

LEASE MANEGEMENT

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 Member1: 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.



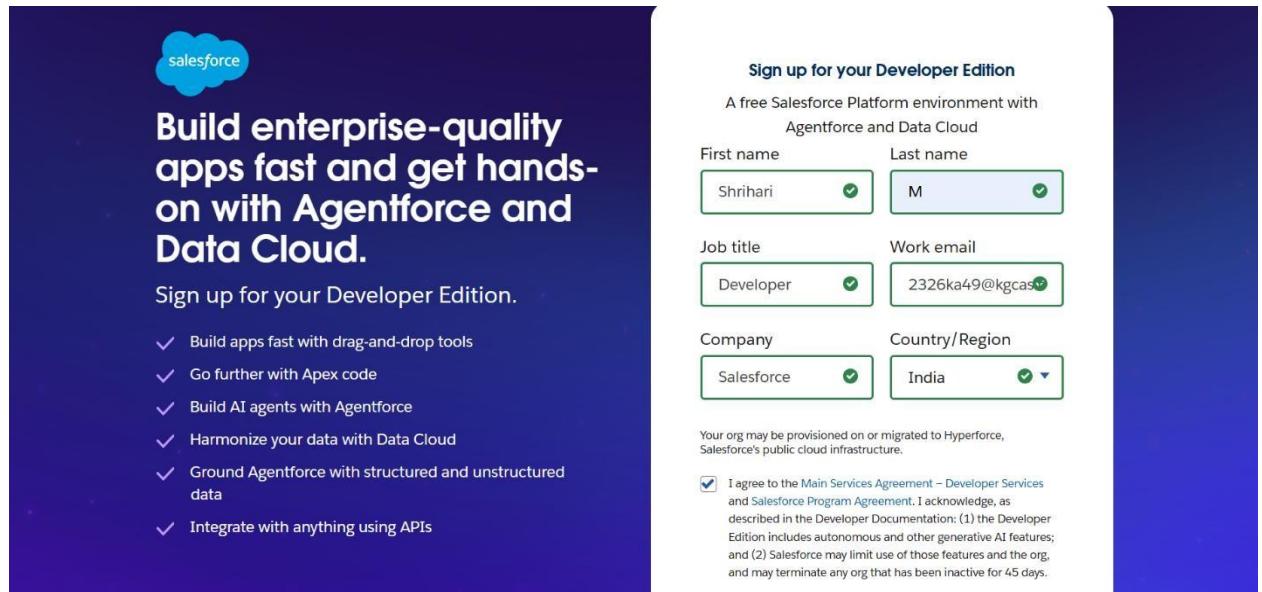
1.2 Purpose

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

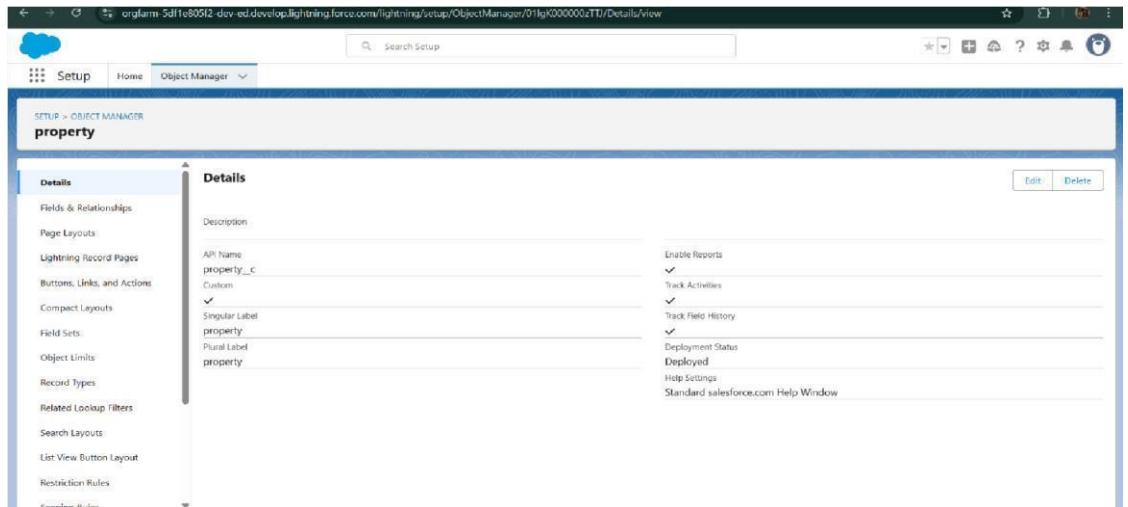
Creating Developer Account:

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



The screenshot shows the "Sign up for your Developer Edition" page. It features a dark blue header with the Salesforce logo and a central text block: "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this is a "Sign up for your Developer Edition" button. To its right is a list of benefits with checkmarks: "Build apps fast with drag-and-drop tools", "Go further with Apex code", "Build AI agents with Agentforce", "Harmonize your data with Data Cloud", "Ground Agentforce with structured and unstructured data", and "Integrate with anything using APIs". On the right side, there are input fields for "First name" (Shrihari), "Last name" (M), "Job title" (Developer), "Work email" (2326ka49@kgcas.v), "Company" (Salesforce), and "Country/Region" (India). A note below states: "Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure." At the bottom is a checkbox for agreeing to the "Main Services Agreement – Developer Services and Salesforce Program Agreement." The background of the page has a subtle gradient.

- Created objects: Property, Tenant, Lease, Payment



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes "Setup", "Home", and "Object Manager". The main area displays the "property" object details. On the left is a sidebar with links like "Fields & Relationships", "Page Layouts", "Lightning Record Pages", "Buttons, Links, and Actions", "Compact Layouts", "Field Sets", "Object Limits", "Record Types", "Related Lookup Filters", "Search Layouts", "List View Button Layout", "Restriction Rules", and "Scoring Rules". The main "Details" section shows the following fields:

- Description:** (empty)
- API Name:** property__c
- Custom:** ✓
- Singular Label:** property
- Plural Label:** property
- Enable Reports:** ✓
- Track Activities:** ✓
- Track Field History:** ✓
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

At the bottom right of the main area are "Edit" and "Delete" buttons.

orgfarm-5dfe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgk00000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface. The left 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, Restriction Rules, and Scoping Rules. The main panel displays the 'Tenant' object's details. The 'Details' section includes fields for Description, API Name (set to 'Tenant__c'), Custom status (checked), Singular Label ('Tenant'), and Plural Label ('Tenants'). On the right, there are sections for Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (set to 'Deployed'), and Help Settings (link to 'Standard salesforce.com Help Window'). At the top right are 'Edit' and 'Delete' buttons.

orgfarm-5dfe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgk00000zTuJ/Details/view

The screenshot shows the Salesforce Object Manager interface. The left 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, Restriction Rules, and Scoping Rules. The main panel displays the 'lease' object's details. The 'Details' section includes fields for Description, API Name (set to 'lease__c'), Custom status (checked), Singular Label ('lease'), and Plural Label ('lease'). On the right, there are sections for Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (set to 'Deployed'), and Help Settings (link to 'Standard salesforce.com Help Window'). At the top right are 'Edit' and 'Delete' buttons.

Object Manager

Payment for tenantat

Details

Description

API Name: Payment_for_tenantat_c
Custom
✓
Singular Label: Payment for tenantat
Plural Label: Payment

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

Edit **Delete**

- Configured fields and relationships

Fields & Relationships
9 Items; Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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		✓

Setup > Object Manager
Payment for tenantat

Fields & Relationships		Fields & Relationships																																								
		FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED																																				
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	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)			
	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 > Object Manager
lease

Fields & Relationships		Fields & Relationships																																								
		FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED																																				
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	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			
	Created By	CreatedById	Lookup(User)																																							
	End date	End_date_c	Date																																							
	Last Modified By	LastModifiedById	Lookup(User)																																							
	lease Name	Name	Text(80)																																							
	Owner	OwnerId	Lookup(User, Group)																																							
	property	property_c	Lookup(property)																																							
start date	start_date_c	Date																																								

Fields & Relationships
7 items: Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone_c	Phone		
status	status_c	Picklist		
Tenant Name	Name	Text(80)		✓

- Developed Lightning App with relevant tabs

App Details & Branding

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

App Details

*App Name: Lease Management

*Developer Name: Lease_Management

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

App Branding

Image:

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

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

App Settings

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

Navigation Items

[User Profiles](#)

Navigation Items

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

Available Items

Type to filter list... [Create](#)

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

Selected Items

- Payment
- Tenants
- property
- lease

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

App Settings

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

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
- Customer Community Login User

Selected Profiles

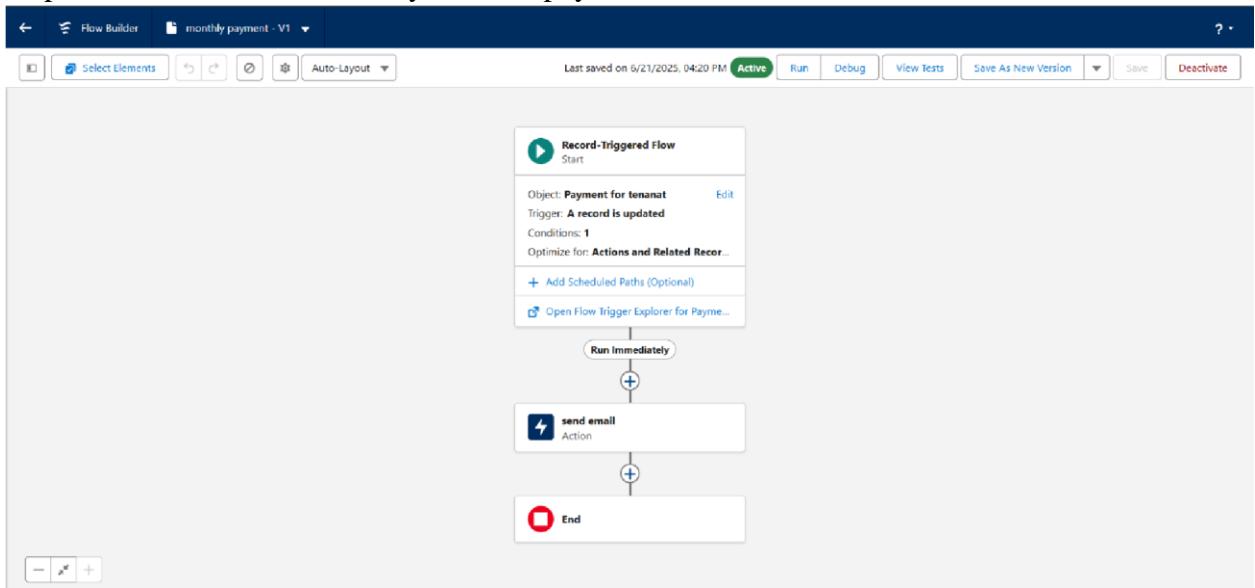
- System Administrator

Lease Management Payment Recently Viewed

5 items • Updated a few seconds ago

	Payment Name	
1	Rahul	
2	Jack	
3	Raj	
4	Sam	
5	Lahari	

- Implemented Flows for monthly rent and payment success

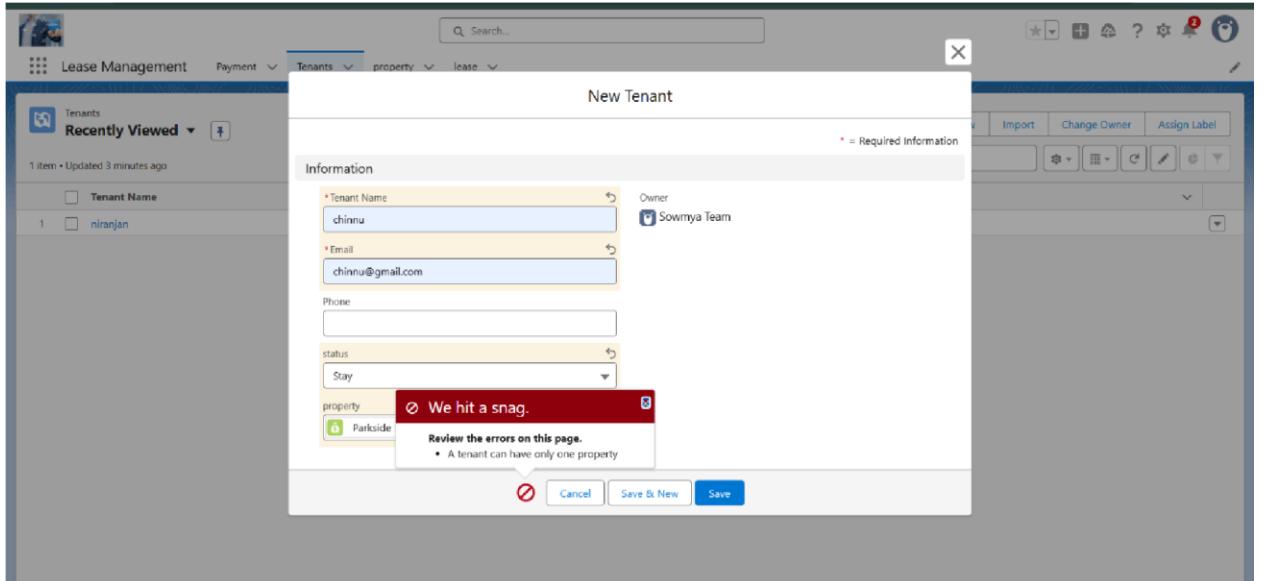


- To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' screen for the 'lease' object. The 'Rule Name' is 'lease_end_date'. The 'Error Condition Formula' is set to 'End_date_c <= start_date_c'. A tooltip for the 'ABS' function is displayed, stating: 'Returns the absolute value of a number, a number without its sign'. The formula bar also shows 'End_date_c < start_date_c'.

The screenshot shows the 'Validation Rule Detail' screen for the 'lease' object. The rule is named 'lease_end_date'. The 'Error Condition Formula' is 'End_date_c <= start_date_c'. The 'Error Message' is 'Your End date must be greater than start date'. The 'Active' checkbox is checked. The 'Created By' field shows 'Sowmya Team' and the 'Modified By' field shows 'Sowmya Team'. The 'Error Location' is 'start_date'.

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```
1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10
11         }
12     }
13
14
15
16     public static void sendMonthlyEmails() {
17
18         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20         for (Tenant__c tenant : tenants) {
21
22             String recipientEmail = tenant.Email__c;
23
24             String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain';
25
26             String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30             email.setToAddresses(new String[]{recipientEmail});
31
32             email.setSubject(emailSubject);
33
34             email.setPlainTextBody(emailContent);
35     }

```

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

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 search result for 'Leave approved' is displayed. The 'Email Template Detail' section shows the following information:

- Email Template Name:** Leave approved
- Template Unique Name:** Leave_approved
- Encoding:** Unicode (UTF-8)
- Author:** Sowmya Team [Changed]
- Description:** Created By Sowmya Team, 6/20/2025, 1:08 AM
- Modified By:** Sowmya Team, 6/20/2025, 1:08 AM
- Available For Use:** checked
- Last Used Date:** N/A
- Times Used:** N/A

The 'Email Template' preview section shows the following content:

```

Subject: Leave approved
Plain Text Preview:
dear{Tenant__c.Name},
I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.
your leave is approved. You can leave now.
  
```

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 search result for 'tenant leaving' is displayed. The 'Email Template Detail' section shows the following information:

- Email Template Name:** tenant leaving
- Template Unique Name:** tenant_leaving
- Encoding:** Unicode (UTF-8)
- Author:** Sowmya Team [Changed]
- Description:** Created By Sowmya Team, 6/20/2025, 1:06 AM
- Modified By:** Sowmya Team, 6/20/2025, 1:06 AM
- Available For Use:** checked
- Last Used Date:** N/A
- Times Used:** N/A

The 'Email Template' preview section shows the following content:

```

Subject: request for approve the leave
Plain Text Preview:
Dear {Tenant__c.CreatedBy},
Please approve my leave
  
```

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

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 [Changed]
Description	
Created By	Sowmya_Team 6/20/2025, 1:11 AM
Modified By	Sowmya_Team 6/20/2025, 1:11 AM

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 following details:

Classic Email Templates

Tenant Email

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Sowmya_Team [Changed]
Description	
Created By	Sowmya_Team 6/20/2025, 1:12 AM
Modified By	Sowmya_Team 6/20/2025, 1:12 AM

Email Template

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.
```

Email Template Detail

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

Email Template

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	
Description			
Entry Criteria	Tenant__c.status EQUALS Stay		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	<input type="checkbox"/>

Approval Assignment Email Template

Initial Submitter	Tenant Owner	Modified By	Sowmya_Team
Created By	Sowmya_Team	Last Modified	6/26/2025, 11:57 PM

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	1	Step 1			User_Sowmya_Team	Final Rejection

For 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__c.status EQUALS Leaving	
Record Entitlability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template	Leave approved	
Initial Submitters	Tenant Owner	
Created By	Sowmya Team	Modified By Sowmya Team
		Created Date 6/20/2025, 3:16 AM Modified Date 6/26/2025, 11:02 PM

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited
Email Alert		please.approve.my.leave

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	Edit	step1			User:Sowmya Team	Final Rejection

- 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	test	
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

```

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

```

Create an Apex Handler class

```

public class testHandler {
    public static void preventInsert(List<Tenant__c> newList) {
        Set<Id> existingPropertyIds = new Set<Id>();
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
            existingPropertyIds.add(existingTenant.Id);
        }
        for (Tenant__c newTenant : newList) {
            if (newTenant.Property__c != null) {
                newTenant.addError('A');
            }
        }
    }
}

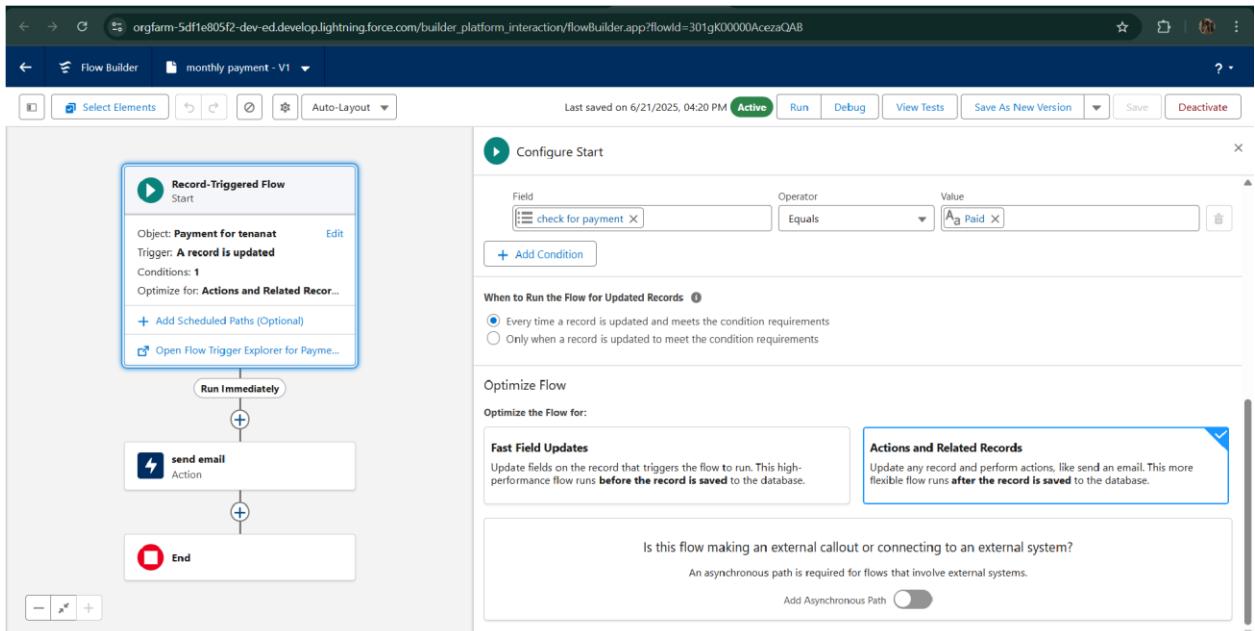
```

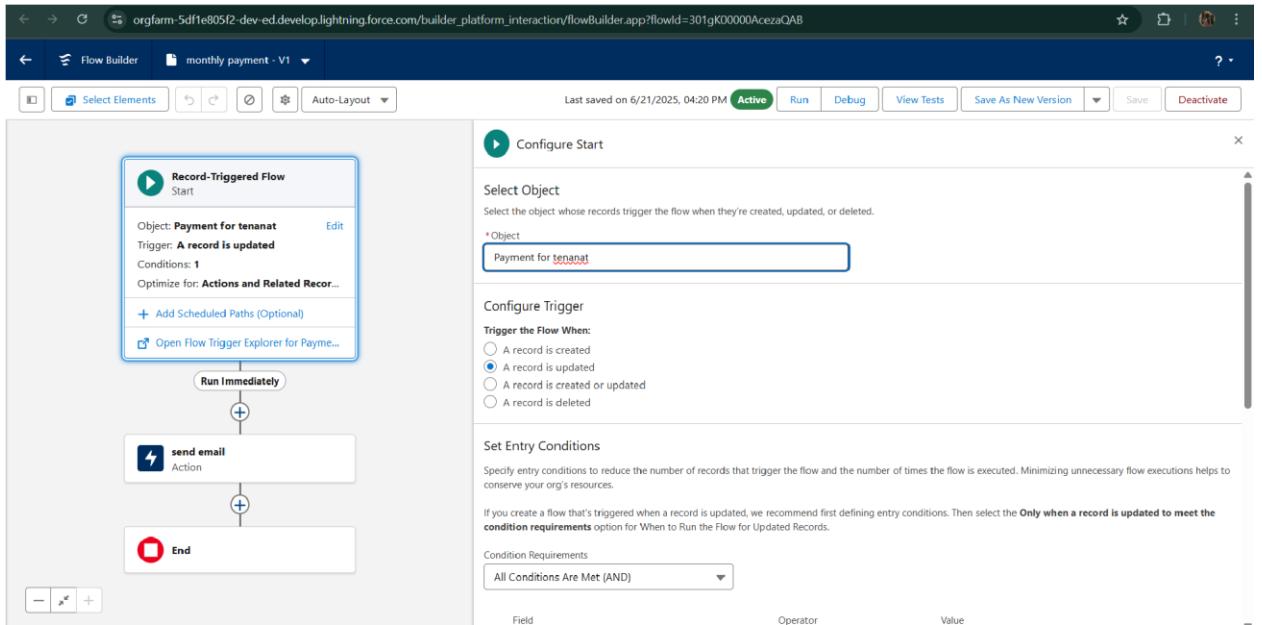
```

1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11        }
12
13
14        for (Tenant__c newTenant : newList) {
15
16
17            if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
18
19                newTenantaddError('A tenant can have only one property');
20
21            }
22
23        }
24
25    }
26
27
28 }

```

● FLOWS





- Schedule class: Create an Apex Class

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10        }
11    }
12 }
13
14
15 public static void sendMonthlyEmail
16 {
17     List<Tenant__c> tenants = [SELECT
18
19         for (Tenant__c tenant : tenants
20
21
22             String recipientEmail = tenant.Email__c;
23

```

The screenshot shows the Salesforce Developer Console with an Apex class named "MonthlyEmailScheduler" open. The class implements the "Schedulable" interface and contains logic to send emails on the first day of every month. A modal window titled "Open" is overlaid on the code editor, showing a list of entity types (Classes, Triggers, Pages, etc.) and a selected item "MonthlyEmailScheduler".

```
Developer Console - Google Chrome
http://orgfarm-5df1e805f2-dev-ed-develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
50% - + Reset

global class MonthlyEmailScheduler implements Scheduleable {
    ...
    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 = 'Recipient: Monthly Rent Payment Due';
        Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
        email.setToRecipients(new String[]{recipientEmail});
        email.setSubject(emailSubject);
        email.setPlainText(emailContent);
        email.setHtmlBody(emailContent);
        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
    }
}
}
```

Schedule Apex class

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' page selected. The sidebar on the left lists various Apex categories like Exception Email, Settings, Test Execution, and Triggers. The main content area displays the details for the 'MonthlyEmailScheduler' class, including its code body.

Apex Class Detail

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

Class Body

```
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
9     public static void sendMonthlyEmails() {
10        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            ...
13        }
14    }
15}
16
17
18
19
20
```

Lease Management Payment Tenants property lease

Search... Tenant Aswini

New Contact Edit New Opportunity

Activity

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

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

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

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

Related Details

* = Required Information

Tenant Name: Aswini Owner: Sowmya Team

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

Cancel Save

javascript:void(0)

Tenant was submitted for approval.

Lease Management Payment Tenants property lease

Search... Tenant Aswini

New Contact Edit New Opportunity

Activity

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

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

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

Related Details

* = Required Information

Tenant Name: Aswini Owner: Sowmya Team

Email: aswiniamaraadi15@gmail.com

Phone: (905) 223-5567

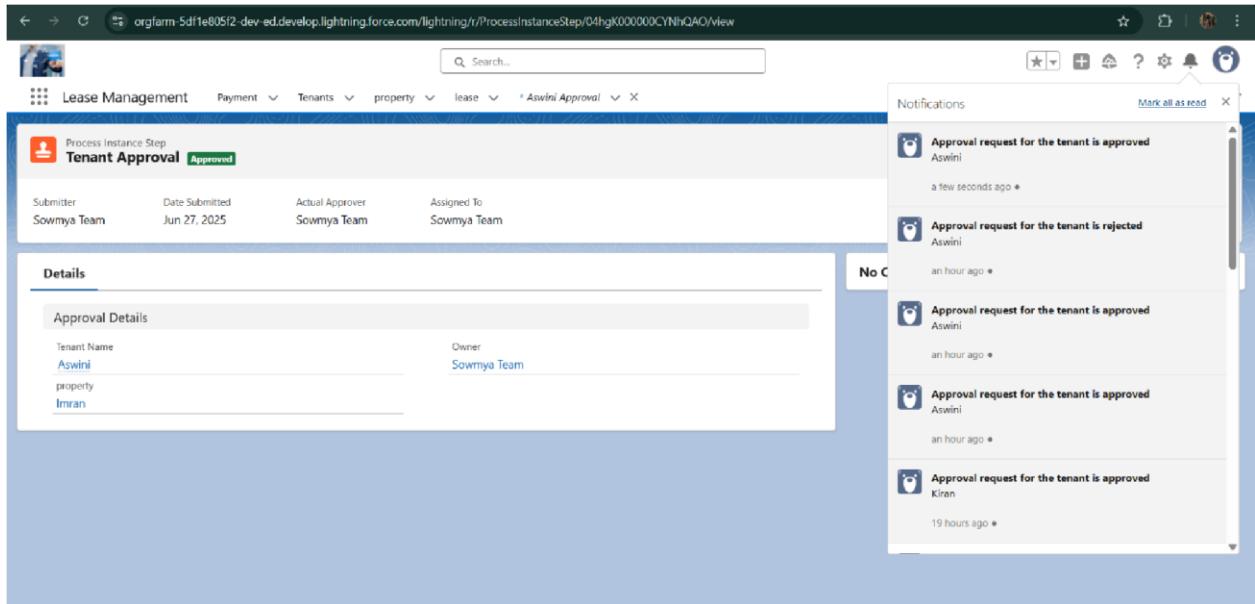
Status: Leaving

Property: Imran

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

Cancel Save

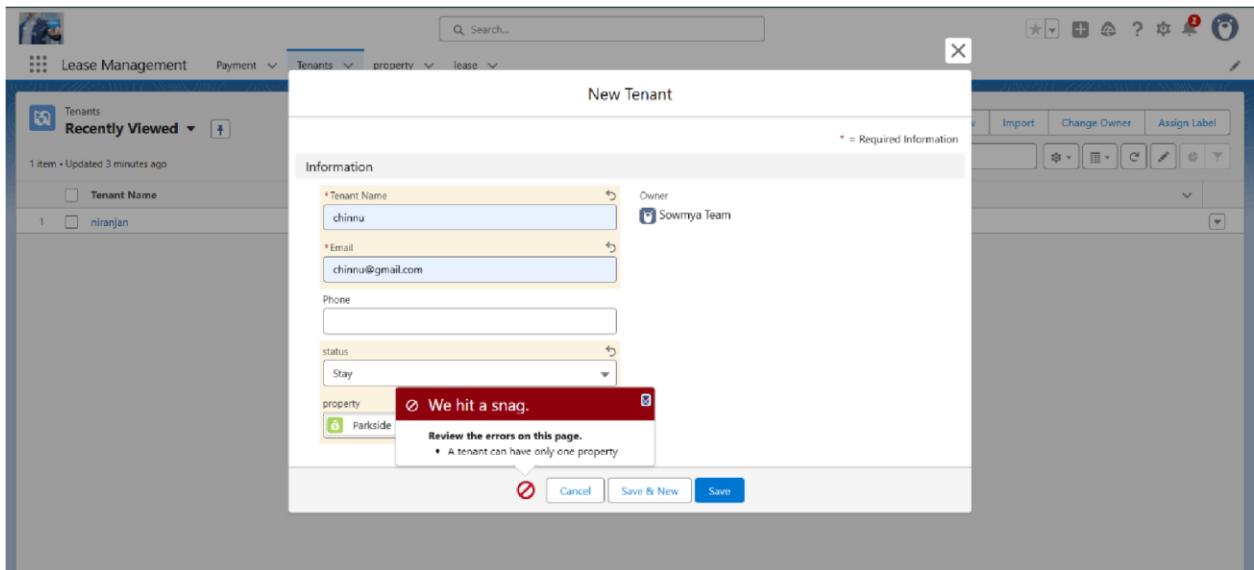
Cloudy 20°C



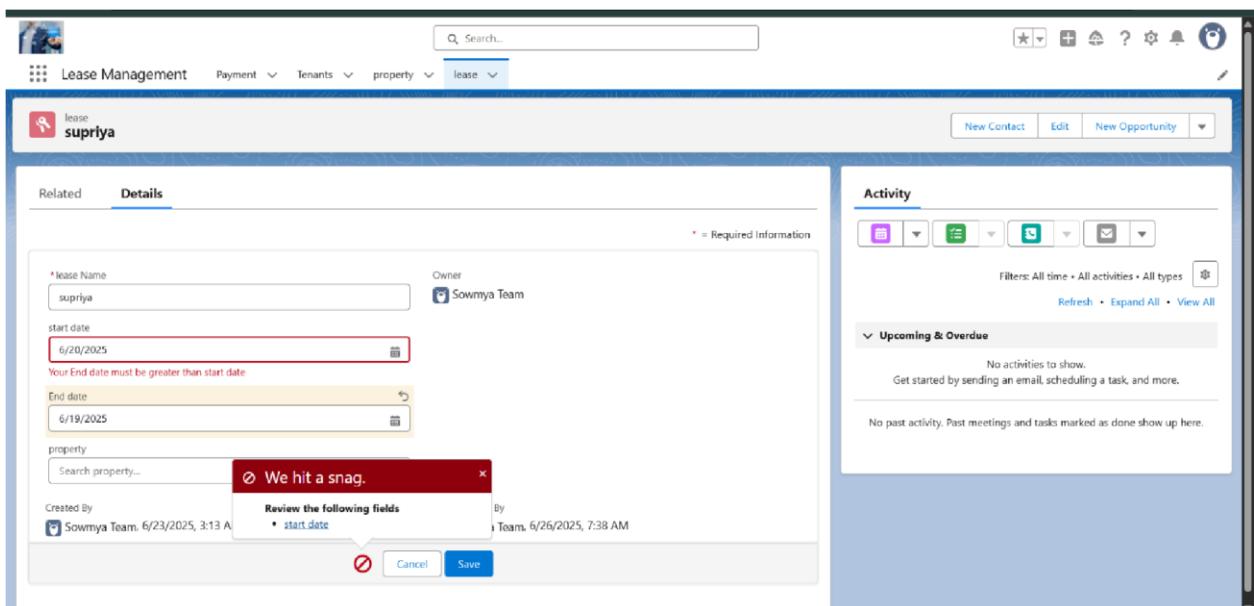
FUNCTIONAL AND PERFORMANCE TESTING

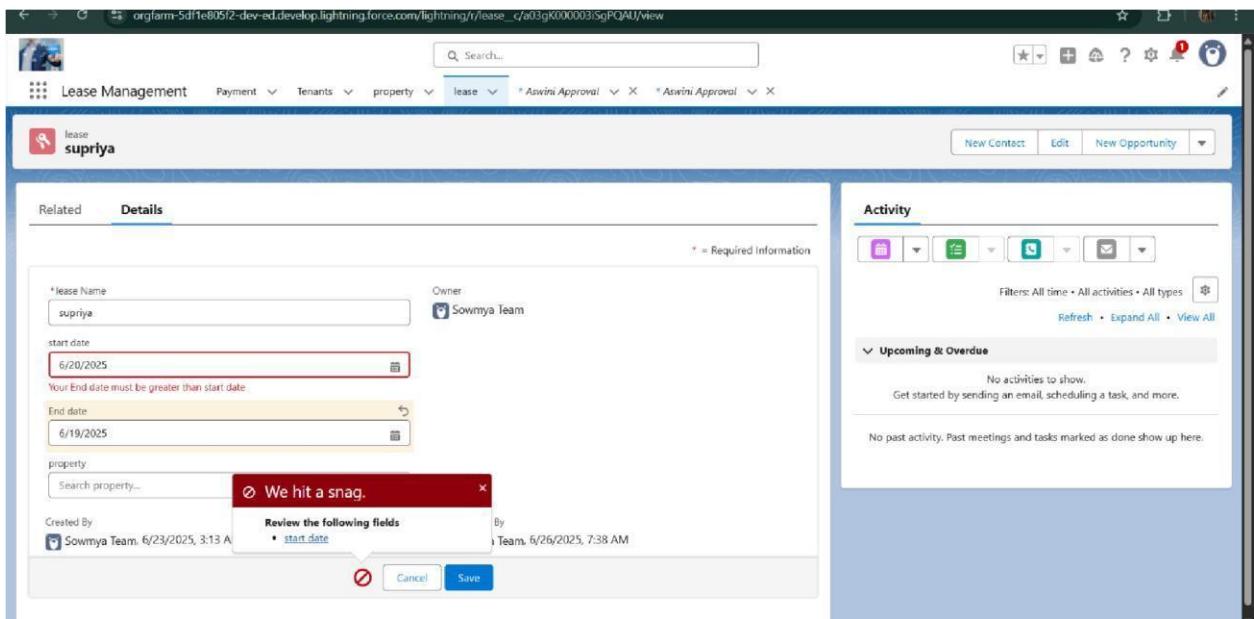
Performance Testing

- Trigger validation by entering duplicate tenant-property records

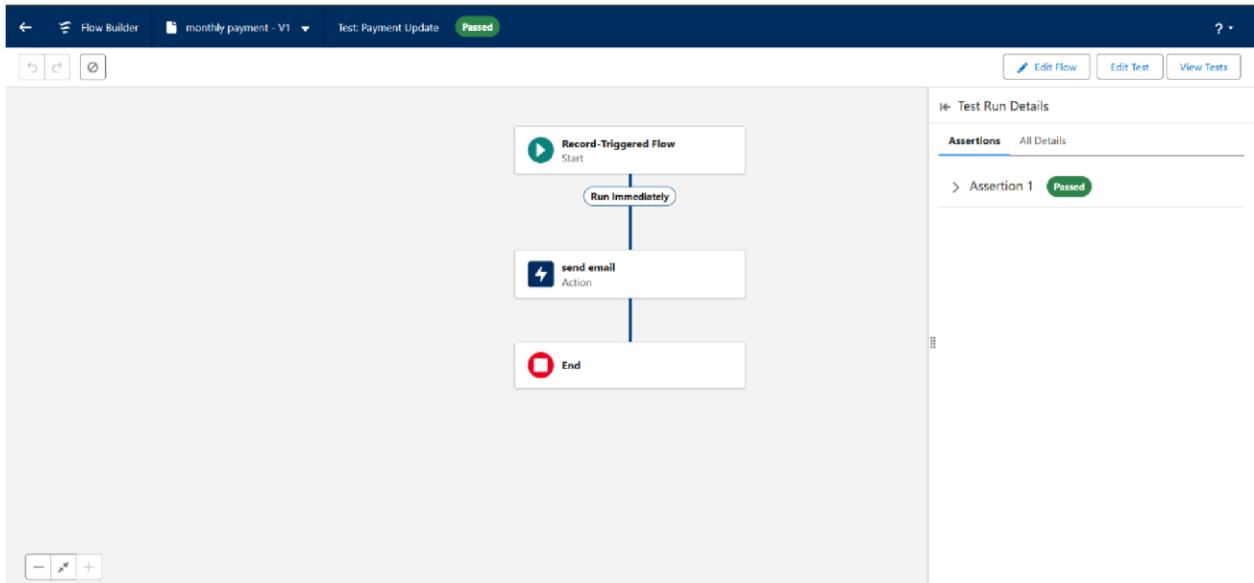


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Tenant niranjan

Details

* = Required Information

Tenant Name	niranjan	Owner	Sowmya Team
Email	niranjan1506@gmail.com		
Phone			
Status	Stay		
Property	Parkside Lofts		

Created By: Sowmya Team, 6/23/2025, 2:33 AM Last Modified By: Sowmya Team, 6/23/2025, 3:58 AM

Cancel **Save**

Notifications

- Approval request for the tenant is approved niranjan (a few seconds ago)
- Approval request for the tenant is rejected niranjan (Jun 23, 2025, 4:29 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:25 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:14 PM)
- New Guidance Center learning resource available Define Your Sales Process (Jun 20, 2025, 1:28 PM)

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

Payment (2)

Payment Name
Jack
Rahul

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

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.

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

Action	Label	Description
No Web Tabs have been defined		

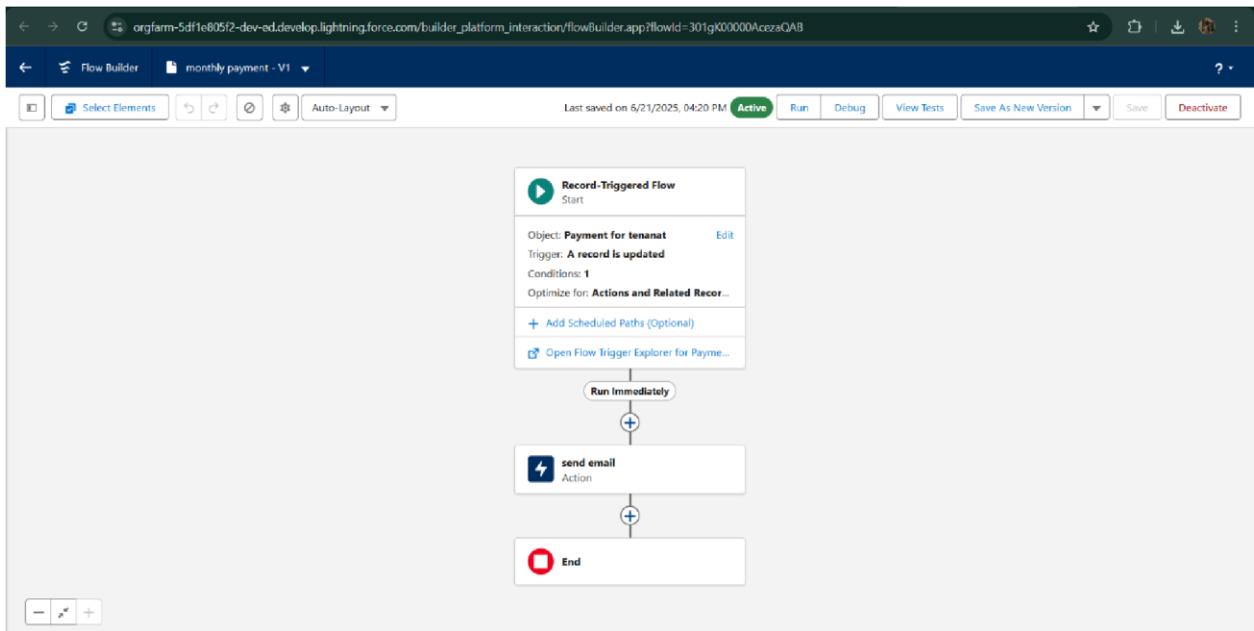
Action	Label	Description
No Visualforce Tabs have been defined		

- Email alerts

Approval History

8 items • Sorted by Is Pending • Updated a few seconds ago

Step Name	Date	Status	Assigned To	Actual Approver	Comments
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team	Sowmya Team	Leaving
Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team	Sowmya Team	Approved
Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team	Sowmya Team	leaving
Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	Leaving

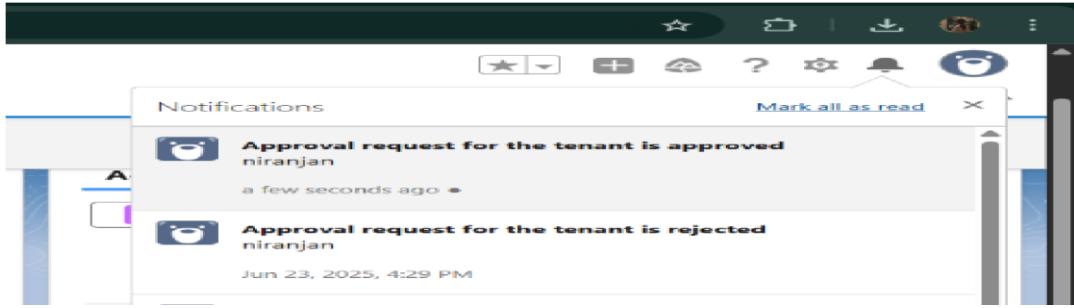


- Trigger error messages

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

- Page:** New Tenant
- Fields:**
 - Tenant Name: chinnu
 - Email: chinnu@gmail.com
 - Phone: (empty)
 - Status: Stay
 - Property: Parkside
- Error Message:** A red error box displays:
 - 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)) { newTenant.addError('A
tenant can have only one property');

}
}

}
```

MothlyEmailScheduler.apxc:

```

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

```