

# **LEASE MANAGEMENT**

**Collage Name:** Sri Krishna Adithya Collage of Arts and Science

**Team ID:** NM2025TMID22482

## **Team Members**

**Team Leader Name:** Janashri P

**Email:** [23bscs218janashrip@skacas.ac.in](mailto:23bscs218janashrip@skacas.ac.in)

**Team Member1:** Inbarasu K

**Email:** [23bscs217inbarasuk@skacas.ac.in](mailto:23bscs217inbarasuk@skacas.ac.in)

**Team Member2:** Jeevanantham R

**Email:** [23bscs219jeevananthamr@skacas.ac.in](mailto:23bscs219jeevananthamr@skacas.ac.in)

**Team Member3:** Krishnapriya K

**Email:** [23bscs221krishnapriyak@skacas.ac.in](mailto:23bscs221krishnapriyak@skacas.ac.in)

## 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, contracts, payments, and communication with automation features such as flows, approval processes and email alerts.



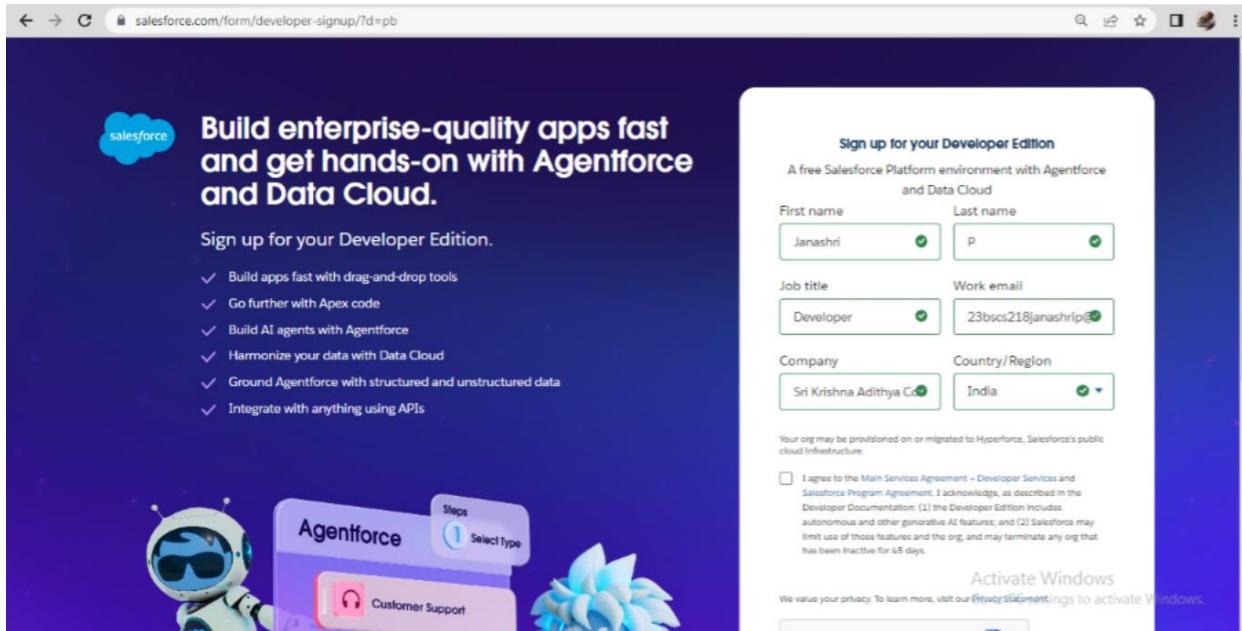
### 1.2 PURPOSE:

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.

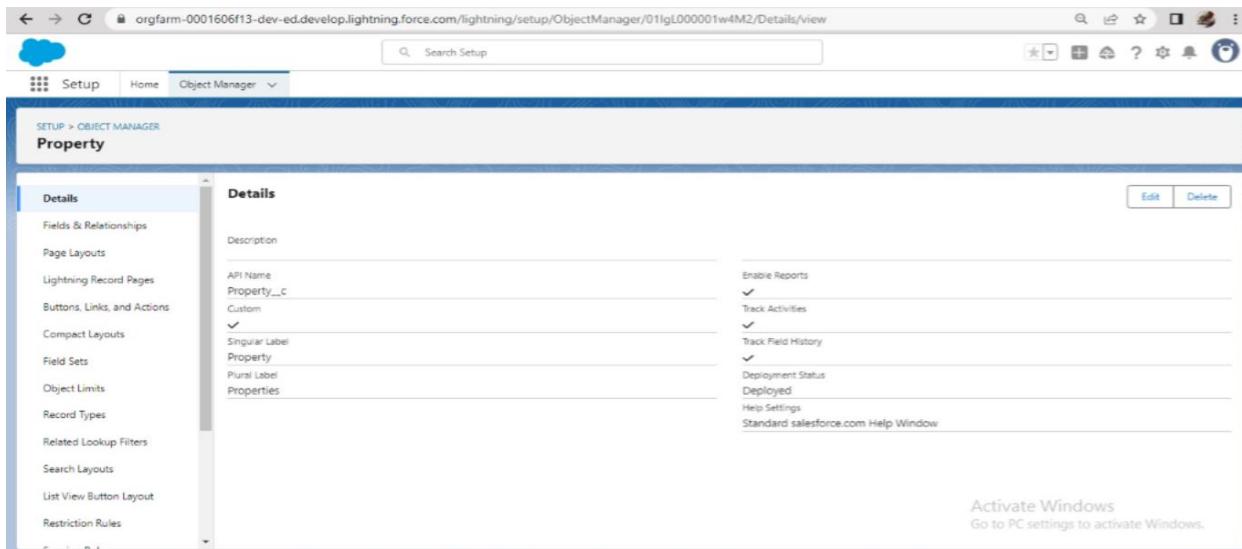
## 2. DEVELOPMENT PHASE:

### Creating Developer Account:

- By using this URL - <https://developer.salesforce.com/signup>



- Created objects: Property, Tenant, Lease, Payment



[orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wDCP/Details/view](http://orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wDCP/Details/view)

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes a cloud icon, a search bar labeled "Search Setup", and various global buttons. The main header says "SETUP > OBJECT MANAGER" and the specific object name is "Lease". On the left, a sidebar lists various configuration tabs: 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 content area is titled "Details" and contains two sections: "Description" and "Enable Reports". The "Description" section includes fields for API Name ("Lease\_\_c"), Singular Label ("Lease"), and Plural Label ("Leases"). The "Enable Reports" section includes checkboxes for "Track Activities", "Track Field History", and "Deployment Status", all of which are checked. At the bottom right of the main content area, there is a message: "Activate Windows" followed by "Go to PC settings to activate Windows." There are "Edit" and "Delete" buttons at the top right of the main content area.

[orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wDSx/Details/view](http://orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wDSx/Details/view)

The screenshot shows the Salesforce Object Manager interface, similar to the previous one but for the "Tenant" object. The top navigation bar, sidebar, and main header ("SETUP > OBJECT MANAGER" and "Tenant") are identical. The left sidebar lists the same configuration tabs as the first screenshot. The main content area is titled "Details" and contains two sections: "Description" and "Enable Reports". The "Description" section includes fields for API Name ("Tenant\_\_c"), Singular Label ("Tenant"), and Plural Label ("Tenants"). The "Enable Reports" section includes checkboxes for "Track Activities", "Track Field History", and "Deployment Status", all of which are checked. At the bottom right of the main content area, there is a message: "Activate Windows" followed by "Go to PC settings to activate Windows." There are "Edit" and "Delete" buttons at the top right of the main content area.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Payment for tenant'. On the left, a sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area is titled 'Details' and contains fields for 'Description', 'API Name' (Payment\_for\_tenant\_\_c), 'Custom' (✓), 'Singular Label' (Payment for tenant), 'Plural Label' (Payment for tenants), and 'Enable Reports' (✓). Other settings include 'Track Activities' (✓), 'Track Field History' (✓), 'Deployment Status' (Deployed), and 'Help Settings' (Standard salesforce.com Help Window). At the bottom right, there is a message: 'Activate Windows' and 'Go to PC settings to activate Windows.'

- Configured fields and relationships.

The screenshot shows the Salesforce Object Manager interface for the 'Property' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Property'. On the left, a sidebar lists various configuration options: 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 content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The fields listed are: Address (Address\_\_c, Long Text Area(32760)), Created By (CreatedById, Lookup(User)), Last Modified By (LastModifiedById, Lookup(User)), Name (Name\_\_c, Text(18)), Owner (OwnerId, Lookup(User;Group)), Property Name (Name, Text(80)), sfqt (sfqt\_\_c, Text(18)), and Type (Type\_\_c, Picklist). There are also checkboxes for 'Controlling Field' and 'Indexed' for each row. At the bottom right, there is a message: 'Activate Windows' and 'Go to PC settings to activate Windows.'

orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wDCP/FieldsAndRelationships/view

Setup Home Object Manager

Lease

Details Fields & Relationships

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		

Activate Windows  
Go to PC settings to activate Windows.

orgfarm-0001606f13-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001wD5x/FieldsAndRelationships/view

Setup Home Object Manager

Tenant

Details Fields & Relationships

Fields & Relationships 8 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		
Property	Property_c	Lookup(Property)		✓
status	status_c	Picklist		
Tenant Name	Name	Text(80)		✓

Activate Windows  
Go to PC settings to activate Windows.

Fields & Relationships				
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 for tenant Name	Name	Text(80)		✓
Property	Property_c	Lookup(Property)		✓
Tenant	Tenant_c	Lookup(Tenant)		

- Developed Lightning App with relevant tabs.

Lightning App Builder

App Settings

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

App Name: Lease Management

Developer Name: Lease\_Management

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

App Branding

Image:  Primary Color Hex Value: #0070D2

Org Theme Options:  Use the app's image and color instead of the org's custom theme

App Launcher Preview

LM Lease Management Application to efficiently handle the processes relat...

Activate Windows

The screenshot shows the Lightning App Builder interface with the URL <https://orgfarm-0001606f13-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02ugL000006FsRHQA0&retUrl=https%3A%2F%2Forgfarm-0001606f13-d...>. The page title is "Lightning App Builder".

**App Settings**

**Navigation Items**

**User Profiles**

**Available Items**

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

**Selected Items**

- Home
- Properties
- Payment for tenants
- Tenants
- Leases

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the Lightning App Builder interface with the URL <https://orgfarm-0001606f13-dev-ed.lightning.force.com/visualEditor/appBuilder.app?id=02ugL000006FsRHQA0&retUrl=https%3A%2F%2Forgfarm-0001606f13-d...>. The page title is "Lightning App Builder".

**App Settings**

**User Profiles**

**Available Profiles**

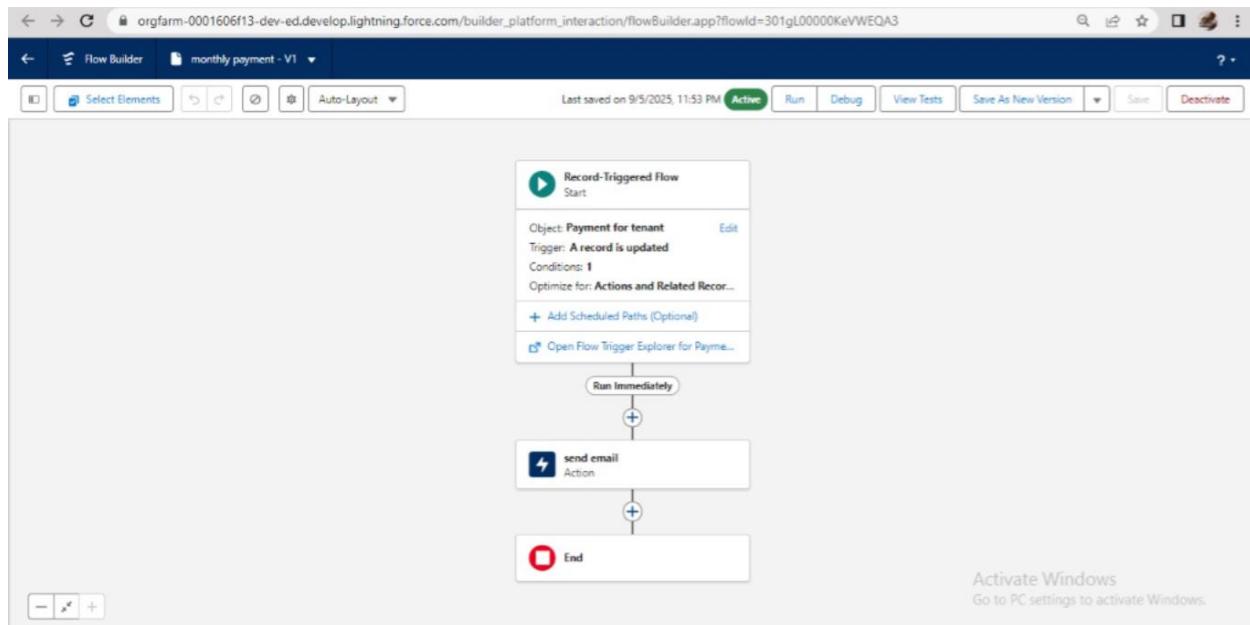
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

**Selected Profiles**

- System Administrator

Activate Windows  
Go to PC settings to activate Windows.

- Implemented Flows for monthly rent and payment success.



SETUP > OBJECT MANAGER Lease

Validation Rule Edit

Rule Name: lease\_end\_date

Active:

Description:

Error Condition Formula

Example: Discount\_Percent\_\_c>0.20 More Examples... Functions

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

Insert Field Insert Operator End\_date\_\_c > start\_date\_\_c

ABS ACOS ADDMONTHS AND ASCII ASIN

Insert Selected Function ABS(number) Returns the absolute value of a number, a number without its sign Help on this function

Check Syntax

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the Salesforce Setup interface under 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 a 'Lease Validation Rule' record with the following details:

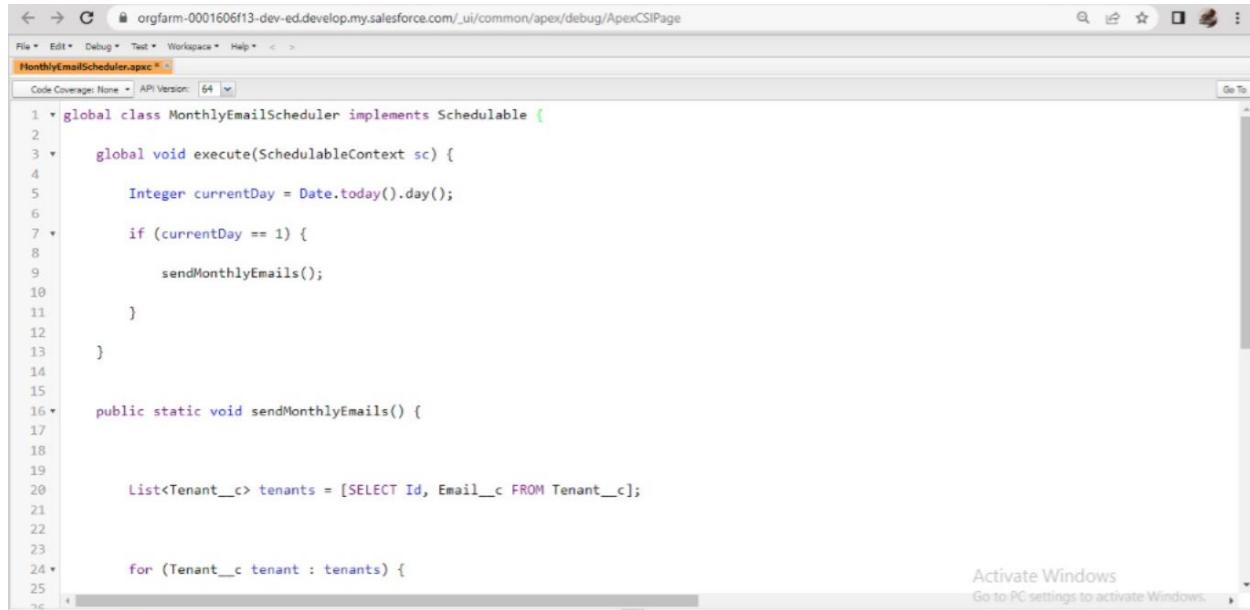
Validation Rule Detail	
Rule Name	lease_end_date
Error Condition Formula	End_date_c > start_date_c
Error Message	Your End date must be greater than start date
Description	
Created By	Janashni P 9/4/2025, 8:56 AM
Modified By	Janashni P 9/4/2025, 8:56 AM

The status is Active (indicated by a checkmark). The right side of the page includes a 'Help for this Page' link and an 'Activate Windows' message.

- Added Apex trigger to restrict multiple tenants per property.

The screenshot shows the Salesforce Lease Management interface. A modal window titled 'New Tenant' is open, displaying an 'Information' section with fields for Tenant Name (priya), Email (priya@gmail.com), Phone (1234567890), and Status (Stay). The Property field is set to Hyd. A red error message box appears, stating: 'We hit a snag.' and 'Review the errors on this page.' with the note: 'A tenant can have only one property.' At the bottom of the modal are 'Cancel', 'Save & New', and 'Save' buttons.

- Scheduled monthly reminder emails using Apex class.



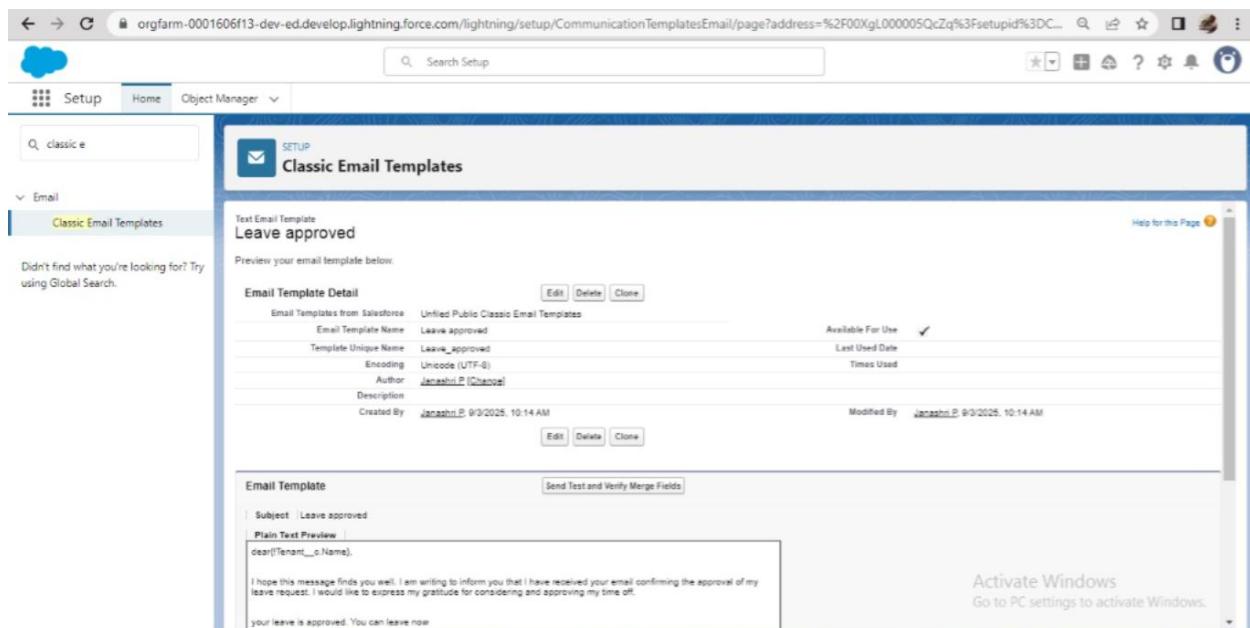
```

1 * global class MonthlyEmailScheduler implements Scheduleable {
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
19     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
20
21
22
23     for (Tenant__c tenant : tenants) {
24
25

```

Activate Windows  
Go to PC settings to activate Windows.

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



**Classic Email Templates**

**Leave approved**

Preview your email template below:

Email Template Detail	
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Jacashit P [Chanc]
Description	
Created By	Jacashit P 9/3/2025, 10:14 AM
Modified By	Jacashit P 9/3/2025, 10:14 AM
<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Clone</a>	

**Email Template**

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
  
```

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the 'Classic Email Templates' page in Salesforce. The template name is 'tenant leaving'. The 'Email Template Detail' section shows the following information:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Jacinto P [Chavez]
Description	
Created By	Jacinto P 9/3/2025, 10:12 AM
Modified By	Jacinto P 9/3/2025, 10:12 AM

The 'Available For Use' checkbox is checked. Below this, the 'Email Template' section displays the email content:

Subject: request for approve the leave  
Plain Text Preview:  
Dear {Tenant\_\_c.CreatedBy},  
  
Please approve my leave

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the 'Classic Email Templates' page in Salesforce. The template name is 'Leave rejected'. The 'Email Template Detail' section shows the following information:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave_rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Jacinto P [Chavez]
Description	
Created By	Jacinto P 9/3/2025, 10:15 AM
Modified By	Jacinto P 9/3/2025, 10:15 AM

The 'Available For Use' checkbox is checked. Below this, the 'Email Template' section displays the email content:

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

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the 'Classic Email Templates' page in Salesforce. The template is named 'Tenant Email'. The 'Email Template Detail' section shows the following information:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Jasash_P [Chennai]
Description	
Created By	Jasash_P 9/3/2025, 10:18 AM
Modified By	Jasash_P 9/3/2025, 10:18 AM

The 'Available For Use' checkbox is checked. Below this, the 'Email Template' section shows the email content:

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, which is currently outstanding. As outlined in

Activate Windows  
Go to PC settings to activate Windows.

The screenshot shows the 'Classic Email Templates' page in Salesforce. The template is named 'tenant payment'. The 'Email Template Detail' section shows the following information:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	Tenant_payment
Encoding	Unicode (UTF-8)
Author	Jasash_P [Chennai]
Description	
Created By	Jasash_P 9/3/2025, 10:17 AM
Modified By	Jasash_P 9/3/2025, 10:17 AM

The 'Available For Use' checkbox is checked. Below this, the 'Email Template' section shows the email 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.

Activate Windows  
Go to PC settings to activate Windows.

- Approval process creation.  
For tenant leaving

The screenshot shows the 'Approval Processes' setup page for a process named 'TenantApproval'. The process is set to 'Active'. It has one approval step named 'step1' assigned to 'User:janeeshi\_P'. There is one initial submission action: 'Record Lock'.

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit   Del	1	step1		Tenant: status EQUALS Stay	User:janeeshi_P	Final Rejection

For check for vacant

The screenshot shows the 'Approval Processes' setup page for a process named 'check\_for\_vacant'. The process is set to 'Active'. It has one approval step named 'step1' assigned to 'User:janeeshi\_P'. There is one initial submission action: 'Email Alert'.

Action	Type	Description	Criteria	Assigned Approver	Reject Behavior
Edit   Remove	Email Alert	08888899999.MV_ISBIS		User:janeeshi_P	Final Rejection

- Apex trigger

### Create an Apex Trigger

```

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

Entity Type	Entities	Related
Entry Type		
Classes		
Triggers	test	← Tenant__c SObject References ← testHandler ApexClass References
Pages		
Page Components		
Objects		
Static Resources		

Open  Filter  Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Activate Windows Go to PC settings to activate Windows.

```

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

Entity Type	Entities	Related
Entry Type		
Classes		
Triggers	test	← Tenant__c SObject References ← testHandler ApexClass References
Pages		
Page Components		
Objects		
Static Resources		

Open  Filter  Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Size

Activate Windows Go to PC settings to activate Windows.

Filter Click here to filter the log list

## Create an apex handler class.

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

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

## • FLOWS

The screenshot shows the Flow Builder interface with a record-triggered flow named "monthly payment - V1". The flow starts with a "Record-Triggered Flow Start" step, which triggers on "A record is updated" for the "Payment for tenant" object. The flow then branches to either "Run Immediately" or "send email" (Action). The "send email" step is connected to an "End" step.

**Configure Start**

**Select Object**  
Select the object whose records trigger the flow when they're created, updated, or deleted.  
Object: Payment for tenant

**Configure Trigger**  
Trigger the Flow When:  
 A record is created  
 A record is updated  
 A record is created or updated  
 A record is deleted

**Set Entry Conditions**  
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.  
Condition Requirements: All Conditions Are Met (AND)

**Activate Windows**  
Go to PC settings to activate Windows.

The screenshot shows the same record-triggered flow "monthly payment - V1". The "Configure Start" section now includes an entry condition: "Field: check for payment Operator: Equals Value: A\_Paid X".

**When to Run the Flow for Updated Records**  
 Every time a record is updated and meets the condition requirements  
 Only when a record is updated to meet the condition requirements

**Optimize Flow**  
Optimize the Flow for:  
**Fast Field Updates**  
Update fields on the record that triggers the flow to run. This high-performance flow runs **before** the record is saved to the database.

**Actions and Related Records**  
Update any record and perform actions, like send an email. This more flexible flow runs **after** the record is **saved** to the database.

**Is this flow making an external callout or connecting to an external system?**  
An asynchronous path is required for flows that involve external systems.

**Activate Windows**  
Go to PC settings to activate Windows.

- SCHEDULE CLASS

## Create an Apex class

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

MonthlyEmailScheduler.apxc ▾

Code Coverage: None ▾ API Version: 64 ▾ Go To

```
33     Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
34
35     email.setToAddresses(new String[]{recipientEmail});
36
37     email.setSubject(emailSubject);
38
39     email.setPlainTextBody(emailBody);
40
41     Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
42
43 }
```

An entity relationship diagram (ERD) is overlaid on the code editor. The diagram shows relationships between entities:

Entity Type	Entities	Relationships
Entity Type	Name Namespace	Name Extent Direction
Classes	testHandler	CronTrigger Referenced
Triggers	MonthlyEmailScheduler	Email CustomField References
Pages		← Tenant__c SObject References
Page Components		← Tenant__c SObject References
Objects		
Static Resources		
Packages		

Open Filter Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

User Application Operation Time Status Read Sis

Activate Windows  
Go to PC settings to activate Windows.



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

- Title Bar:** orgfarm-0001606f13-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage
- File Menu:** File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
- Toolbar:** Code Coverage: None API Version: 64 ▾ Go To
- Code Area:** The code is for a class named `MonthlyEmailScheduler` which implements `Schedulable`. It contains a `execute` method that checks if the current day is 1, and if so, calls `sendMonthlyEmails`. The `sendMonthlyEmails` method queries for `Tenant__c` records and loops through them to perform some action.

```
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) {
    }
```

## Schedule apex class

The screenshot shows the Salesforce Setup interface under the Apex Classes section. The class `MonthlyEmailScheduler` is selected. The code body contains the following Apex 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    }
11 }
```

The class is active and has 0% code coverage. A tooltip for the Windows activation bar says "Activate Windows Go to PC settings to activate Windows."

The screenshot shows the Salesforce Lease Management interface for a Tenant record named `Sam`. The Details tab is selected. The record contains the following fields:

- Tenant Name: Sam
- Email: sdfghjk@gmail.com
- Phone: 1234567890
- Status: Leaving
- Property: jhgf

The Activity sidebar shows no upcoming or overdue activities. A context menu is open over the sidebar, with the "Submit for Approval" option highlighted.

Lease Management Home Properties Payment for tenants Tenants Leases \* Tenant\_c X

Tenant Sam

Tenant was submitted for approval.

Related Details

Tenant Name: Sam Owner: Janashri P

Email: sdifghjk@gmail.com

Phone: (123) 456-7890

Status: Leaving

Property: jhgf

Created By: Janashri P. 9/12/2025, 9:44 AM

Last Modified By: Janashri P. 9/12/2025, 9:57 PM

Activity

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.

Activate Windows Go to PC settings to activate Windows.

Lease Management Home Properties Payment for tenants Tenants Leases \* Sem Approval X \* Tenant\_c X

Process Instance Step Tenant Approval Approved

Submitter: Janashri P Date Submitted: Sep 13, 2025 Actual Approver: Janashri P Assigned To: Janashri P

Details

Approval Details

Tenant Name: Sam Owner: Janashri P

Notifications

Approval request for the tenant is approved Sam a few seconds ago \*

Approval request for the tenant is rejected Anu 16 minutes ago \*

Approval request for the tenant is rejected Sam 11 hours ago \*

Approval request for the tenant is rejected Anu 11 hours ago \*

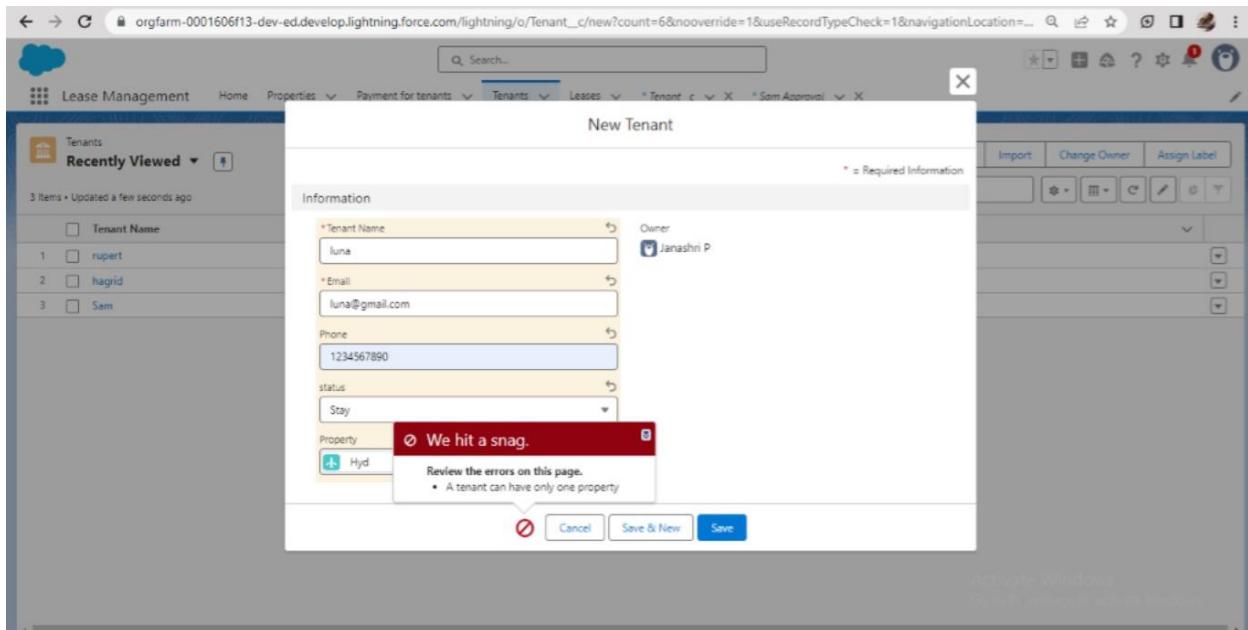
Approval request for the tenant is approved Anu Sep 6, 2025, 2:44 PM

Activate Windows Go to PC settings to activate Windows.

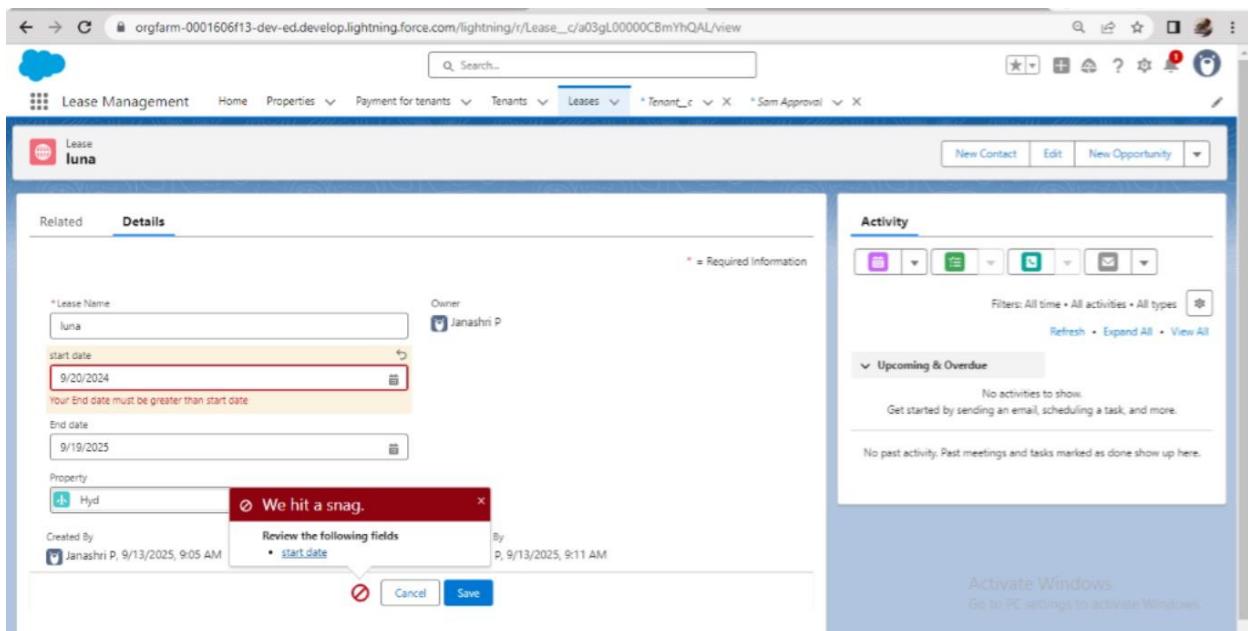
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

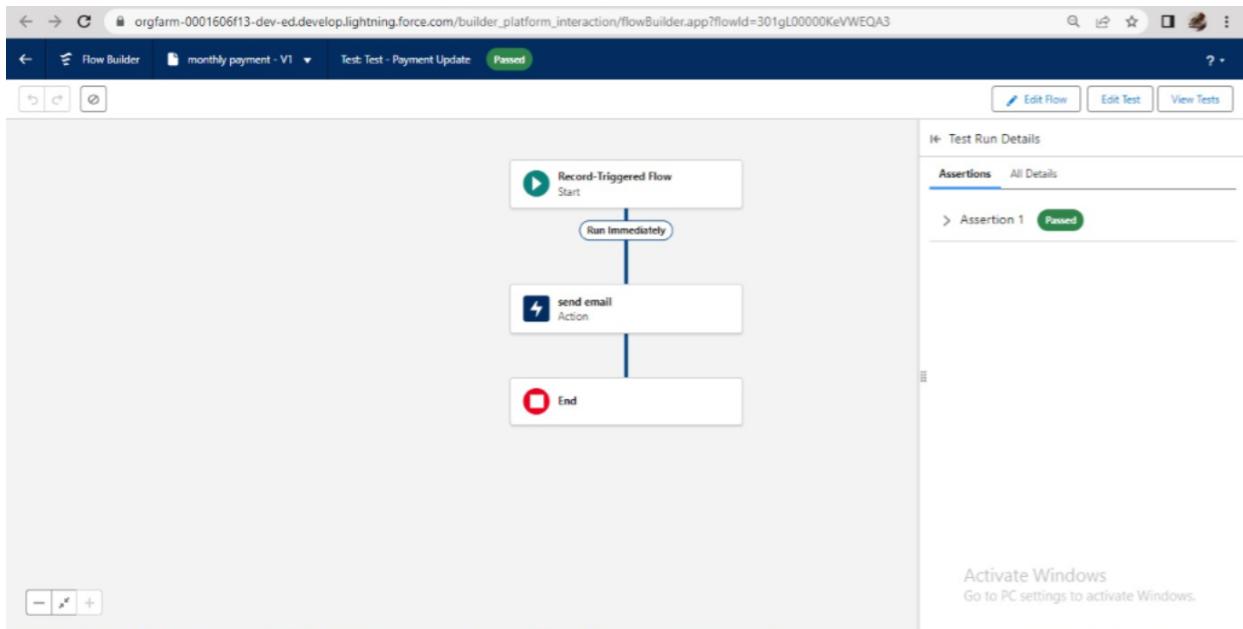
- Trigger validation by entering duplicate tenant-property records.



- Validation Rule checking.



- Test Flows on Payment update.



- Approval process validated through email alerts and status updates.

The screenshot shows the Salesforce Lightning interface for a Tenant record. The record details are as follows:

Related	Details
Tenant Name	hagrid
Email	hagrid@gmail.com
Phone	(123) 456-7890
status	Leaving
Property	Hyd
Created By	<a href="#">janashri P.</a> 9/4/2025, 10:29 AM
Last Modified By	<a href="#">janashri P.</a> 9/13/2025, 8:59 AM

A sidebar on the right shows a list of notifications:

- Approval request for the tenant is approved hagrid a few seconds ago \*
- Approval request for the tenant is approved Sam a day ago \*
- Approval request for the tenant is rejected rupert a day ago \*
- Approval request for the tenant is rejected Sam Sep 13, 2025, 10:30 AM \*
- Approval request for the tenant is rejected hagrid Sep 13, 2025, 10:27 AM \*

Lease Management Home Properties Payment for tenants Tenants Leases \* Tenant\_c X

**Related** Details

**Payments (0)** **New**

**Approval History (4)**

Step Name	Date	Status	Assigned To
step1	9/14/2025, 9:23 AM	Approved	Janashri P
Approval Request Submitted	9/14/2025, 8:03 AM	Submitted	Janashri P
step1	9/12/2025, 9:57 PM	Rejected	Janashri P
Approval Request Submitted	9/12/2025, 9:55 PM	Submitted	Janashri P

**Payment for tenants (0)** **New**

**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.

Activate Windows Go to PC settings to activate Windows.

## RESULTS

### Output Screenshots

- Tabs for property , Tenant, Lease , Payment.

Setup Home Object Manager

**Custom Tabs**

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

**Custom Object Tabs**

Action	Label	Tab Style	Description
Edit   Del	Leases	Globe	
Edit   Del	Payment for tenants	Hot Air Balloon	
Edit   Del	Expenses	Airplane	
Edit   Del	Tenant	Bank	

**Web Tabs**

No Web Tabs have been defined.

**Visualforce Tabs**

No Visualforce Tabs have been defined.

Activate Windows Go to PC settings to activate Windows.

Lease Management    Home    Properties    Payment for tenants    Tenants    Leases    Tenant\_c

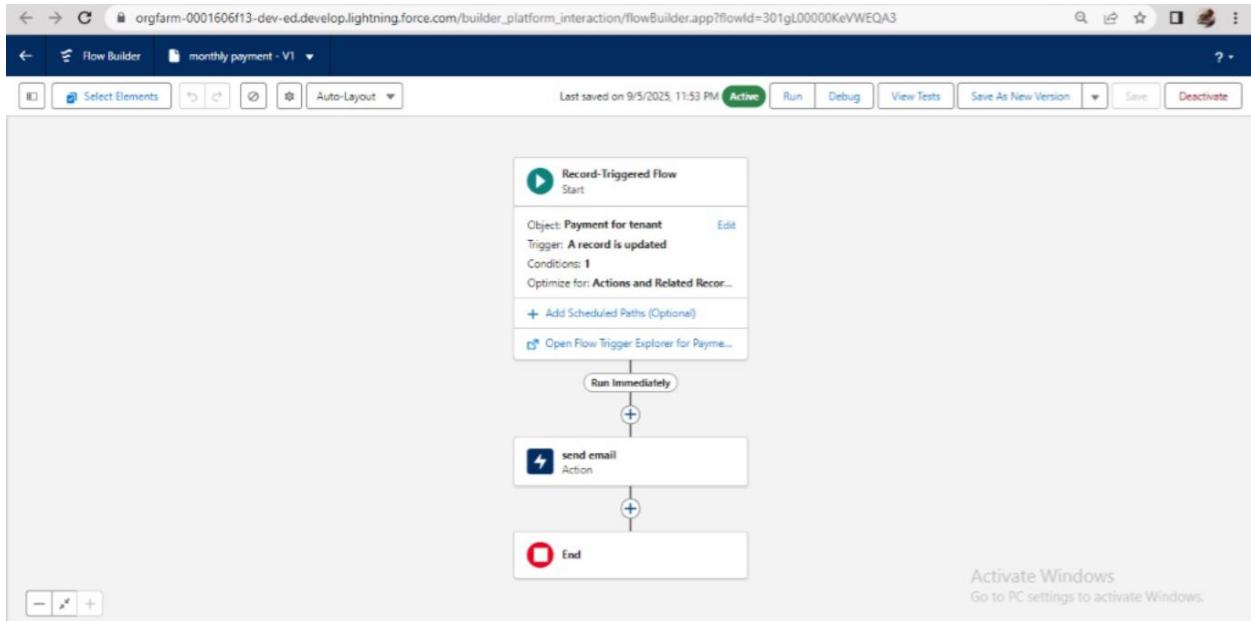
### Approval History

4 Items • Sorted by Is Pending • Updated 29 minutes ago

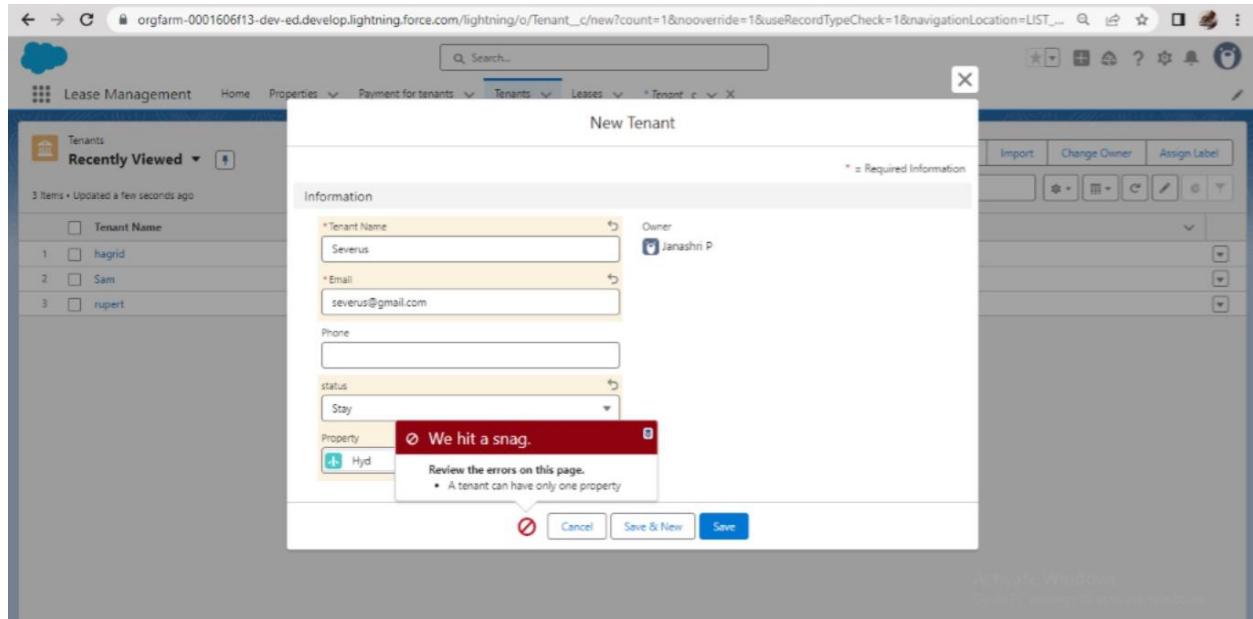
Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 step1	9/14/2025, 9:23 AM	Approved	Janashri P	Janashri P	Approved
2 Approval Request Submitted	9/14/2025, 8:03 AM	Submitted	Janashri P	Janashri P	Leaving
3 step1	9/12/2025, 9:57 PM	Rejected	Janashri P	Janashri P	Approval rejected.
4 Approval Request Submitted	9/12/2025, 9:55 PM	Submitted	Janashri P	Janashri P	Leaving

Activate Windows  
Go to PC settings to activate Windows.

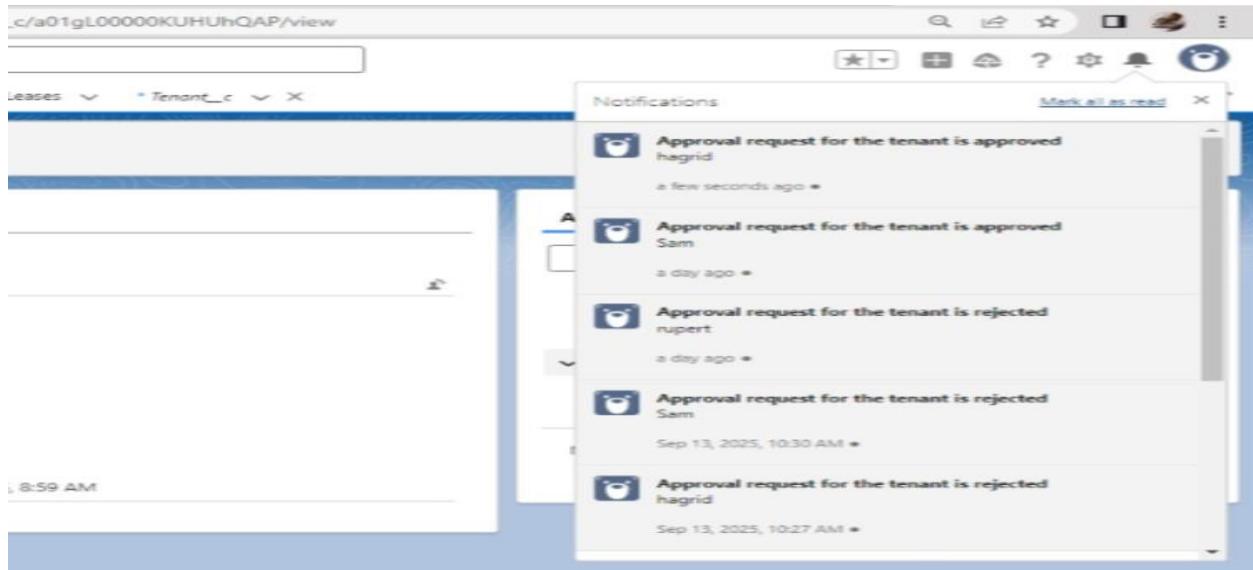
- Flow runs.



- Trigger error messages.



- Approval process notification.



## **ADVANTAGES**

1. Centralized Data Management– Stores all property, tenant, and lease information in one Salesforce platform, making it easy to access and update.
2. Automation of Routine Tasks – Flows, triggers, and scheduled Apex classes reduce manual work such as sending reminders and approving requests.
3. Improved Accuracy & Compliance– Validation rules and approval processes help avoid duplicate tenants for a property and maintain correct records.
4. Real-Time Notifications– Email alerts keep tenants and admins informed about payments, approvals, and lease status changes.
5. Scalability– Easily adapts to manage multiple properties and a growing number of tenants as the organization expand.
6. Enhanced Communication– Automated email templates ensure consistent and timely communication between management and tenants.

## **DISADVANTAGES**

1. Initial Setup Complexity – Requires Salesforce knowledge to configure objects, flows, and Apex triggers correctly.
2. Cost of Salesforce Licensing – Organizations must pay for Salesforce subscriptions and may need higher editions for advanced features.
3. Dependency on Internet & Platform – System availability depends on Salesforce uptime and a stable internet connection.
4. Customization Limitations – Some unique business needs may require advanced Apex coding or third-party integrations.
5. Training Requirement – Staff must learn Salesforce navigation and basic administration, which can take time.

## 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');
            }
        }
    }
}
```

## **MonthlyEmailScheduler.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});  
        }  
    }  
}
```