

# **Lease Management**

**College Name:** Sree Narayana Guru College,Chavadi, Coimbatore.

**College Code:** Bru36

**TEAM ID:**NM2025TMID26552

## **TEAM MEMBERS:**

**Team LeaderName:** MUHAMMED HARSHAD M.H

**Email:** [achuachuzz479@gmail.com](mailto:achuachuzz479@gmail.com)

**Team Member:** MANU R

**Email:** [manur9963@gmail.com](mailto:manur9963@gmail.com)

**Team Member:** MUHAMMED AJMAL A

**Email:** [ajmal20122005@gmail.com](mailto:ajmal20122005@gmail.com)

## INTRODUCTION

### 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.



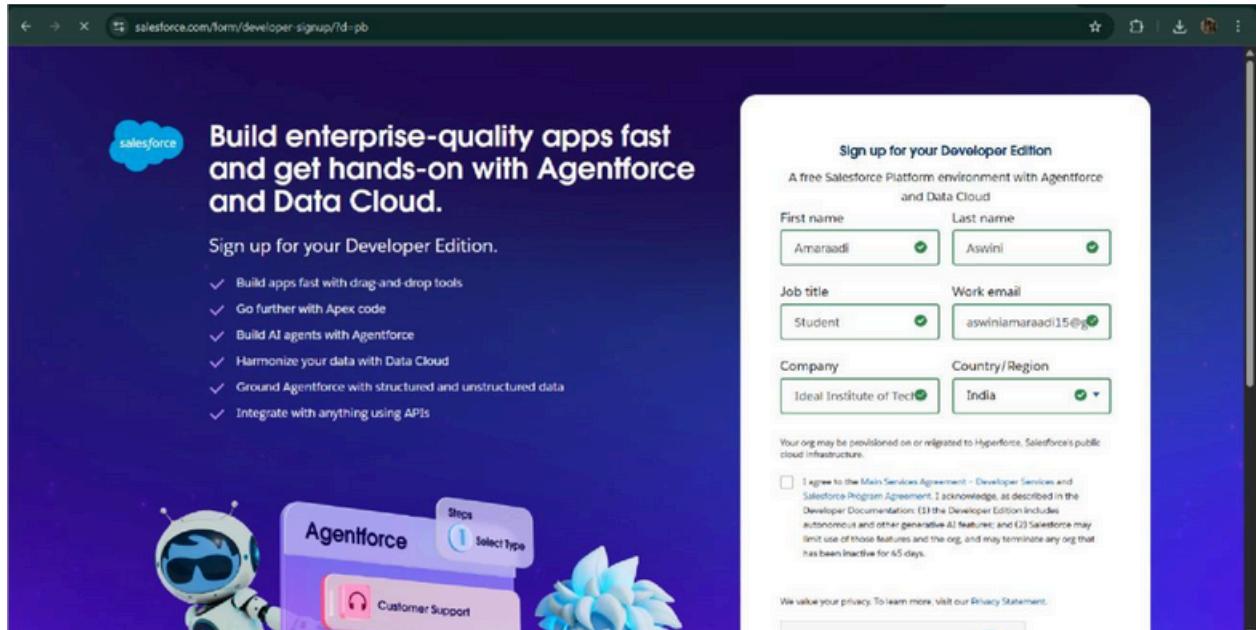
### Purpose

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.

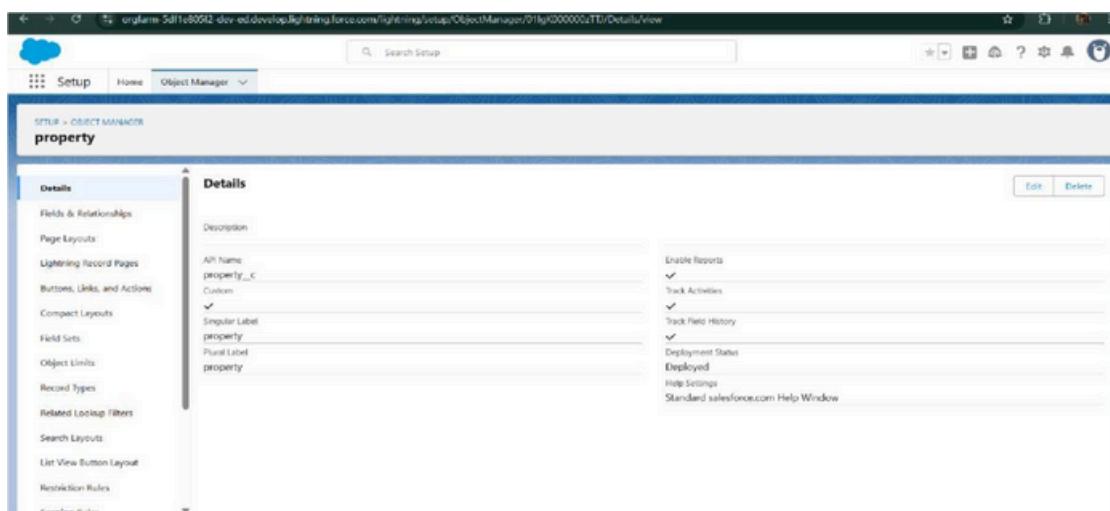
## DEVELOPMENT PHASE

Creating Developer Account:

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



- Created objects: Property, Tenant, Lease, Payment



orgId=5dfe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lg000000z7bN/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER Tenant

Details

Description

API Name: Tenant\_c

Custom: ✓

Singular Label: Tenant

Plural Label: Tenants

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

orgId=5dfe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lg000000z7uj/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER Lease

Details

Description

API Name: lease\_c

Custom: ✓

Singular Label: lease

Plural Label: lease

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

**Payment for tenant**

**Details**

Description

API Name: Payment\_for\_tenant\_\_c  
Custom: ✓  
Singular Label: Payment for tenant  
Plural Label: Payment

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

**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)		
sft	sft__c	Text(18)		
Type	Type__c	Picklist		

- Configured fields and relationships

**property**

**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)		
sft	sft__c	Text(18)		
Type	Type__c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER  
**Payment for tenantat**

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

**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 Home Object Manager

SETUP > OBJECT MANAGER  
**lease**

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 Scoping Rules

**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		

**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](#) [Leave 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](#) [Leave Management](#)

**App Settings**

[App Details & Branding](#)

[App Options](#)

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

**User Profiles**

**User Profiles**

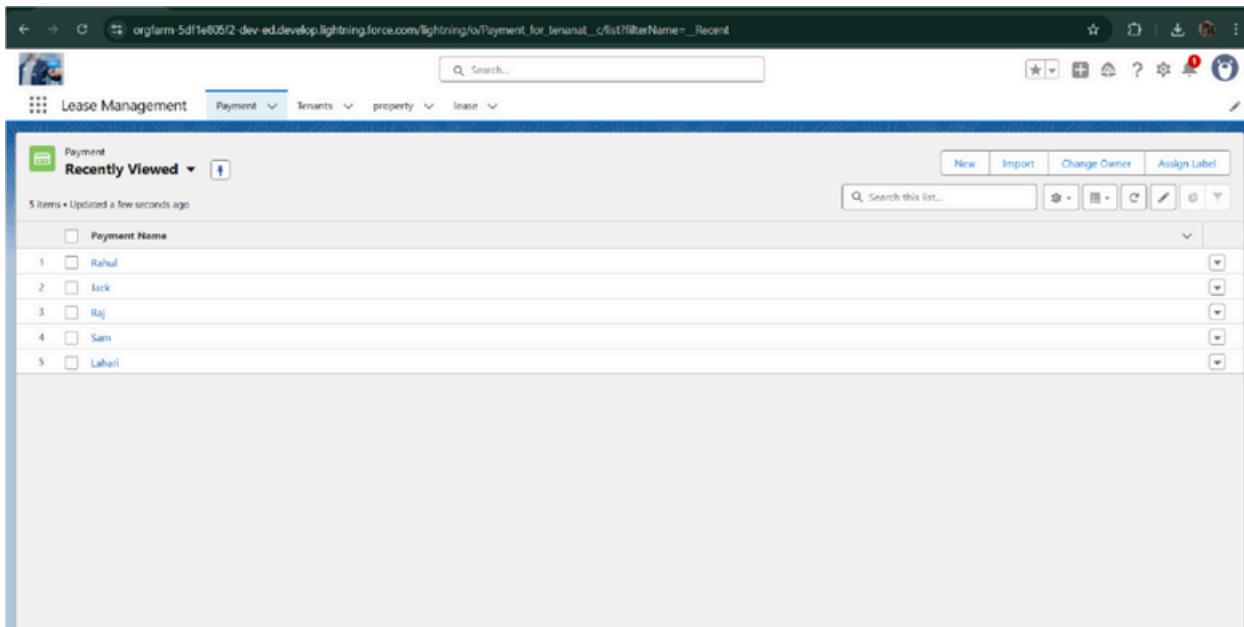
Choose the user profiles that can access this app.

**Available Profiles**

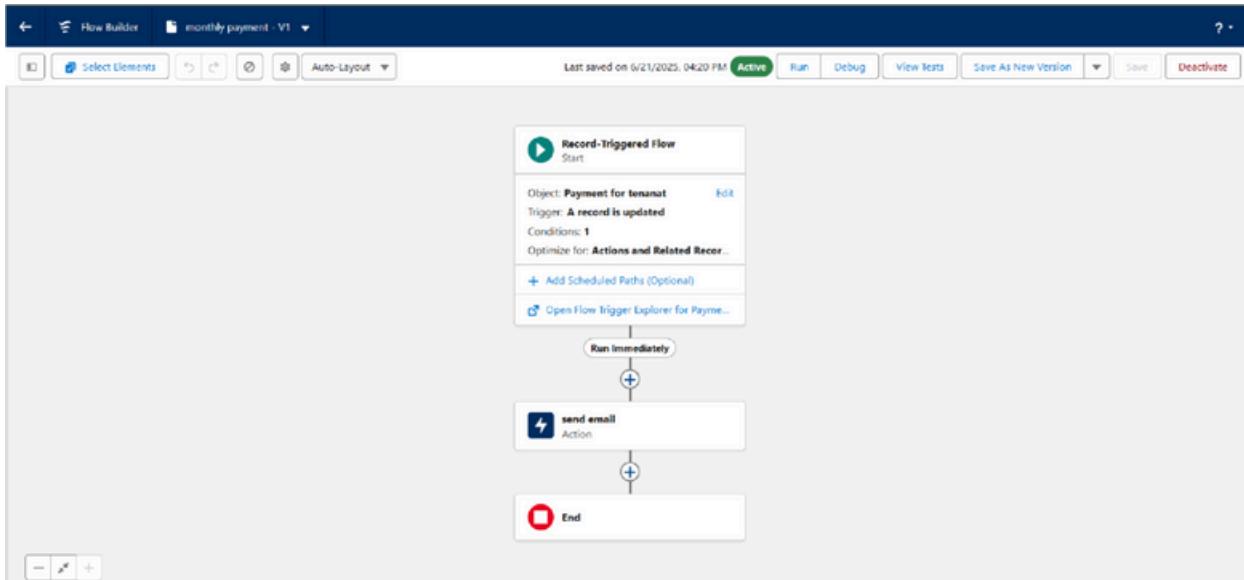
- [Analytics Cloud Integration User](#)
- [Analytics Cloud Security User](#)
- [Anypoint Integration](#)
- [Authenticated Website](#)
- [Authenticated Website](#)
- [B2B Reordering Portal Buyer Profile](#)
- [Contract Manager](#)
- [Custom: Marketing Profile](#)
- [Custom: Sales Profile](#)
- [Custom: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

**Selected Profiles**

- [System Administrator](#)



- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

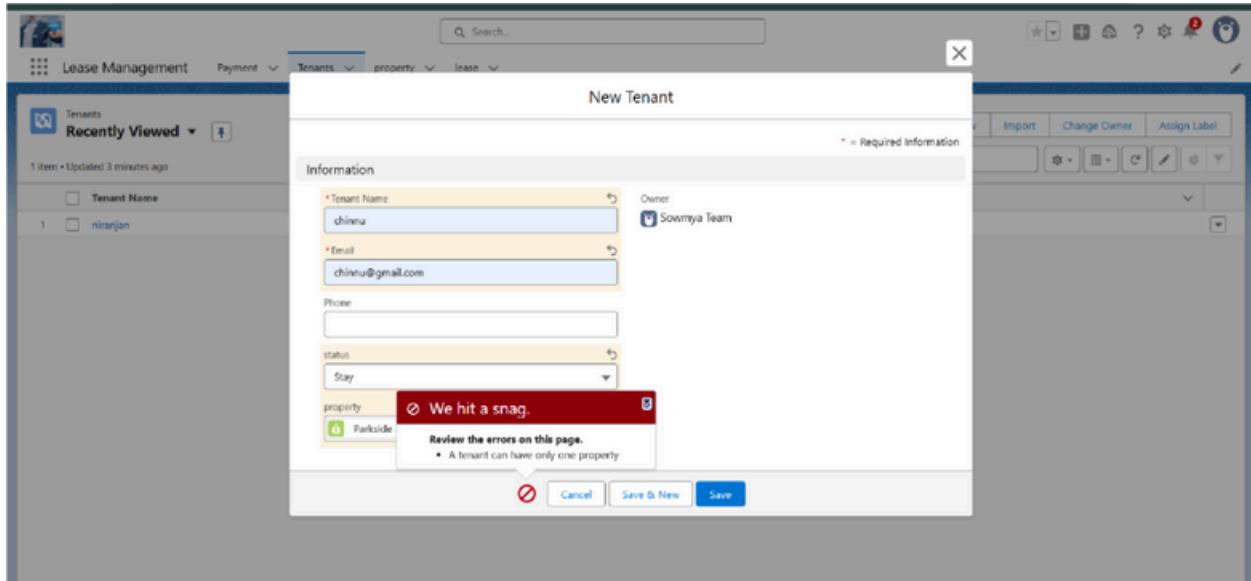
The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** orgfarm-Sdf1e8052-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTu/ValidationRules/page?address=%2F03dgk000000CgjQAG%2Fe%3FrefURL%3D92...
- Section:** SETUP > OBJECT MANAGER
- Object:** lease
- Form:** Validation Rule Edit
- Role Name:** lease\_end\_date
- Active:** checked
- Description:** (empty)
- Error Condition Formula:**
  - Example: `Discount_Percent__c < 0.30`
  - Display an error if Discount is more than 30%
  - If this formula expression is true, display the text defined in the Error Message area
- Formula Editor:**
  - Insert Field: End\_date\_c
  - Insert Operator: <=
  - Insert Selected Function: ABS(number)
  - Functions dropdown: ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN
- Error Message:** Your End date must be greater than start date.

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

- Page Header:** orgfarm-Sdf1e8052-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgk000000zTu/ValidationRules/03dgk000000CgjQAG/view
- Section:** SETUP > OBJECT MANAGER
- Object:** lease
- Form:** lease Validation Rule
- Validation Rule Detail:**
  - Validation Rule:** lease\_end\_date
  - Error Condition Formula:** End\_date\_c <= start\_date\_c
  - Error Message:** Your End date must be greater than start date.
  - Description:** (empty)
  - Created By:** Scormva Team, 6/19/2025, 5:37 AM
  - Modified By:** Scormva 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 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';
24
25         String emailSubject = 'Reminder: Monthly Rent Payment Due';
26
27         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28
29         email.setToAddresses(new String[]{recipientEmail});
30
31         email.setSubject(emailSubject);
32
33         email.setPlainTextBody(emailContent);
34
35     }
36 }
```

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

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', a specific template named 'Leave approved' is displayed. The template details include:

- Email Template Name:** Leave approved
- Template Unique Name:** Leave\_approved
- Encoding:** Unicode (UTF-8)
- Author:** Scorma Team [Changed]
- Description:** Created By: Scorma Team, 6/20/2025, 1:05 AM
- Last Used Date:** Not specified
- Times Used:** Not specified
- Modified By:** Scorma Team, 6/20/2025, 1:05 AM

The preview pane shows the email content:

```

Subject: Leave approved
Main 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.

where {tenant} is account. You can leave now.

```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', a specific template named 'tenant leaving' is displayed. The template details include:

- Email Template Name:** tenant leaving
- Template Unique Name:** tenant\_leaving
- Encoding:** Unicode (UTF-8)
- Author:** Scorma Team [Changed]
- Description:** Created By: Scorma Team, 6/20/2025, 1:05 AM
- Last Used Date:** Not specified
- Times Used:** Not specified
- Modified By:** Scorma Team, 6/20/2025, 1:05 AM

The preview pane shows the email content:

```

Subject: request for approve the leave
Main Text Preview:
Dear {Tenant__c.CreatedBy}.

Please approve my leave.

```

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

**Page Header:** Search Setup, Help for this Page

**Left Navigation:** Setup, Home, Object Manager, Email (selected), Classic Email Templates, Lightning Email Templates.

**Search Bar:** Q\_email template

**Section Header:** SETUP Classic Email Templates

**Section:** Text Email Template

**Email Template Name:** Leave rejected

**Description:** Preview your email template below.

**Email Template Detail:**

Email Template from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Sonoma Team (Changed)
Description	
Created By	Sonoma Team, 6/20/2025, 1:11 AM
Modified By	Sonoma Team, 6/20/2025, 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:

**Page Header:** Search Setup, Help for this Page

**Left Navigation:** Setup, Home, Object Manager, Email (selected), Classic Email Templates, Lightning Email Templates.

**Search Bar:** Q\_email template

**Section Header:** SETUP Classic Email Templates

**Section:** Text Email Template

**Email Template Name:** Tenant Email

**Description:** Preview your email template below.

**Email Template Detail:**

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

**Email Template:**

Send Test and Verify Merge Fields

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.

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. The left sidebar under 'Email' has 'Classic Email Templates' selected. A specific template, 'tenant payment', is highlighted in the list. The main content area displays the 'Email Template Detail' for 'tenant payment'. The template is 'Untitled Public Classic Email Templates' with the unique name 'tenant\_payment'. It is marked as 'Available For Use' and was last used on 6/20/2025 at 1:13 AM. The author is 'Screamer Team [Change]'. The subject is 'Confirmation of Successful Monthly Payment'. The plain text preview contains a message to the tenant confirming successful payment.

- Approval Process creation

### For Tenant Leaving:

The screenshot shows the Salesforce Setup interface with the search bar set to 'approval'. The left sidebar under 'Data' has 'Approval Processes' selected. A specific process, 'Tenant: TenantApproval', is highlighted. The 'Process Definition Detail' for 'TenantApproval' shows the process name is 'TenantApproval', unique name is 'TenantApproval', and it is active. The entry criteria is 'Is Active & status equals Stay'. The record editability is 'Administrator ONLY'. The approval assignment email template is 'Initial Submitter' and the initial owner is 'Tenant Owner'. The process was created by 'Screamer Team' on 6/20/2025 at 3:41 AM and modified by 'Screamer Team' on 6/20/2025 at 11:57 PM. The 'Initial Submission Actions' section includes a 'Record Lock' action type with the description 'Lock the record from being edited'. The 'Approval Steps' section shows one step named 'Step 1' with an assigned approver of 'User:Screamer Team' and a reject behavior of 'Final Rejection'.

- For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The process is named 'Tenant: check for vacant' and is set to run before insert. It has one step and an initial submission action of 'Record Lock'.

Action	Type	Description
Record Lock		Lock the record from being edited.

- Apex Trigger

## Create an Apex Trigger

```

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

```

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'testHandler.apex'. The trigger prevents insertions for Tenant\_\_c records. A modal window is open, showing a search results table for entity types like Classes, Triggers, Pages, etc.

Developer Console - Google Chrome  
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

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

testHandler.apc [ MonthlyTaskScheduler.apc ]

Code Coverage: None • API Version: 41 • Go To

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

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

## Create an Apex Handler class

Developer Console - Google Chrome  
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

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

testHandler.apc [ MonthlyTaskScheduler.apc ]

Code Coverage: None • API Version: 41 • 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(existin
10     }
11
12
13
14+     for (Tenant__c newTenant : newList) {
15
16
17         if (newTenant.Property__c != null) {
18             newTenantaddError('A t
19
20         }
21
22     }
23 }
```

Entity Type Entity Name Namespace Related

Entity Type	Entity Name	Namespace	Related
Empty List			
Classes	testHandler	MonthlyTaskScheduler	+< test ApexTrigger References
Triggers			+< property CustomerId References
Pages			+< Tenant__c Subject References
Page Components			+< Tenant__c Subject References
Objects			
Item Resources			
Packages			

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

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-Sd1t0e05f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

Test.Javascript testHandler.apc [ ] HostedDynamicScheduler.apc [ ]

Code Coverage Class Name API Version 64 Go To

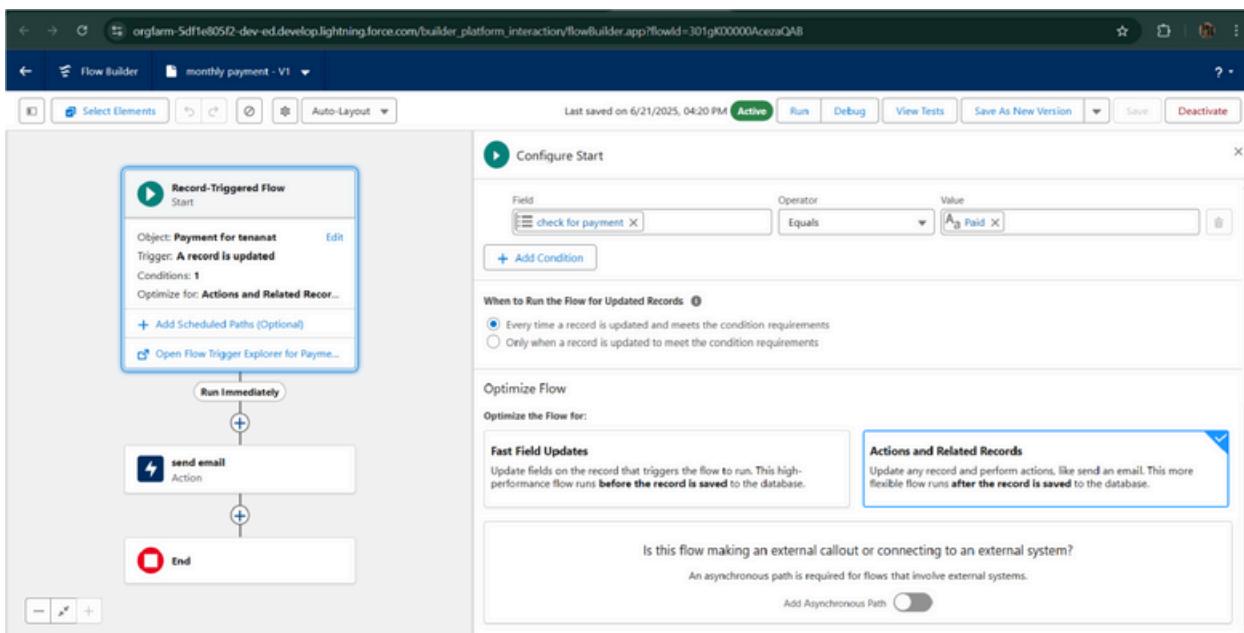
```

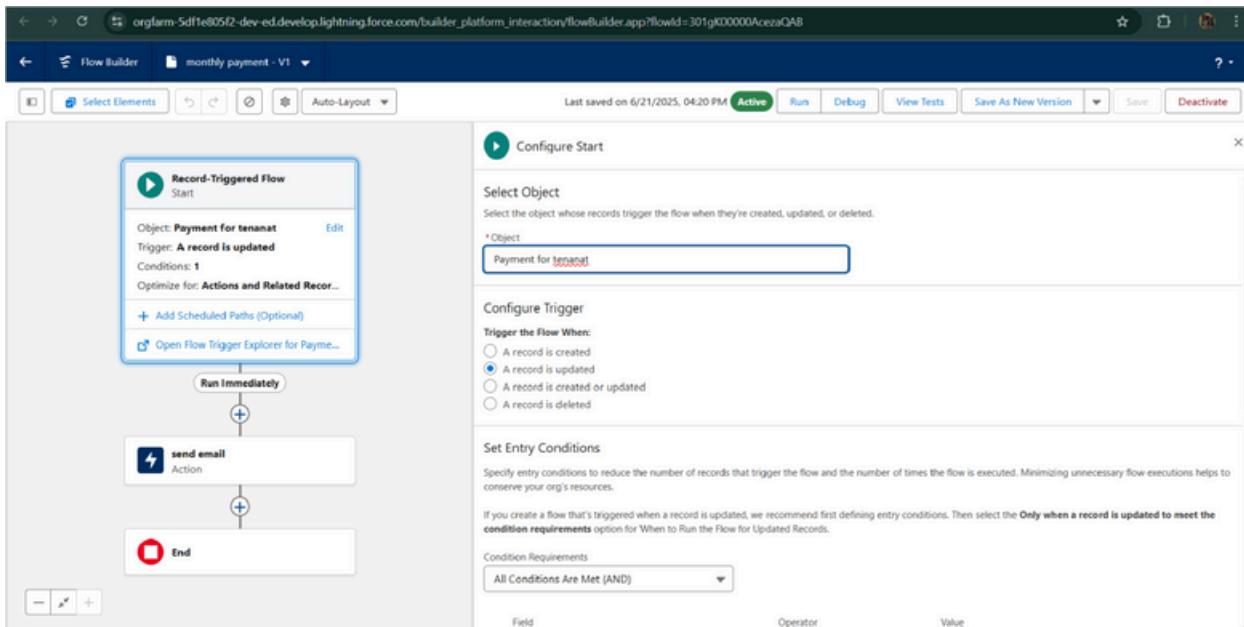
1 • public class testHandler {
2
3 •     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

Name 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             String recipientEmail = tenant.Email__c;
  
```

Entity Type: Classes  
Name: MonthlyEmailScheduler Namespace: **testhandler**

Entity Type	Edition	Related
Classes	Name: MonthlyEmailScheduler	Name: testhandler CustomType: References Email: References ← Tenant__c: Collected ← Tenant__c: Subject: References
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

```

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 = 'Dear Sir/Madam, 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 = 'Rental Reminders: Monthly Rent Payment Due';
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailContent();
33
34             messaging.send(mail);
35         }
36     }
37 }
38

```

## Schedule Apex class

The screenshot shows the Salesforce Setup interface with the search bar set to "apex". The left sidebar is expanded to show categories like Email, Custom Code, and Environments. Under "Apex Classes", the "MonthlyEmailScheduler" class is selected. The main pane displays the Apex Class Detail for "MonthlyEmailScheduler". The "Name" field is set to "MonthlyEmailScheduler". The "Status" is "Active". The "Created By" field shows "Sonoya Team" and the "Created Date" is "6/23/2025, 2:46 AM". The "Last Modified By" field shows "Sonoya Team" and the "Last Modified Date" is "6/23/2025, 2:47 AM". The "Code Coverage" is listed as "0% (0/15)". Below the details, there are tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags". The "Class Body" tab is active, showing 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 = 'Dear Sir/Madam, 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 = 'Rental Reminders: Monthly Rent Payment Due';
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailContent();
33
34             messaging.send(mail);
35         }
36     }
37 }
38

```

The screenshot shows a Salesforce Lightning page for a 'Tenant' record. The record details are as follows:

- Tenant Name:** Aswini
- Email:** aswinimaraadi15@gmail.com
- Phone:** (905) 223-5567
- Status:** Leaving
- Property:** Imran

The record was created by Sowmya Team on 6/26/2025, 6:05 AM, and last modified by Sowmya Team on 6/26/2025, 11:06 PM.

In the top right corner, there is a context menu with options like New Case, New Lead, Delete, Close, Change Owner, Printable View, Submit for Approval, and Edit Labels.

The screenshot shows the same Salesforce Lightning page for the 'Tenant' record for 'Aswini'. A green success message at the top indicates that the record has been submitted for approval.

The record details remain the same as in the previous screenshot.

In the top right corner, the context menu now includes 'Submit for Approval' and 'Edit Labels' under the 'Submit for Approval' section.

The system status bar at the bottom shows the date as 27-06-2025 and the time as 12:46.

Lease Management

Process Instance Step Tenant Approval Approved

Submitter: Sowmya Team Date Submitted: Jun 27, 2025 Actual Approver: Sowmya Team Assigned To: Sowmya Team

Details

Approval Details

Tenant Name: Aswini  
Owner: Sowmya Team  
Property: Imran

Notifications

Approval request for the tenant is approved Aswini a few seconds ago

Approval request for the tenant is rejected Aswini an hour ago

Approval request for the tenant is approved Aswini an hour ago

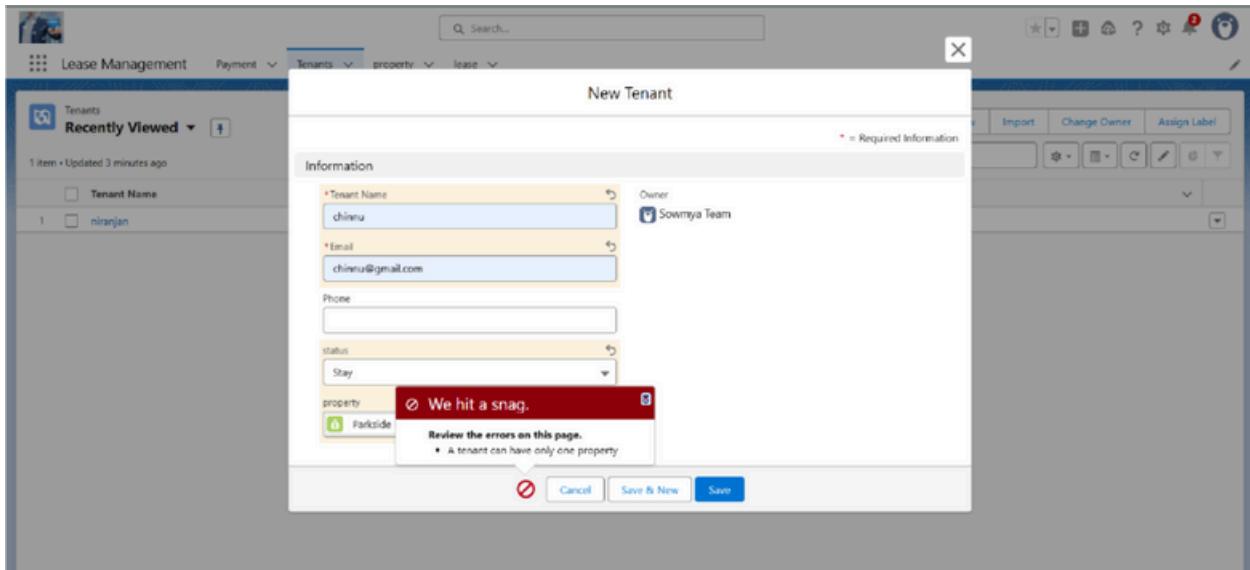
Approval request for the tenant is approved Aswini an hour ago

Approval request for the tenant is approved Kiran 19 hours ago

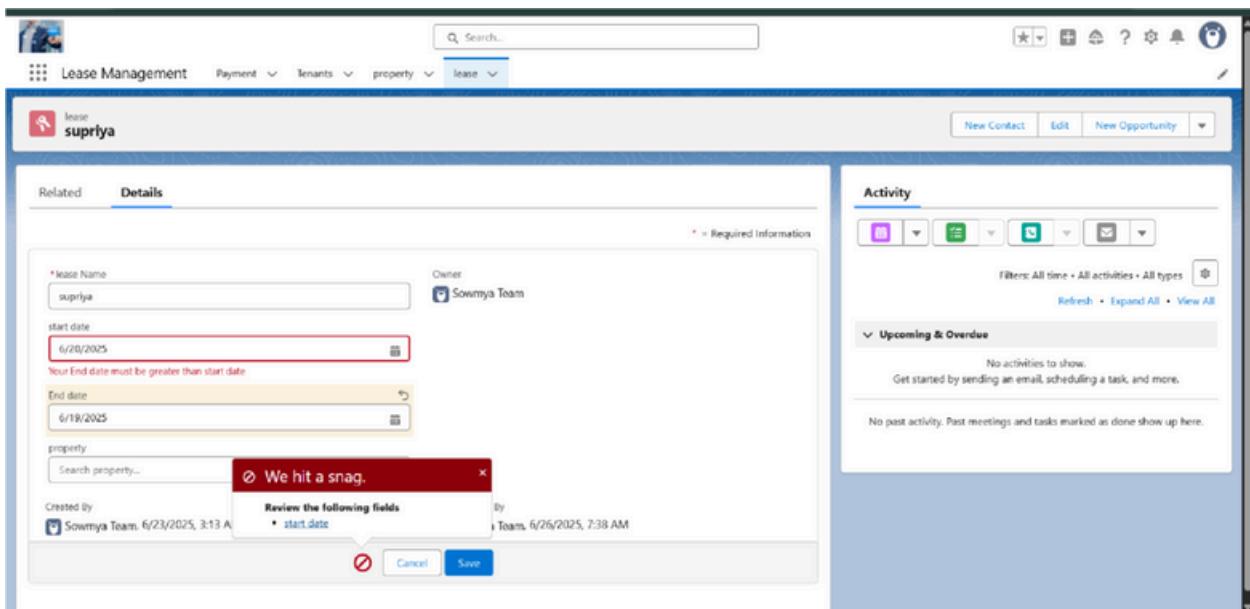
# FUNCTIONAL AND PERFORMANCE TESTING

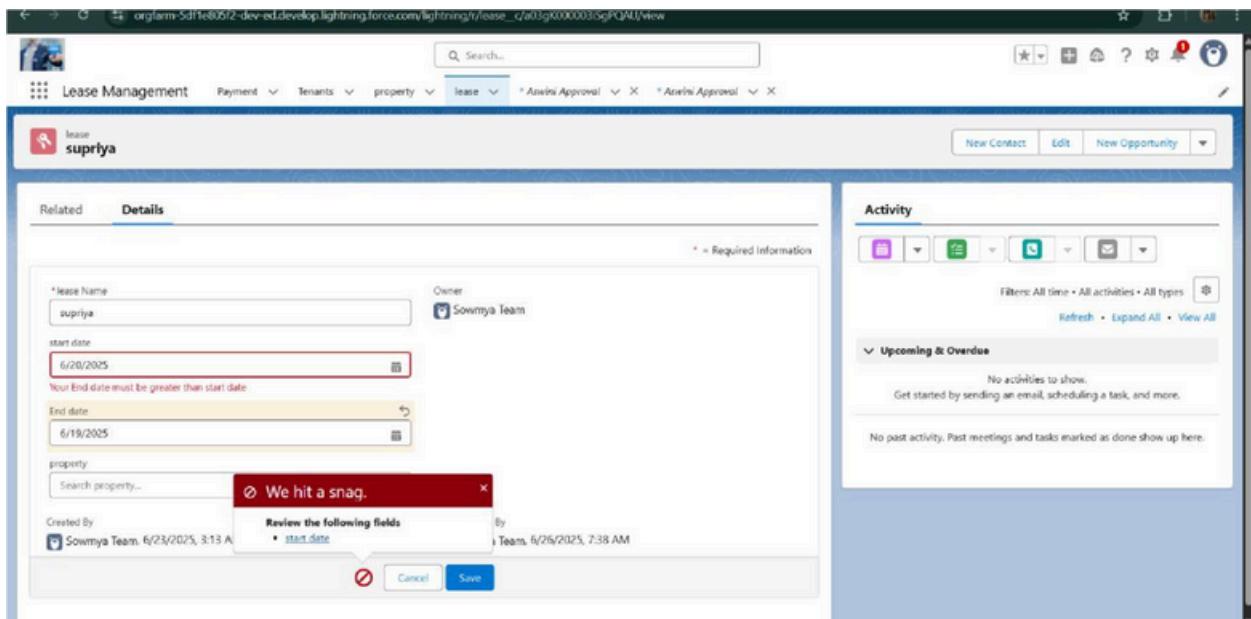
## Performance Testing

- Trigger validation by entering duplicate tenant-property records

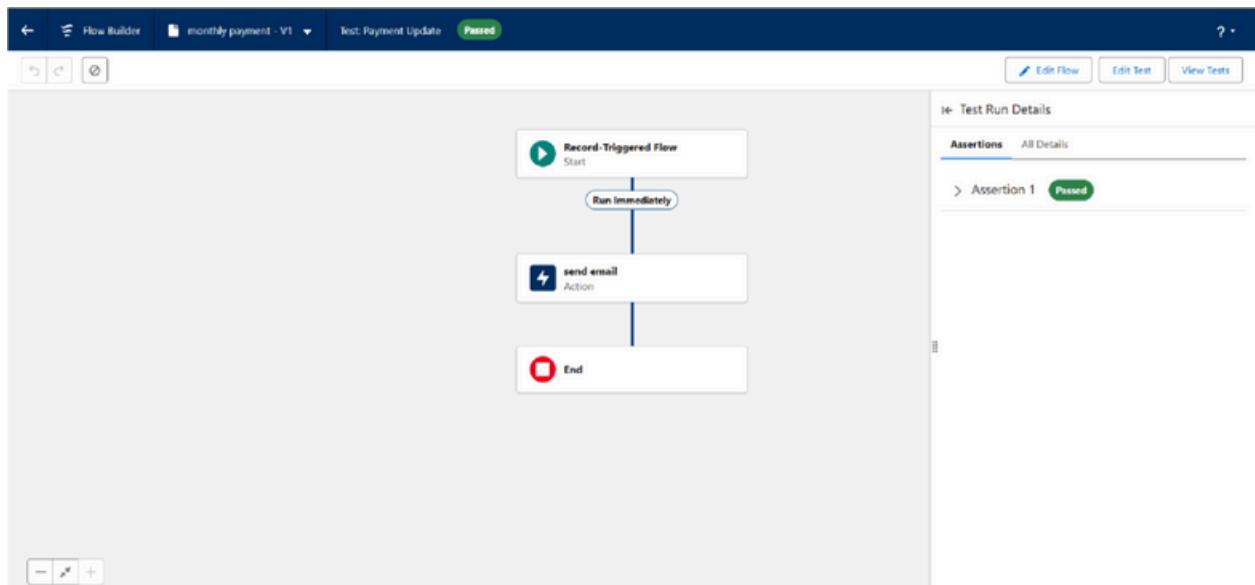


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows a CRM interface for 'Lease Management'. On the left, a 'Tenant' record for 'niranjan' is displayed with fields for Name, Email, Phone, Status, and Property. The property is listed as 'Parkside Lofts'. The right side shows a 'Notifications' sidebar with several messages:

- 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)

The screenshot shows a CRM interface for 'Lease Management'. On the left, the 'Approval History' section lists six steps, all of which have been approved. The steps are:

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

Below this, the 'Payment' section shows two entries for 'Jark' and 'Rahul'.

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

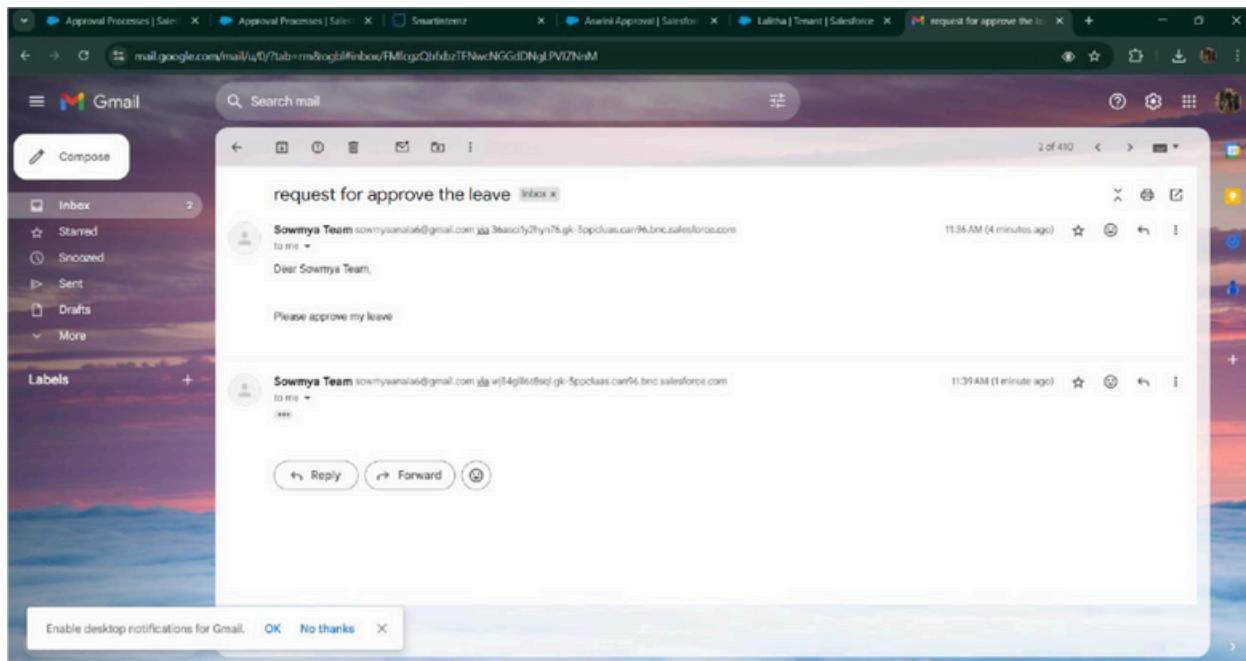
The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. It displays three main sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Lease', 'Payment', 'Account', and 'Contact', each with a 'Tab Style' icon (e.g., Keys, Credit card, Back, Map) and a 'Description' column. The 'Web Tabs' and 'Visualforce Tabs' sections both indicate 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

- Email alerts

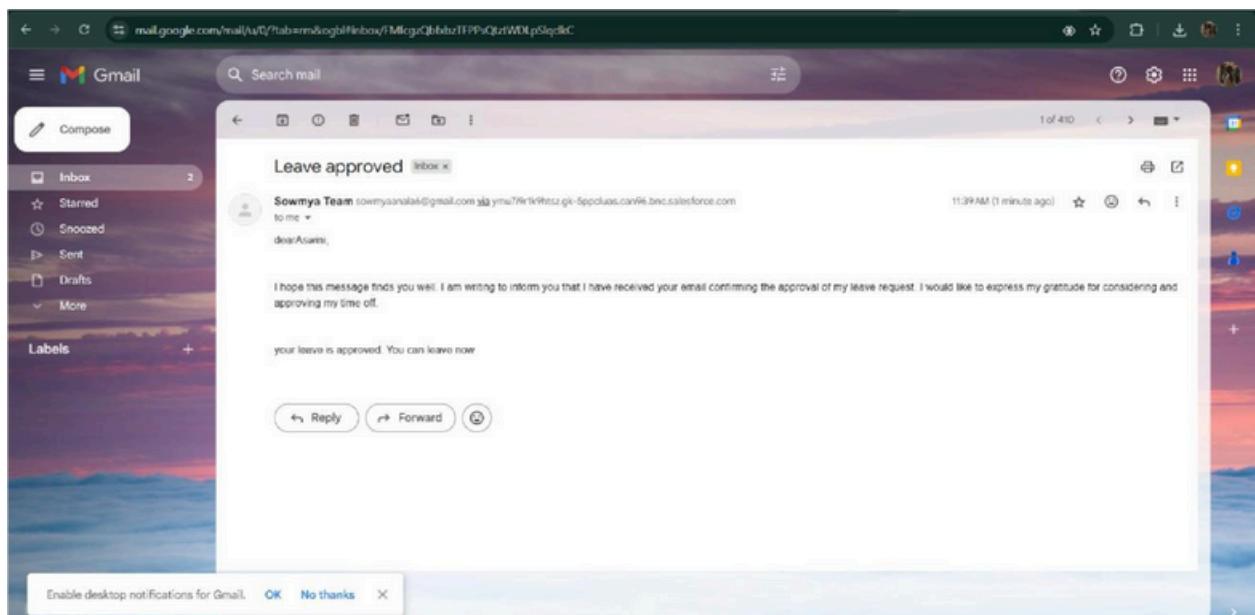
The screenshot shows the 'Approval History' list view within the 'Lease Management' application. The table has columns: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data shows a sequence of steps involving 'Step 1' and 'Approval Request Submitted' at various times, with statuses like Approved, Submitted, Rejected, and Pending, and comments such as 'approved', 'leaving', 'Rejected', 'Leaving', 'Approved', 'leaving', 'Approval Approved', and 'Leaving'.

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/23/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/23/2025, 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

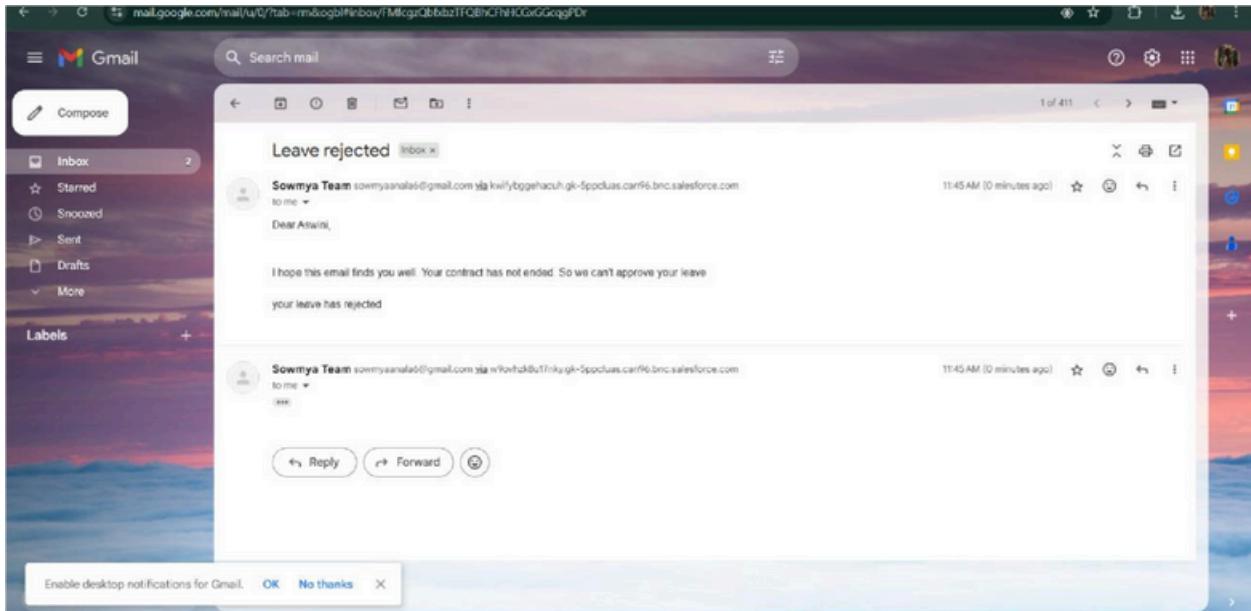
- Request for approve the leave



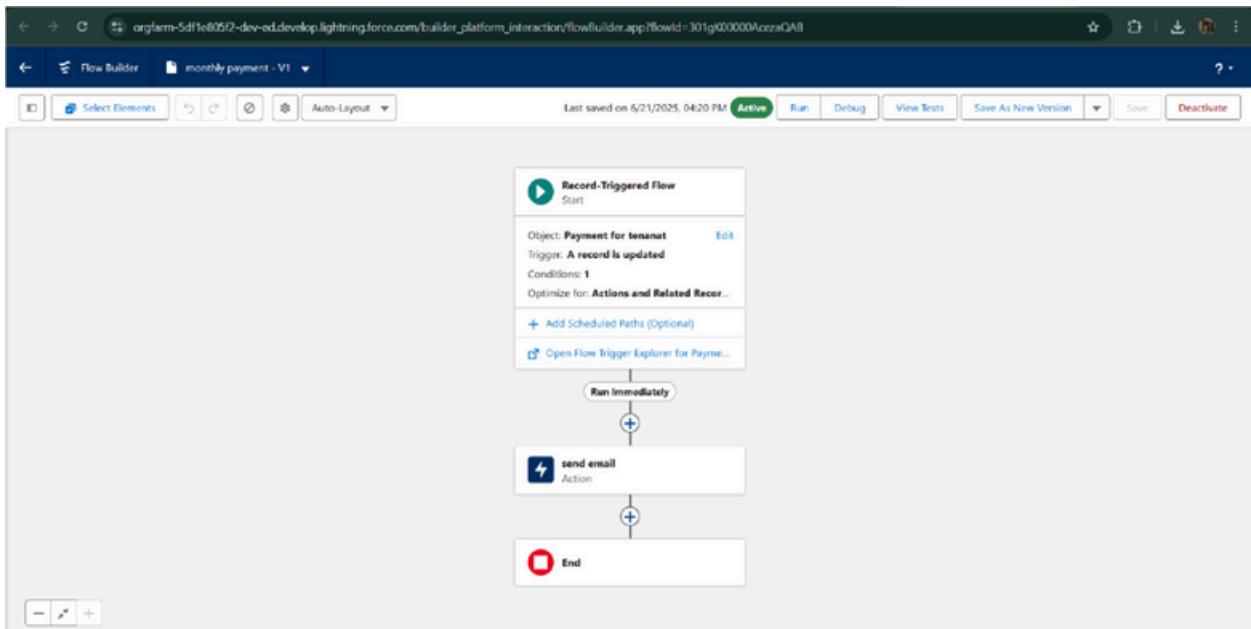
- Leave approved



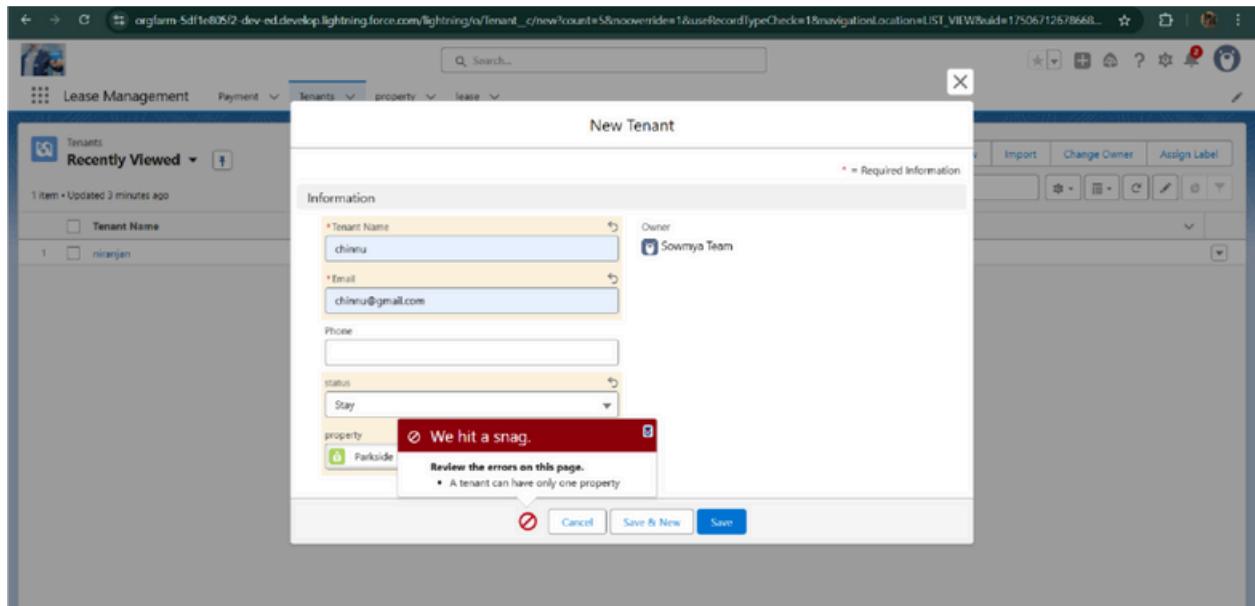
- Leave rejected



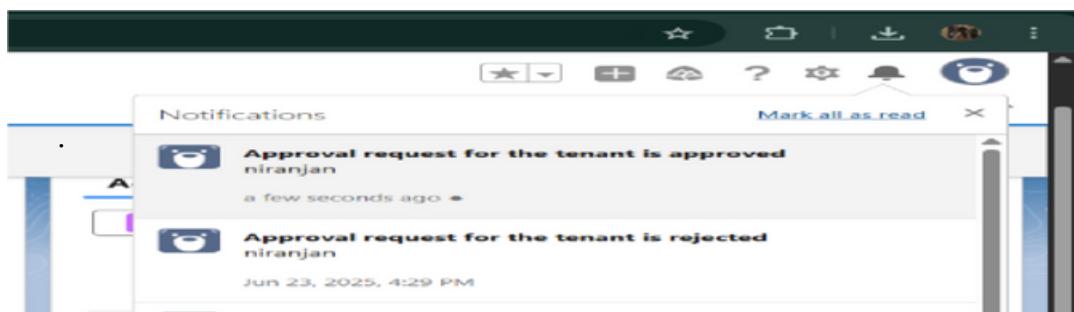
- Flow runs



- Trigger error messages



- Approval process notifications



# 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 teston 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});  
  
    }  
  
}
```