

# LEASEMANAGEMENT

College Name: Hindusthan College Of Science And Commerce College Code: CA

**TEAM ID: NM2025TMID28177 TEAM MEMBERS:**

**Team Leader Name: DHARANIDHARAN S**

**Email: [21dharani21s@gmail.com](mailto:21dharani21s@gmail.com)**

**Team Member: HARISH RAGAVENDRA S**

**Email:[harishragavendra9306@gmail.com](mailto:harishragavendra9306@gmail.com)**

**Team Member : SAKTHIVEL**

**Email: [sakthivelseeni743@gmail.com](mailto:sakthivelseeni743@gmail.com)**

**Team Member : INDHIRESH**

**Email: [indhiresh@gmail.com](mailto:indhiresh@gmail.com)**

**Team Member : SANTHOSH S**

**Email: [santoshabariss513@gmail.com](mailto:santoshabariss513@gmail.com)**

## 1. INTRODUCTION

### 1. Project Overview

This project is focused on the development and implementation of a Lease Management System using Salesforce, designed to address the challenges of manual property lease tracking, tenant management, payment processing, and

communication within the organization. The goal is to deliver a comprehensive solution that streamlines and automates various aspects of lease management, from contract creation to payment tracking, approval workflows, and reporting.



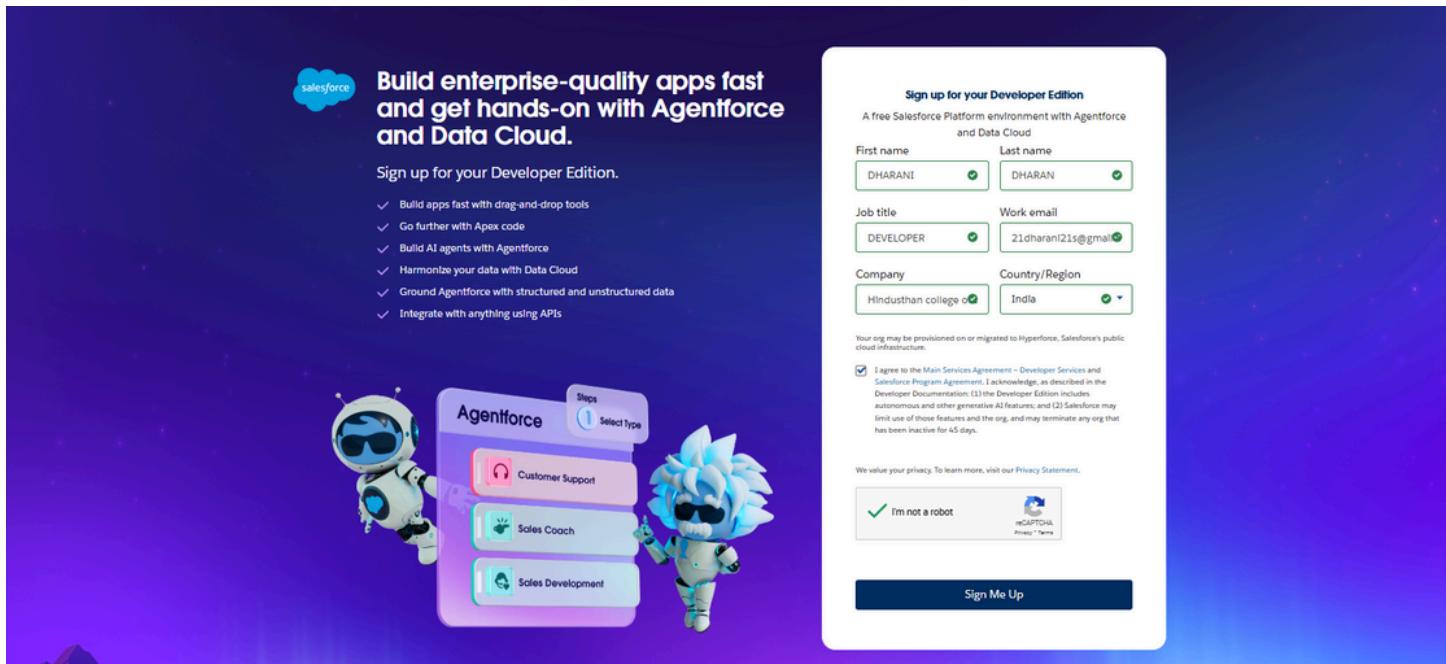
## 1. Purpose

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



- Created objects: Property, Tenant, Lease, Payment

A screenshot of the Salesforce Object Manager. The top navigation bar shows "SETUP &gt; OBJECT MANAGER" and the object name "Tenant". The main area displays the "Details" tab for the Tenant object. On the left is a sidebar with links like "Field &amp; Relationships", "Page Layouts", "Lightning Record Pages", "Buttons, Links, and Actions", "Compact Layouts", "Field Sets", "Object Limits", "Record Types", "Related Lookups", "Search Layouts", "List View Button Layout", "Restriction Rules", and "Sourcing Rules". The main "Details" section has a "Description" field and a "Fields &amp; Relationships" panel on the right. The "Fields &amp; Relationships" panel lists fields such as "API Name" (Tenant\_\_c), "Custom" (checked), "Singular Label" (Tenant), and "Plural Label" (Tenants). The "Fields &amp; Relationships" panel also includes sections for "Enable Reports" (checked for Track Activities, Track Feed History, and Deployment Status), "Deployment Status" (Deployed), and "Help Settings" (Standard salesforce.com Help Window).

**New Custom Object**

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label:  Example: Account

Plural Label:  Example: Accounts

Starts with vowel sound:

The Object Name is used when referencing the object via the API.

Object Name:  Example: Account

Description:

Context-sensitive Help Setting:  Open the standard Salesforce.com Help & Training window  
                                  Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format.

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is 'Account Name' and for Case it is 'Case Number'. Note that the Record Name field is always called 'Name' when referenced via the API.

Record Name:  Example: Account Name

Data Type:  Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

- Allow Record
- Allow New Activities
- Task Field History
- Allow in Chatter Groups
- Enable Licensing

**lease**

Details

Description

API Name:  Custom:

Singular Label:  Plural Label:

Enable Reports:  Stock Activities:  Track Field History:

Deployment Status: Deployed Help Settings: Standard salesforce.com Help Window

**Payment for tenantat**

Details

Description

API Name:  Custom:

Singular Label:  Plural Label:

Enable Reports:  Stock Activities:  Track Field History:

Deployment Status: Deployed Help Settings: Standard salesforce.com Help Window

## ■ Configured fields and relationships

Fields & Relationships				
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(30)		✓
sfp	sfp_c	Text(18)		
Type	Type_c	Picklist		

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 Name	Name	Text(30)		✓

Setup Home Object Manager

SETUP > OBJECT MANAGER

## lease

Details Fields & Relationships

Fields & Relationships

	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	End date	End_date__c	Date		
Buttons, Links, and Actions	Last Modified By	LastModifiedById	Lookup(User)		
Compact Layouts	Lease Name	Name	Text(80)		✓
Field Sets	Owner	OwnerId	Lookup(User/Group)		✓
Object Limits	property	property__c	Lookup(property)		✓
Record Types	start date	start_date__c	Date		
Related Lookup Filters					
Search Layouts					
List View Button Layout					
Restriction Rules					
Scoping Rules					

Setup Home Object Manager

SETUP > OBJECT MANAGER

## Tenant

Details Fields & Relationships

Fields & Relationships

	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Email	Email__c	Email		
Buttons, Links, and Actions	Last Modified By	LastModifiedById	Lookup(User)		
Compact Layouts	Owner	OwnerId	Lookup(User/Group)		✓
Field Sets	Phone	Phone__c	Phone		
Object Limits	status	status__c	Picklist		
Record Types	Tenant Name	Name	Text(80)		✓
Related Lookup Filters					
Search Layouts					
List View Button Layout					
Restriction Rules					
Scoping Rules					

- Developed Lightning App with relevant tabs

**App Details & Branding**

**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: Placeholder image of a person in a suit.

Primary Color Hex Value: #0070C2

**Org Theme Options**

Use the app's image and color instead of the org's custom theme

**App Launcher Preview**

Lease Management  
Application to efficiently handle the processes relate...

**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**

C: 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

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

**User Profiles**

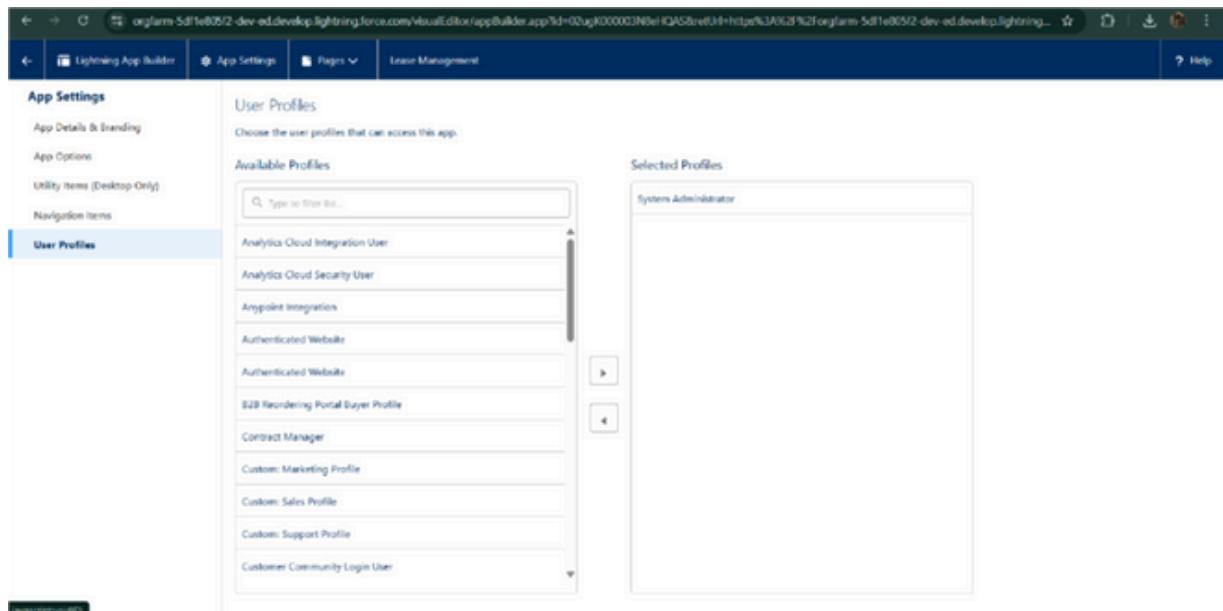
Choose the user profiles that can access this app.

**Available Profiles**

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

**Selected Profiles**

- System Administrator



Lease Management | Payment | Tenants | property | Lease | ?

**Payment**

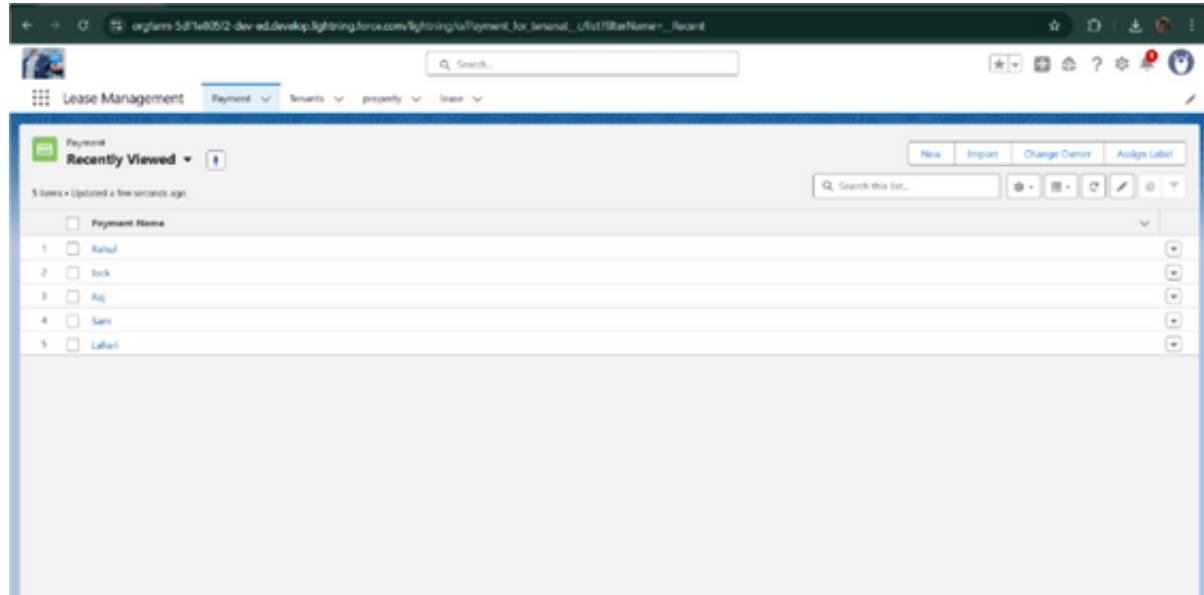
Recently Viewed

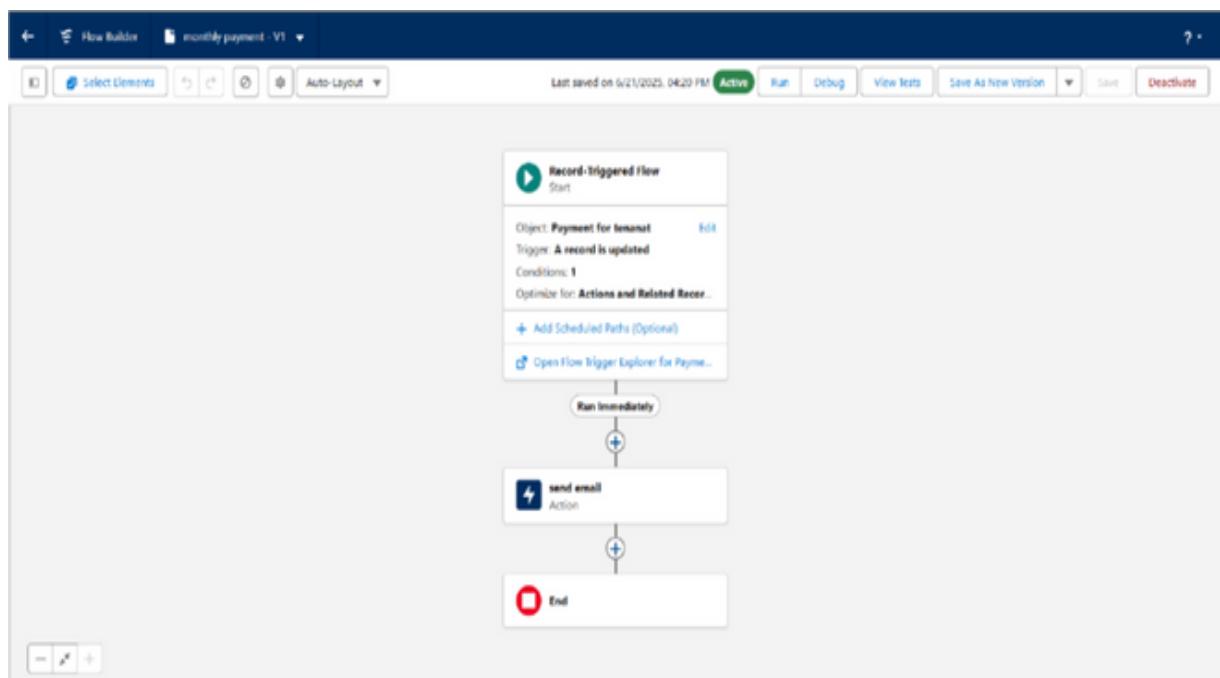
5 items • Updated a few seconds ago

	Payment Name
1	Rahul
2	Rock
3	Raj
4	Sam
5	Lakshmi

New Import Change Owner Assign Label

Search this list...



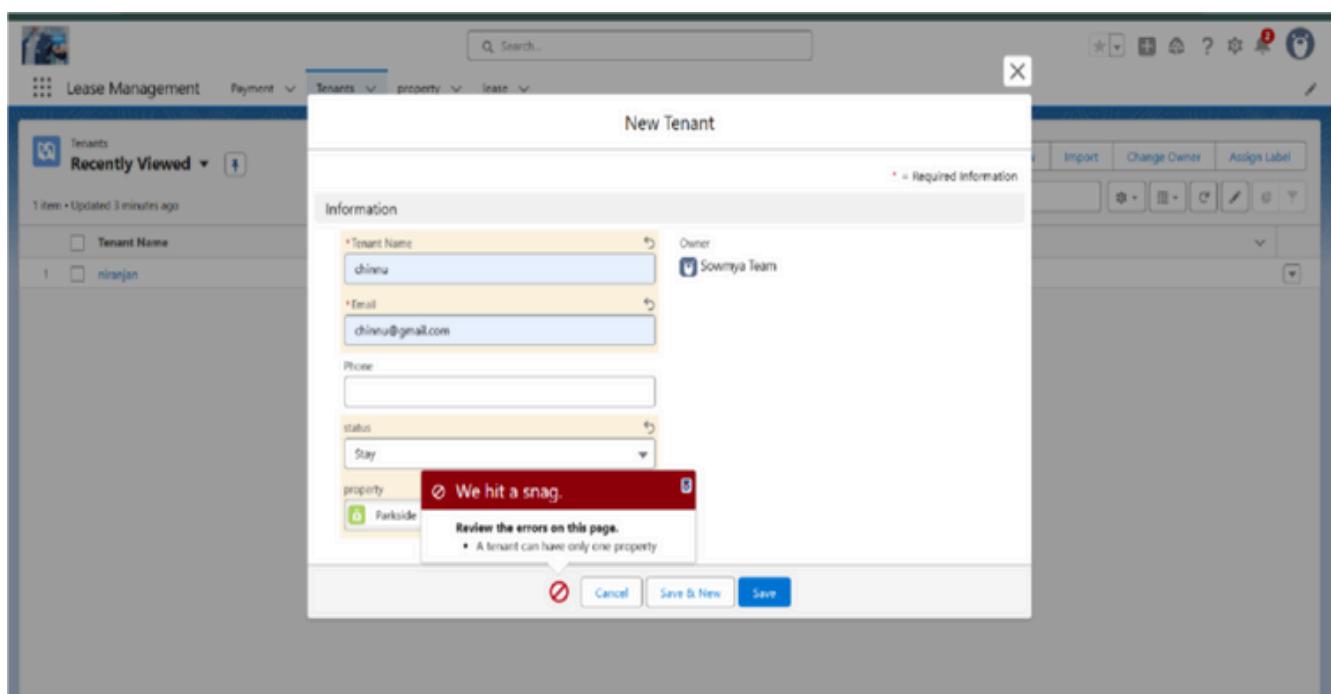


- Implemented Flows for monthly rent and payment success
- To create a validation rule to a Lease Object

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. On the left, a sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main area is titled 'Validation Rule Edit' for the rule named 'lease\_end\_date'. The 'Error Condition Formula' field contains the expression 'End\_date\_\_c < start\_date\_\_c'. A dropdown menu for functions is open, showing options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. The status bar at the bottom right indicates 'Help on this screen'.

The screenshot shows the 'lease Validation Rule' detail page. The 'Validation Rule Detail' section displays the following information: 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: Senthil.Than, 9/19/2025, 5:37 AM', Active: checked, Error Location: 'Start Date', and Modified By: 'Senthil.Than, 6/6/2026, 7:47 AM'. The sidebar on the left is identical to the one in the previous screenshot.

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```

1 * global class MonthlyRentalScheduler 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    public static void sendMonthlyEmails() {
15
16        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
17
18        for (Tenant__c tenant : tenants) {
19
20            String recipientEmail = tenant.Email__c;
21
22            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';
23
24            String emailSubject = 'Reminder: Monthly Rent Payment Due';
25
26            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
27
28            email.setToAddresses(new String[]{recipientEmail});
29
30            email.setSubject(emailSubject);
31
32            email.setPlainTextBody(emailContent);
33
34        }
35    }

```

- 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	Actions
Email Templates from Salesforce Email Template Name: Leave approved Template Unique Name: Leave_approved Preexisting Author: Sumeet_Sabot Description: Created By: Sumeet_Sabot 6/20/2020, 1:06 AM	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Close</a>
Available For Use: <input checked="" type="checkbox"/> Last Used Date: <span style="font-size: small;">Never used</span> Times Used: <span style="font-size: small;">0</span>	
<a href="#">Help for this Page</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.  
  

Sincerely,  
[Signature]

```

Q Search Setup

Setup Home Object Manager

Q\_email template

v Email

Classic Email Templates Lightning Email Templates

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

Test Email Template **Tenant leaving**

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Email Template Name: Tenant leaving	Available For Use: <input checked="" type="checkbox"/>
Template Unique Name:	tenant_leaving	Last Used Date:
Encoding:	Unicode (UTF-8)	Times Used:
Author:	SOMETHING_SFDC	
Description:	SOMETHING_SFDC	
Created By:	SOMETHING_SFDC, 6/09/2025, 1:06 AM	Modified By: SOMETHING_SFDC, 6/09/2025, 1:06 AM

Email Template

Send Test and Verify Merge Fields

Subject: request for approve the leave

Plain Text Preview

Dear [Tenant\_\_c.CreationDate],  
Please approve my leave.

Q Search Setup

Setup Home Object Manager

Q\_email template

v Email

Classic Email Templates Lightning Email Templates

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

Test Email Template **Leave rejected**

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Email Template Name: Leave rejected	Available For Use: <input checked="" type="checkbox"/>
Template Unique Name:	Leave_rejected	Last Used Date:
Encoding:	Unicode (UTF-8)	Times Used:
Author:	SOMETHING_SFDC	
Description:	SOMETHING_SFDC	
Created By:	SOMETHING_SFDC, 6/09/2025, 1:11 AM	Modified By: SOMETHING_SFDC, 6/09/2025, 1:11 AM

Email Template

Send Test and Verify Merge Fields

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	Available For Use	<input checked="" type="checkbox"/>
Template Unique Name	Tenant_Email	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Sistema, Team (Chatter)		
Description			
Created By	Sistema, Team (6/20/2025, 1:12 AM)	Modified By	Sistema, Team (6/20/2025, 1:12 AM)

**Email Template**

Subject: Urgent: Monthly Rent Payment Reminder

Plain Text Preview:

```
Dear {Tenant__c.Name},  
  
I hope this email finds you well. We appreciate your continued tenancy of our property and I hope you have been comfortable in your residence.
```

This template is used in a Tenant account. It has been used once. Last modified: 6/20/2025, 1:12 AM by Sistema, Team.

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

**Classic Email Templates**

**Tenant payment**

Email Template Detail

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

The screenshot shows the 'Approval Processes' page in the Salesforce Setup. The process is named 'Tenant: TenantApproval'. It has one step, 'Step 1', which is a 'Record Lock' action. The process is active and was created by 'Screena Team' on 6/25/2025 at 3:01 AM.

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	Step 1			User:Screena Team	Final Rejection

For Check for Vacant:

The screenshot shows the 'Approval Processes' page in the Salesforce Setup. The process is named 'Tenant: check for vacant'. It has two steps: 'Record Lock' and 'Email Alert'. The process is active and was created by 'Screena Team' on 6/25/2025 at 3:18 AM.

Action	Type	Description
Record Lock	Lock the record from being edited	
Email Alert	DEBBD_ABDSVVE.MV.JOBJS	

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	Step 1			User:Screena Team	Final Rejection

- Apex Trigger

## Create an Apex Trigger

The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a dropdown for Tooling API Version. The main area displays the code for a trigger:

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

A modal dialog titled "Open" is displayed, listing various types of objects in the repository:

Object Name	Edition	Reflected
Body Type	Standard	
Classes		
Triggers	test	
Pages		
Page Components		
Objects		
Static Resources		
Packages		

At the bottom of the dialog are buttons for Open, Filter, Refresh, and a note about Managed Packages.

The bottom navigation bar includes tabs for Logs, Tests, Checkstyle, Query Editor, View Status, Progress, Problems, and a dropdown for "None". Below this is a status bar with Line and Problem indicators.

Developer Console - Google Chrome

orgfarm-5d11e052 - dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help < >

testHandler.apex - /apex/testHandler

Code Coverage Name API Version 14

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

Log Tests Checkpoints Query Editor View Status Progress Problems

## 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(exist
10
11     }
12
13
14     for (Tenant__c newTenant : newList) {
15
16
17
18         if (newTenant.Property__c != null) {
19
20             newTenant.addError('A tenant must have a property')
21
22     }
23 }
```



The screenshot shows the Developer Console in Google Chrome with the URL <https://orgname-SelfHelp052-dev-ed.development.salesforce.com/sf/common/sales/debug/ApexCSPage>. A modal dialog box is open, listing three validation errors:

- Property\_\_c is required
- Property\_\_c must be a valid object
- Property\_\_c must be a valid object

The 'Prevent Insert' checkbox at the bottom of the dialog is checked.

Developer Console - Google Chrome

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

Toolbars: testHandler.apex HeadlessBatchScheduler.apex

Code Coverage: None • API Version: 54 • Go To

```

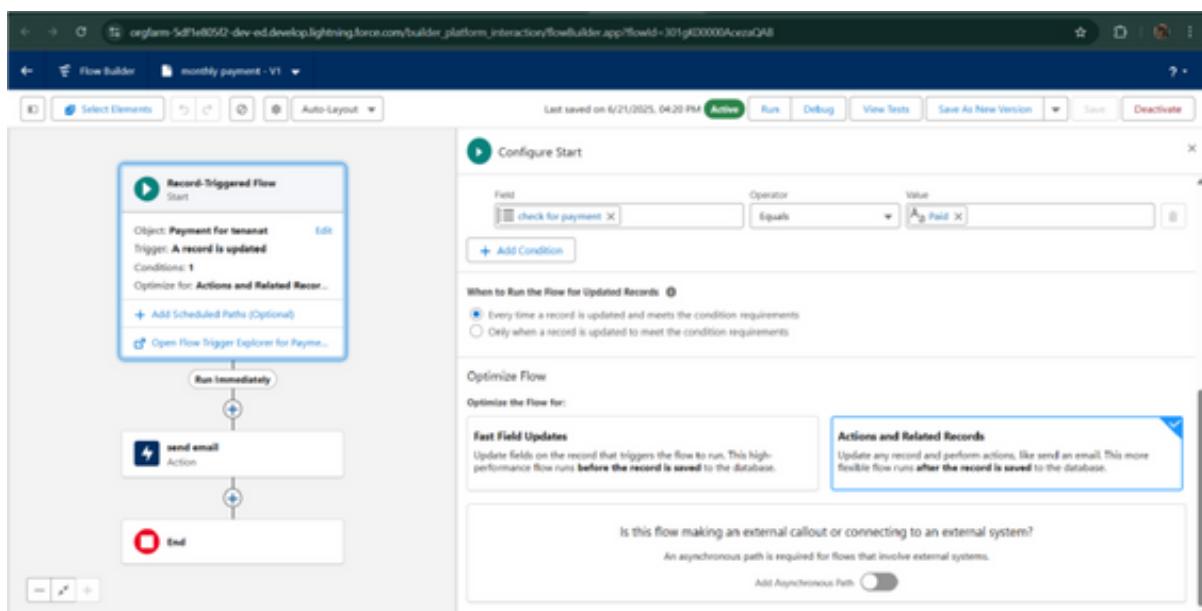
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

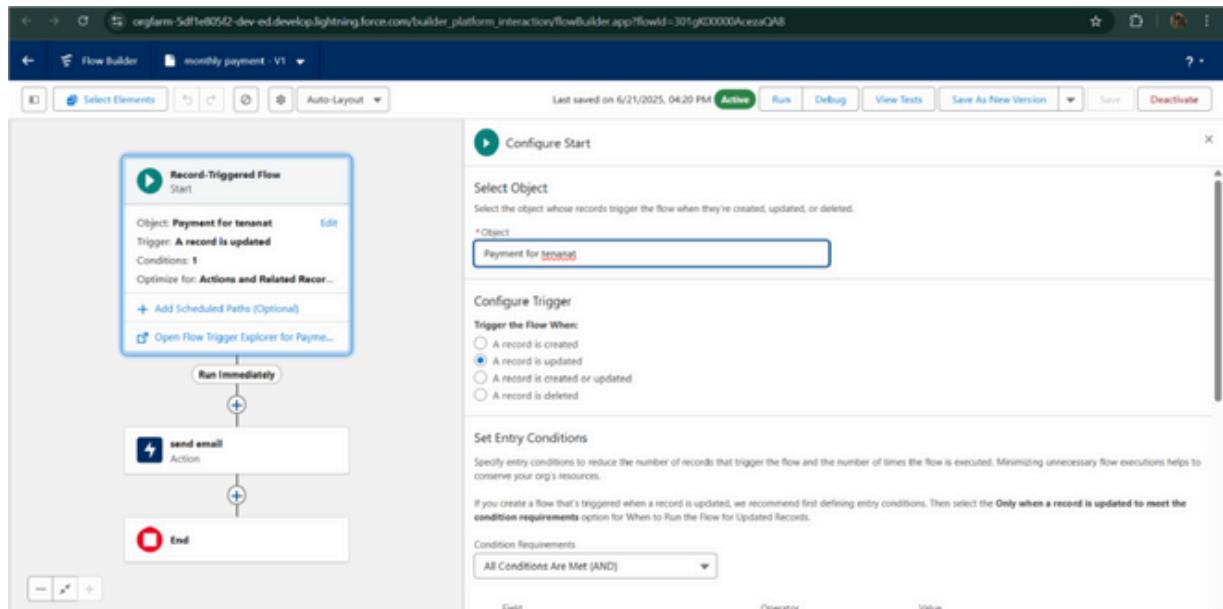
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

None Line Problem

- 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
16 public static void sendMonthlyEmail(
17
18     List<Tenant__c> tenants = [SELECT
19
20         for (Tenant__c tenant : tenants
21
22             String recipientEmail = tenant.Email__c;

```

## Schedule Apex class

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' section selected. The 'MonthlyEmailScheduler' class is displayed. The class detail page includes tabs for Class Body, Class Summary, Version Settings, and Trace Flags. The code editor shows the following Apex code:

```
1 global class MonthlyEmailScheduler implements Scheduleable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendEmail();
6         }
7     }
8 }
9
10
11
12
13
14
15
16 public static void sendEmail() {
17     List<Tenant__c> tenants = [SELECT M.Email__c FROM Tenant__c];
18     for (Tenant__c tenant : tenants) {
19         // Email logic here
20     }
21 }
```

The screenshot shows a web-based application interface for lease management. At the top, there's a navigation bar with links to Gmail, YouTube, Maps, News, Translate, and precht.at. The main header includes a blue cloud icon, the title "Lease Management", and a search bar with placeholder text "Search...". Below the header, a breadcrumb trail shows the current path: Tenants > properties > lease > \*Tenant\_c. The main content area displays a tenant record for "Dharanidharan". On the left, a sidebar titled "Related" lists fields like Tenant Name, Email, Phone, status, and Property, each with an edit icon. The "Details" tab is selected, showing the Owner as "DHARANIDHARAN S". On the right, there's a section titled "Activity" with icons for different activity types (calendar, list, messages, etc.) and a message stating "No activities to show. Get started by sending an email, scheduling a task, and more.".

The screenshot shows a web-based application interface for 'Lease Management'. At the top, there's a navigation bar with links to Gmail, YouTube, Maps, News, Translate, and precht.at. A search bar is also at the top. The main header includes a blue cloud icon, the text 'Lease Management', and a dropdown menu for 'Tenant\_\_c'. Below the header, a banner indicates an 'Approval Request' for 'Tenant Approval' status 'Pending'. The main content area has two sections: 'Details' on the left and 'Submitter Comments' on the right. The 'Details' section shows the submitter as 'DHARANIDHARAN S', date submitted as 'Sep 13, 2025', actual approver as 'DHARANIDHARAN S', and assigned to 'DHARANIDHARAN S'. The 'Submitter Comments' section shows a comment from 'DHARANIDHARAN S' stating 'Leaving' on 'Sep 13, 2025, 2:22:46 AM'. Action buttons for 'Approve', 'Reject', and 'Reassign' are located at the top right of the main content area.

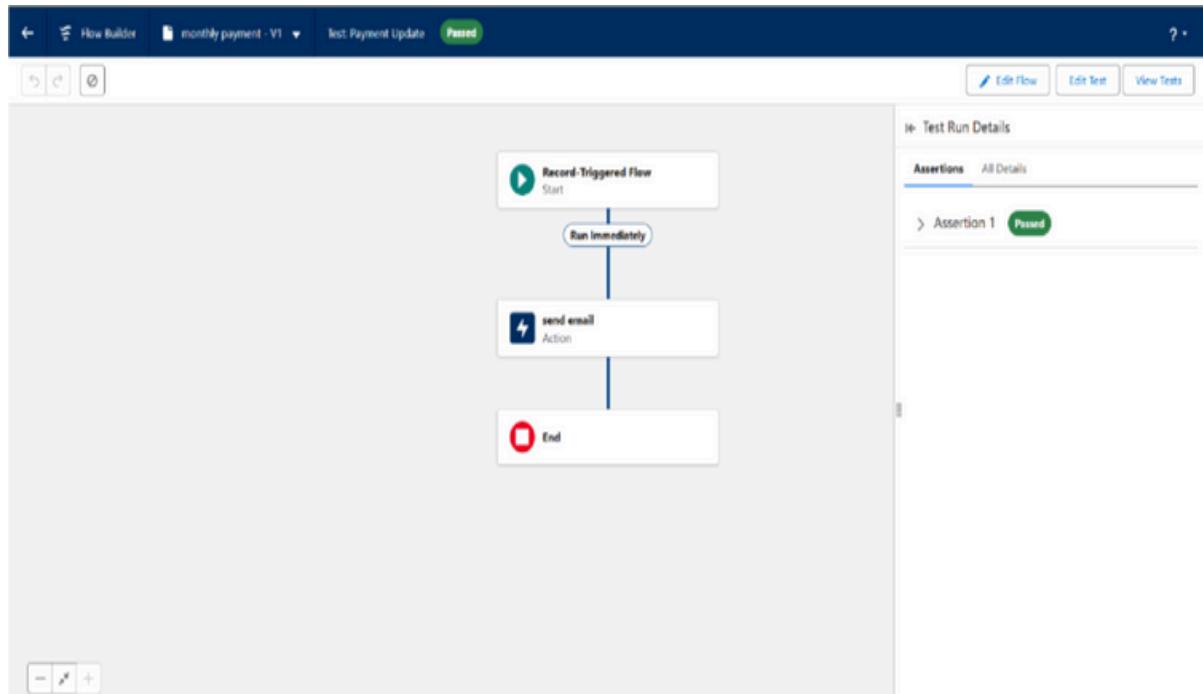
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

- Trigger validation by entering duplicate tenant-property records



- Validation Rule checking



Sales

Home Opportunities Leads Tasks Files Accounts Contacts Campaigns Dashboards Reports

Search...

Notifications

DHARANIDHARAN S is requesting approval for tenant  
Tenant Name: Dharanidharan • Owner: DHARANIDHARAN S  
6 hours ago

Seller Home Good evening, DHARANIDHARAN. Let's get selling!

**Close Deals**  
Opportunities owned by me and closing this quarter

Total Pipeline \$0

- \$0 Open
- \$0 Won
- \$0 Lost

**Plan My Accounts**  
Accounts owned by me

0 Accounts

- 0 Upcoming Activity
- 0 Past Activity
- 0 No Activity

**Build Pipeline**  
Leads owned by me and created in the last 30 days

**My Goals**  
Set personal weekly or monthly goals for emails, calls, and meetings.

**Today's Events**

- Test flows on payment update

**Lease Management**

Tenant: niranjan

**Approval History (6+)**

Step Name	Date	Status	Assigned To
Step 1	6/25/2015, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2015, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2015, 2:59 AM	Rejected	Sowmya Team
Approval Request Submitted	6/23/2015, 2:58 AM	Submitted	Sowmya Team
Step 1	6/23/2015, 2:55 AM	Approved	Sowmya Team
Approval Request Submitted	6/23/2015, 2:55 AM	Submitted	Sowmya Team

**View All**

**Payment (2)**

Payment Name	Link	Rahul
link		
Rahul		

**View All**

New Contact Edit New Opportunity

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

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

# RESULTS

Setup

Home Object Manager

User Interface

Rename Tabs and Labels

Custom Tabs

Custom Object Tabs

Action	Label	Tab Style	Description
Link	Basic	Light	
Link	Payment	Credit Card	
Link	Assess	Dark	
Link	Success	Web	

Web Tabs

No Web Tabs have been defined.

Visualforce Tabs

No Visualforce Tabs have been defined.

## Output Screenshots

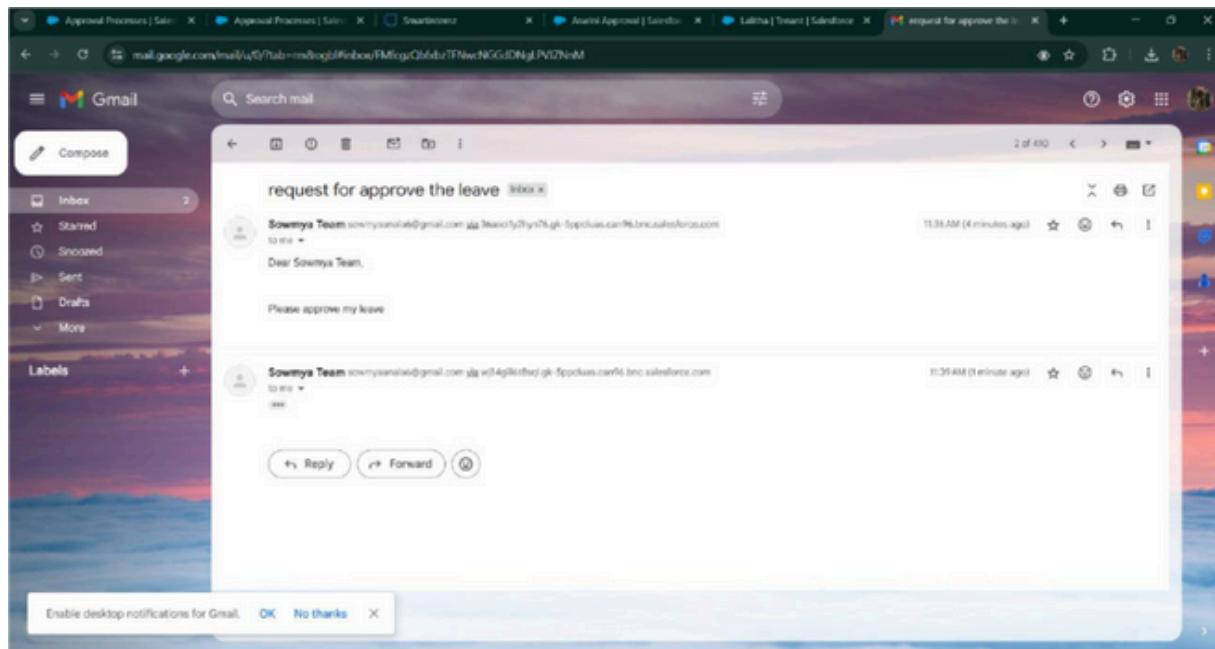
Lease Management    Payment    Tenants    property    lease

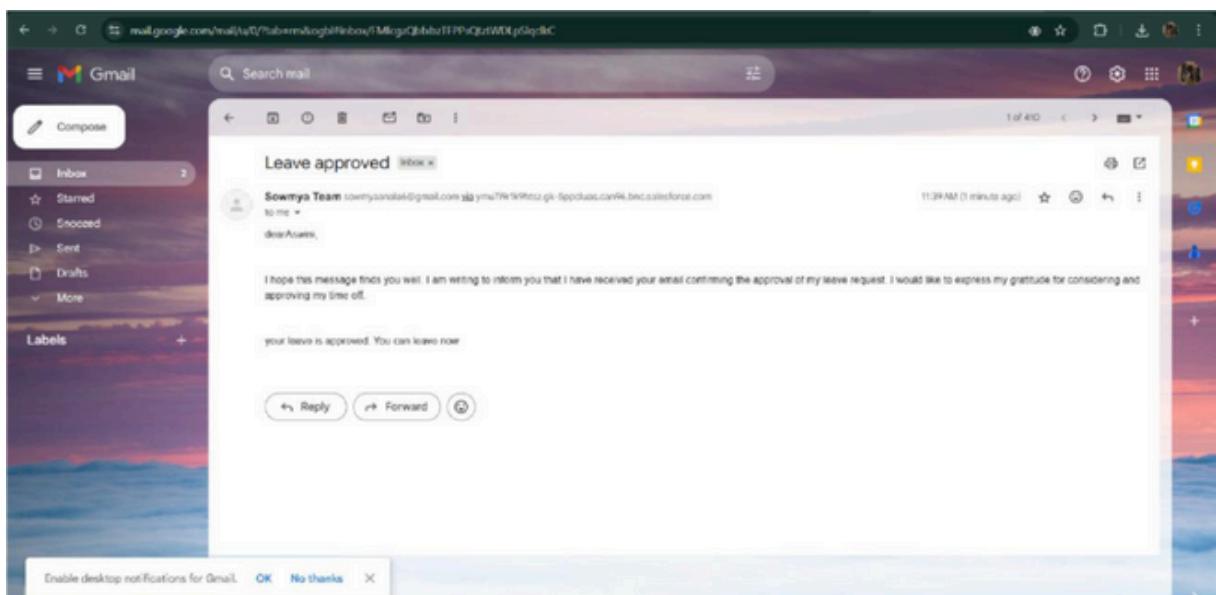
Approval History

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

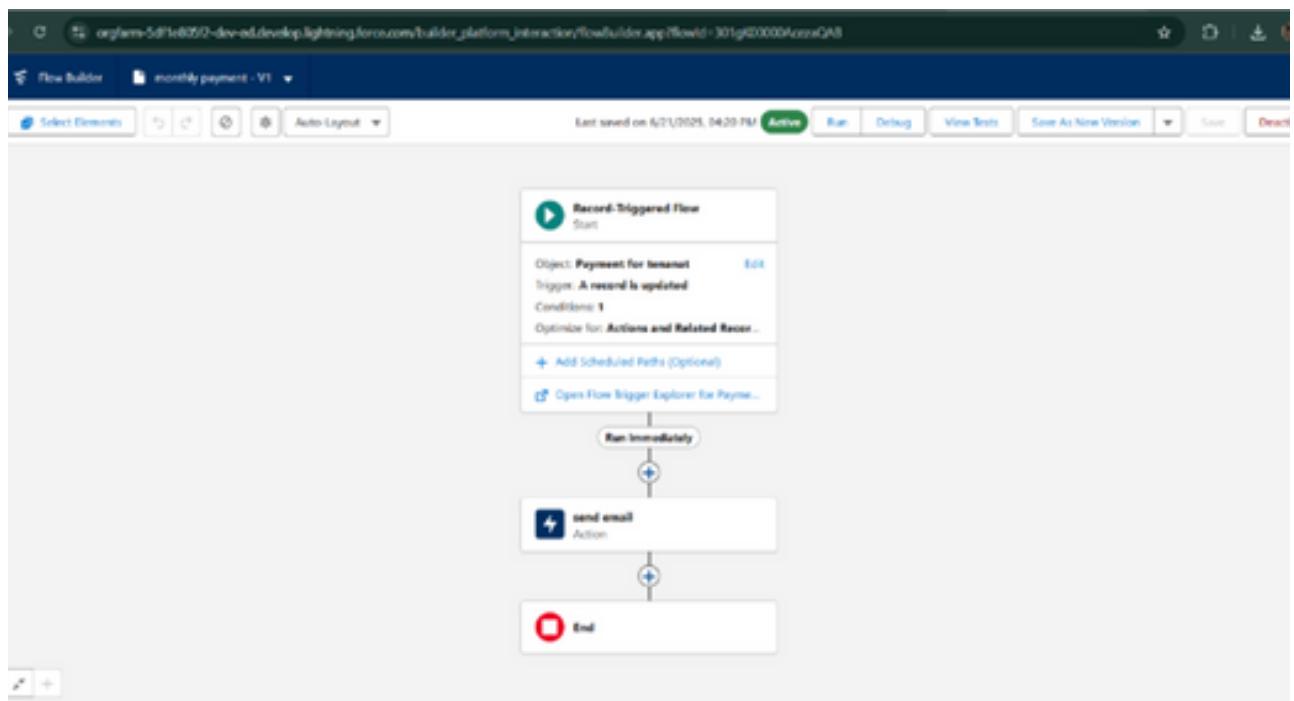
Step Name	Date	Status	Assigned To	Actual Approver	Comments
Step 1	6/23/2025, 5:39 AM	Approved	Soumya Team	Soumya Team	approved
Approval Request Submitted	6/23/2025, 5:39 AM	Submitted	Soumya Team	Soumya Team	leaving
Step 2	6/23/2025, 5:51 AM	Rejected	Soumya Team	Soumya Team	Rejected
Approval Request Submitted	6/23/2025, 5:52 AM	Submitted	Soumya Team	Soumya Team	leaving
Step 3	6/23/2025, 5:55 AM	Approved	Soumya Team	Soumya Team	Approved
Approval Request Submitted	6/23/2025, 5:55 AM	Submitted	Soumya Team	Soumya Team	leaving
Step 4	6/23/2025, 5:44 AM	Approved	Soumya Team	Soumya Team	Approval Approved
Approval Request Submitted	6/23/2025, 5:42 AM	Submitted	Soumya Team	Soumya Team	leaving

- Request for approve the leave





## ■ Leave rejected



mail.google.com/mail/u/0/?tab=rm&ogblInbox/IMigrQMBz2TQ8hC7hICGgGoogfDr

Gmail

Compose

Inbox

Starred

Snoozed

Sent

Drafts

More

Labels

Leave rejected

Sowmya Team <sowmyaanalad@gmail.com> 11:45 AM (0 minutes ago)

to me

Dear Arun,

I hope this email finds you well. Your contract has not ended. So we can't approve your leave.

Your leave has rejected.

Sowmya Team <sowmyaanalad@gmail.com> 11:45 AM (0 minutes ago)

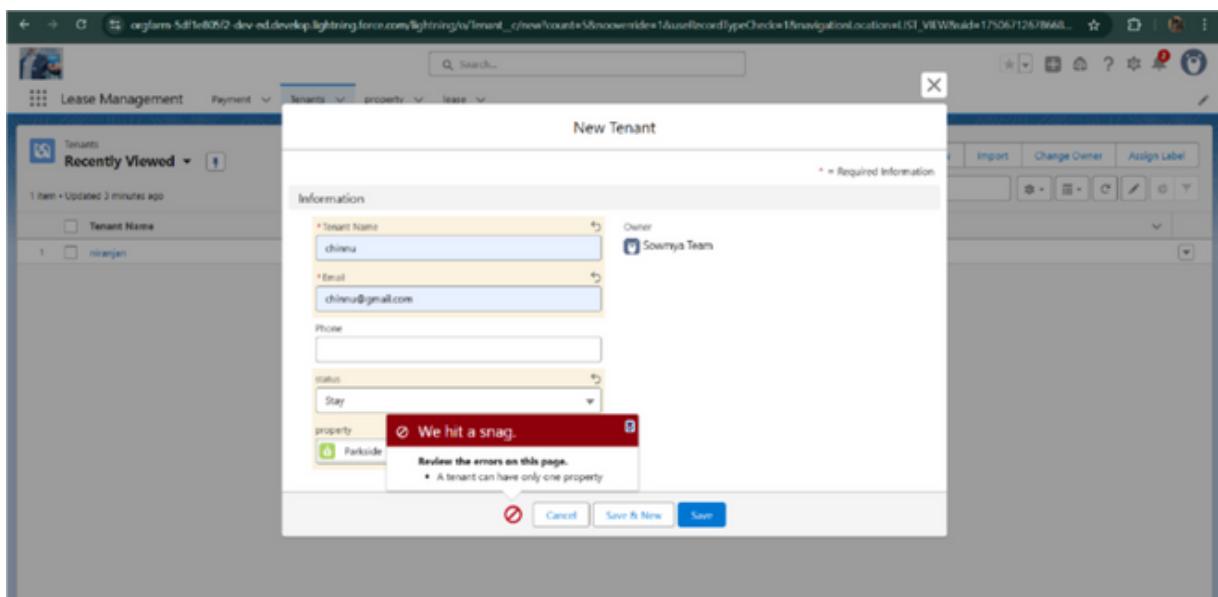
to me

...

Reply Forward

Enable desktop notifications for Gmail. OK No thanks

- Trigger error messages



- Approval process notifications

# ADVANTAGE

**Lower Upfront Costs:** Leasing eliminates large capital expenditures, allowing businesses to preserve cash for other needs.

**Flexibility:** Businesses can easily upgrade assets at the end of a lease term helping to avoid obsolescence.

**Tax Benefits:** Lease payments are often deductible as operating expenses, providing a tax advantage for the lessee.

# DISADVANTAGE

**Higher Long-Term Costs:** While initial costs are lower, the total cost of leasing can be higher than buying over the asset's lifespan.

**No Ownership Equity:** At the end of the lease, lessees do not gain ownership of the asset, and their payments do not build equity.

**Restrictions:** Leased assets often come with limitations, such as mileage caps for vehicles or restrictions on customization.

# CONCLUSION

## Summary of Achievements

The Salesforce implementation for Lease Management has successfully streamlined and automated critical aspects of the leasing lifecycle. Key accomplishments include:

### 1. Comprehensive Lease Lifecycle Management

- Implemented a robust system to manage lease agreements, from creation and amendments to renewals and terminations.

### 1. Automated Payment Processes

- Enabled scheduled rent tracking, payment reminders, and overdue alerts, reducing manual oversight and improving cash flow management.

## 1. Efficient Expiration and Renewal Workflows

- Automated notifications and streamlined renewal processes, ensuring timely follow-ups and improved lease retention rates.

## 1. Enhanced Property and Tenant Management

- Centralized information for properties, units, and tenants, improving operational visibility and collaboration.

## 1. Actionable Insights and Reporting

- Delivered custom reports and dashboards to monitor revenue, lease performance, and occupancy metrics, supporting data-driven decision-making.

## 1. Improved User Experience

- Provided intuitive interfaces and mobile accessibility for leasing agents, property managers, and tenants, enhancing productivity and satisfaction.

## 1. Secure and Scalable Solution

- Ensured role-based access, field-level security, and compliance with legal requirements, enabling a reliable and scalable system for lease management.

## 1. Integration with External Systems

- Established seamless connections with payment gateways and accounting systems for end-to-end process automation.

