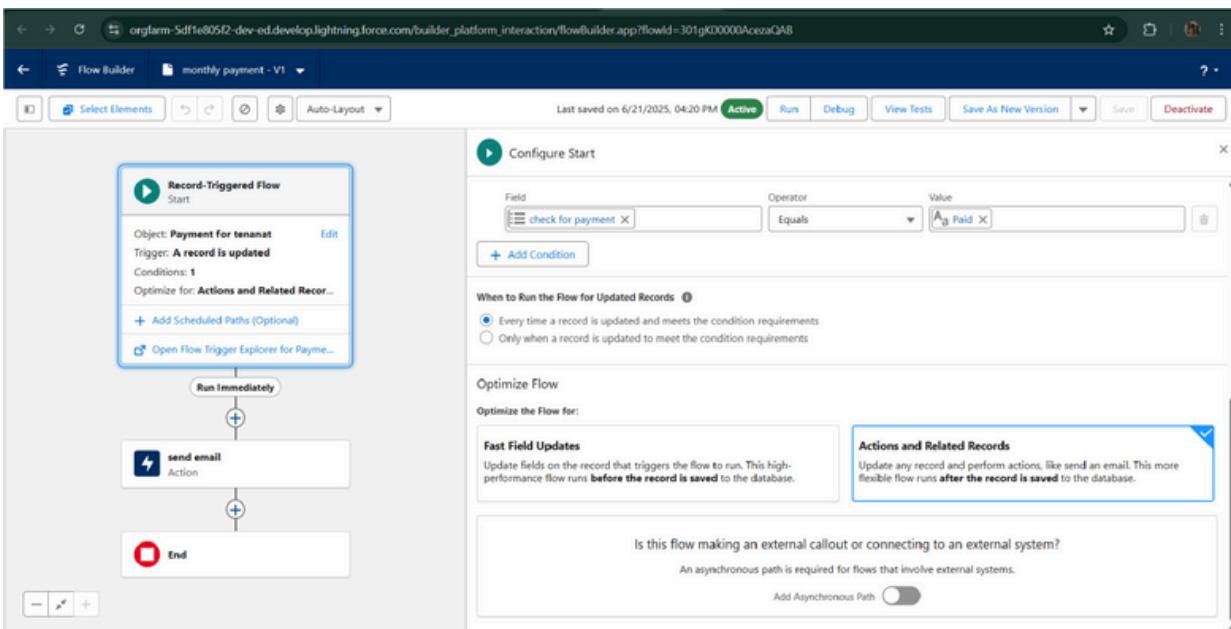
```

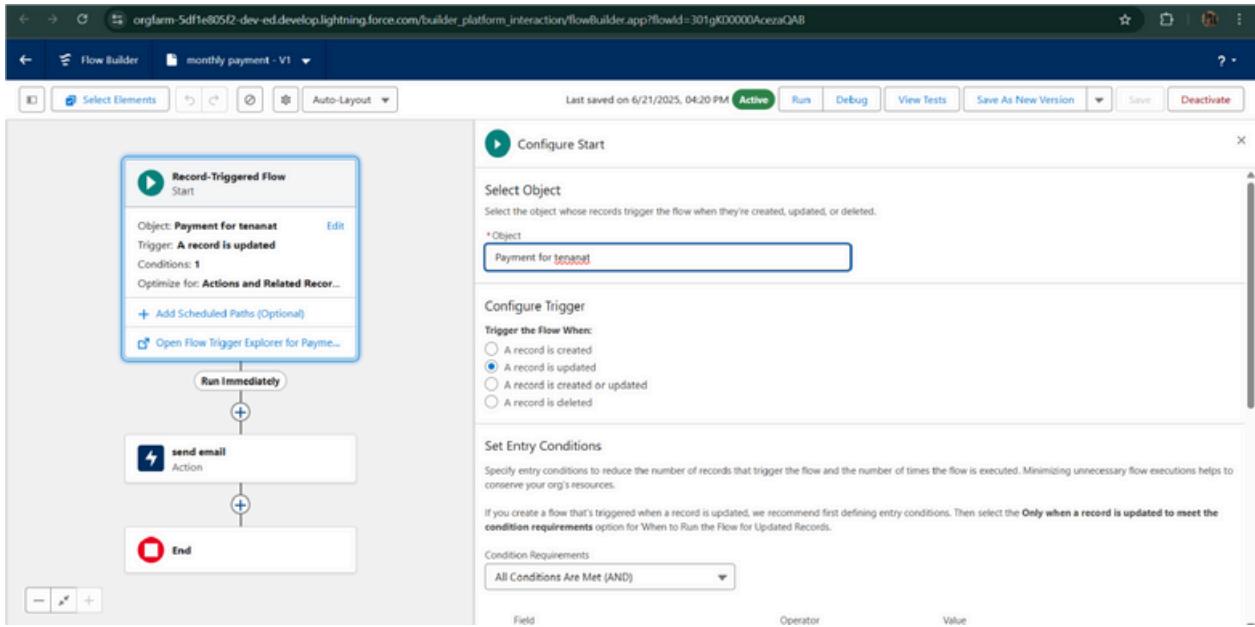
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

## ● FLOWS





- Schedule class:  
Create an Apex Class

Developer Console - Google Chrome

orgfarm-5dfe805f2-dev-ed.develop.lightning.force.com/builder\_platform\_interaction/flowBuilder.app?flowId=301gKD00000AceaQAB

```
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
19
20         for (Tenant__c tenant : tenants
21
22             String recipientEmail = tenant.Email__c;
```

The screenshot shows the Developer Console in a browser window. The code editor displays an Apex class named 'MonthlyEmailScheduler'. A modal window titled 'Open' is overlaid on the screen, showing a list of entities in the repository, with 'MonthlyEmailScheduler' selected. The modal includes tabs for 'Entity Type', 'Edition', and 'Related'.

```

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             sendMonthlyEmails();
9         }
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 hope 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.";
26
27         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28
29         email.setToRecipients(new String[]{recipientEmail});
30
31         email.setSubject(emailSubject);
32
33         email.setHtmlBody(emailContent);
34
35         messaging.sendEmail(new Messaging.SingleEmailMessage[] {email});
36
37     }
38
39 }
40
41 }

```

## Schedule Apex class

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

- Search Bar:** Q apex
- Left Navigation:**
  - Email
  - Custom Code
  - Apex Classes (selected)
  - Apex Settings
  - Apex Test Execution
  - Apex Test History
  - Apex Triggers
  - Environments
  - Jobs
  - Apex Flex Queue
  - Apex Jobs
- Apex Classes Page:**
  - Apex Class Detail:** Name: MonthlyEmailScheduler, Status: Active, Created By: Soamya\_Teams, Last Modified By: Soamya\_Teams.
  - Class Body:** Displays the Apex code for MonthlyEmailScheduler.

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

\* = Required Information

Tenant Name: Aswini  
Email: aswiniamaraadi15@gmail.com  
Phone: (905) 223-5567  
Status: Leaving  
Property: Imran

Owner: Sowmya Team

Created By: Sowmya Team, 6/26/2025, 6:05 AM Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

New Contact Edit New Opportunity

Activity

Filters: All time • All activities • All types Refresh

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.

New Case New Lead Delete Clone Change Owner Printable View Submit for Approval Edit Labels

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

\* = Required Information

Tenant Name: Aswini  
Email: aswiniamaraadi15@gmail.com  
Phone: (905) 223-5567  
Status: Leaving  
Property: Imran

Owner: Sowmya Team

Created By: Sowmya Team, 6/26/2025, 6:05 AM Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

Cancel Save

✓ Tenant was submitted for approval.

New Contact Edit New Opportunity

Activity

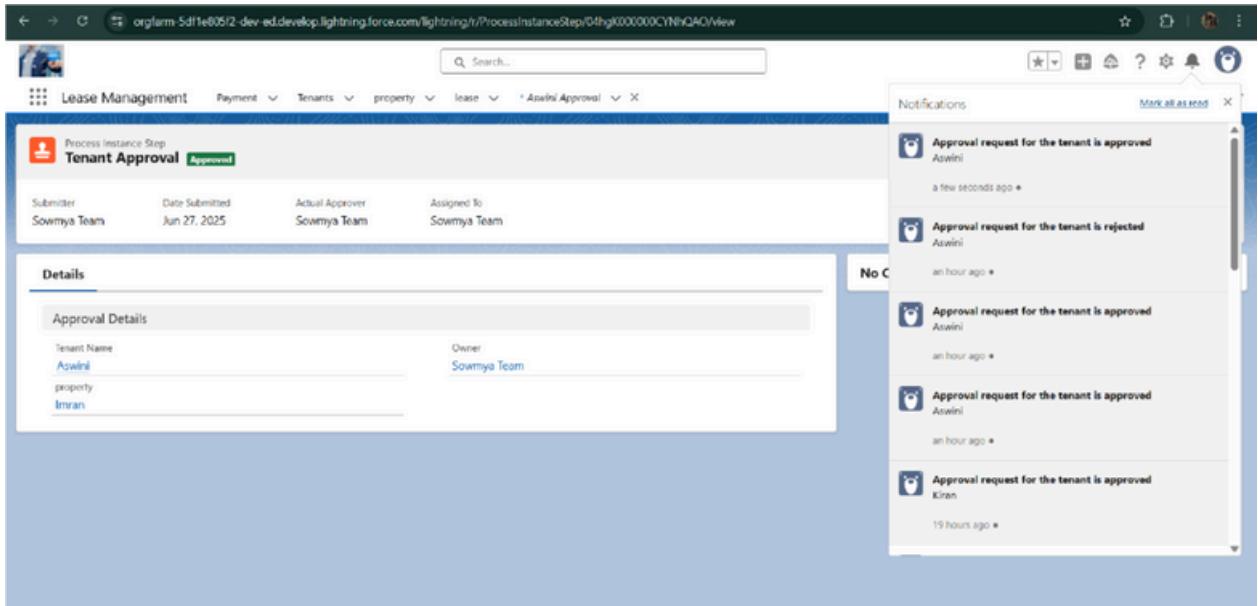
Filters: All time • All activities • All types Refresh Expand All View All

Upcoming & Overdue

No activities to show. Get started by sending an email, scheduling a task, and more.

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

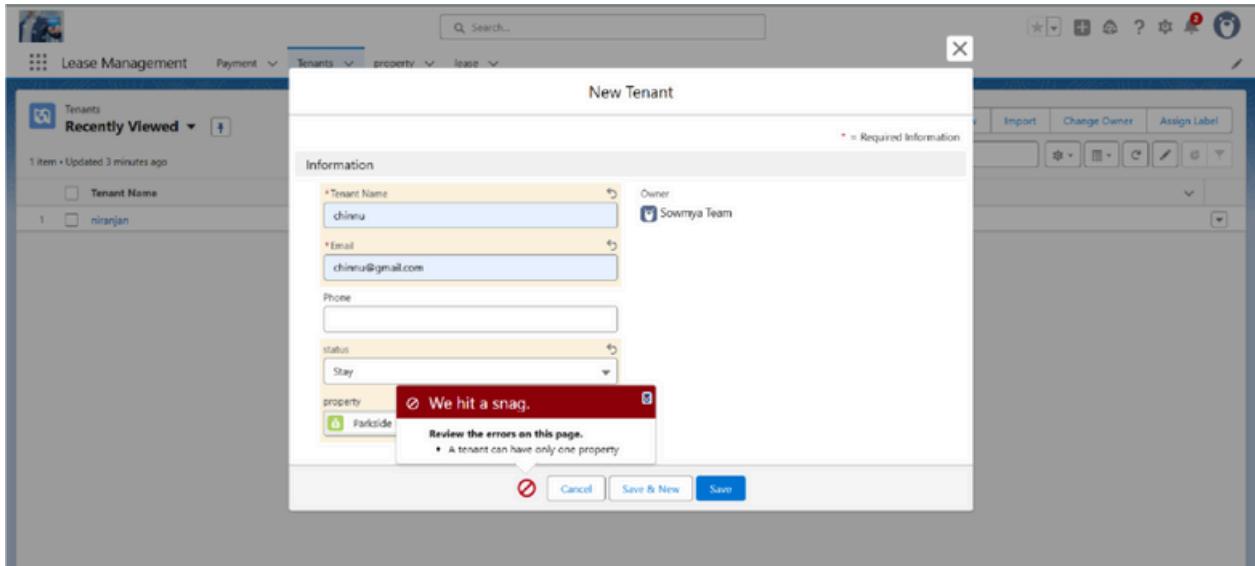
Cloudy 30°C ENG IN 12:46 27-06-2025



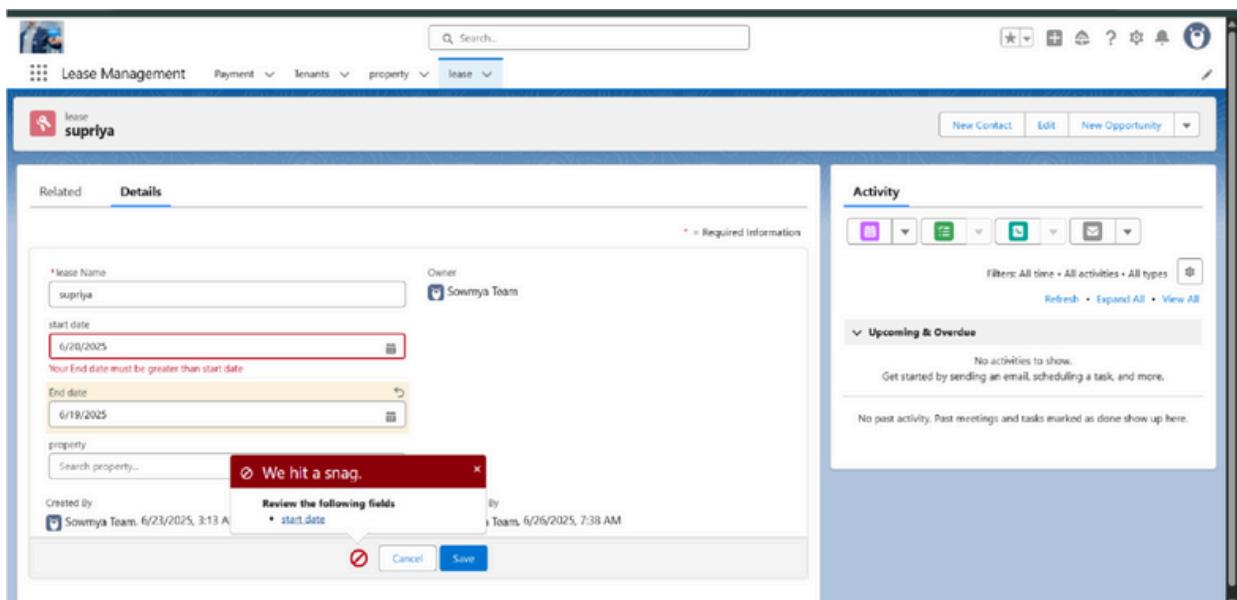
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

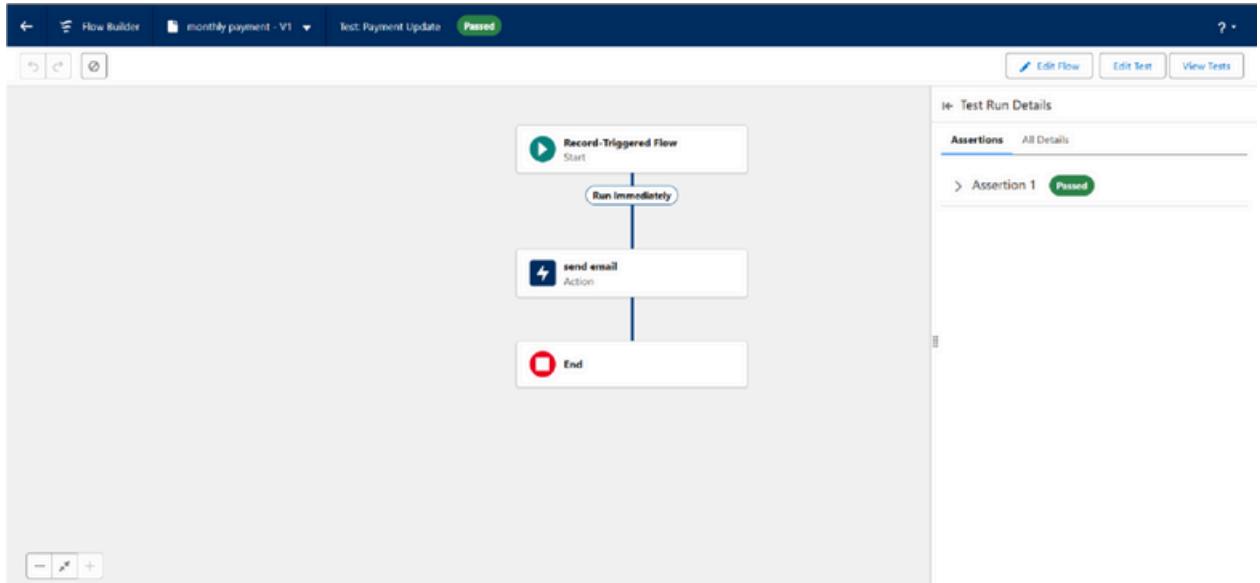
- Triggervalidationbyenteringduplicate tenant-property records



- Validation Rule checking



- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows a tenant detail page for 'niranjan' in the 'Lease Management' system. The 'Details' tab is selected, displaying fields for Tenant Name (niranjan), Email (niranjan1506@gmail.com), Phone, status (Stay), property (Parkside Lofts), and created/modified by (Sowmya Team). A sidebar on the right lists notifications: 'Approval request for the tenant is approved' (a few seconds ago), 'Approval request for the tenant is rejected' (Jun 23, 2025, 4:29 PM), 'Approval request for the tenant is approved' (Jun 23, 2025, 4:25 PM), 'Approval request for the tenant is approved' (Jun 23, 2025, 4:14 PM), and a guidance center learning resource available.

The screenshot shows a tenant detail page for 'niranjan' in the 'Lease Management' system. The 'Details' tab is selected, displaying fields for Tenant Name (niranjan), Email (niranjan1506@gmail.com), Phone, status (Stay), property (Parkside Lofts), and created/modified by (Sowmya Team). A sidebar on the right lists notifications: 'Approval request for the tenant is approved' (a few seconds ago), 'Approval request for the tenant is rejected' (Jun 23, 2025, 4:29 PM), 'Approval request for the tenant is approved' (Jun 23, 2025, 4:25 PM), 'Approval request for the tenant is approved' (Jun 23, 2025, 4:14 PM), and a guidance center learning resource available.

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

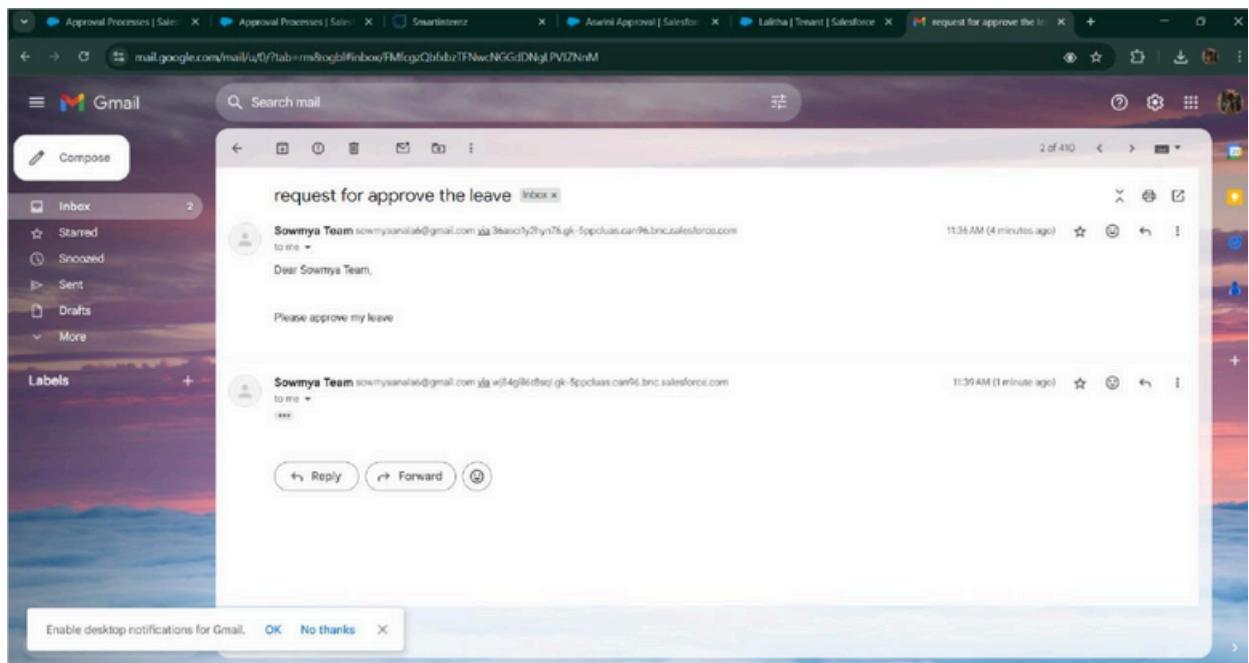
The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. The 'Custom Object Tabs' section displays four tabs: 'Lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'Receipts' (Tab Style: Back), and 'Receipts' (Tab Style: Map). The 'Web Tabs' and 'Visualforce Tabs' sections are currently empty.

- Email alerts

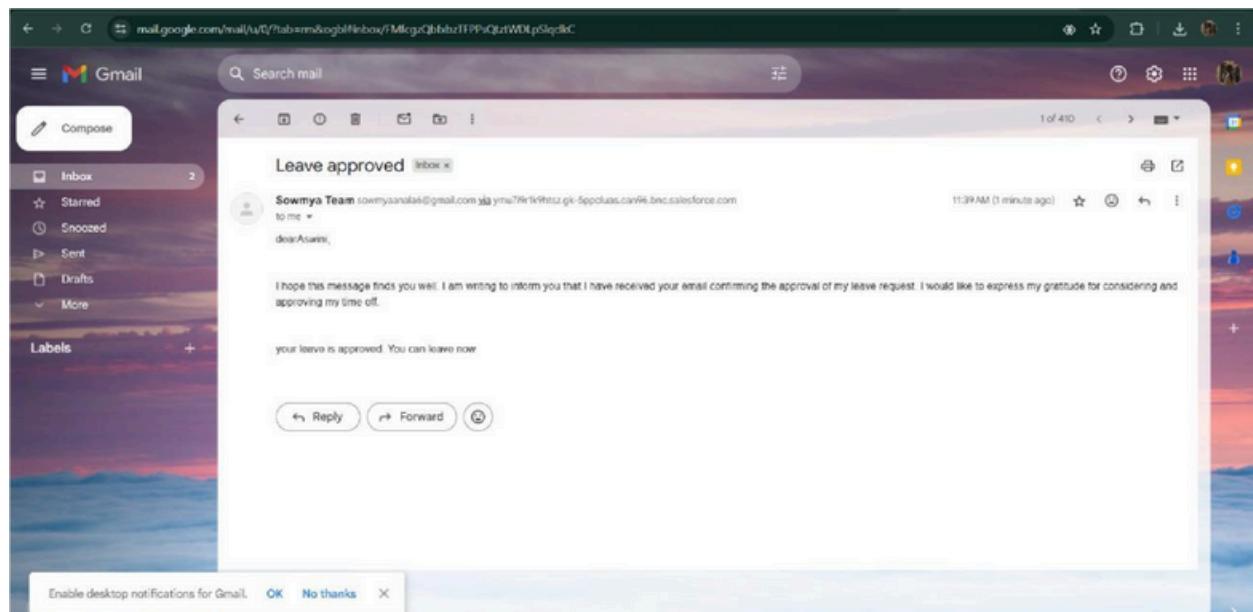
The screenshot shows the 'Approval History' section of the Lease Management application. It lists 8 items, all sorted by 'Is Pending' and updated a few seconds ago. The columns include Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The history shows various steps for approval requests, with statuses like Approved, Submitted, Rejected, and Pending, and comments indicating actions like 'leaving' or 'Approved'.

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

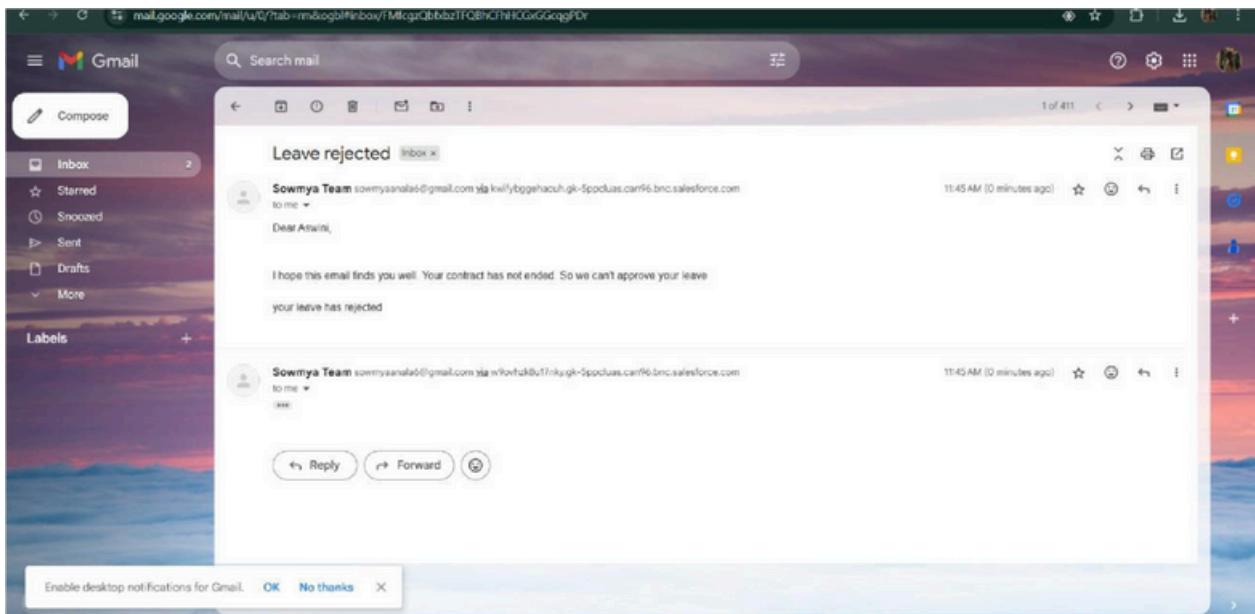
- Request for approve the leave



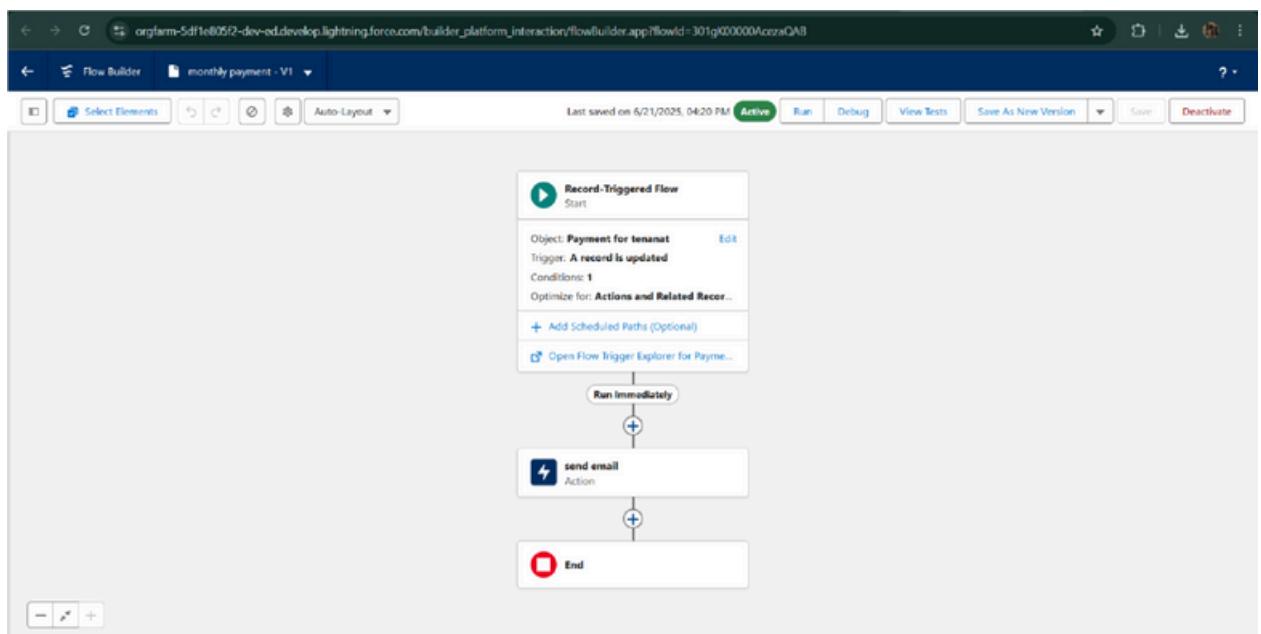
- Leave approved



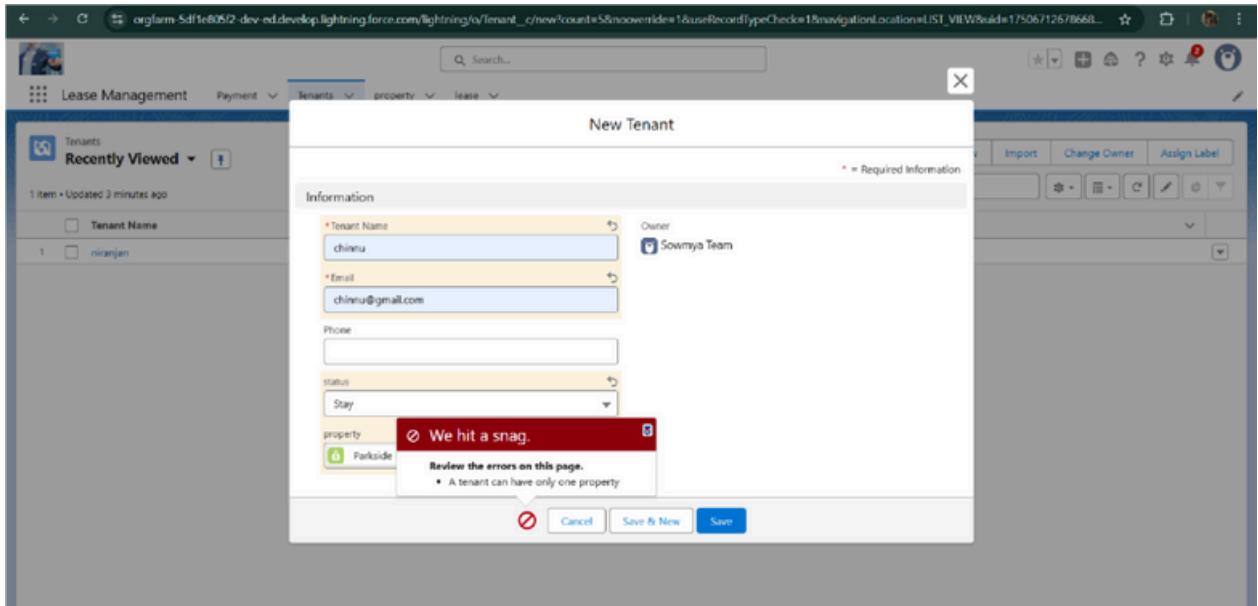
- Leave rejected



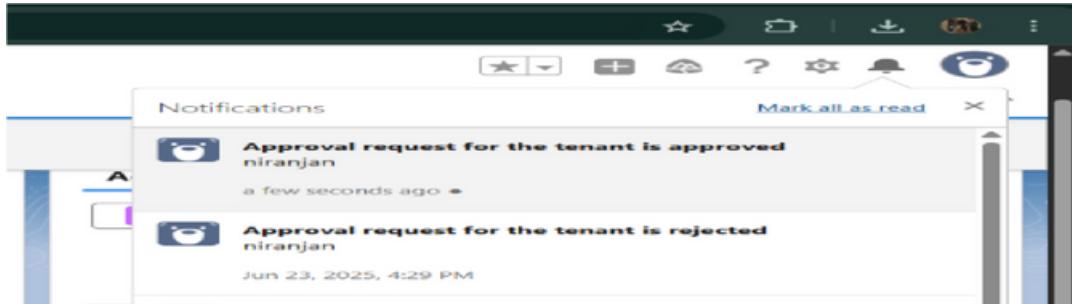
- Flow runs



- Trigger error messages



- Approval process notifications



## ADVANTAGES & DISADVANTAGES

---

### Advantages

1. Cost Predictability
  - Lease agreements typically involve fixed payments, making budgeting and forecasting easier.
2. Access to High-Value Assets
  - Allows businesses to use expensive assets (e.g., real estate, machinery, vehicles) without large upfront capital investment.

### 3. Flexibility

- Leasing can offer flexibility in contract length, upgrades, and expansion options.

### 4. Tax Benefits

- Lease payments can often be deducted as business expenses, reducing taxable income.

### 5. Reduced Maintenance Responsibility

- Depending on the lease type, maintenance and repairs may be the responsibility of the lessor (e.g., in triple net or full-service leases).

## Disadvantages

### 1. Long-Term Cost

- Over time, leasing can be more expensive than purchasing the asset outright.

### 2. Limited Control

- Leased assets are not owned, so modifications or usage may be restricted by lease terms.

### 3. Contractual Obligations

- Lease agreements may lock businesses into long-term commitments, limiting flexibility if business needs change.

### 4. Complexity

- Managing multiple lease agreements across locations or asset types can be administratively complex without the right system.

### 5. Penalties

- Early termination, overuse, or violations of lease terms can result in penalties or legal disputes.

## 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});  
    }  
}  
}
```