

LEASE MANAGEMENT

College Name: Texcity Atrs & Science College

College Code: BRU4E

TEAM ID: NM2025TMID27666

TEAM MEMBERS:

Team Leader Name: MOHAMMED SABITH S

Email: ssabith448@gmail.com

Team Member1: AHAMED SUHAIB S

Email: suhaibsahvi1122@gmail.com

Team Member: UMAMAHESHWARI R

Email: umamaheshwarimanjunathan@gmail.com

Team Member: MAHAGANAPATHI S

Email: msmmahaganapathi@gmail.com

Team Member: MUSHRAF KHAN M

Email: mushrraf.0310.siya@gmail.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.



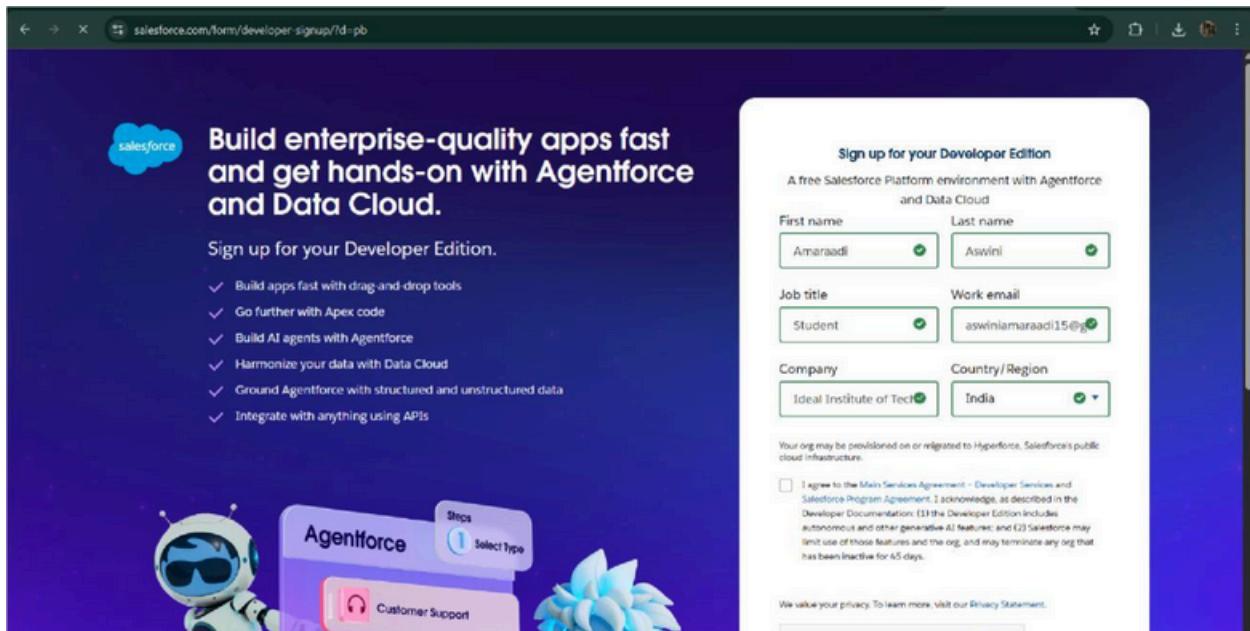
1.2 Purpose

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>



Created objects: Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The left sidebar lists various setup options 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 Accesible Roles. The main panel displays the 'Details' section for the 'property' object. It shows the API Name (property__c), Custom status (checked), Singular Label (property), Plural Label (property), and several checkboxes for configuration: Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (Deployed), Help Settings, and Standard salesforce.com Help Window. There are 'Edit' and 'Delete' buttons at the top right of the details panel.

org://arm-5d1fe005f2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. 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 Scrolling Rules. The main details pane shows the following fields:

Field	Value
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

At the top right of the details pane are 'Edit' and 'Delete' buttons.

org://arm-5d1fe005f2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTuJ/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'Lease' object. The left sidebar lists the same configuration options as the Tenant object. The main details pane shows the following fields:

Field	Value
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

At the top right of the details pane are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various configuration options under 'SETUP > OBJECT MANAGER'. The main area is titled 'Payment for tenantat' and displays the 'Details' tab. The 'Description' section contains fields for API Name ('Payment_for_tenantat_c'), Singular Label ('Payment for tenant'), and Plural Label ('Payment'). To the right, there's a list of checkboxes for object settings like 'Enable Reports', 'Track Activities', and 'Deployment Status'. At the bottom right of the main area are 'Edit' and 'Delete' buttons.

Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The sidebar on the left lists configuration options. The main area is titled 'property' and shows the 'Fields & Relationships' tab. A table lists the fields with their labels, names, data types, controlling fields, and indexing status. The fields listed are Address, Created By, Last Modified By, Name, Owner, property, property Name, sflq, and Type.

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)		
sflq	sflq__c	Text(18)		
Type	Type__c	Picklist		

Setup > Object Manager

Payment for tenantat

Fields & Relationships				
7 items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check for payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)		✓

Setup > Object Manager

lease

Fields & Relationships				
7 items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
property	property__c	Lookup(property)		✓
start date	start_date__c	Date		

SETUP > OBJECT MANAGER
Tenant

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

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

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: #007002

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...

[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

[Create](#)

- [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 ▼](#)

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

App Settings

[App Details & Branding](#)

[App Options](#)

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

User Profiles

[Navigation Items](#)

User Profiles

Choose the user profiles that can access this app.

Available Profiles

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

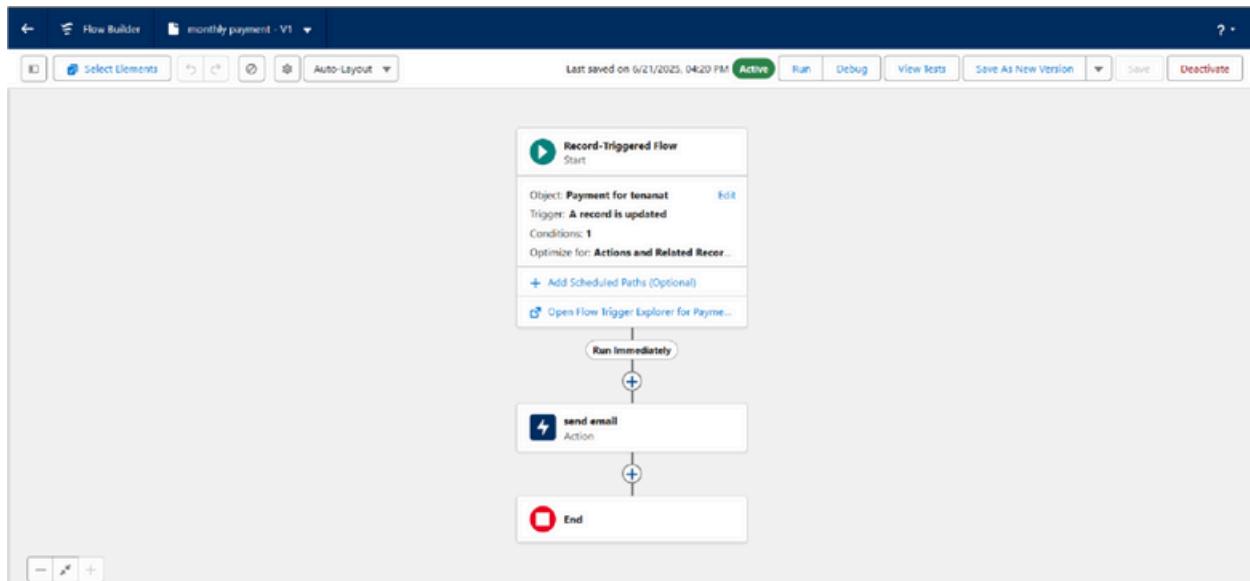
Selected Profiles

- [System Administrator](#)

[Up ▲](#) [Down ▼](#)

Screenshot of the Salesforce Lightning interface showing a list view of payments. The header shows 'Lease Management' and 'Payment'. The list view is titled 'Recently Viewed' and contains 5 items. The columns include 'Payment Name' and numbered rows 1 through 5 with names: Rahul, Jack, Raj, Sam, and Lahari.

Implemented Flows for monthly rent and payment success



To create a validation rule to a Lease Object

Validation Rule Edit

Role Name: lease_end_date

Active:

Description:

Error Condition Formula:

Example: Discount_Percent_c>0.30 More Examples...
Display an error if Discount is more than 30%

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

Functions:

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

Insert Field Insert Operator

End_date_c <= start_date_c

Check Syntax

Quick Tips

- Operators & Functions

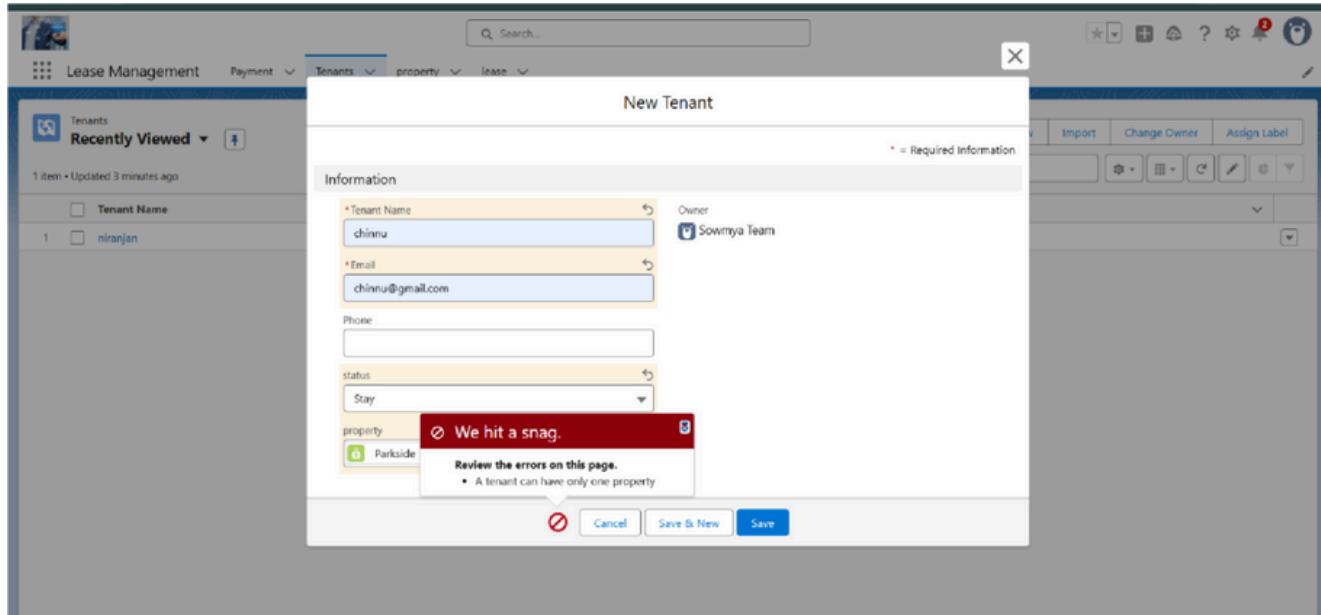
Help on this function

lease Validation Rule

Validation Rule Detail

Role Name	lease_end_date	Active	<input checked="" type="checkbox"/>
Error Condition Formula	End_date_c <= start_date_c	Error Location	start date
Error Message	Your End date must be greater than start date	Created By	Soumya Team, 6/19/2025, 5:37 AM
Description		Modified By	Soumya Team, 6/26/2025, 7:47 AM

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	Available For Use	✓
Template Unique Name	Leave_approved	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Soumya.T Team [Changed]		
Description			
Created By	Soumya.T Team	Created	6/20/2025, 1:06 AM
		Modified By	Soumya.T Team
			6/20/2025, 1:06 AM

The "Email Template" section contains the following details:

Subject	Leave approved	Send Test and Verify Merge Fields
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.		

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	Available For Use	✓
Template Unique Name	tenant_leaving	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Soumya.T Team [Changed]		
Description			
Created By	Soumya.T Team	Created	6/20/2025, 1:06 AM
		Modified By	Soumya.T Team
			6/20/2025, 1:06 AM

The "Email Template" section contains the following details:

Subject	request for approve the leave	Send Test and Verify Merge Fields
Plain Text Preview	Dear {Tenant__c.CreatedBy},	
Please approve my leave		

Setup Home Object Manager

Q_email template

Classic Email Templates

Lightning Email Templates

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

Classic Email Templates

Text Email Template Leave rejected

Preview your email template below.

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

Available For Use Last Used Date Times Used

Send Test and Verify Merge Fields

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

Setup Home Object Manager

Q_email template

Classic Email Templates

Lightning Email Templates

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

Classic Email Templates

Text Email Template Tenant Email

Preview your email template below.

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 [Channe]
Description	
Created By	Sowmya_Team 6/20/2025, 1:12 AM
Modified By	Sowmya_Team 6/20/2025, 1:12 AM

Available For Use Last Used Date Times Used

Send Test and Verify Merge Fields

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.

Approval Process creation

For Tenant Leaving :

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 equals Leaving		
Record Entitlity:	Administrator ONLY	Allow Submitters to Recall Approval Requests:	<input type="checkbox"/>
Approval Assignment Email Template:	Leave approved		
Initial Submitters:	Tenant Owner		
Created By:	Sowmya Team	Created Date:	6/20/2025, 3:18 AM
Modified By:	Sowmya Team	Modified Date:	6/26/2025, 11:02 PM

Initial Submission Actions

Action:	Type:	Description:
Record Lock		Lock the record from being edited
Email Alert		REASS2.B00R0V2.MV.JC8V2

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	Edit	1	step1		User:Sowmya.Team	Final Rejection

Apex Trigger

Create an Apex Trigger

```

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

```

Open

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

Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apex testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 64 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Use Problems

Create an Apex Handler class

Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apex testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 64 Go To

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

Open Entity Type Entity Related

Entity Type	Entity	Related
Entry Type		
Classes	testHandler	Name Extent Direct
Triggers	MonthlyEmailScheduler	test ApexTrigger Referenced
Pages		property CustomField References
Page Components		Tenant__c Subject References
Objects		Tenant__c Subject References
Static Resources		
Packages		

Open Filter the repository (* = any string) Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Use Problems

Developer Console - Google Chrome

orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/ui/common/apex/debug/ApexCSPage

File Edit Test Workspace Help < >

test.apex [] testHandler.apex [] MonthlyEmailScheduler.apex []

Code Coverage: None API Version: 64 Go To

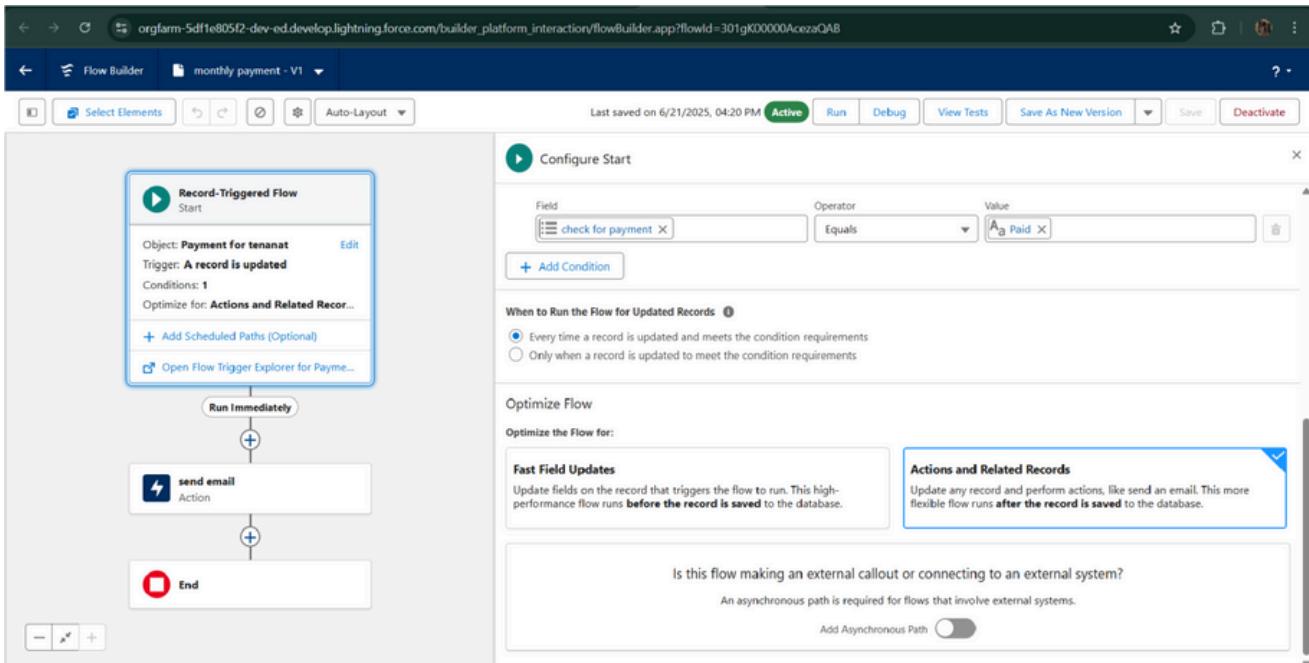
```

1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13
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 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

FLows



Flow Builder - monthly payment - V1

Last saved on 6/21/2025, 04:20 PM Active Run Debug View Tests Save As New Version Save Deactivate

Record-Triggered Flow Start

Object: Payment for tenant Edit

Trigger: A record is updated

Conditions: 1

Optimize for: Actions and Related Recor...

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Payne...

Run Immediately

send email Action

End

Configure Start

Select Object

* Object: Payment for tenant

Configure Trigger

Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements: All Conditions Are Met (AND)

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;
23

```

Open

Entity Type	Edition	Related
Classes	Name Namespace	Name Extent Direction
Triggers		CronTrigger Referenced...
Pages		CustomField...
Page Components		References
Objects		
Static Resources		
Packages		

Entity Type: Classes Edition: testHandler Related: MonthlyEmailScheduler

```

1  global class MonthlyEmailScheduler implements Schedulable {
2
3      global void execute(SchedulableContext sc) {
4
5          Integer currentDay = Date.today().day();
6
7          if (currentDay == 1) {
8
9              sendMonthlyEmails();
10         }
11     }
12 }
13
14
15 public static void sendMonthlyEmails() {
16
17     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19     for (Tenant__c tenant : tenants) {
20
21         String recipientEmail = tenant.Email__c;
22
23         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
24
25         String emailSubject = 'Re: Monthly Rent Payment Due';
26
27         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28
29         email.setPlainText(emailContent);
30
31         email.setSubject(emailSubject);
32
33         email.setTo(recipientEmail);
34
35         messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
36
37     }
38 }
39
40
41 }
42

```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left has categories like Email, Custom Code, Environments, and Jobs. Under Custom Code, Apex Classes is selected. The main area displays the Apex Class Detail for 'MonthlyEmailScheduler'. The class is active and was created by 'Somya.Team' on 6/23/2025 at 2:46 AM. The code editor shows the Apex code for the class.

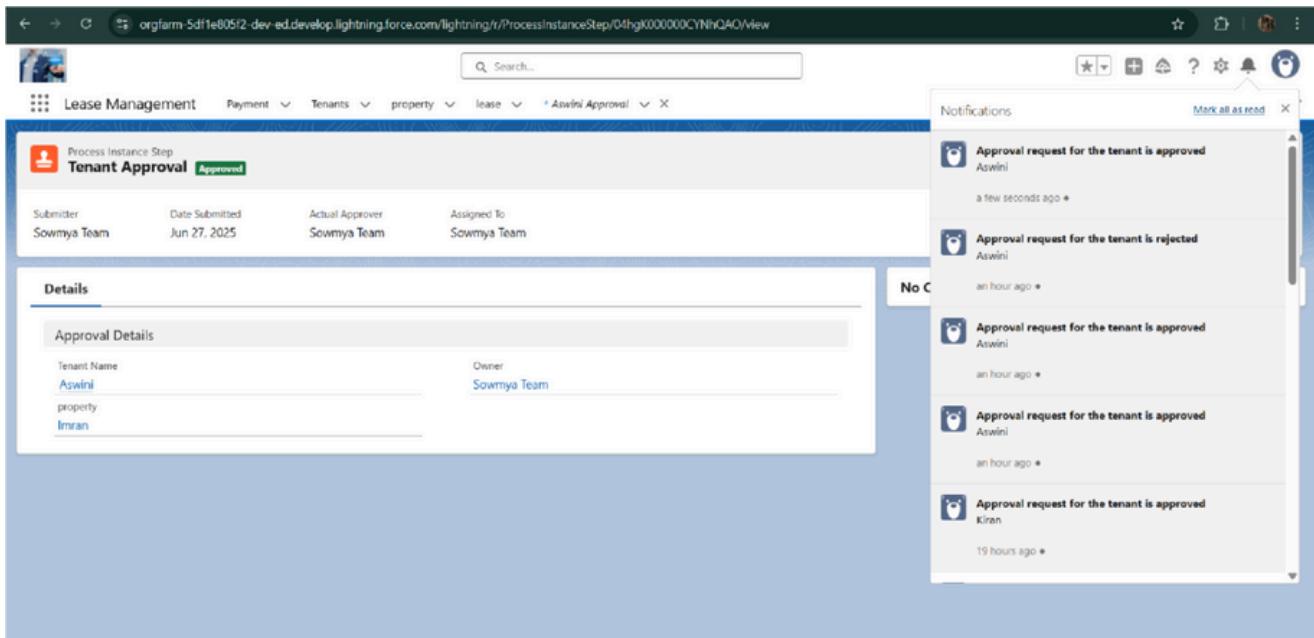
```

1  global class MonthlyEmailScheduler implements Schedulable {
2
3      global void execute(SchedulableContext sc) {
4
5          Integer currentDay = Date.today().day();
6
7          if (currentDay == 1) {
8
9              sendMonthlyEmails();
10         }
11     }
12 }
13
14
15 public static void sendMonthlyEmails() {
16
17     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19     for (Tenant__c tenant : tenants) {
20
21         String recipientEmail = tenant.Email__c;
22
23         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
24
25         String emailSubject = 'Re: Monthly Rent Payment Due';
26
27         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28
29         email.setPlainText(emailContent);
30
31         email.setSubject(emailSubject);
32
33         email.setTo(recipientEmail);
34
35         messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
36
37     }
38 }
39
40
41 }
42

```

The screenshot shows a Salesforce Lightning page for a 'Tenant' record named 'Aswini'. The page has a header with tabs: 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A search bar is at the top right. On the left, there's a sidebar with 'Related' and 'Details' tabs. The 'Details' tab is active, displaying fields for 'Tenant Name' (Aswini), 'Email' (aswiniamaraadi15@gmail.com), 'Phone' ((905) 223-5567), 'status' (Leaving), and 'property' (Imran). It also shows 'Owner' as 'Sowmya Team'. Below these are 'Created By' and 'Last Modified By' sections. At the bottom are 'Cancel' and 'Save' buttons. To the right is an 'Activity' sidebar with a 'Upcoming & Overdue' section. A context menu is open on the right side of the page, listing options like 'New Case', 'New Lead', 'Delete', 'Clone', 'Change Owner', 'Printable View', 'Submit for Approval', and 'Edit Labels'. The status bar at the bottom shows 'javascript:void(0)'.

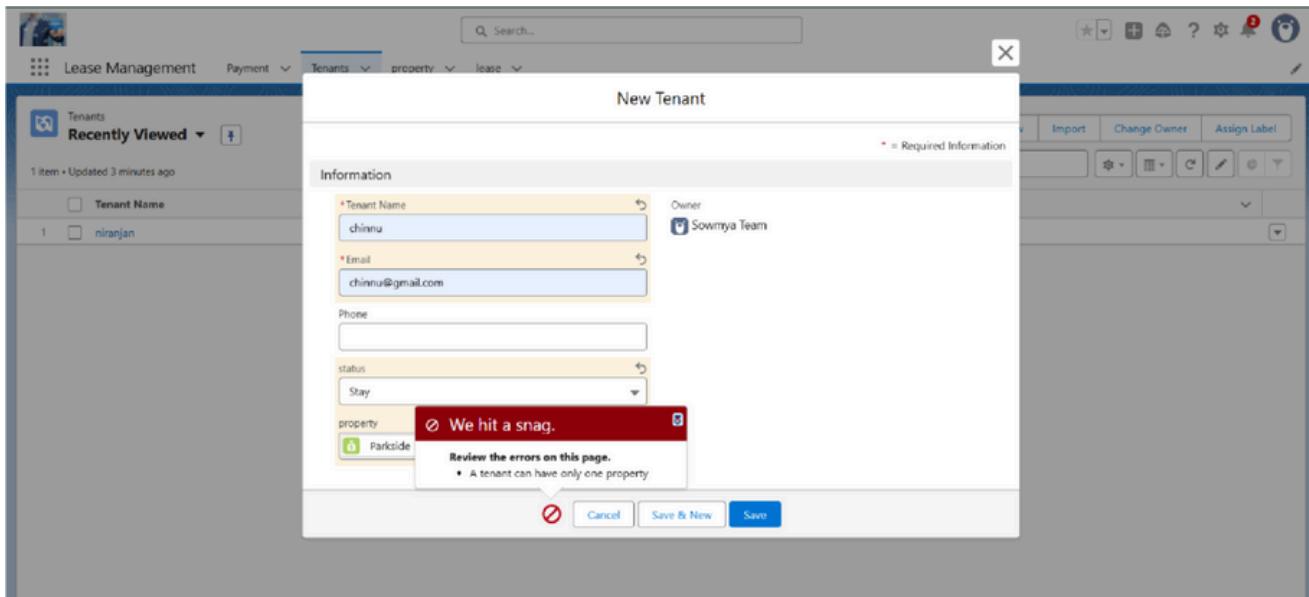
This screenshot is similar to the first one but includes a green success message at the top center: 'Tenant was submitted for approval.' The rest of the page content is identical to the first screenshot, showing the 'Aswini' tenant record details and the activity sidebar. The status bar at the bottom shows weather information (30°C Cloudy), system icons, and the date/time (27-06-2025).



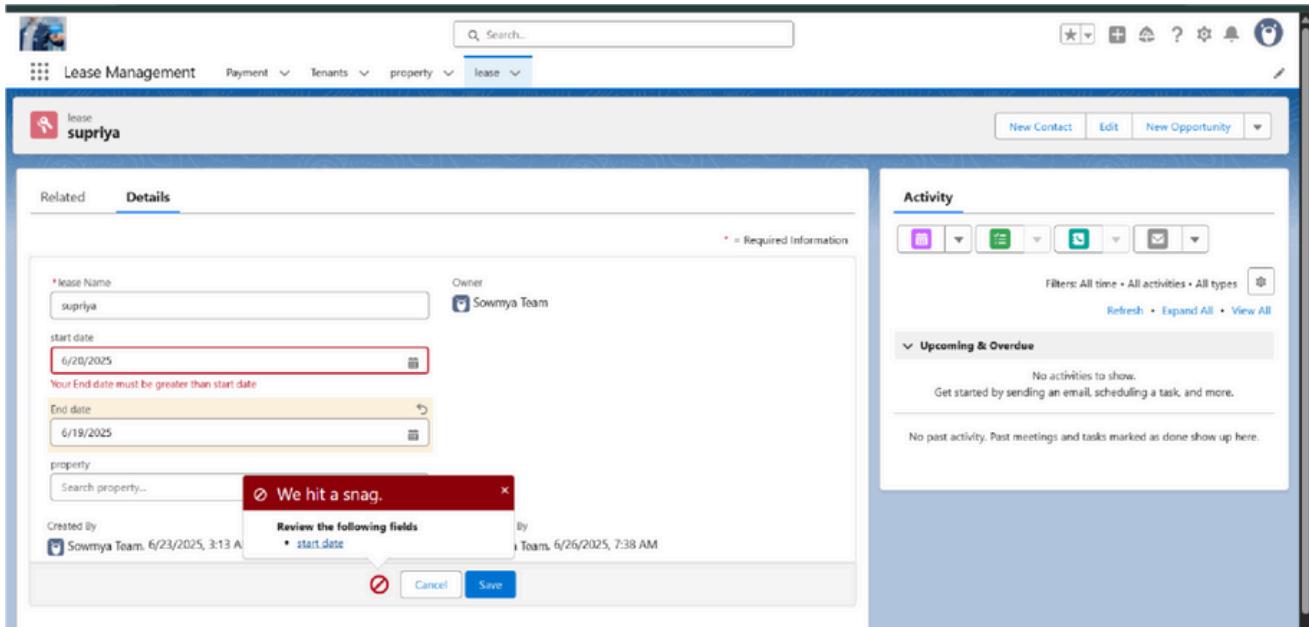
FUNCTIONAL AND PERFORMANCE TESTING

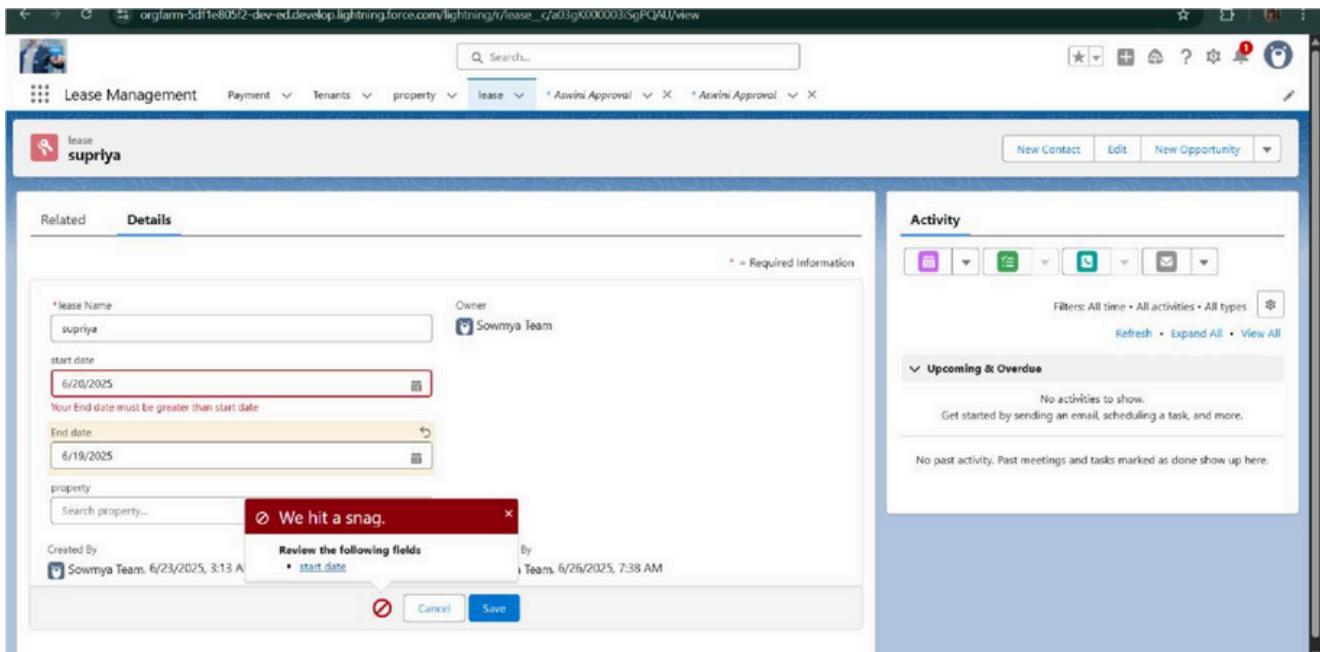
Performance Testing :

Trigger validation by entering duplicate tenant–property records

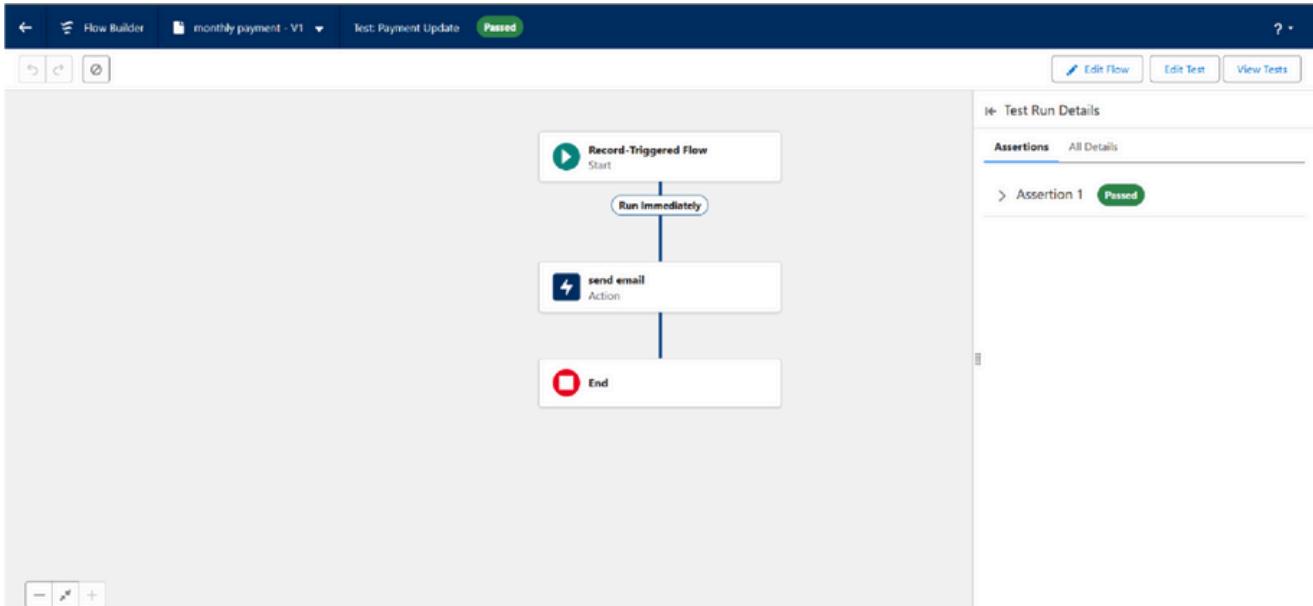


Validation Rule checking





Test flows on payment update



Approval process validated through email alerts and status updates

Lease Management Payment Tenants property lease *niranjan Approval

Tenant: niranjan

Related Details

* = Required Information

Tenant Name:	niranjan	Owner:	Sowmya Team
Email:	niranjan1506@gmail.com		
Phone:			
Status:	Stay		
Property:	Parkside Lofts	X	
Created By:	Sowmya Team, 6/23/2025, 2:33 AM	Last Modified By:	Sowmya Team, 6/23/2025, 3:58 AM
<input type="button" value="Cancel"/> <input type="button" value="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
Learn how to guide reps through the sales process.
Jun 20, 2025, 1:28 PM *

Lease Management Payment Tenants property lease

Tenant: niranjan

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, 3:53 AM	Approved	Sowmya Team
Approval Request Submitted	6/23/2025, 3:53 AM	Submitted	Sowmya Team

[View All](#)

Payment (2)

Payment Name	New
Jack	
Rahul	

[View All](#)

New Contact Edit New Opportunity

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

RESULTS

Output Screenshots

Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. On the left, there's a sidebar with 'User Interface' expanded, showing 'Rename Tabs and Labels' and 'Tabs' selected. The main area displays 'Custom Tabs' with a sub-section for 'Custom Object Tabs'. It lists four tabs: 'Lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'property' (Tab Style: Sack), and 'Tenant' (Tab Style: Map). Each tab has 'Edit | Del' links. Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both of which currently show 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

Email alerts

The screenshot shows the 'Lease Management' application within Salesforce. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. The main content area is titled 'Approval History' and shows a table of 8 items. The table columns are: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data in the table is as follows:

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

Request for approve the leave

The screenshot shows a Gmail inbox with a sunset background. There are two unread emails:

- request for approve the leave** from Sowmya Team (sowmyaanala6@gmail.com) at 11:36 AM (4 minutes ago). The message body says "Dear Sowmya Team, Please approve my leave".
- Sowmya Team** (sowmyaanala6@gmail.com) at 11:39 AM (1 minute ago). The message body says "to me" followed by three dots.

At the bottom left, there is a notification bar: "Enable desktop notifications for Gmail." with "OK" and "No thanks" buttons.

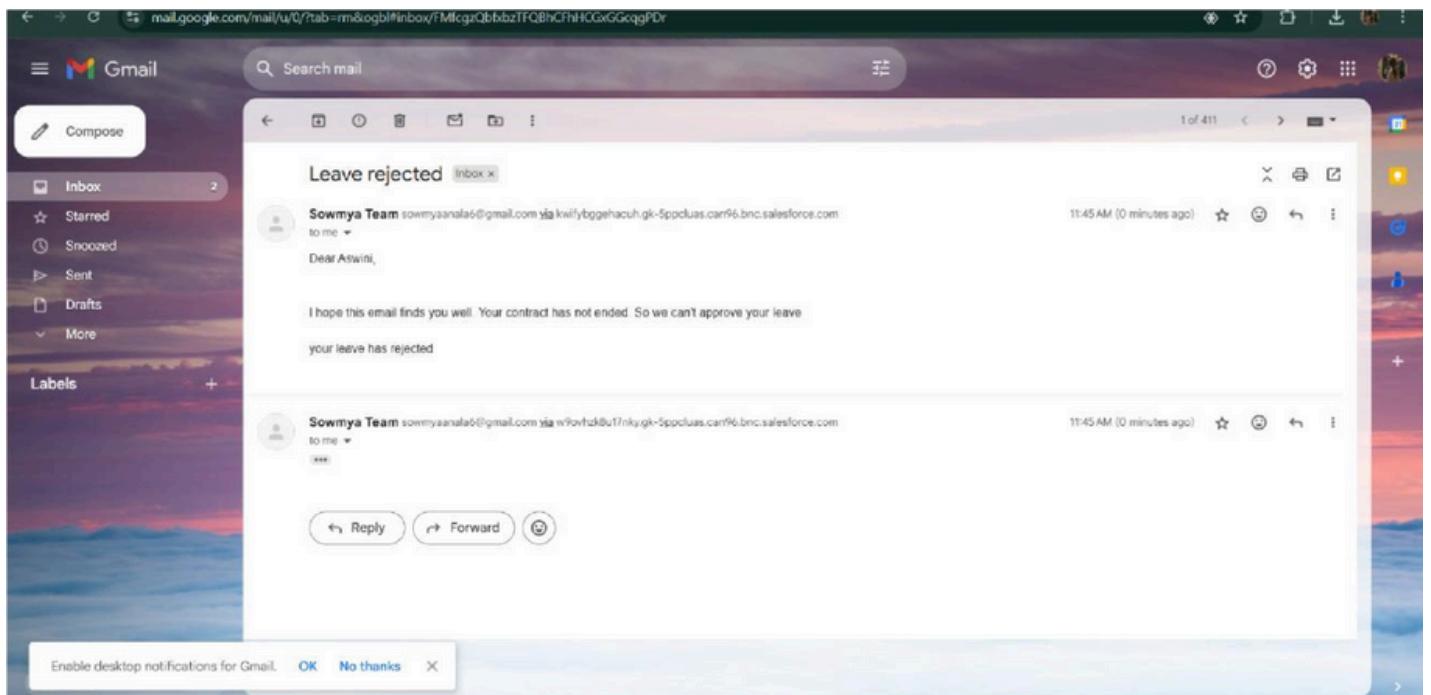
Leave approved

The screenshot shows a Gmail inbox with a sunset background. There is one unread email:

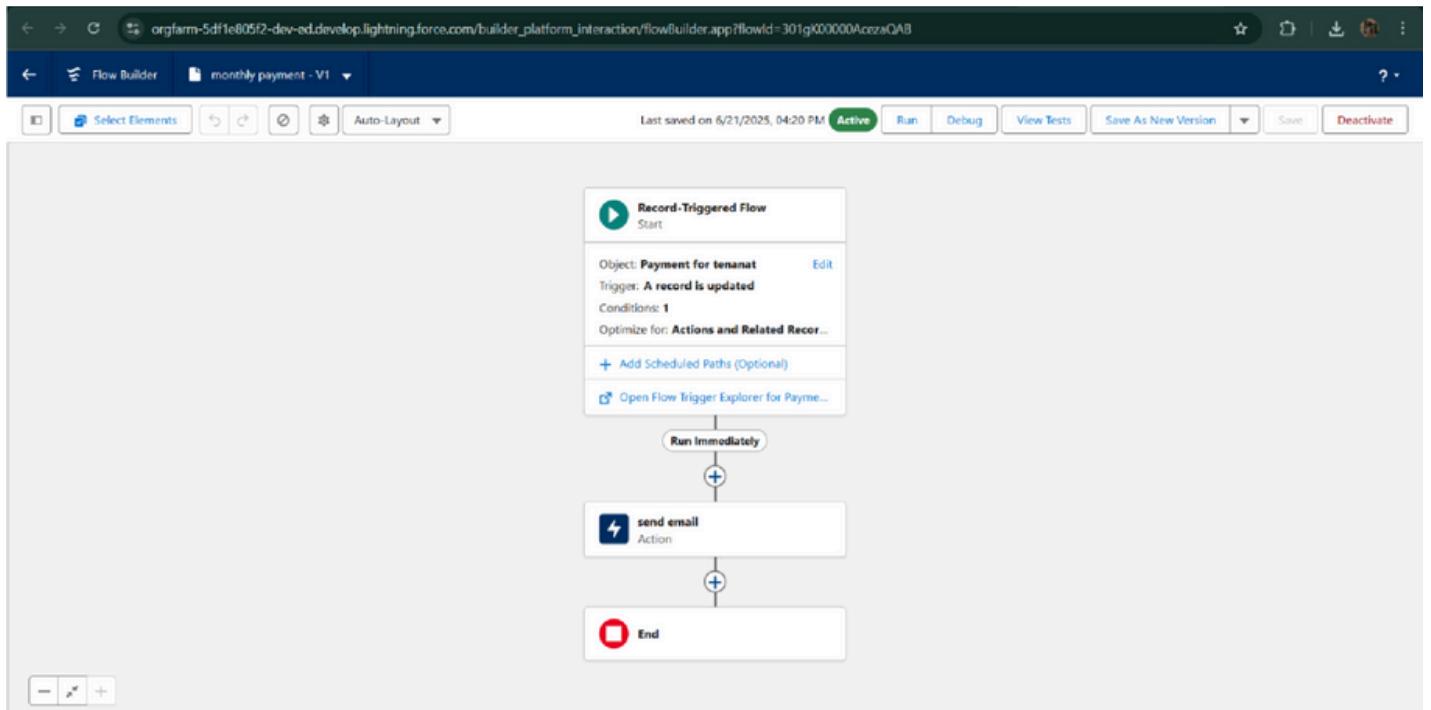
- Leave approved** from Sowmya Team (sowmyaanala6@gmail.com) at 11:39 AM (1 minute ago). The message body says "dearAswini, I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off. your leave is approved. You can leave now."

At the bottom left, there is a notification bar: "Enable desktop notifications for Gmail." with "OK" and "No thanks" buttons.

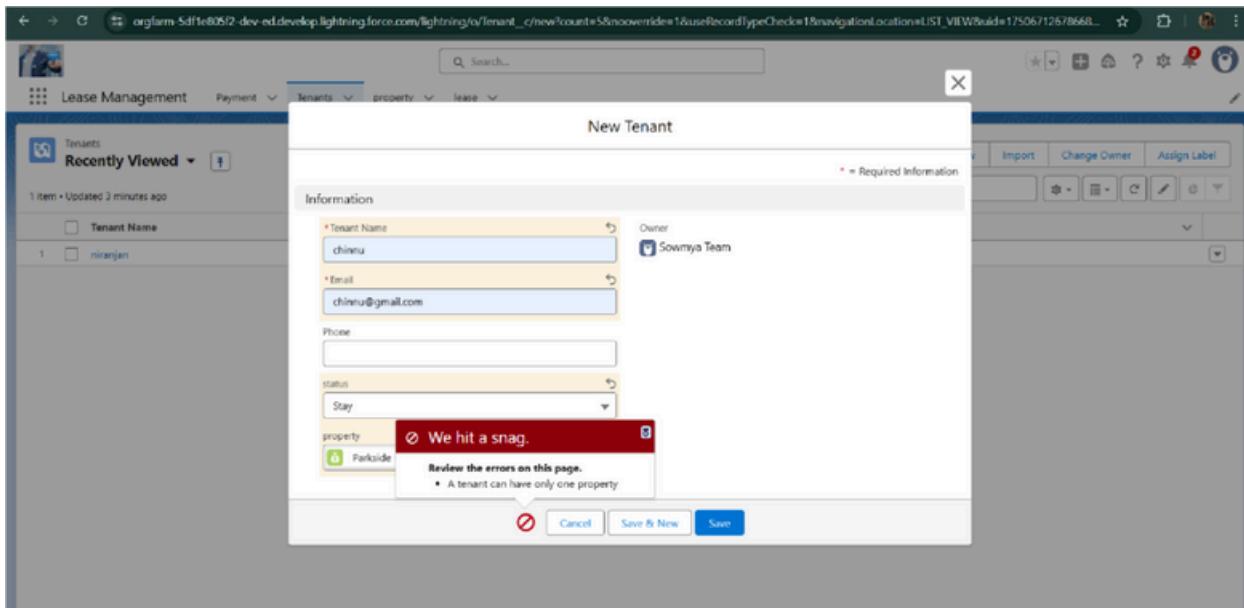
Leave rejected



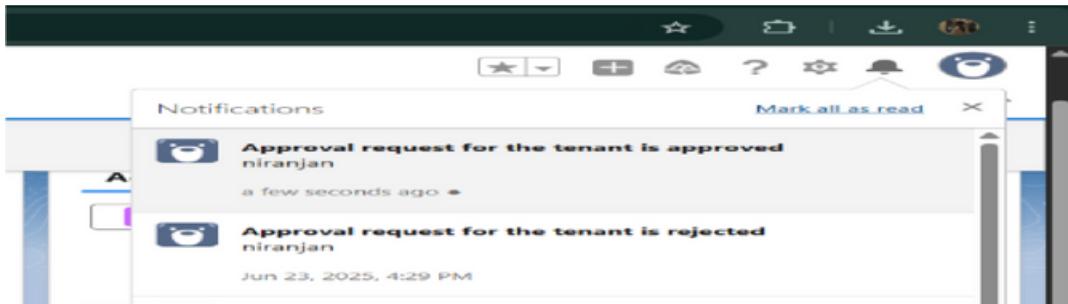
Flow runs



Trigger error messages



● Approval process notifications



ADVANTAGES & DISADVANTAGES

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});  
        }  
    }  
}
```