

# **LEASE MANAGEMENT**

**College Name: K.G.College of Arts and Science**

**College Code: 641035**

**TEAM ID: NM2025TMID23769**

**TEAM MEMBERS: FOUR**

**Team LeaderName: SHRIHARI M**

**Email: 2326ka49@kgcas.com**

**Team Member: Sree Varshini A**

**Email: 2326ka50@kgcas.com**

**Team Member: SUDHARSON R**

**Email: 2326ka51@kgcas.com**

**Team Member: Suseela S**

**Email: 2326ka52@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>

The screenshot shows the 'Sign up for your Developer Edition' page. The left side features a dark blue header with the Salesforce logo and a promotional message: 'Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.' Below this is a button labeled 'Sign up for your Developer Edition.' followed by a bulleted list of benefits. The right side contains form fields for personal information and a checkbox for agreeing to terms. A note at the bottom states that the org may be provisioned on or migrated to Hyperforce.

**Sign up for your Developer Edition**  
A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Shrihari Last name: M

Job title: Developer Work email: 2326ka49@kgcas.v

Company: Salesforce Country/Region: India

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

- Created objects: Property, Tenant, Lease, Payment

Setup > Object Manager

### property

Details

Description

API Name: property\_c  
Custom: ✓  
Singular Label: property  
Plural Label: property

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

Enable Reports

- ✓ Track Activities
- ✓ Track Field History
- ✓ Deployment Status
- Deployed
- Help Settings
- Standard salesforce.com Help Window

Edit Delete

Setup > Object Manager

### Tenant

Details

Description

API Name: Tenant\_c  
Custom: ✓  
Singular Label: Tenant  
Plural Label: Tenants

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

Enable Reports

- ✓ Track Activities
- ✓ Track Field History
- ✓ Deployment Status
- Deployed
- Help Settings
- Standard salesforce.com Help Window

Edit Delete

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTuj/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. The left sidebar lists various configuration tabs: 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, and Restriction Rules. The main 'Details' tab is selected. The right panel displays the object's details, including its API name ('lease\_c'), which is custom and singular ('lease'). It also shows the plural label ('lease'). On the far right, there are checkboxes for enabling reports, tracking activities, and field history, along with deployment status ('Deployed') and help settings. A link to the standard Salesforce help window is also present.

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zToH/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'Payment for tenant' object. The left sidebar lists the same configuration tabs as the previous screenshot. The main 'Details' tab is selected. The right panel displays the object's details, including its API name ('Payment\_for\_tenant\_c'), which is custom and singular ('Payment for tenant'). It also shows the plural label ('Payment'). On the far right, there are checkboxes for enabling reports, tracking activities, and field history, along with deployment status ('Deployed') and help settings. A link to the standard Salesforce help window is also present.

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The left sidebar lists various configuration options under 'Fields & Relationships'. The main area displays a table titled 'Fields & Relationships' with 9 items, sorted by Field Label.

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(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

The screenshot shows the Salesforce Object Manager interface for the 'Payment for tenant' object. The left sidebar lists various configuration options under 'Fields & Relationships'. The main area displays a table titled 'Fields & Relationships' with 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		Fields & Relationships		
7 items, Sorted by Field Label		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		

Details  
Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout  
Restriction Rules  
Scoping Rules

Setup Home Object Manager

SETUP > OBJECT MANAGER  
Tenant

Fields & Relationships		Fields & Relationships		
7 items, Sorted by Field Label		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	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
status	status__c	Picklist		
Tenant Name	Name	Text(80)		✓

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

- Developed Lightning App with relevant tabs

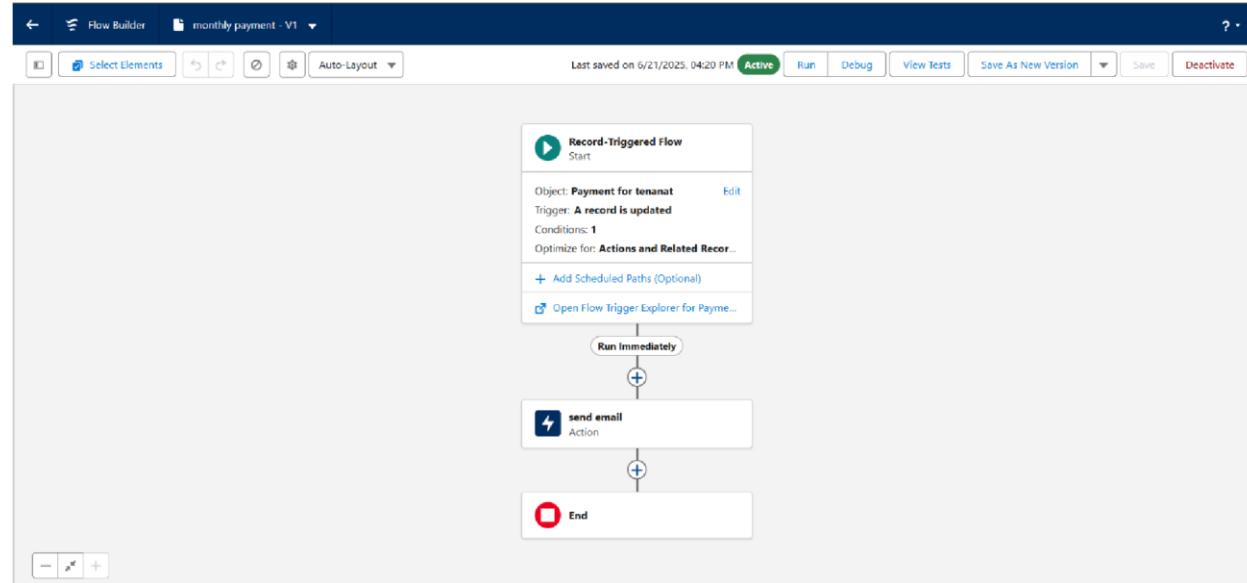
The screenshot shows the 'Lightning App Builder' interface with the 'App Details & Branding' tab selected. On the left, a sidebar lists 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main area contains fields for 'App Name' (Lease Management), 'Developer Name' (Lease\_Management), and a 'Description' box (Application to efficiently handle the processes related to leasing real estate properties). To the right, there's an 'App Branding' section with an 'Image' (a person interacting with a screen), a 'Primary Color Hex Value' (#0070D2), and an 'Org Theme Options' checkbox. Below this is an 'App Launcher Preview' showing a card with the app's name, image, and description.

The screenshot shows the 'Lightning App Builder' interface with the 'Navigation Items' tab selected. The left sidebar includes 'App Options', 'Utility Items (Desktop Only)', and 'User Profiles'. The main area displays the 'Navigation Items' configuration. It features two columns: 'Available Items' (Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests) and 'Selected Items' (Payment, Tenants, property, lease). Navigation arrows between the columns allow items to be moved.

The screenshot shows the Lightning App Builder interface with the URL <https://orgfarm-5df1e005f2-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02ugK00003N9elQAS&retUrl=https%3A%2F%2Forgfarm-5df1e005f2-dev-ed.lightning...>. The page title is "Lightning App Builder". The left sidebar has sections: App Details & Branding, App Options, Utility Items (Desktop Only), Navigation Items, and User Profiles (which is selected). The main content area is titled "User Profiles" and says "Choose the user profiles that can access this app." It shows two panels: "Available Profiles" (listing various profiles like Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, etc.) and "Selected Profiles" (listing "System Administrator").

The screenshot shows the Lease Management application with the URL [https://orgfarm-5df1e005f2-dev-ed.lightning.force.com/lightning/o/Payment\\_for\\_tenant\\_c/list/filterName=\\_Recent](https://orgfarm-5df1e005f2-dev-ed.lightning.force.com/lightning/o/Payment_for_tenant_c/list/filterName=_Recent). The top navigation bar includes "Lease Management", "Payment", "Tenants", "property", and "lease". The main content area is titled "Payment" and shows a "Recently Viewed" section with 5 items. The list includes entries: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, and 5. Lahari. There are checkboxes next to each entry. The right side of the screen has standard Salesforce search and filter tools.

- 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 in the Object Manager. The validation rule is named 'lease\_end\_date'.

**Validation Rule Edit**

**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**

**Validation Rule Details**

**Rule Name:** lease\_end\_date

**Active:**

**Description:** (empty)

**Error Condition Formula:**

**Example:** Discount\_Percent\_\_c>30 | More Examples...  
Display an error if Discount is more than 30%

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

**Functions:**

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

**Formula:** end\_date\_c < start\_date\_c

**Check Syntax**

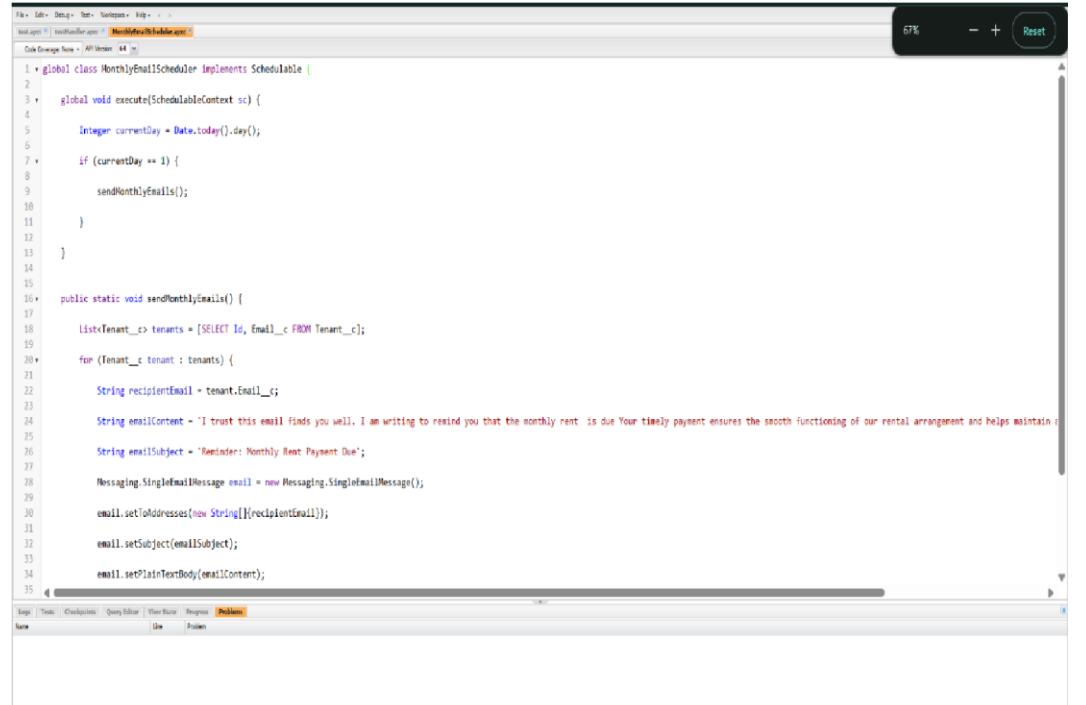
**Error Message:**

The screenshot shows the Salesforce Setup Object Manager 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 the 'Validation Rule Detail' for a rule named 'lease\_end\_date'. The rule's condition formula is 'End\_date\_\_c <= start\_date\_\_c', and its message is '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 6:37 AM.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the 'Lease Management' application's 'Tenants' screen. A new tenant is being added with the name 'chinnu' and email 'chinnu@gmail.com'. A validation error message 'We hit a snag.' appears, stating 'A tenant can have only one property'. The property dropdown shows 'Parkside' is selected. The 'Information' section includes fields for Tenant Name, Email, Phone, status (Stay), and property (Parkside). The owner is listed as 'Sowmya Team'.

- Scheduled monthly reminder emails using Apex class

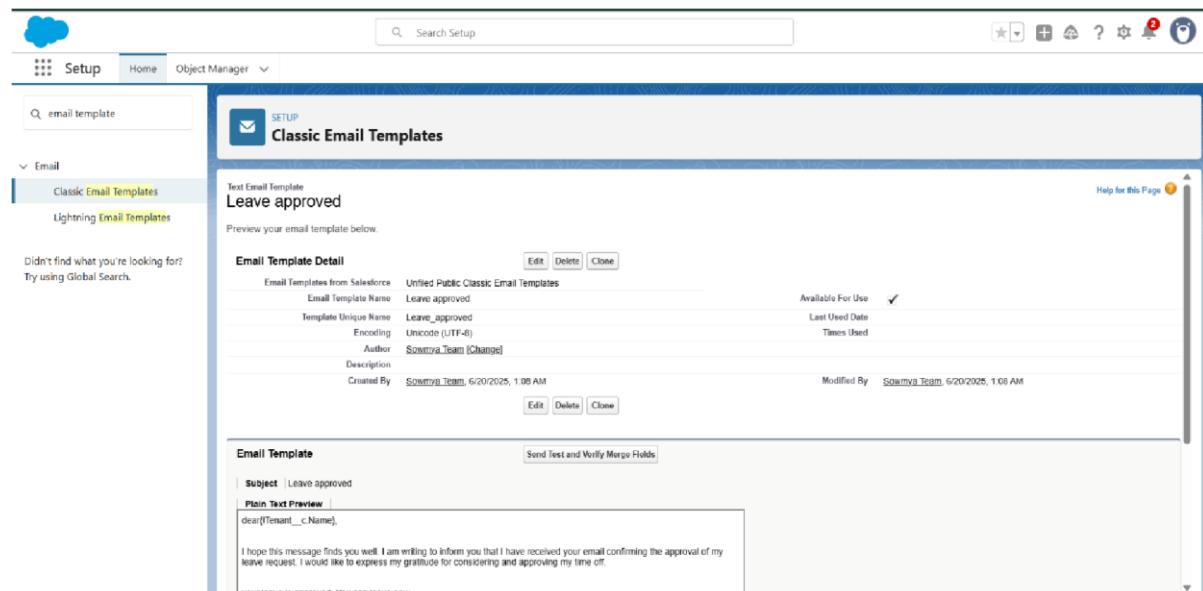


```

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 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 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 following details:

- Page Header:** Search Setup
- Left Navigation:** Home, Object Manager, Setup (selected), Email, Classic Email Templates (selected), Lightning Email Templates.
- Search Bar:** Search for "email template".
- Central Content:**
  - Section Title:** Classic Email Templates
  - Email Template Detail:**
    - Name:** Leave approved
    - Description:** Preview your email template below.
    - Buttons:** Edit, Delete, Close.
    - Details:** Email Templates from Salesforce, Unified Public Classic Email Templates, Available For Use (checked), Last Used Date, Times Used.
    - Created By:** Sowmya Team, Created: 6/20/2025, 1:08 AM
    - Modified By:** Sowmya Team, Modified: 6/20/2025, 1:08 AM
  - Email Template Preview:**
    - Subject:** Leave approved
    - Plain Text Preview:**

```
dear(Tenant__c.Name),
I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.
Your leave is approved. You can leave now.
```
    - Buttons:** Send Test and Verify Merge Fields, Edit, Delete, Close.

**Classic Email Templates**

**tenant leaving**

Email Template Detail

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

Available For Use: ✓

Last Used Date: Times Used:

**Email Template**

Subject: request for approve the leave

Plain Text Preview

Dear {Tenant\_\_c.CreatedBy},  
Please approve my leave

**Classic Email Templates**

**Leave rejected**

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team 6/20/2025, 1:11 AM
Modified By	Sowmya Team 6/20/2025, 1:11 AM

Available For Use: ✓

Last Used Date: Times Used:

**Email Template**

Subject: Leave rejected

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 search bar set to "email template". Under the "Email" category, "Classic Email Templates" is selected. A new template named "Tenant Email" is being created under "Text Email Template".

**Email Template Detail:**

- Email Template Name: Tenant Email
- Template Unique Name: tenant\_email
- Encoding: Unicode (UTF-8)
- Author: Sowmya\_Team [Change]
- Description: Created By: Sowmya\_Team 6/20/2025, 1:12 AM
- Modified By: Sowmya\_Team, 6/20/2025, 1:12 AM

**Email Template Preview:**

**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.
```

This communication is a friendly reminder regarding your monthly rent payment. Please let us know if you have any questions or concerns.

The screenshot shows the Salesforce Setup interface with the search bar set to "email template". Under the "Email" category, "Classic Email Templates" is selected. A new template named "tenant payment" is being created under "Text Email Template".

**Email Template Detail:**

- Email Template Name: tenant payment
- Template Unique Name: tenant\_payment
- Encoding: Unicode (UTF-8)
- Author: Sowmya\_Team [Change]
- Description: Created By: Sowmya\_Team 6/20/2025, 1:13 AM
- Modified By: Sowmya\_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	Active	<input checked="" type="checkbox"/>
Unique Name	TenantApproval	Next Automated Approver Determined By	<input type="checkbox"/>
Description		Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Entry Criteria	Tenant: status EQUALS Stay		
Record Editability	Administrator ONLY		

**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   Edit	1	Step 1			User: Sowmya Team	Final Rejection

## For Check for Vacant:

**Process Definition Detail**

Process Name	check_for_vacant	Active	<input checked="" type="checkbox"/>
Unique Name	check_for_vacant	Next Automated Approver Determined By	<input type="checkbox"/>
Description		Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Entry Criteria	Tenant: status EQUALS Leaving		
Record Editability	Administrator ONLY		

**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   Edit	1	step1			User: Sowmya Team	Final Rejection

- Apex Trigger

Create an Apex Trigger

The screenshot shows the Salesforce IDE interface. In the top-left, there's a code editor window titled "testApex" containing the following Apex trigger code:

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

Below the code editor is a modal dialog titled "Open" which lists entities. The "Entity Type" dropdown is set to "Triggers". The "Entities" table shows one entry:

Name	Namespace	Name	Extent	Direction
test				

At the bottom of the modal are buttons for "Open", "Filter", "Hide Managed Packages", and "Refresh".

Below the modal is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The "Problems" tab is selected. Underneath the navigation bar is a search bar with the placeholder "Name" and a dropdown menu showing "Line" and "Problem".

The screenshot shows the Developer Console in Google Chrome. The URL is "orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The page displays the same Apex trigger code as the IDE.

Below the code is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The "Problems" tab is selected. Underneath the navigation bar is a search bar with the placeholder "Name" and a dropdown menu showing "Line" and "Problem".

Create an Apex Handler class

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

testApex1 testHandler.apex HomeMonthlyEmailScheduler.apxc

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.Property__c);
10
11         }
12
13         for (Tenant__c newTenant : newList) {
14
15             if (newTenant.Property__c != null) {
16                 newTenantaddError('A tenant can have only one property');
17             }
18         }
19     }
20
21 }
```

Open Entity Type Entity Namespace Related

Entity Type	Entity	Namespace	Related
Classes	testHandler	Namespace	← test ApexTrigger References
Triggers	MonthlyEmailScheduler		property CustomField References
Pages			← Tenant__c SObject References
Page Components			← Tenant__c SObject References
Objects			
Static Resources			
Profiles			

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

testApex1 testHandler.apex HomeMonthlyEmailScheduler.apxc

Code Coverage: None API Version: 64 Go To

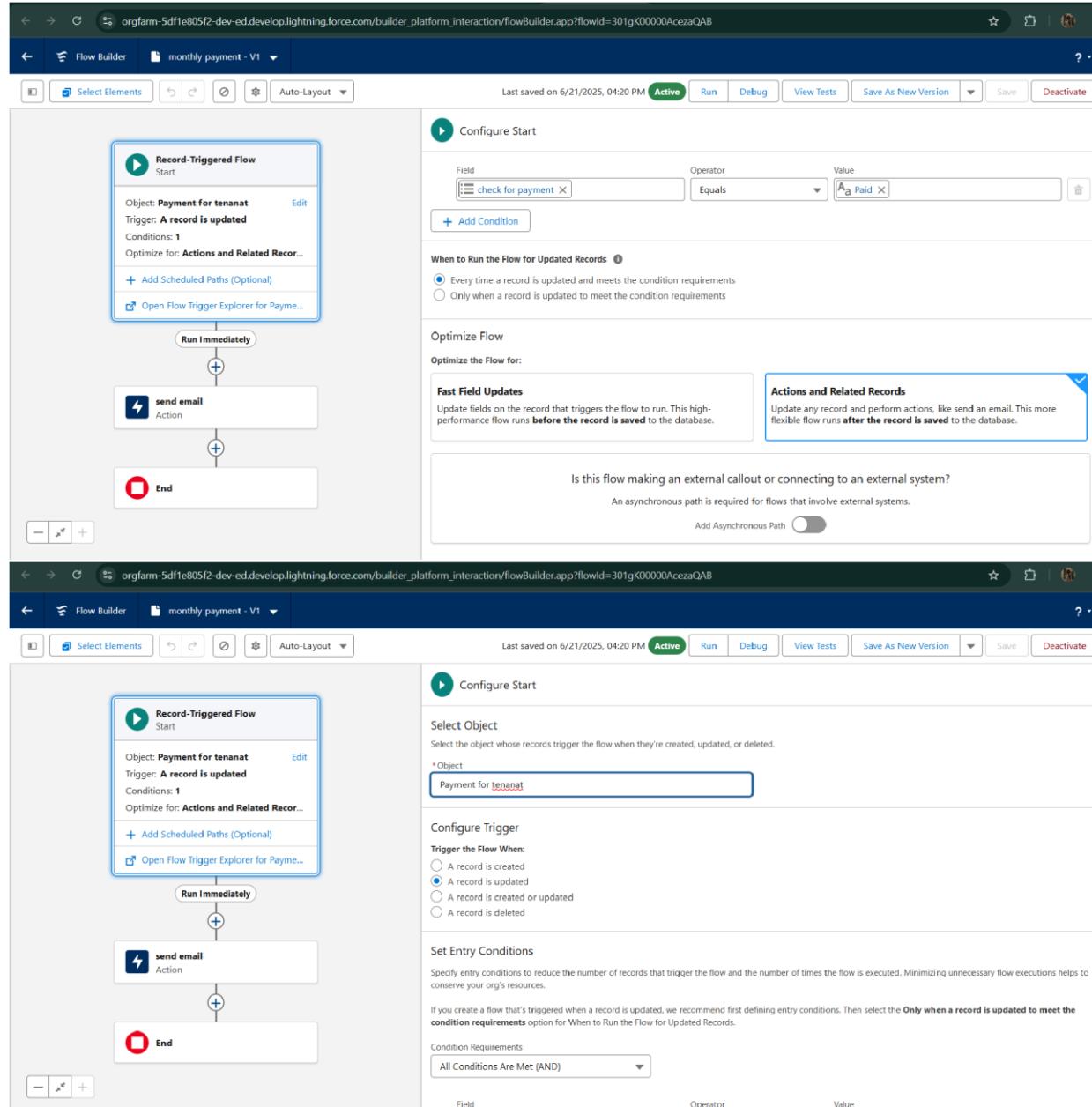
```

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.Property__c);
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 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

- FLOWS



- Schedule class: Create an Apex Class

Developer Console - Google Chrome  
orgfarm-Sdf1e8052-dev-ed develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ <   
test.apex ▾ testHandler.apex [ MonthlyEmailScheduler.apxc ]   
Code Coverage: None API Version: 64 ▾

```
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  
16 public static void sendMonthlyEmail  
17  
18     List<Tenant__c> tenants = [SEL  
19  
20         for (Tenant__c tenant : tenants)  
21  
22             String recipientEmail = tenant.Email__c;  
23 }
```

Open

Entity Type	Entities	Related
Entity type	Name Namespace +	Name Extent Direction
Classes	testHandler	Email CronTrigger... References
Triggers	MonthlyEmailScheduler	CustomField... References
Pages		← Tenant__c SObject References
Page Components		← Tenant__c SOBJECT References
Objects		
Static Resources		
Packages		

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

The screenshot shows the Google Chrome Developer Console with the title "Developer Console - Google Chrome". The URL is "orgfarm-5dfe805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The code editor displays an Apex class named "MonthlyEmailScheduler". The class implements the "Scheduleable" interface and contains two methods: "execute" and "sendMonthlyResults". The "execute" method checks if the current day is the 1st, and if so, sends an email to all tenants. The "sendMonthlyResults" method queries all tenants and sends them a monthly reminder email. The code uses the Messaging API to send the emails.

```
1 * @Job(1 Class MonthlyEmailScheduler implements Scheduleable {
2 *
3 *     global void execute(SchedulableContext sc) {
4 *
5 *         Integer currentday = Date.today().day();
6 *
7 *         if (currentday == 1) {
8 *             sendEmails();
9 *         }
10    }
11
12
13
14
15
16
17    public static void sendEmails() {
18
19        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
20
21
22        for (Tenant__c tenant : tenants) {
23
24            String recipientEmail = tenant.Email__c;
25
26            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.';
27
28            String emailSubject = 'Fee Under: monthly Rent Payment Due';
29
30            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
31
32            email.setToAddresses(new String[]{recipientEmail});
33
34            email.setSubject(emailSubject);
35
36            email.setPlainTextBody(emailContent);
37
38            Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
39
40        }
41    }
42 }
```

## Schedule Apex class

Screenshot of the Salesforce Setup Apex Classes page.

**Apex Class Detail**

**MonthlyEmailScheduler**

**Code:**

```
1 global class MonthlyEmailScheduler implements Schedulable {  
2     global void execute(SchedulableContext sc) {  
3         Integer currentDay = Date.today().day();  
4         if (currentDay == 1) {  
5             sendMonthlyEmails();  
6         }  
7     }  
8     public static void sendMonthlyEmails() {  
9         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];  
10        for (Tenant__c tenant : tenants) {  
11            ...  
12        }  
13    }  
14}  
15  
16  
17  
18  
19  
20}
```

**Details:**

- Name: MonthlyEmailScheduler
- Namespace Prefix:
- Created By: Sowmya Team, 6/23/2025, 2:46 AM
- Status: Active
- Code Coverage: 0% (0/15)
- Last Modified By: Sowmya Team, 6/23/2025, 2:47 AM

**Actions:**

- Edit
- Delete
- Download
- Security
- Show Dependencies

**Related:**

- Email
- Custom Code
- Apex Classes
- Apex Settings
- Apex Test Execution
- Apex Test History
- Apex Triggers
- Environments
- Jobs
- Apex Flex Queue
- Apex Jobs

Didn't find what you're looking for?  
Try using Global Search.

Screenshot of the Salesforce Lease Management Tenant record detail page.

**Tenant: Aswini**

**Details:**

* Tenant Name: Aswini	Owner: Sowmya Team
* Email: aswiniamaraadi15@gmail.com	
Phone: (905) 222-5567	
status: Leaving	
property: Imran	
Created By: Sowmya Team, 6/26/2025, 6:05 AM	Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

**Activity:**

- New Case
- Edit
- New Opportunity

**Upcoming & Overdue:**

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

**Recent Activity:**

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

**Actions:**

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

Screenshot of a Salesforce Lightning interface showing a tenant record for "Aswini".

The top navigation bar includes tabs for "Lease Management", "Payment", "Tenants", "property", and "lease". A search bar is at the top right.

The main content area shows the following details:

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

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

A green banner at the top right says "Tenant was submitted for approval." with a checkmark icon.

Buttons at the bottom are "Cancel" and "Save".

The right sidebar shows the "Activity" section with no recent activity.

The bottom status bar shows the weather as 30°C Cloudy, system notifications, and the date/time as 27-06-2025 12:46.

Screenshot of a Salesforce Lightning interface showing a process instance step for "Tenant Approval".

The top navigation bar includes tabs for "Lease Management", "Payment", "Tenants", "property", "lease", and "Aswini Approval".

The main content area shows the following details:

Submitter	Date Submitted	Actual Approver	Assigned To
Sowmya Team	Jun 27, 2025	Sowmya Team	Sowmya Team

A green banner at the top right says "Tenant Approval Approved" with a checkmark icon.

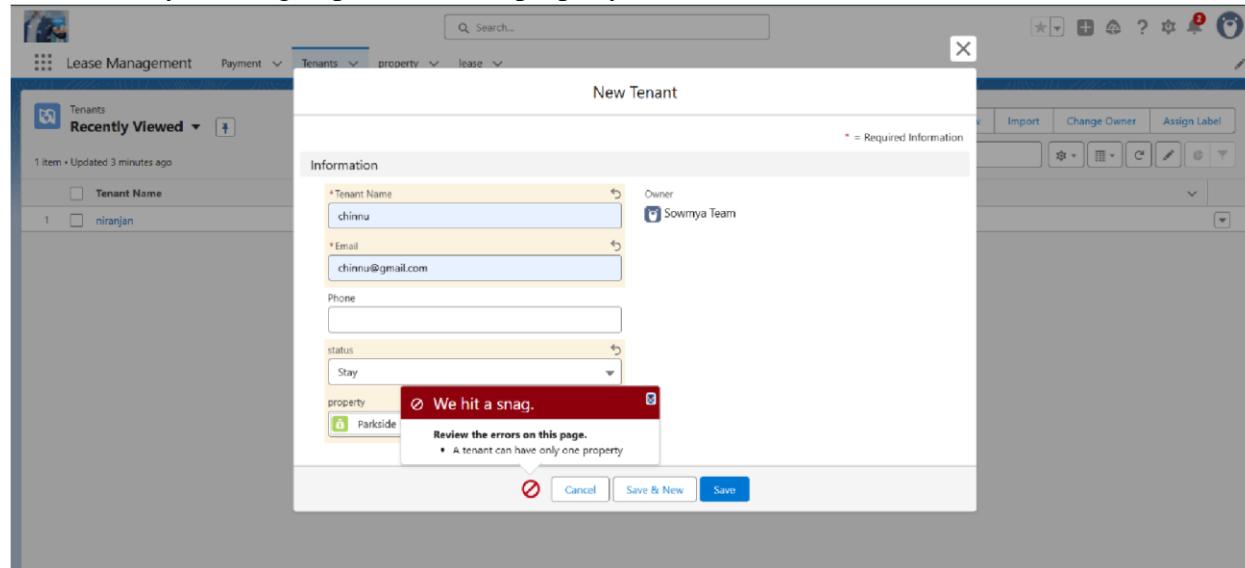
The right sidebar shows a "Notifications" list with the following entries:

- Approval request for the tenant is approved Aswini a few seconds ago
- Approval request for the tenant is rejected Aswini an hour ago
- Approval request for the tenant is approved Aswini an hour ago
- Approval request for the tenant is approved Aswini an hour ago
- Approval request for the tenant is approved Kiran 19 hours ago

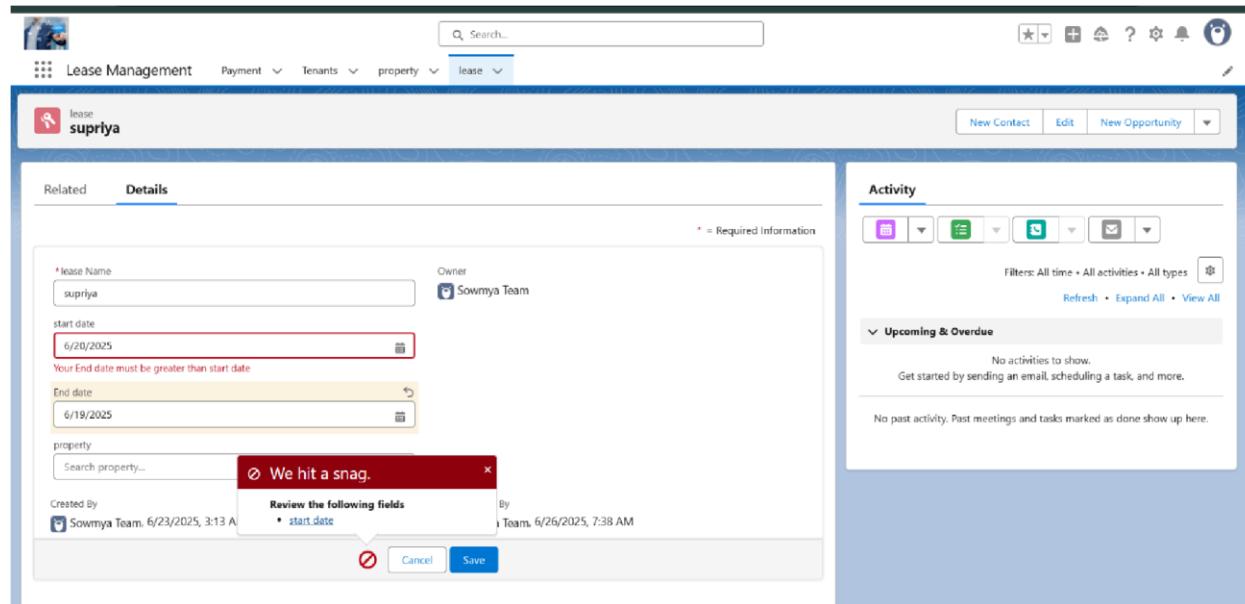
# FUNCTIONAL AND PERFORMANCE TESTING

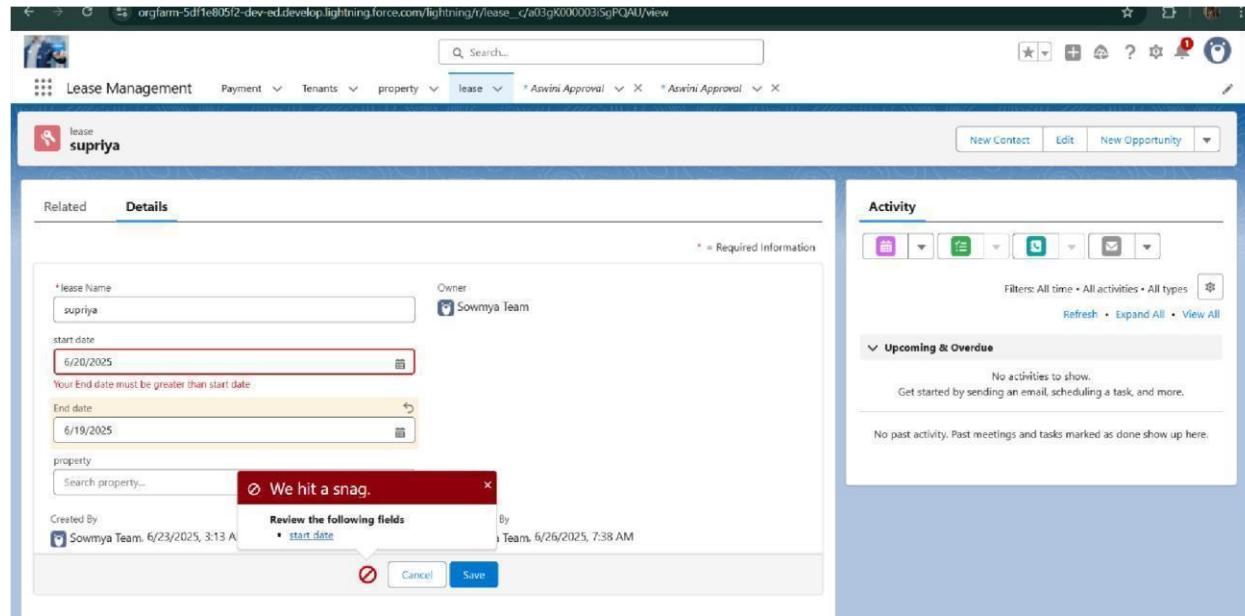
## Performance Testing

- Trigger validation by entering duplicate tenant-property records

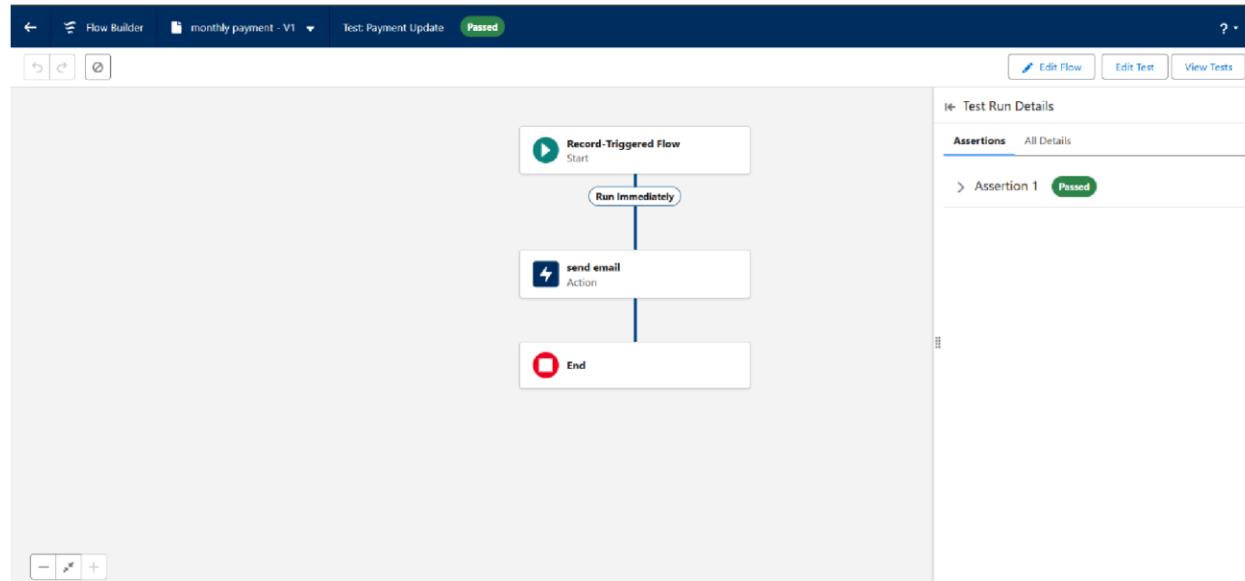


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Lease Management Payment Tenants property lease \* nirajan Approval

Tenant nirajan

Related Details

\* = Required Information

Tenant Name	nirajan	Owner	Sowmya Team
Email	niranjan1506@gmail.com		
Phone			
Status	Stay		
Property	Parkade Lofts		
Created By	Sowmya Team	Last Modified By	Sowmya Team
Jun 23, 2025, 2:33 AM			
Jun 23, 2025, 3:58 AM			
<input type="button" value="Cancel"/> <input type="button" value="Save"/>			

Notifications

Approval request for the tenant is approved nirajan  
a few seconds ago

Approval request for the tenant is rejected nirajan  
Jun 23, 2025, 4:29 PM

Approval request for the tenant is approved nirajan  
Jun 23, 2025, 4:25 PM

Approval request for the tenant is approved nirajan  
Jun 23, 2025, 4:14 PM

New Guidance Center learning resource available  
Define Your Sales Process  
Learn how to guide reps through the sales process.  
Jun 20, 2025, 1:20 PM

Lease Management Payment Tenants property lease

Tenant nirajan

**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
Jack	
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

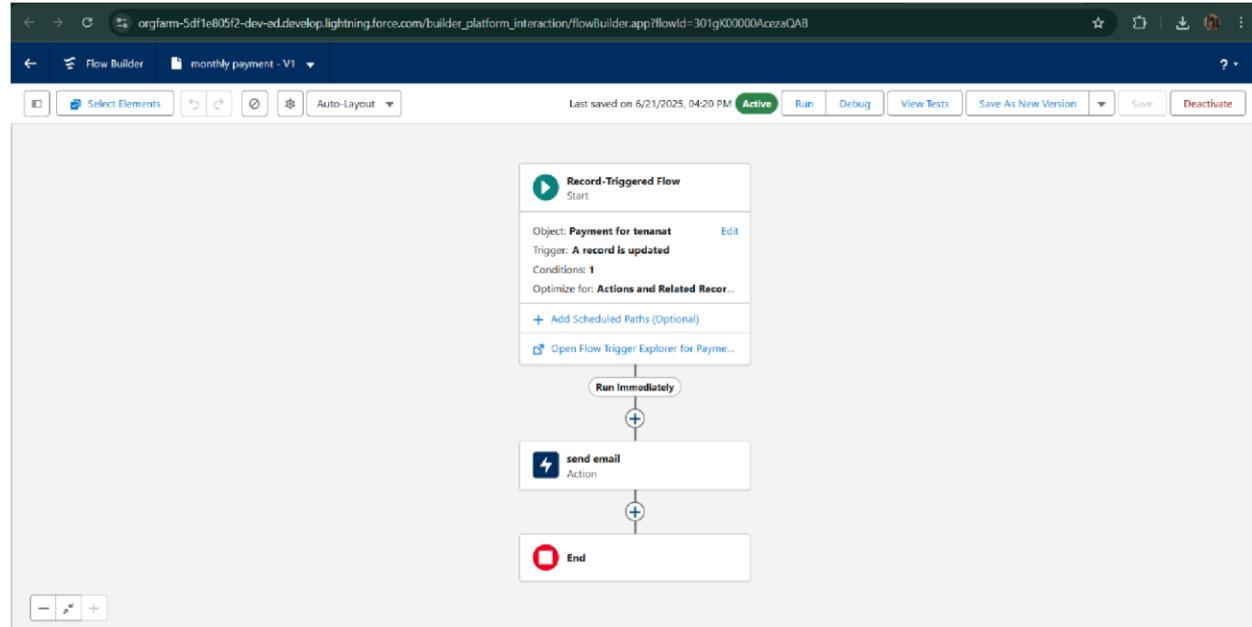
The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. It displays a table of 'Custom Object Tabs' with columns for Action, Label, Tab Style, and Description. The tabs listed are:

Action	Label	Tab Style	Description
Edit   Del	lease	Keys	
Edit   Del	Payment	Credit Card	
Edit   Del	property	Sack	
Edit   Del	Tenants	Map	

- Email alerts

The screenshot shows the Salesforce Lease Management interface under the 'Approval History' section for tenant 'niranjan'. It displays a table of approval steps with columns for Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The steps listed are:

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

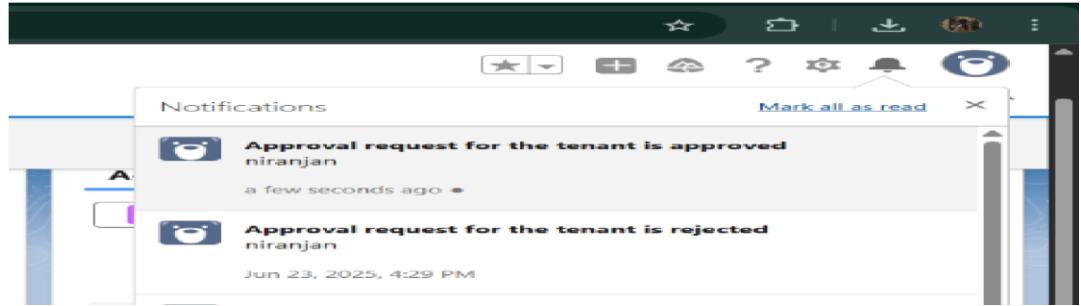


- Trigger error messages

The screenshot shows the Lease Management application with the following details:

- Page:** New Tenant
- Fields:** Tenant Name (chinnu), Email (chinnu@gmail.com), Phone, status (Stay), property (Parkside)
- Error Message:** "We hit a snag.  
Review the errors on this page.  
\* A tenant can have only one property"
- Buttons:** Cancel, Save & New, Save

- Approval process notifications



## ADVANTAGES & DISADVANTAGES

### Advantages

#### 1. Centralized Information Management

- All lease agreements, documents, and related data are stored in one place.
- Reduces reliance on paper records or scattered spreadsheets.

#### 2. Time and Cost Efficiency

- Automates repetitive tasks such as payment reminders, rent calculations, or generating reports. Reduces administrative overhead.

## **Disadvantages**

### **1. High Initial Investment □**

- Development or licensing costs for software can be significant.
- Training staff and migrating legacy data add to the initial expense.

### **2. Complex Implementation □**

- Complex lease terms and regional compliance laws may require customization.
- Integration with other systems (e.g., accounting, CRM, ERP) can be challenging.

---

## **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)) { newTenantaddError('A  
tenant can have only one property');  
}  
}
```

```
}
```

### **MothlyEmailScheduler.apxc:**

```
global class MonthlyEmailScheduler implements Schedulable { global void
execute(SchedulableContext sc) { Integer currentDay =
Date.today().day(); if (currentDay == 1) { sendMonthlyEmails();
}
} public static void sendMonthlyEmails()
{ List<Tenant__c> tenants
= [SELECT Id, Email__c FROM Tenant__c];
for (Tenant__c tenant :
tenants) {
String recipientEmail = tenant.Email__c;
String emailContent = 'I trust this email finds you well. I am writing to remind you
that the monthly rent is due. Your timely payment ensures the smooth functioning of our
rental arrangement and helps maintain a positive living environment for all.';
String emailSubject = 'Reminder: Monthly Rent Payment Due';
Messaging.SingleEmailMessage email = new
Messaging.SingleEmailMessage(); email.setToAddresses(new
String[]{recipientEmail}); email.setSubject(emailSubject);
email.setPlainTextBody(emailContent);
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
}
}
```

