

LEASE MANAGEMENT

College Name: Sri Krishna Adithya College of Arts and Science

College Code:

TEAM ID:

TEAM MEMBERS:

Team LeaderName: PRITIKA G

Email: 23bscs241pritikag@skacas.ac.in

Team Member 1: RENI S

Email: 23bscs242renis@skacas.ac.in

Team Member 2: PAVITHRAN VK

Email: 23bscs239pavithranvk@skacas.ac.in

Team Member 3: PRANEET M

Email: 23bscs240praneetm@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, lease contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



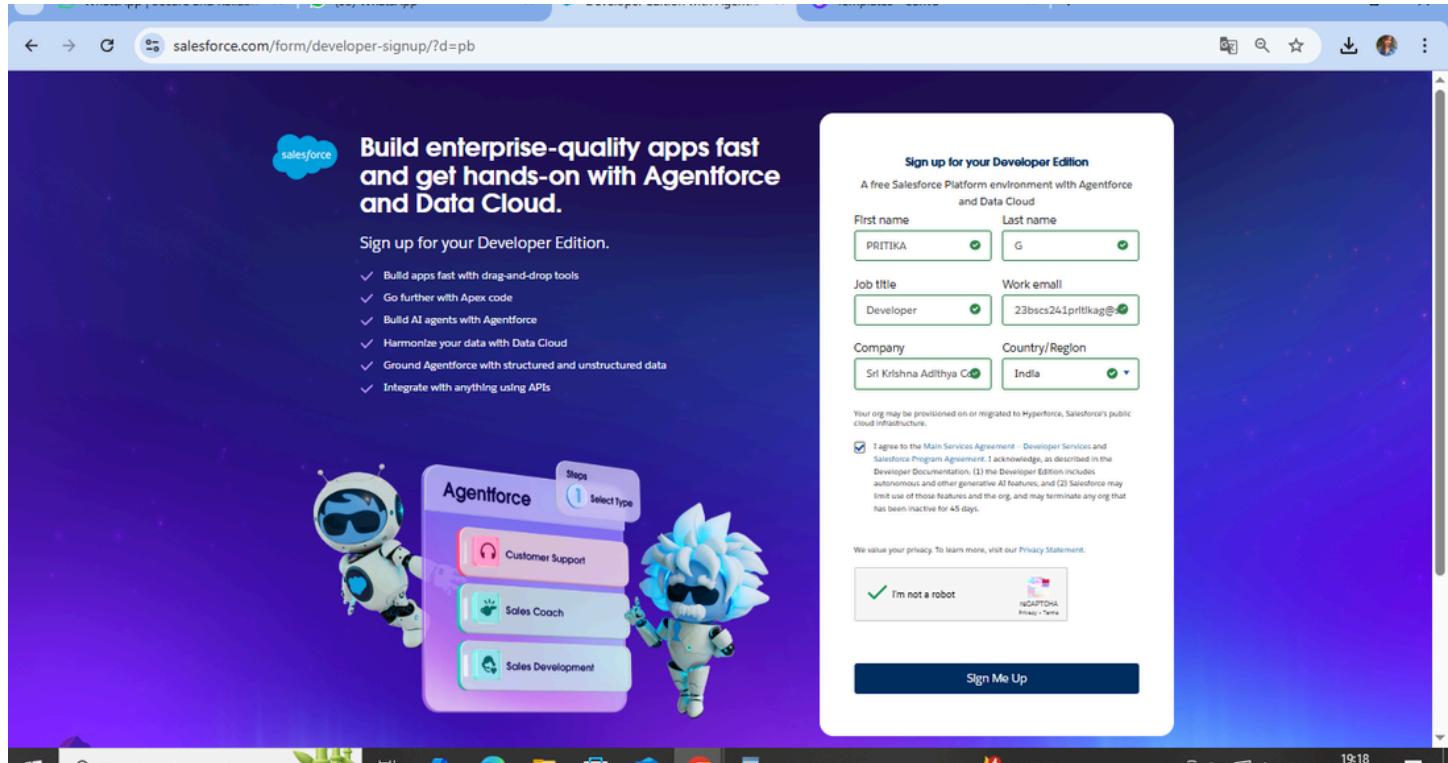
1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

DEVELOPMENT PHASE

Creating Developer Account:

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



- Created objects: Property, Tenant, Lease, Payment

The screenshot shows the 'Object Manager' section of the Salesforce setup. The left sidebar lists various setup categories like Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The main area displays the 'Property' object details. The 'Details' tab is selected, showing the API Name (Property__c), Custom status (✓), Singular Label (Property), and Plural Label (Property). On the right, there are checkboxes for enabling Reports, Track Activities, and Track Field History, all of which are checked. Deployment Status is set to 'Deployed'. Help Settings link to the 'Standard salesforce.com Help Window'. At the top right are 'Edit' and 'Delete' buttons.

WhatsApp | Secure and Reli... (53) WhatsApp FINAL REPORT - Lease Mana... Recently Viewed | Payment Tenant | Salesforce

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000272Lh/Details/view

Cloud Search Setup Object Manager

SETUP > OBJECT MANAGER Tenant

Details

Description

API Name: Tenant_c
Custom: ✓
Singular Label: Tenant
Plural Label: Tenants

Enable Reports: ✓
Track Activities: ✓
Track Field History: ✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts

Edit Delete

Type here to search Rain warning 09:40 17-09-2025

WhatsApp | Secure and Reli... (53) WhatsApp FINAL REPORT - Lease Mana... Recently Viewed | Payment lease | Salesforce

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000272QX/Details/view

Cloud Search Setup Object Manager

SETUP > OBJECT MANAGER lease

Details

Description

API Name: lease_c
Custom: ✓
Singular Label: lease
Plural Label: lease

Enable Reports: ✓
Track Activities: ✓
Track Field History: ✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts

Edit Delete

Type here to search Rain warning 09:41 17-09-2025

The screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various configuration options under 'Details'. The main area displays the 'Details' tab for the 'Payment for tenant' object. The API Name is set to 'Payment_for_tenant__c'. The singular label is 'Payment for tenant' and the plural label is 'Payment'. Under the 'Enable Reports' section, 'Track Activities' and 'Track Field History' are checked. Deployment status is marked as 'Deployed'.

Field	Value
API Name	Payment_for_tenant__c
Singular Label	Payment for tenant
Plural Label	Payment
Enable Reports	✓ Track Activities ✓ Track Field History
Deployment Status	Deployed

- Configured fields and relationships.

The screenshot shows the 'Fields & Relationships' section for the 'Property' object. It lists nine fields: Address, Created By, Last Modified By, Name, Owner, Property, Property Name, sfqt, and Type. The 'Owner' field is a lookup to 'User,Group'. The 'Property' field is a lookup to 'Property'. The 'Property Name' field is a text field. The 'sfqt' field is a text field. The 'Type' field is a picklist.

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		

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000272Nj/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenant

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

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000272QX/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	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		

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000272Lh/FieldsAndRelationships/view

SETUP > OBJECT MANAGER

Tenant

Fields & Relationships		8 Items, Sorted by Field Label			
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Details	Created By	CreatedById	Lookup(User)		
Fields & Relationships	Email	Email__c	Email		
Page Layouts	Last Modified By	LastModifiedById	Lookup(User)		
Lightning Record Pages	Owner	OwnerId	Lookup(User,Group)		✓
Buttons, Links, and Actions	Phone	Phone__c	Phone		
Compact Layouts	Property	Property__c	Lookup(Property)		✓
Field Sets	Status	Status__c	Picklist		
Object Limits	Tenant Name	Name	Text(80)		✓
Record Types					
Related Lookup Filters					
Search Layouts					
List View Button Layout					
Restriction Rules					

- Developed Lightning App with relevant tabs

orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02ugK000006QxlnQAC&retUrl=https%3A%2F%2Forgfarm-5e651cb... ? Help

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 Branding
* App Name	Image Primary Color Hex Value
Lease Management	#0070D2
* Developer Name	
Lease_Management	
Description	
A Lightning application to track and manage property lease agreements, tenants, and...	
Org Theme Options	
<input type="checkbox"/> Use the app's image and color instead of the org's custom theme	
App Launcher Preview	
<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Lease Management A Lightning application to track and manage property ... </div>	

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

App Settings

Navigation Items

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

Available Items

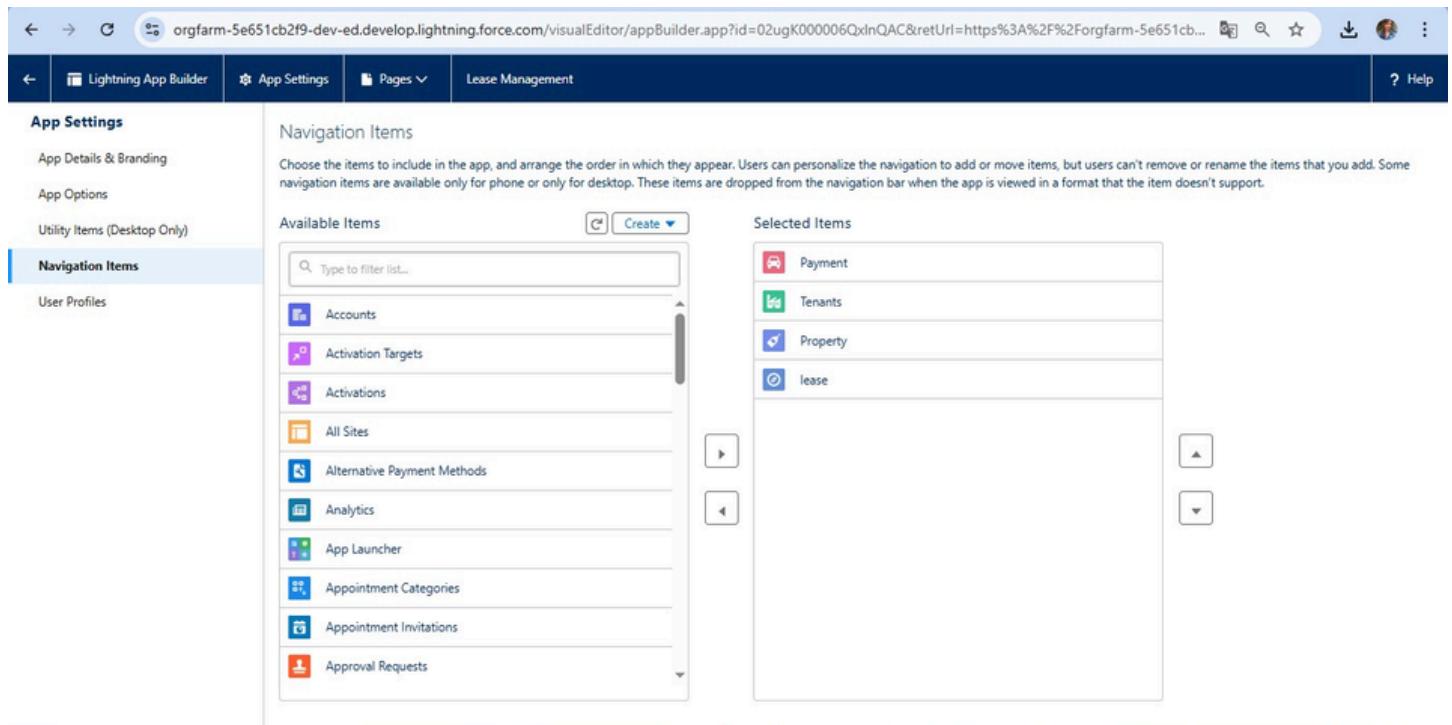
Type to filter list...

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

Selected Items

- Payment
- Tenants
- Property
- Lease

Up ▲ Down ▼



WhatsApp | WhatsApp | FINAL REPO | Recently Viewed | Lease Management | Tenant | Sales | Salesforce | Search results | + | - | X

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

App Settings

User Profiles

Choose the user profiles that can access this app.

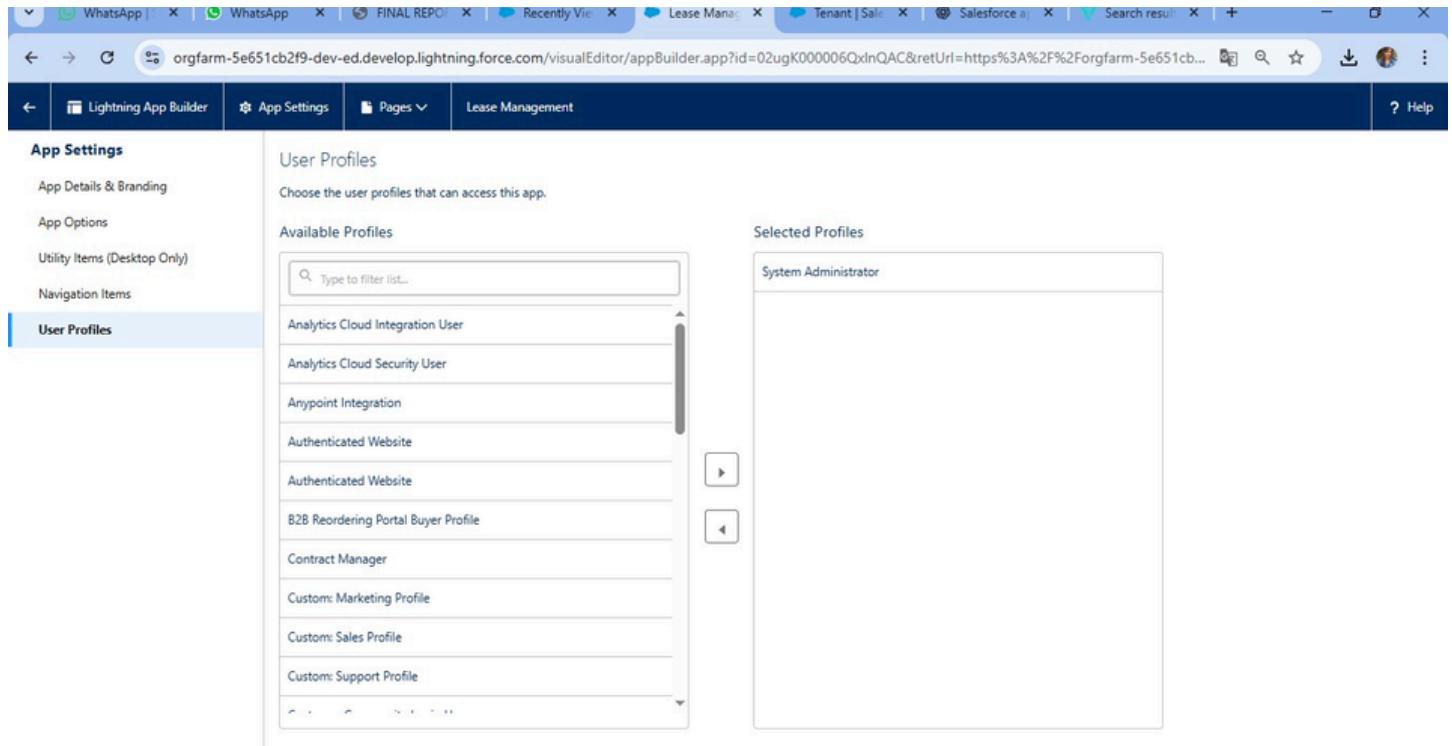
Available Profiles

Type to filter list...

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

Selected Profiles

- System Administrator



Lease Management Payment Tenants Property lease

Payment Recently Viewed

4 items • Sorted by Payment for tenant Name • Updated a few seconds ago

	Payment for tenant Name	Actions
1	Pavithran	[]
2	Praneet	[]
3	Priti	[]
4	Reni	[]

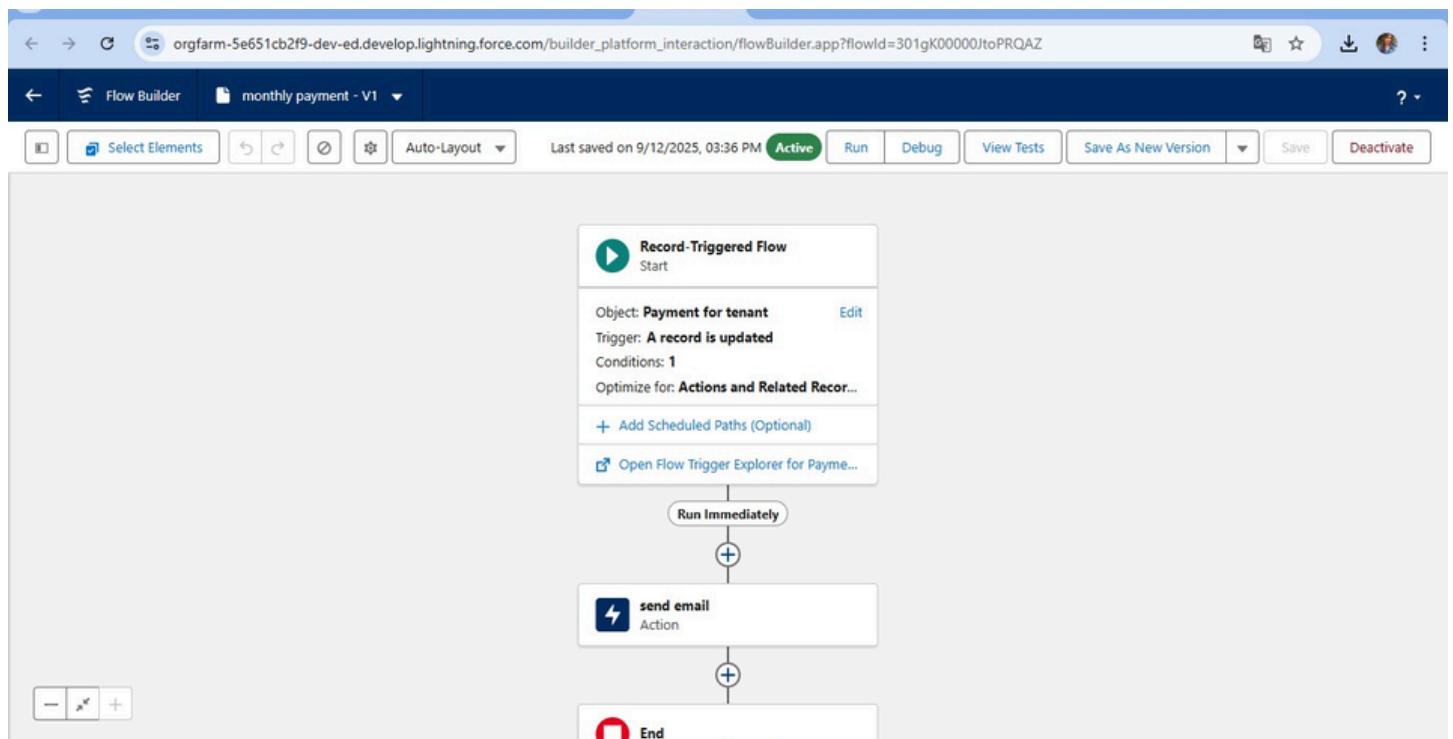
New Import Change Owner Assign Label

Search this list...

Search...

Print Export Copy Edit Delete

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

The screenshot shows the Salesforce Setup interface for creating a Validation Rule. The left sidebar lists various configuration options for the 'lease' object. The main area is titled 'lease Validation Rule' and contains the following fields:

- Validation Rule Edit:**
 - Rule Name: lease_end_date
 - Active: checked
 - Description: (empty)
- Error Condition Formula:**

Example: `Discount_Percent_c > 0.30` [More Examples...](#)

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

Formula input field: `End_date__c > start_date__c`

Function dropdown menu (open): ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN

Help text for ABS: Returns the absolute value of a number, a number without its sign.
- Error Message:**

Your End date must be greater than start date

The screenshot shows the details of the created Validation Rule for the 'lease' object. The left sidebar lists various configuration options for the 'lease' object. The main area is titled 'Validation Rule Detail' and displays the following information:

Field	Value	Action
Rule Name	lease_end_date	Active <input checked="" type="checkbox"/>
Error Condition Formula	<code>End_date__c > start_date__c</code>	
Error Message	Your End date must be greater than start date	Error Location <code>start date</code>
Description		
Created By	PRITIKA G. 9/10/2025, 9:49 PM	Modified By <code>PRITIKA G. 9/10/2025, 9:49 PM</code>

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

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

Classic Email Templates

Text Email Template: **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	PRITIKA G [Change]
Description	
Created By	PRITIKA G, 9/10/2025, 10:00 PM
Modified By	PRITIKA G, 9/10/2025, 10:00 PM

Email Template

Send Test and Verify Merge Fields

Subject: Leave approved

Plain Text Preview: dear {!Tenant__c.Name},

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

Classic Email Templates

Text Email Template: **tenant leaving**

Preview your email template below.

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	PRITIKA G [Change]
Description	
Created By	PRITIKA G, 9/10/2025, 9:55 PM
Modified By	PRITIKA G, 9/10/2025, 11:14 PM

Email Template

Send Test and Verify Merge Fields

Subject: request for approve the leave

Plain Text Preview: Dear {!Tenant__c.CreatedBy},

← → ⌂ orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK000006Akcp%3Fsetupid%3DComm... ⌂ ☆ + ? ⚙ 🔍 🔔 🔍

SETUP Classic Email Templates

Text Email Template **Leave rejected**

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	PRITIKA.G [Change]
Description	
Created By	PRITIKA.G, 9/10/2025, 10:05 PM
Modified By	PRITIKA.G, 9/10/2025, 10:05 PM

Available For Use Last Used Date Times Used

Email Template

Send Test and Verify Merge Fields

Subject | Leave rejected

Plain Text Preview |

Dear {!Tenant__c.Name},

← → ⌂ orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK000006AIAH%3Fsetupid%3DComm... ⌂ ☆ + ? ⚙ 🔍 🔔 🔍

SETUP Classic Email Templates

Text Email Template **Tenant Email**

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	PRITIKA.G [Change]
Description	
Created By	PRITIKA.G, 9/10/2025, 10:10 PM
Modified By	PRITIKA.G, 9/10/2025, 10:10 PM

Available For Use Last Used Date Times Used

Email Template

Send Test and Verify Merge Fields

Subject | Urgent: Monthly Rent Payment Reminder

Plain Text Preview |

Dear {!Tenant__c.Name},

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

- Page Header:** orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK000006AIYT%3Fsetupid%3DComm...
- Page Title:** SETUP Classic Email Templates
- Search Bar:** Search Setup
- Left Navigation:** Email (Classic Email Templates, Lightning Email Templates)
- Message Bar:** Didn't find what you're looking for? Try using Global Search.
- Table:** Email Template Detail (for tenant payment)

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	PRITIKA G [Change]
Description	
Created By	PRITIKA G, 9/10/2025, 10:12 PM
Modified By	PRITIKA G, 9/10/2025, 10:12 PM
- Section:** Email Template (Subject: Confirmation of Successful Monthly Payment, Plain Text Preview: Dear {Tenant__c Email__c},)
- Help:** Help for this Page

- Approval Process creation For Tenant Leaving:

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

- Page Header:** orgfarm-5e651cb2f9-dev-ed.develop.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04agK00000DWGb
- Page Title:** SETUP Approval Processes
- Search Bar:** Search Setup
- Left Navigation:** Data (Mass Transfer Approval Requests), Feature Settings (Approval Settings), Process Automation (Approval Processes)
- Message Bar:** Didn't find what you're looking for? Try using Global Search.
- Table:** Process Definition Detail (for Tenant Approval)

Process Name	Tenant Approval	Active
Unique Name	Tenant_Approval	Next Automated Approver Determined By
Description		
Entry Criteria		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template		
Initial Submitters	Tenant Owner	
Created By	PRITIKA G, 9/13/2025, 9:21 AM	Modified By PRITIKA G, 9/13/2025, 9:21 AM
- Section:** Initial Submission Actions (Add Existing, Add New)

Action	Type	Description
	Record Lock	Lock the record from being edited
- Help:** Help for this Page

- For Check for Vacant:

Approval Processes

Tenant: check for vacant

Process Definition Detail

Process Name	check for vacant	Active
Unique Name	check_for_vacant	Next Automated Approver Determined By
Description		
Entry Criteria	Tenant: Status NOT EQUAL TO Leaving	
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template		
Initial Submitters	Tenant Owner	
Created By	PRITIKA G.	9/10/2025, 10:42 PM
Modified By	PRITIKA G.	9/11/2025, 4:28 AM

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited

- Apex Trigger

Create an Apex Trigger

```

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

```

Open

Entity Type	Entities	Related
Entity Type	Name Namespace	Name Extent Direction
Classes		
Triggers	test	
Pages		
Page Components		
Objects		
Static Resources		
Packages		

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

MonthlyEmailScheduler.apxc test.apxc

Code Coverage: None API Version: 64 Go To

```

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

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size

Filter Click here to filter the log list

- Create an Apex Handler class

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

MonthlyEmailScheduler.apxc test.apxc testHandler.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(exist
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16             if (!newTenant.Property__c.equals(null)) {
17
18                 if (existingPropertyIds.contains(newTenant.Id)) {
19                     newTenant.Property__c = null;
20
21                 }
22
23             }
24
25         }
26
27     }
28
29 }
```

Open

Entity Type	Entities	Related		
Name	Namespace	Name	Extent	Direction
testHandler		test	ApexTrigger	Referenced
MonthlyEmailScheduler		Property	CustomField	References
		<- Tenant__c	SObject	References
		<- Tenant__c	SObject	References

Logs Tests Checkpoints Query Editor View State Progress Problems

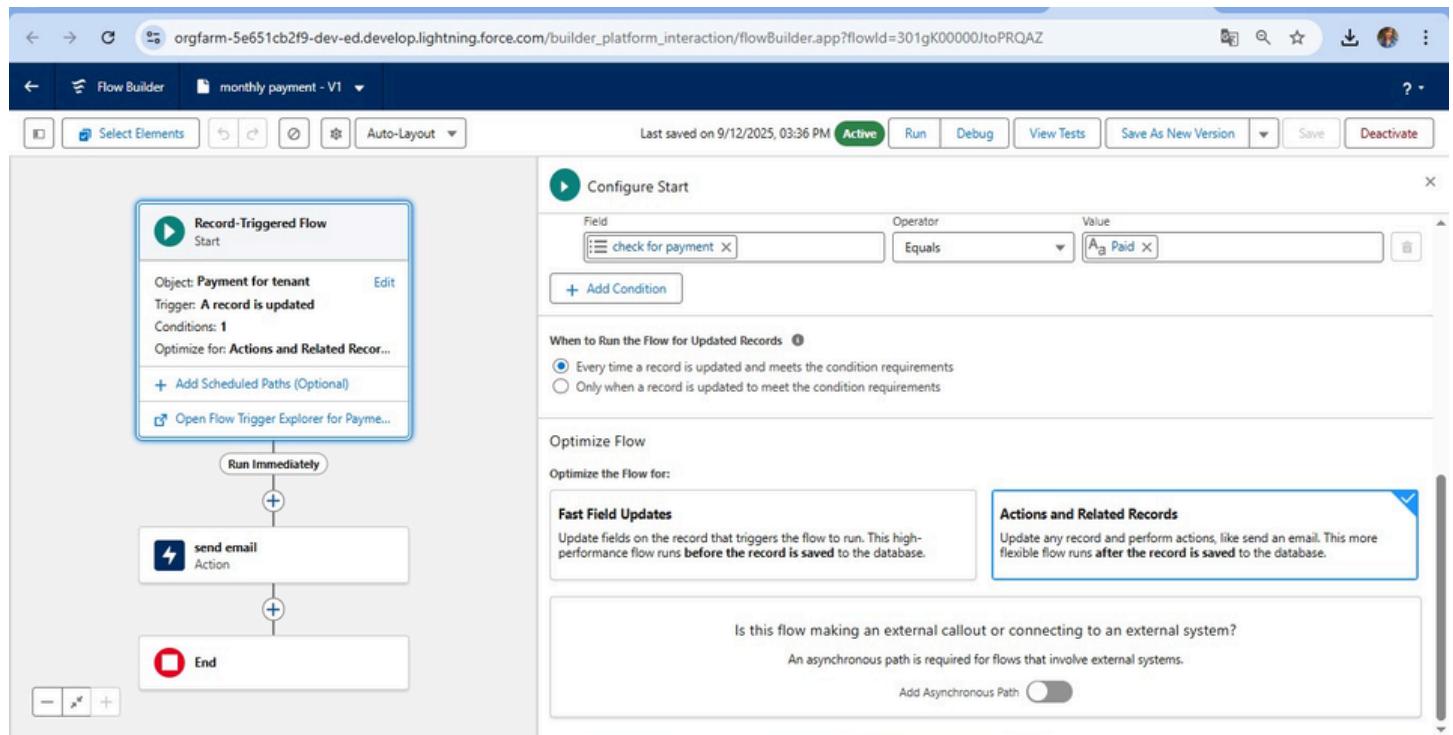
User	Application	Operation	Time	Status	Read	Size

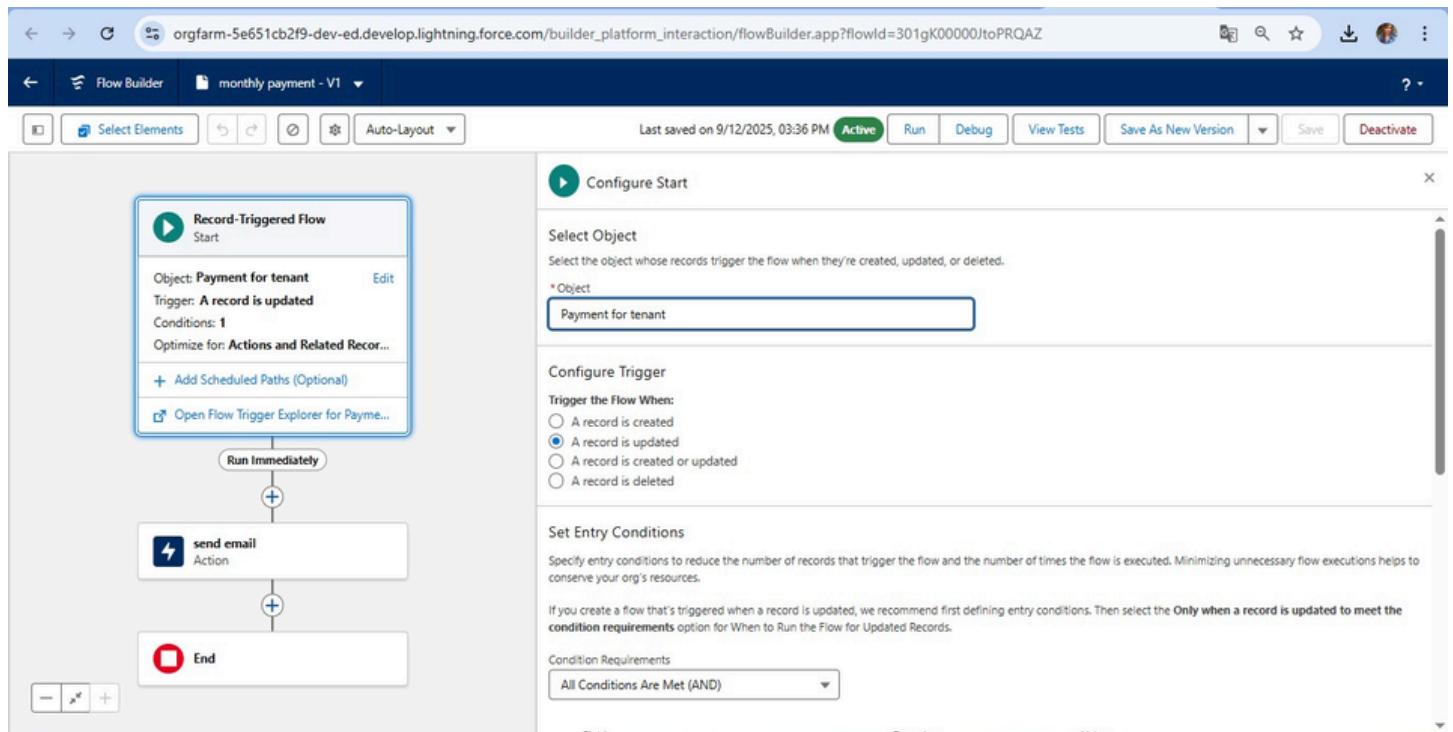
Filter Click here to filter the log list

The screenshot shows the Salesforce Apex code editor. The current file is 'testHandler.apxc'. The code implements a static method 'preventInsert' that checks if a new tenant record has a property record associated with it. If so, it prevents the insert.

```
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             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
17
18                 // Prevent insertion
19             }
20
21         }
22
23     }
24 }
```

• FLOWS





- Schedule class:

Create an Apex Class

```

1 global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8             sendMonthlyEmails();
9         }
10    }
11
12 }
13
14
15 public static void sendMonthlyEmails() {
16 }
```

Entity Type

Entity Type	Entities	Related
Classes	testHandler	Email CronTrigger CustomField... References
Triggers	MonthlyEmailScheduler	← Tenant__c SObject ← Tenant__c SObject
Pages		
Page Components		
Objects		
Static Resources		
Packages		

The screenshot shows the Salesforce IDE interface with the code editor open. The file is named `MonthlyEmailScheduler.apc`. The code implements a scheduled apex class:

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10
11        }
12
13    }
14
15
16    public static void sendMonthlyEmails() {
17
18
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.';
27
28            Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
29
30            message.setTo(recipientEmail);
31            message.setSubject('Monthly Rent Remind');
32            message.setPlainTextEmailBody(emailContent);
33
34            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{message});
35        }
36    }
37
38

```

The code defines a scheduled apex class `MonthlyEmailScheduler` that sends monthly emails to all tenants. It uses the `Schedulable` interface and the `Messaging` class to send emails.

- Schedule Apex class

The screenshot shows the Salesforce Setup page for Apex Classes. The class `MonthlyEmailScheduler` is selected. The details are as follows:

Apex Class Detail		Status	Active
Name	<code>MonthlyEmailScheduler</code>	Code Coverage	0% (0/15)
Namespace Prefix		Last Modified By	<code>PRITIKA.G</code> , 9/12/2025, 6:28 AM
Created By	<code>PRITIKA.G</code> , 9/12/2025, 6:28 AM		
<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Download"/> <input type="button" value="Security"/> <input type="button" value="Show Dependencies"/>			

The class body contains the same Apex code as shown in the previous screenshot:

```

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
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.';
27
28            Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
29
30            message.setTo(recipientEmail);
31            message.setSubject('Monthly Rent Remind');
32            message.setPlainTextEmailBody(emailContent);
33
34            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{message});
35        }
36    }
37
38

```

Lease Management

Tenant Pritika

Related Details

Tenant Name: Pritika

Email: 23bscs241pritikag@skacas.ac.in

Phone: (123) 456-7890

Status: Leaving

Property: Hyd

Created By: PRITIKA G, 9/13/2025, 6:50 AM

Last Modified By: PRITIKA G, 9/18/2025, 6:08 PM

Owner: PRITIKA G

Activity

New Case
New Lead
Delete
Clone
Change Owner
Printable View
Sharing
Sharing Hierarchy
Submit for Approval

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.

Lease Management

Tenant Pritika

Related Details

Tenant Name: Pritika

Email: 23bscs241pritikag@skacas.ac.in

Phone: (123) 456-7890

Status: Leaving

Property: Hyd

Created By: PRITIKA G, 9/13/2025, 6:50 AM

Last Modified By: PRITIKA G, 9/18/2025, 6:08 PM

Owner: PRITIKA G

Activity

Tenant was submitted for approval.

New Case
New Lead
Delete
Clone
Change Owner
Printable View
Sharing
Sharing Hierarchy
Submit for Approval

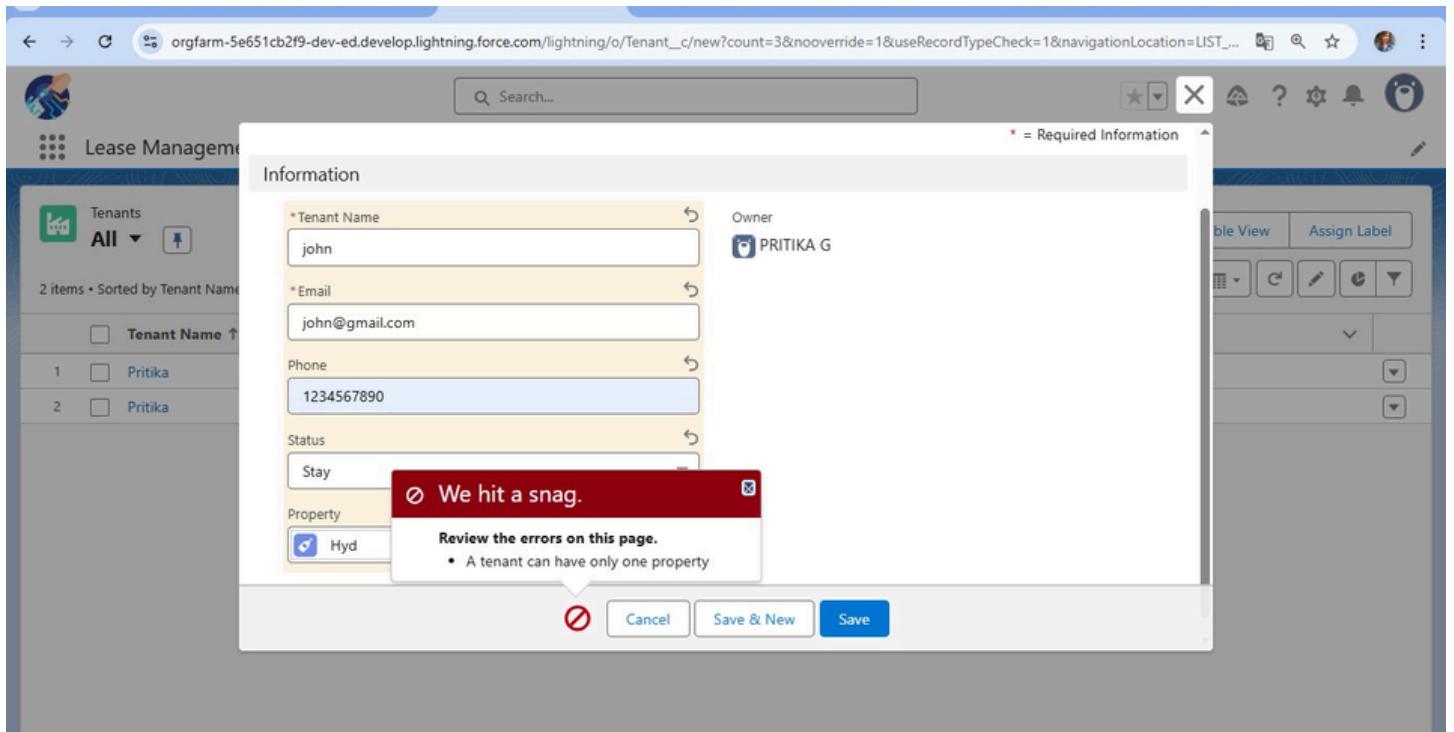
No activities to show. Get started by sending an email, scheduling a task, and more.

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

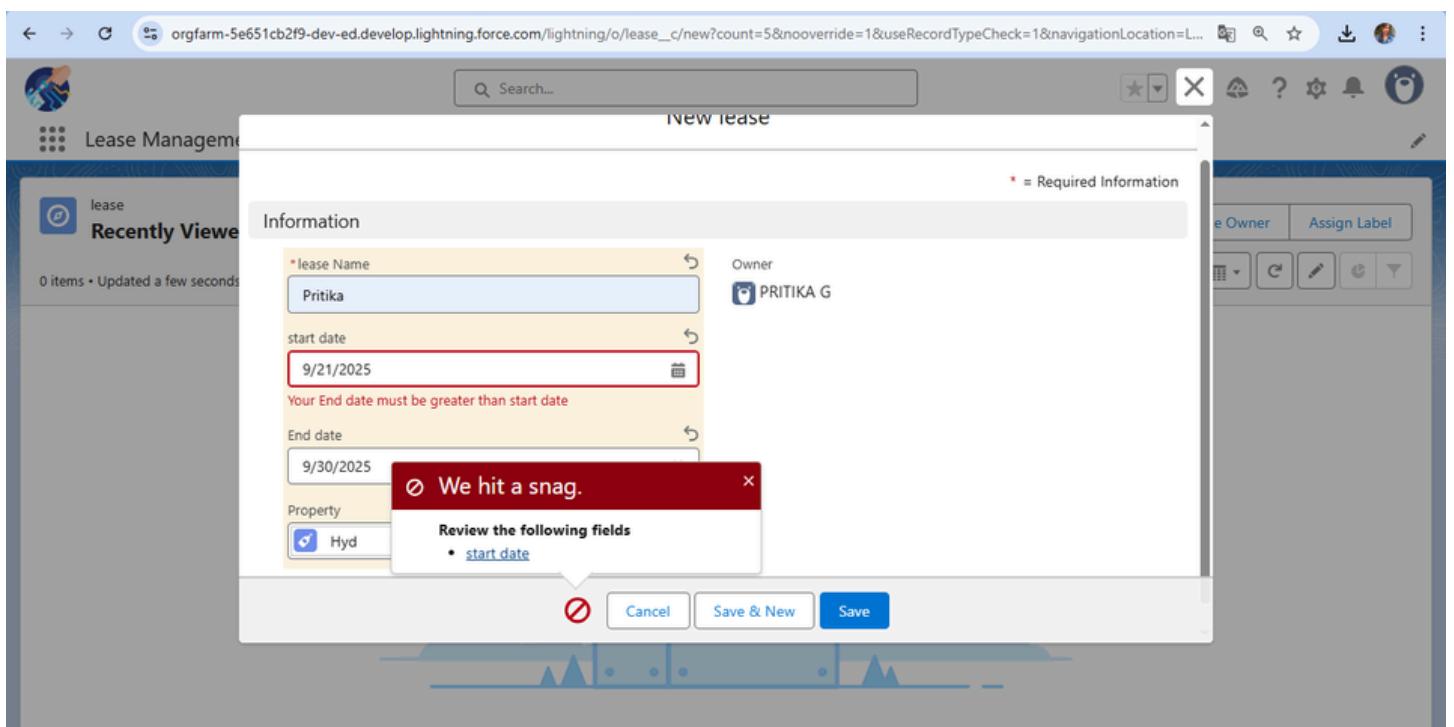
FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

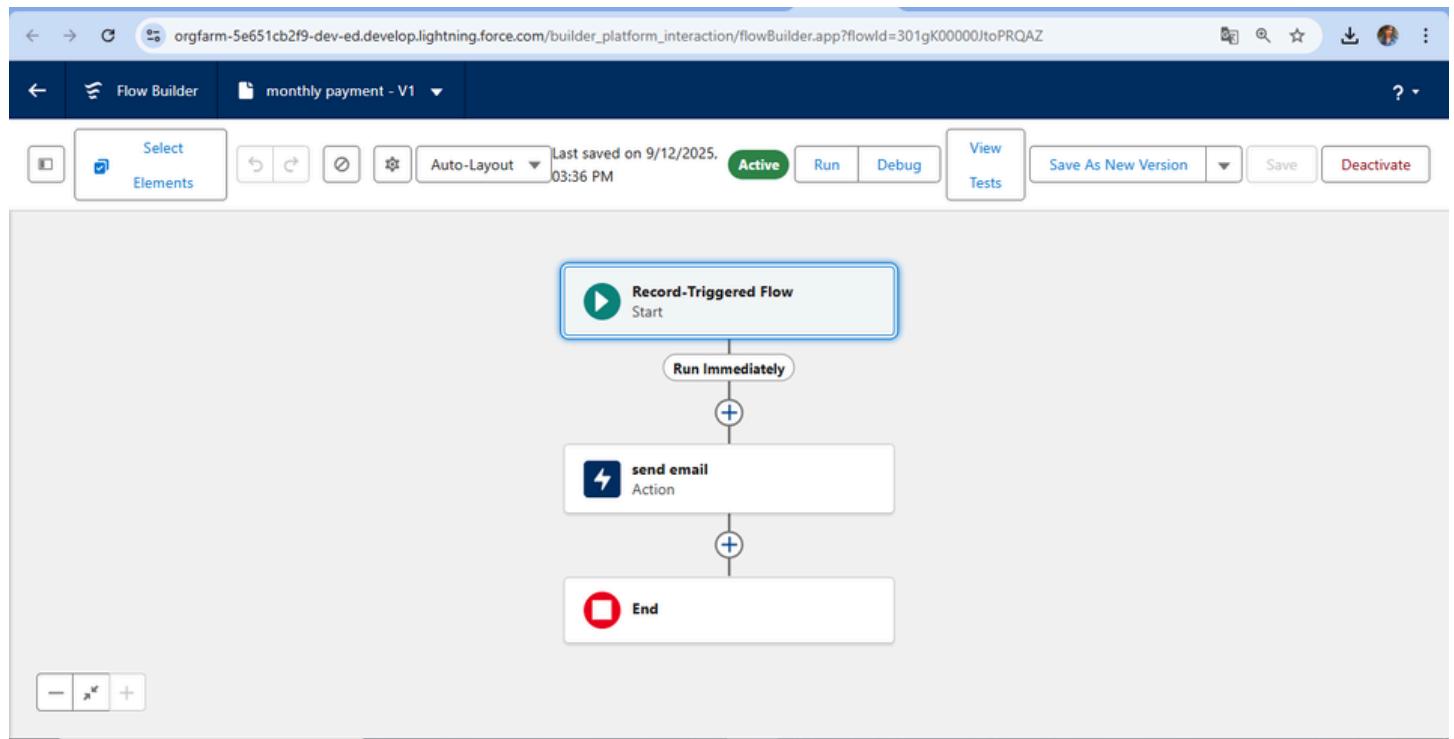
- Trigger validation by entering duplicate tenant-property records.



- Validation Rule checking



- Test flows on payment update



- Approval process validated through email alerts and status updates.

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface for managing Custom Tabs. The left sidebar has a 'User Interface' section with 'Tabs' selected. The main content area is titled 'Custom Tabs' and contains a table for 'Custom Object Tabs'. The table lists four tabs:

Action	Label	Tab Style	Description
Edit Del	Lease	Compass	
Edit Del	Payment	Car	
Edit Del	Property	Bottle	
Edit Del	Tenant	Factory	

Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both of which currently have no tabs defined.

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 <tenants_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});  
        }  
    }  
}
```