

**CollegeName: POLLACHICOLLEGEARTSANDSCIENCE**

**CollegeCode:bruaf**

**TEAMID: NM2025TMID21337**

**TEAM MEMBERS:3**

**TeamLeaderName:** NIYASH AHAMED A

**Email:** 23nmcsniyashahamed@gmail.com

**TeamMember1:** Mugilnilavan M

**Email:** 23mncsmugilnilavan@gmail.Com

**TeamMember2:** Mohan raj M

**Email:** [23nmcsmohanraj@gmail.com](mailto:23nmcsmohanraj@gmail.com)

**TeamMember3:** : Meganathan S

**Email:** [23nmcsmeganathan@gmail.com](mailto:23nmcsmeganathan@gmail.com)

## **1.INTRODUCTION**

### **ProjectOverview**

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.



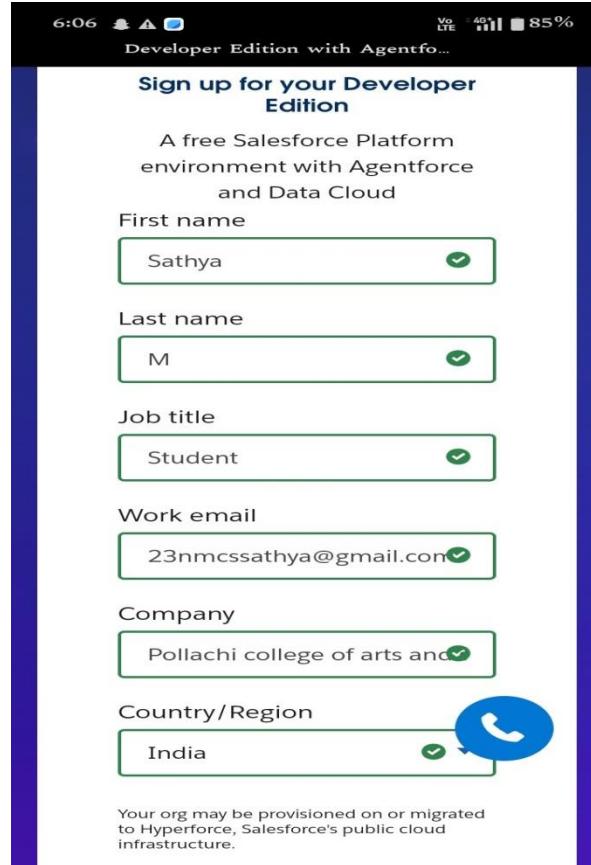
## Purpose

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:

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



- Created objects: Property, Tenant, Lease, Payment

SETUP > OBJECT MANAGER  
property

**Details**

Description

API Name: property\_\_c  
Custom: ✓  
Singular Label: property  
Plural Label: property

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

**Fields & Relationships**

- 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
- Scoring Rules

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

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists various configuration tabs like Fields & Relationships, Page Layouts, and Record Types. The main 'Details' tab is selected, showing the API name 'Tenant\_\_c', a custom label 'Tenant', and a plural label 'Tenants'. Under 'Enable Reports', 'Track Activities' and 'Track Field History' are checked. Deployment status is set to 'Deployed'. Help settings point to the standard Salesforce help window.

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

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'lease' object. Similar to the 'Tenant' object, it has an API name 'lease\_\_c', a custom label 'lease', and a plural label 'lease'. It also includes 'Enable Reports' settings for tracking activities and field history, and is currently deployed. The configuration tabs on the left are identical to the 'Tenant' screen.

The screenshot shows the Salesforce Setup interface under 'Object Manager'. A sidebar on the left lists various setup categories like Fields & Relationships, Page Layouts, and Lightning Record Pages. The main panel displays the 'Details' for the 'Payment for tenant' object. The 'Description' section includes the API Name 'Payment\_for\_tenant\_c' and Singular Label 'Payment for tenant'. On the right, there's a section for 'Enable Reports' with several checkboxes checked, including 'Track Activities', 'Track Field History', and 'Deployment Status'. Buttons for 'Edit' and 'Delete' are at the top right.

- Configured fields and relationships

The screenshot shows the Salesforce Setup interface under 'Object Manager'. The sidebar lists various setup categories. The main panel shows the 'Fields & Relationships' section for the 'property' object. It lists nine fields: Address, Created By, Last Modified By, Name, Owner, property, property\_Name, sfqt, and Type. The 'Address' field is of type 'Long Text Area(32768)'. The 'Created By' and 'Last Modified By' fields are of type 'Lookup(User)'. The 'Name' field is of type 'Text(25)'. The 'Owner' field is of type 'Lookup(User,Group)'. The 'property' field is of type 'Lookup(property)'. The 'property\_Name' field is of type 'Text(80)'. The 'sfqt' field is of type 'Text(18)'. The 'Type' field is of type 'Picklist'. A 'Quick Find' bar and buttons for 'New', 'Deleted Fields', 'Field Dependencies', and 'Set History Tracking' are at the top right.

Setup > Object Manager

### Payment for tenant

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(80)		✓

Setup > Object Manager

### lease

Fields & Relationships				
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 Details & Branding**

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

**App Details**

\*App Name: Lease Management

\*Developer Name: Lease\_Management

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

**App Branding**

Image:

Primary Color Hex Value: #0070D2

Org Theme Options:  Use the app's image and color instead of the org's custom theme

**App Launcher Preview**

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

**App Settings**

App Details & Branding  
App Options  
Utility Items (Desktop Only)  
**Navigation Items**  
User Profiles

**Navigation Items**

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

**Available Items**

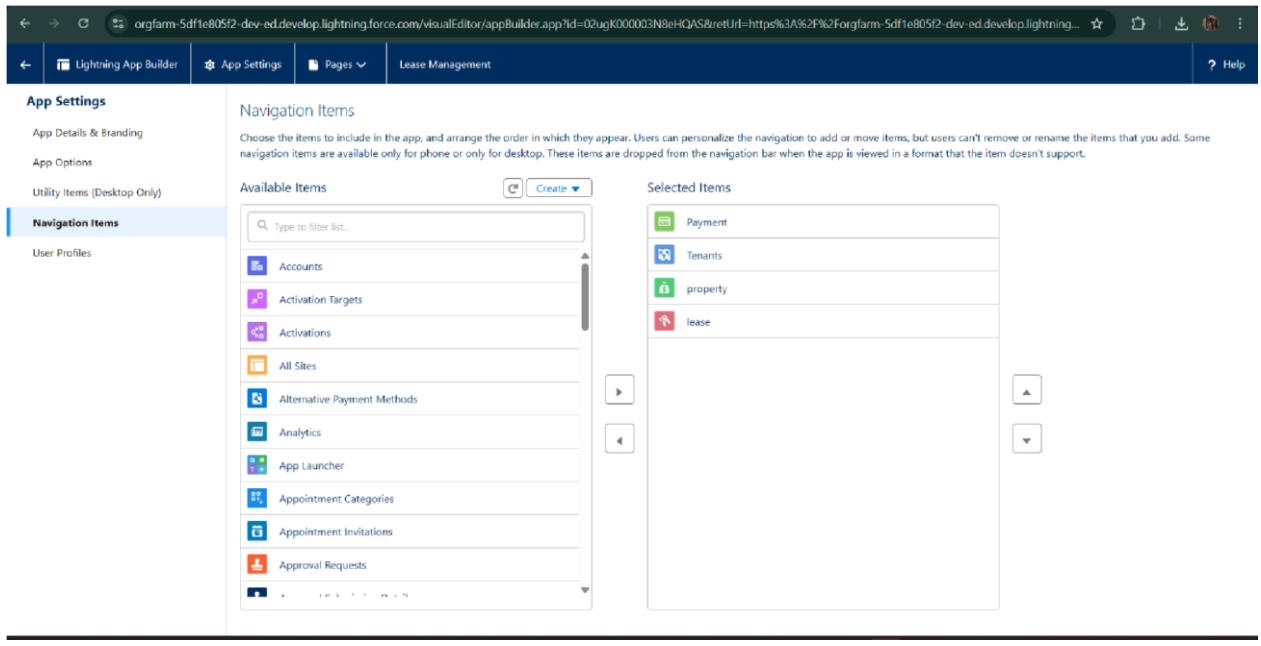
Type to filter list...  Create ▾

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

**Selected Items**

Payment  
Tenants  
property  
lease

Up ▾ Down ▾



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

**App Settings**

App Details & Branding  
App Options  
Utility Items (Desktop Only)  
**User Profiles**

**User Profiles**

Choose the user profiles that can access this app.

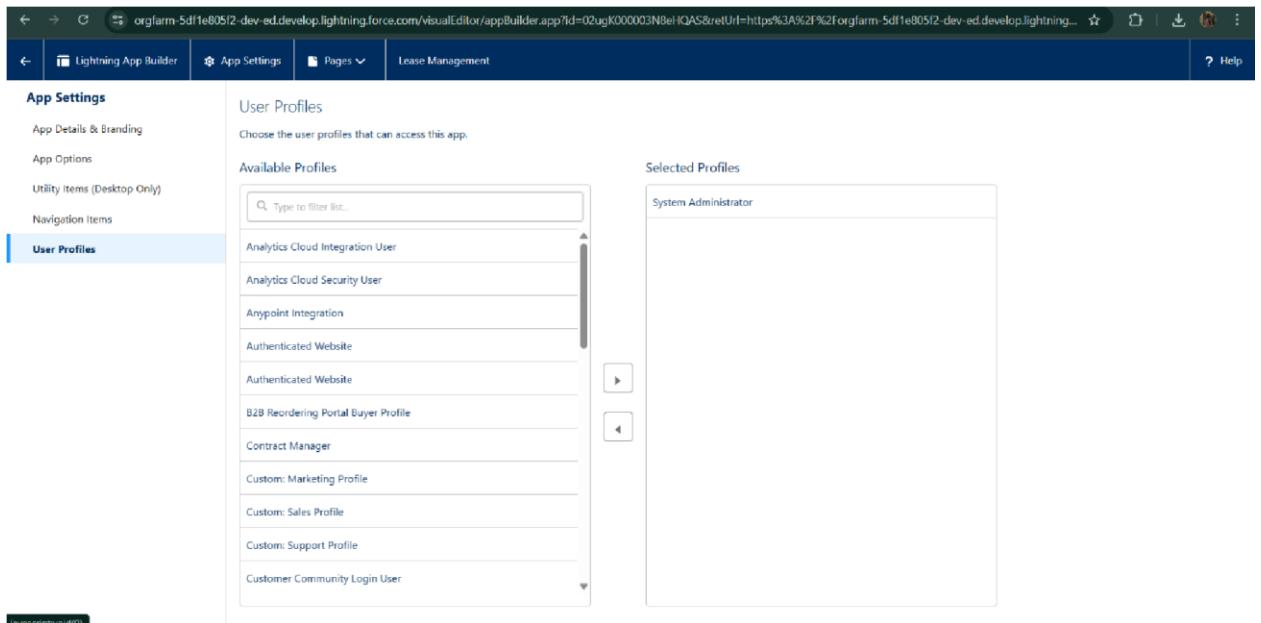
**Available Profiles**

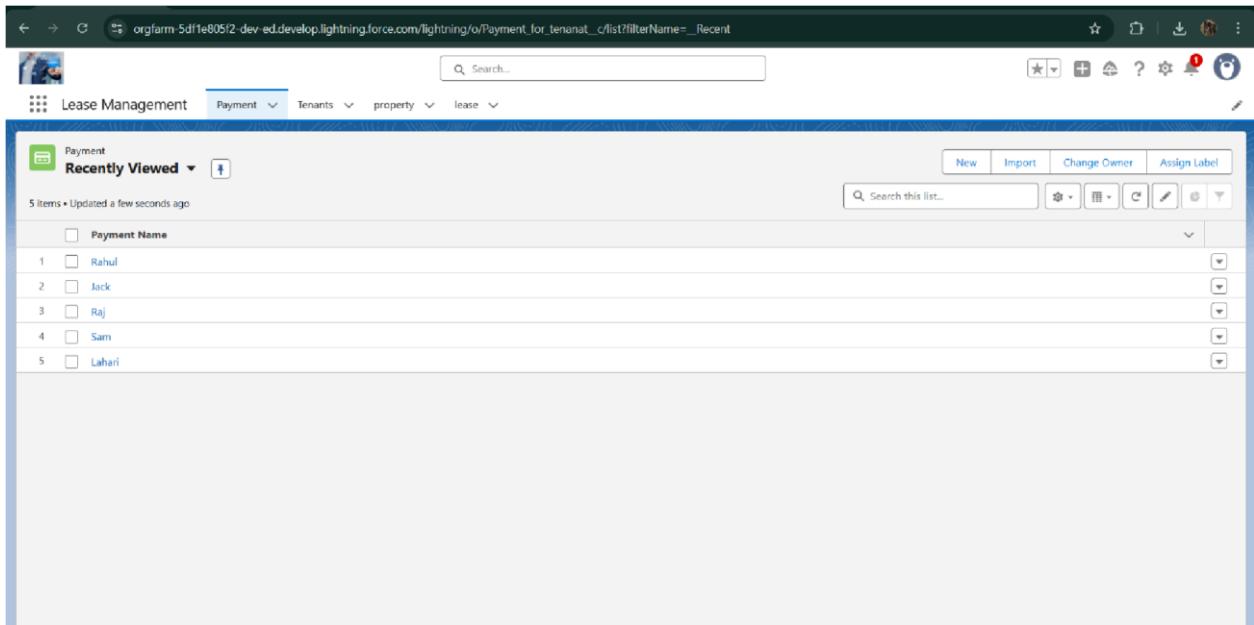
Type to filter list...

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

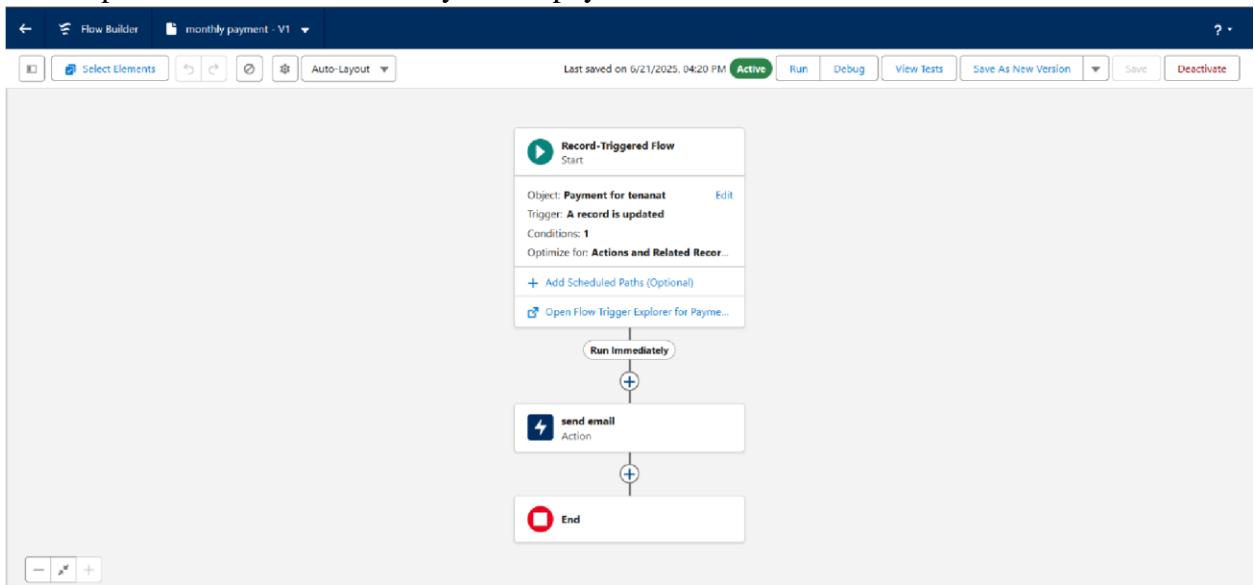
**Selected Profiles**

System Administrator





- 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-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTujValidationRules/page?address=%2F03dgK00000C8qPQAG%2Fe%3freURL%3D%2...
- Page Title:** SETUP > OBJECT MANAGER
- Object:** lease
- Section:** Validation Rule Edit
- Fields:**
  - Rule Name: lease\_end\_date
  - Active: checked
  - Description: (empty)
- Error Condition Formula:**

```
End_date_c <= start_date_c
```

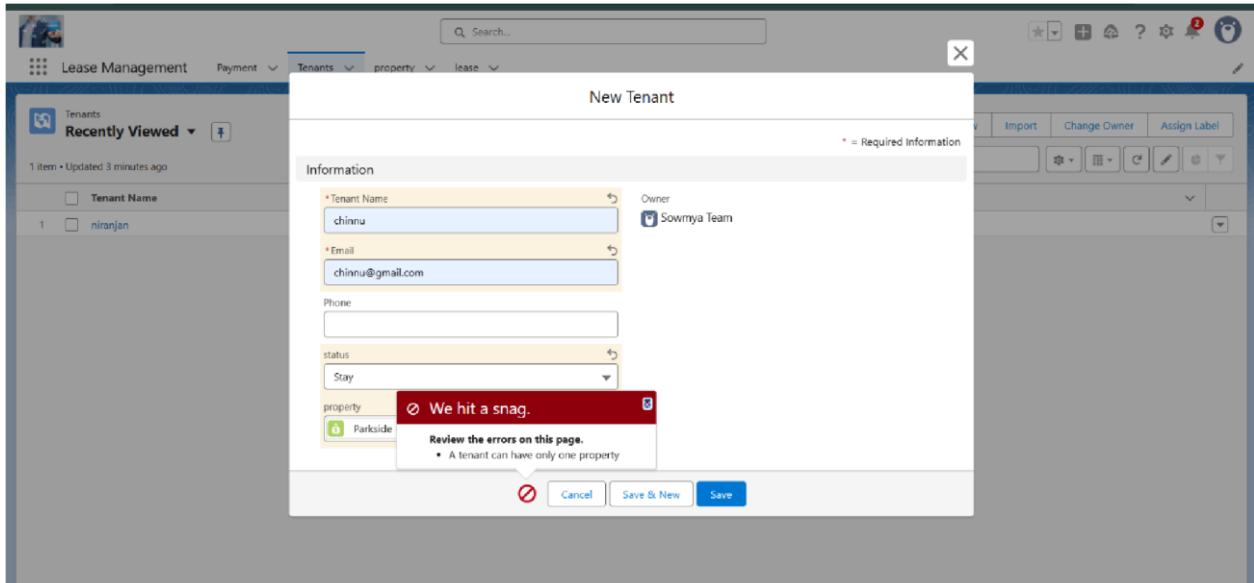
Example: Discount\_Percent\_C>30 More Examples...  
Display an error if Discount is more than 30%  
If this formula expression is true, display the text defined in the Error Message area
- Functions:**
  - All Function Categories
  - ABS
  - ACOS
  - ADDMONTHS
  - AND
  - ASCII
  - ASIN
- Buttons:**
  - Save
  - Save & New
  - Cancel
  - Check Syntax
- Help:** Quick Tips (Operators & Functions)

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

- Page Header:** orgfarm-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTujValidationRules/03dgK00000C8qPQAG/view
- Page Title:** SETUP > OBJECT MANAGER
- Object:** lease
- Section:** lease Validation Rule
- Fields:**

Field	Value
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	(empty)
Created By	Sowmya Team, 6/19/2025, 5:37 AM
Modified By	Sowmya Team, 6/26/2025, 7:47 AM
- Buttons:**
  - Edit
  - Close
- Help:** Help for this Page

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

- 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' section, 'Classic Email Templates' is selected. A search result for 'Leave approved' is displayed. The 'Email Template Detail' section shows the following information:

- Email Template Name:** Leave approved
- Template Unique Name:** Leave\_approved
- Encoding:** Unicode (UTF-8)
- Author:** Sowmya Team [Changed]
- Description:** Created By Sowmya Team, 6/20/2025, 1:06 AM
- Modified By:** Sowmya Team, 6/20/2025, 1:06 AM
- Available For Use:** checked
- Last Used Date:** Not specified
- Times Used:** Not specified

The 'Email Template' preview area shows the following 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.  

your leave is approved. You can leave now.
  
```

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. Under the 'Email' section, 'Classic Email Templates' is selected. A search result for 'tenant leaving' is displayed. The 'Email Template Detail' section shows the following information:

- Email Template Name:** tenant leaving
- Template Unique Name:** tenant\_leaving
- Encoding:** Unicode (UTF-8)
- Author:** Sowmya Team [Changed]
- Description:** Created By Sowmya Team, 6/20/2025, 1:06 AM
- Modified By:** Sowmya Team, 6/20/2025, 1:06 AM
- Available For Use:** checked
- Last Used Date:** Not specified
- Times Used:** Not specified

The 'Email Template' preview area shows the following 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:

**Classic Email Templates**

**Leave rejected**

**Email Template Detail**

Field	Value
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team 6/20/2025, 1:11 AM
Modified By	Sowmya Team 6/20/2025, 1:11 AM

**Email Template**

**Subject:** Leave rejected

**Plain Text Preview:**

Dear {Tenant\_c\_Name},  
I hope this email finds you well. Your contract has not ended. So we can't approve your leave.  
your leave has rejected

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

**Classic Email Templates**

**Tenant Email**

**Email Template Detail**

Field	Value
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team 6/20/2025, 1:12 AM
Modified By	Sowmya Team 6/20/2025, 1:12 AM

**Email Template**

**Subject:** Urgent: Monthly Rent Payment Reminder

**Plain Text Preview:**

Dear {Tenant\_c\_Name},  
I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Email', 'Classic Email Templates' is chosen. A search bar at the top left shows 'email template'. The main content area displays a 'Text Email Template' named 'tenant payment'. The 'Email Template Detail' section shows the following information:

- Email Templates from Salesforce: Unified Public Classic Email Templates
- Email Template Name: tenant payment
- Template Unique Name: tenant\_payment
- Encoding: Unicode (UTF-8)
- Author: Sowmya Team [Change]
- Description: Created By: Sowmya Team 6/20/2025, 1:13 AM
- Available For Use: ✓
- Last Used Date: Times Used: 0
- Modified By: Sowmya Team 6/20/2025, 1:13 AM

The 'Email Template' preview shows the subject 'Confirmation of Successful Monthly Payment' and the plain text content:

```

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.

```

## ● ApprovalProcesscreation

For TenantLeaving:

The screenshot shows the Salesforce Setup interface with the 'Data' section selected. Under 'Data', 'Approval Processes' is chosen. A search bar at the top left shows 'approval'. The main content area displays an 'Approval Process' named 'Tenant: TenantApproval'. The 'Process Definition Detail' section shows the following information:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Tenant: status equals Stay
- Entry Criteria: Record Editability: Administrator ONLY
- Active: ✓
- Next Automated Approver Determined By: [empty]
- Allow Submitters to Recall Approval Requests:
- Created By: Sowmya Team 6/20/2025, 3:41 AM
- Modified By: Sowmya Team 6/26/2025, 11:57 PM

The 'Initial Submission Actions' section shows a single action: 'Record Lock' with the description 'Lock the record from being edited'.

The 'Approval Steps' section shows one step:

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

For CheckforVacant:

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

- Approval Processes**: Tenant: check for vacant
- Process Definition Detail**:
  - Process Name: check for vacant
  - Unique Name: check\_for\_vacant
  - Description: Tenant: status EQUALS Leaving
  - Entry Criteria: Record Editability - Administrator ONLY
  - Next Automated Approver Determined By: Active (checked)
  - Approval Assignment Email Template: Leave\_approved
  - Initial Submitters: Tenant Owner
  - Created By: Sowmya Team
  - Modified By: Sowmya Team
- Initial Submission Actions**:
 

Action	Type	Description
Record Lock		Lock the record from being edited
Email Alert		please approve my leave
- Approval Steps**:
 

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

## ● ApexTrigger

### CreateanApexTrigger

The screenshot shows the Salesforce Developer Console with the following details:

- File • Edit • Debug • Test • Workspace • Help • < >**
- Code Coverage: None API Version: 54**
- trigger test on Tenant\_c (before insert)**
- Open Entity Type** modal window:
 

Entity Type	Edition	Related
Triggers	test	
Closes		
Pages		
Page Components		
Objects		
Static Resources		
Packages		
- Logs Tests Checkpoints Query Editor View Status Progress Problems**

Developer Console - Google Chrome  
 orgfarm-5d1fe805f2-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 Name: API Version: 64 Go To
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
11 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

## CreateanApex Handlerclass

Developer Console - Google Chrome  
 orgfarm-5d1fe805f2-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 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(exi
10      }
11
12
13
14 +   for (Tenant__c newTenant : newList) {
15
16
17
18 +     if (!newTenant.Property__c)
19
20       newTenantaddError('A t
21
22
23 }
```

Entity Type Entity Name Namespace Related

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

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-5d1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

test.apex testHandler.apxc MonthlyEmailScheduler.apac

Code Coverage Name API Version 64 Go To

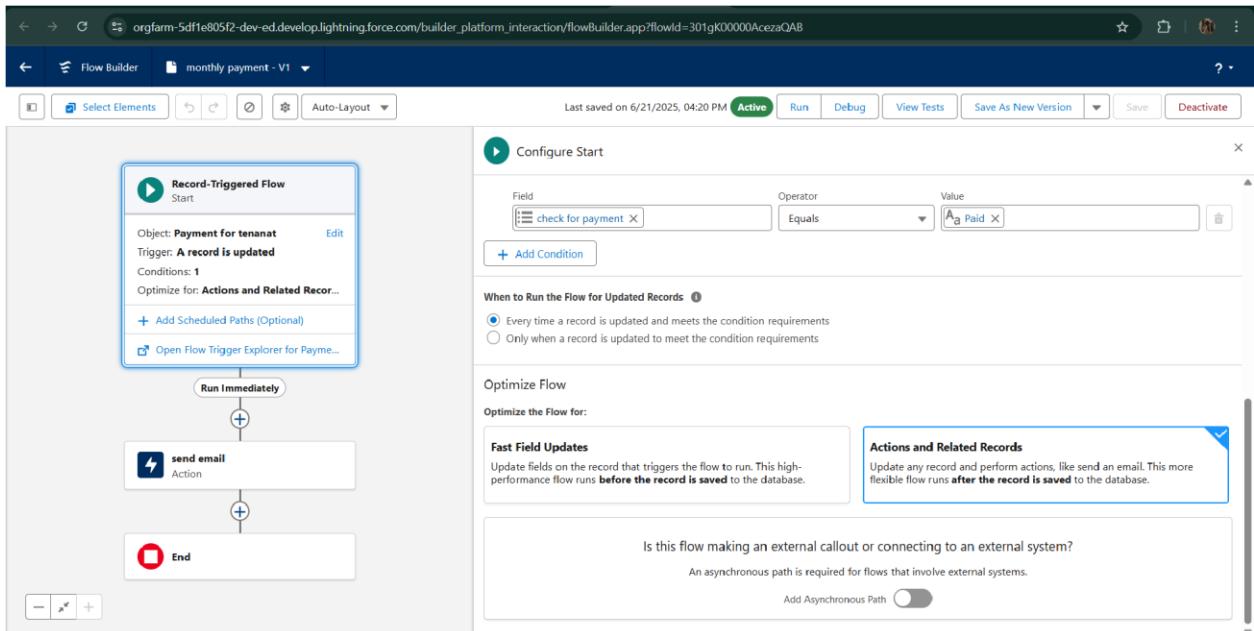
```

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 State Progress Problems

Name Line Problem

## ● FLOWS



The screenshot shows the Salesforce Flow Builder interface. On the left, a flow diagram titled "Record-Triggered Flow Start" is displayed. It starts with a "Run Immediately" step, followed by an "Action" step labeled "send email", and ends with an "End" step. On the right, a configuration panel titled "Configure Start" is open. It shows the "Select Object" section with "Payment for tenant" selected. Below it, the "Configure Trigger" section is set to trigger "A record is updated". The "Set Entry Conditions" section indicates "All Conditions Are Met (AND)".

- Schedule class:  
CreateanApexClass

The screenshot shows the Salesforce Developer Console. The code editor contains the following 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

```

A modal window titled "Open" is overlaid on the code editor, showing the repository browser with the entity "MonthlyEmailScheduler" selected. The "Related" section shows relationships to "Email", "CustomField", "Object", and "SOObject".

```

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 a positive living environment for all.';
25
26            String emailSubject = 'Rent Remind: Monthly Rent Payment Due';
27
28            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30            email.setToNames(recipientEmail);
31            email.setSubject(emailSubject);
32            email.setPlainTextBody(emailContent);
33
34            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
35
36        }
37
38    }
39
40 }
41
42

```

## ScheduleApex class

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left has a search bar with 'apex' typed in. The main area displays the 'Apex Classes' section for the 'Apex Class' named 'MonthlyEmailScheduler'. The 'Apex Class Detail' table shows the following information:

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

The 'Class Body' tab is selected, 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
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 a positive living environment for all.';
25
26            String emailSubject = 'Rent Remind: Monthly Rent Payment Due';
27
28            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30            email.setToNames(recipientEmail);
31            email.setSubject(emailSubject);
32            email.setPlainTextBody(emailContent);
33
34            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
35
36        }
37
38    }
39
40 }
41
42

```

Screenshot of the Salesforce Lightning interface showing the 'Lease Management' tab selected. A context menu is open over a tenant record named 'Aswini'. The menu options include:

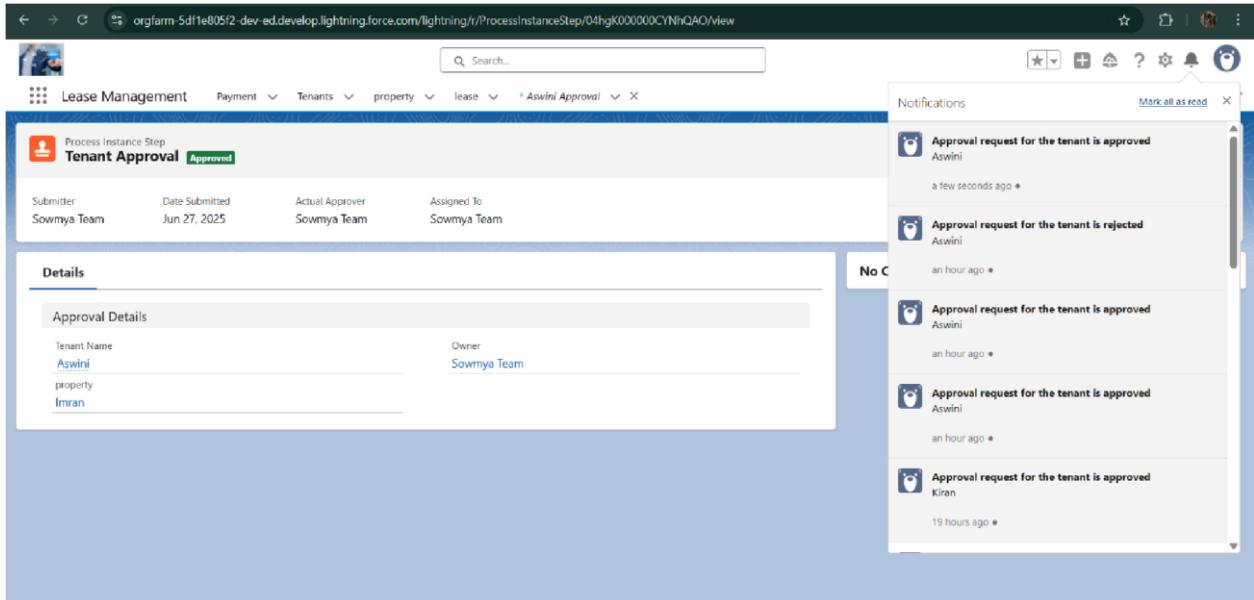
- New Case
- New Lead
- Delete
- Clone
- Change Owner
- Printable View
- Submit for Approval**
- Edit Labels

The 'Submit for Approval' option is highlighted.

Screenshot of the Salesforce Lightning interface showing the 'Lease Management' tab selected. A success message 'Tenant was submitted for approval.' is displayed in a green banner at the top right. The tenant record for 'Aswini' is shown with all fields filled in, including:

- \* Tenant Name: Aswini
- \* Email: aswiniamaraadi15@gmail.com
- Phone: (905) 223-5567
- Status: Leaving
- Property: Imran

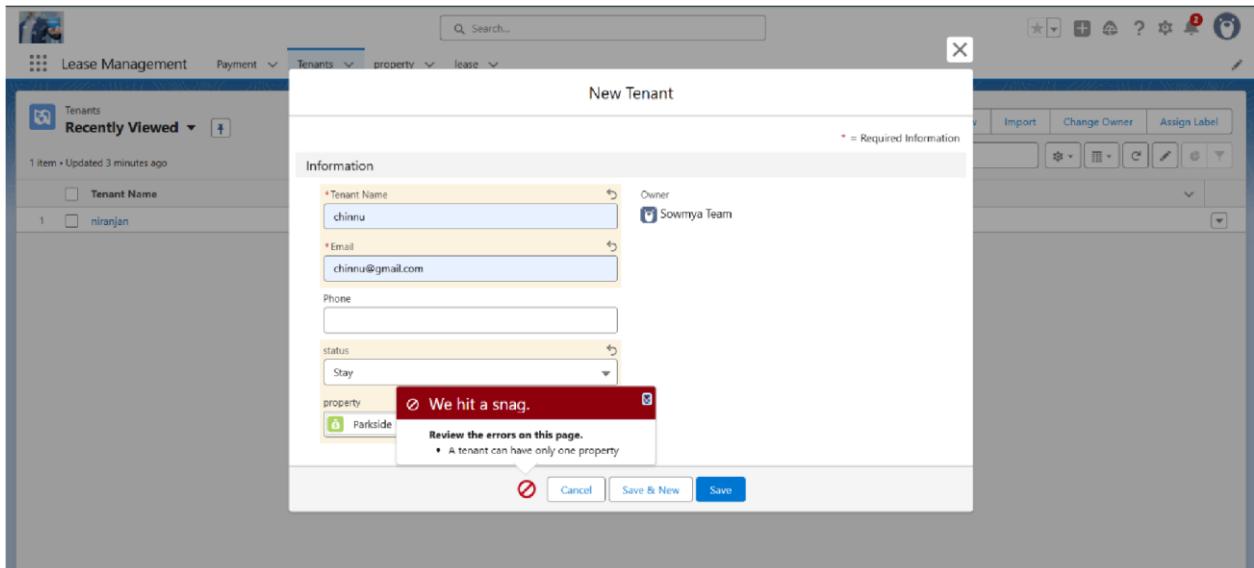
The 'Save' button is visible at the bottom right.



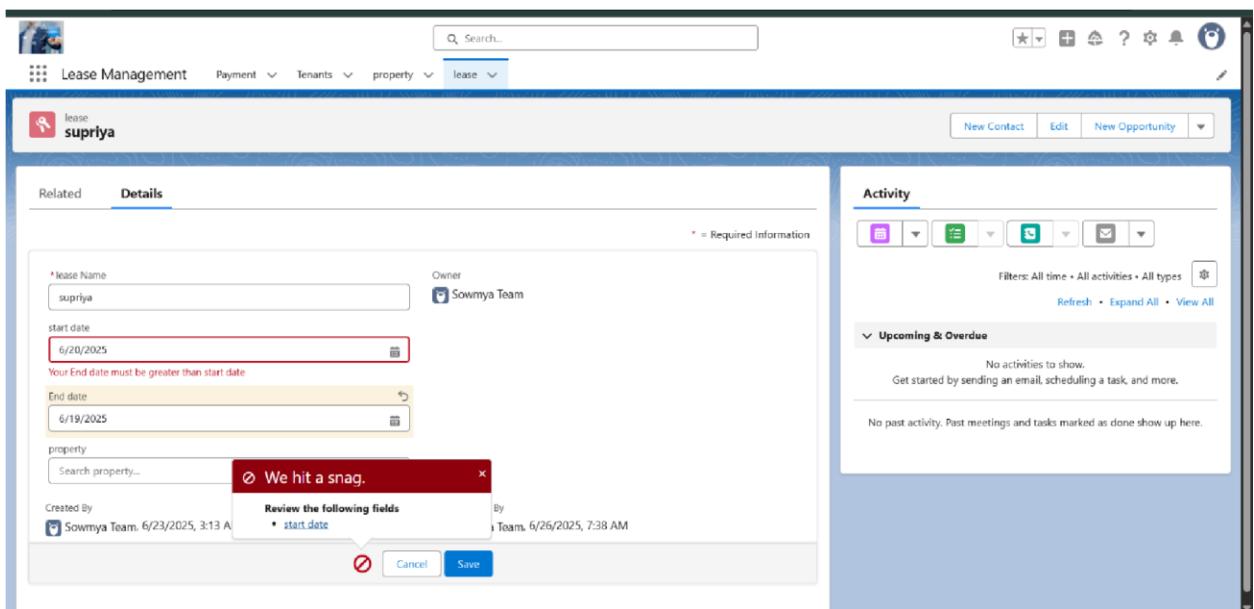
# FUNCTIONAL AND PERFORMANCE TESTING

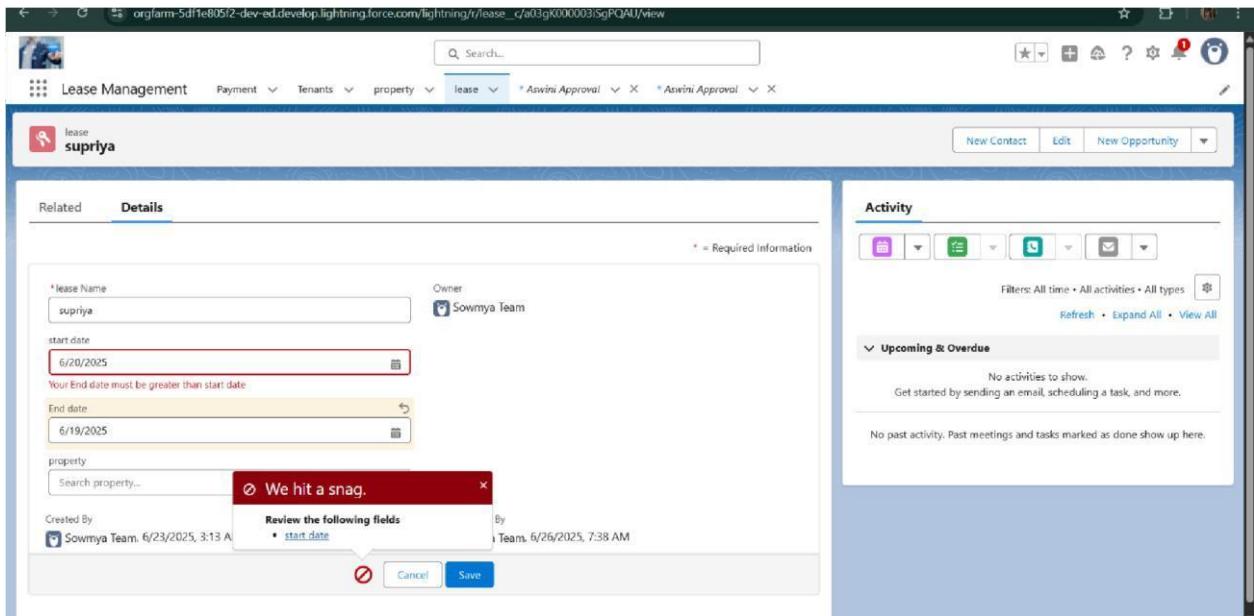
## Performance Testing

- Trigger validation by entering duplicate tenant-property records

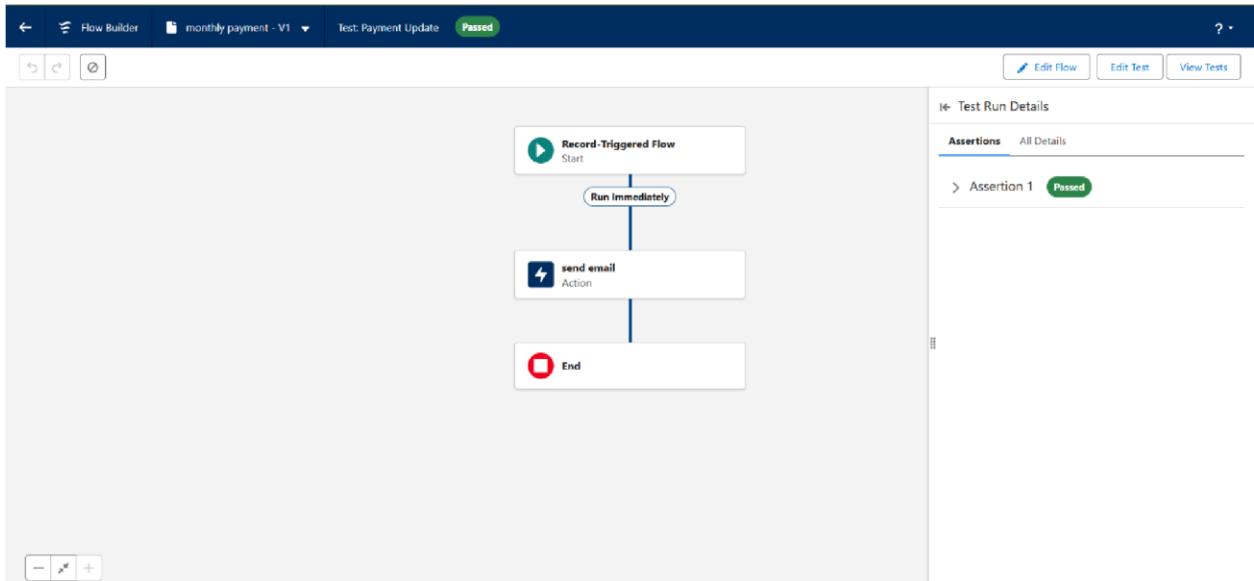


- ValidationRulechecking





- Testflows 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 (Parkside Lofts). The status is set to 'Stay'. The right side shows a 'Notifications' sidebar with several items:

- 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 with details:

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 is a 'Payment' section showing two entries for 'Jack' and 'Rahul'.

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

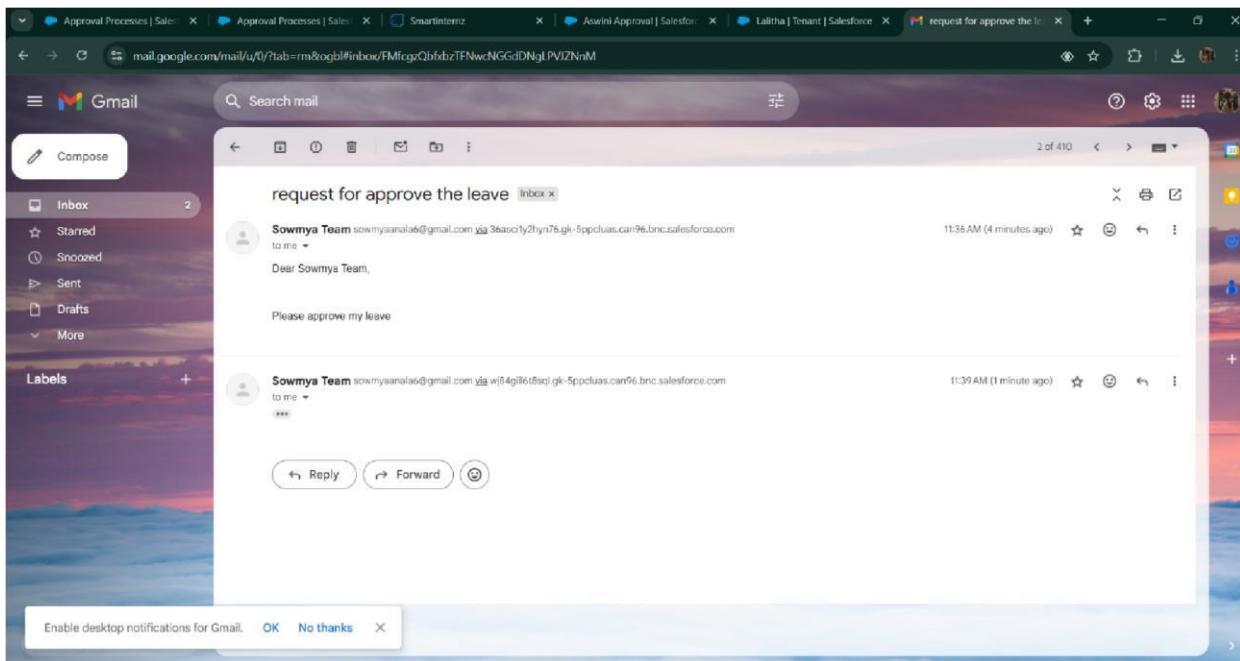
The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It displays four custom tabs: 'Lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'property' (Tab Style: Sack), and 'Tenant' (Tab Style: Map). The 'Custom Object Tabs' section also lists these tabs with their respective actions (Edit | Del) and labels.

- Email alerts

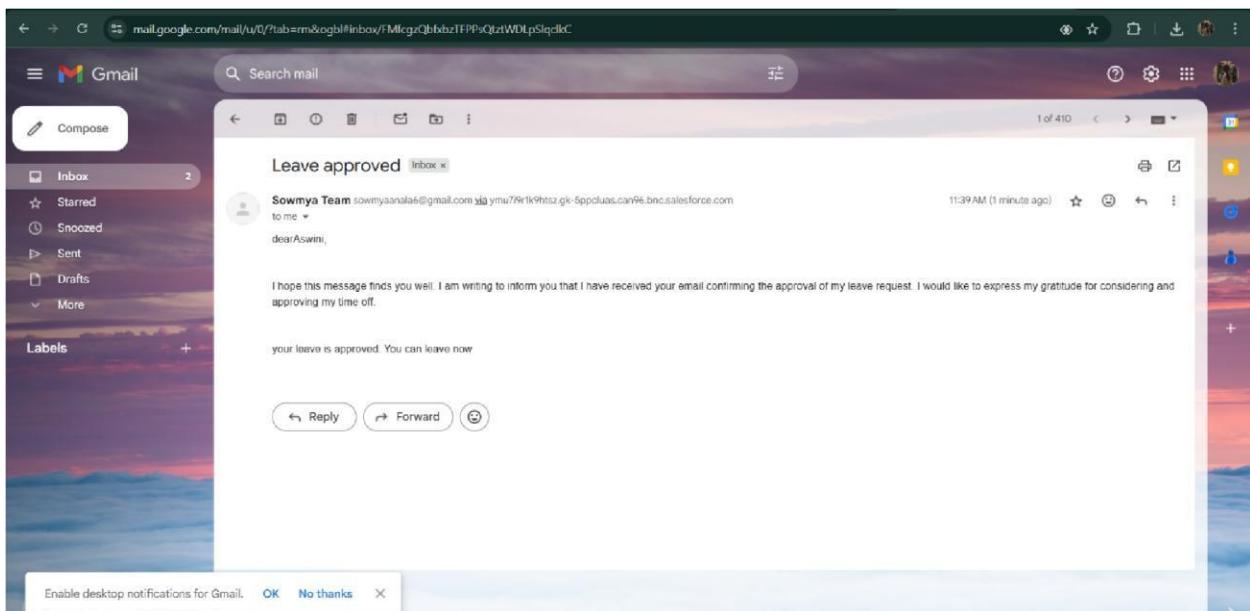
The screenshot shows the 'Lease Management' application interface. It displays an 'Approval History' table for a leave request. The table includes columns for Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The history shows multiple steps, mostly 'Step 1', with various statuses like Approved, Submitted, and Rejected, 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	6/25/2025, 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

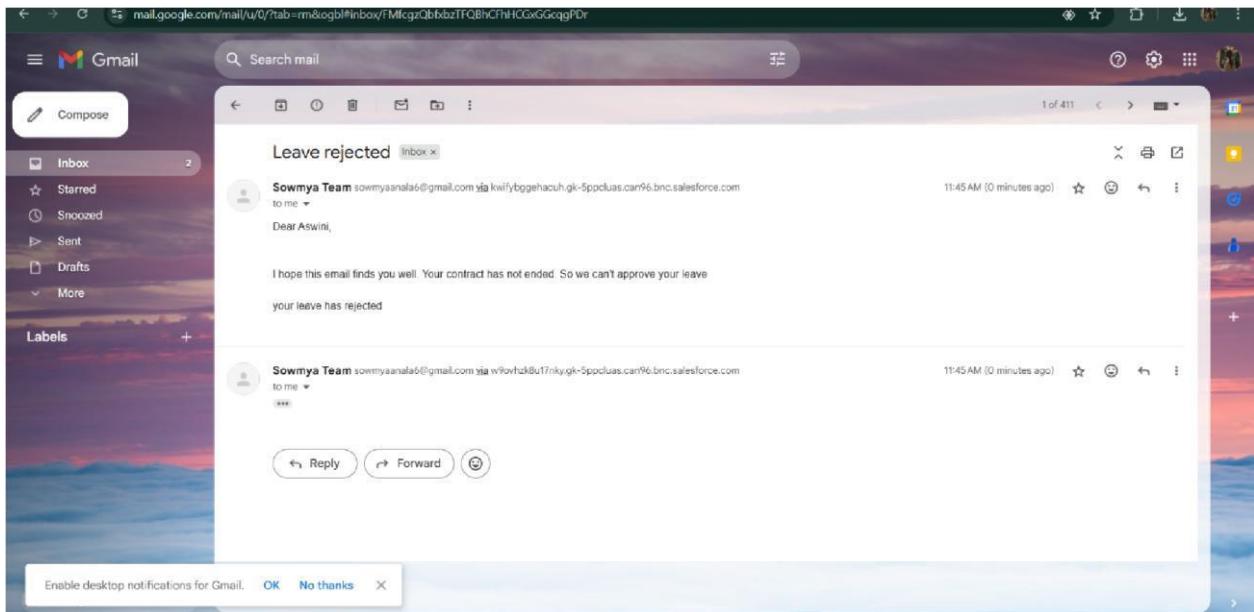
- Request for approval the leave



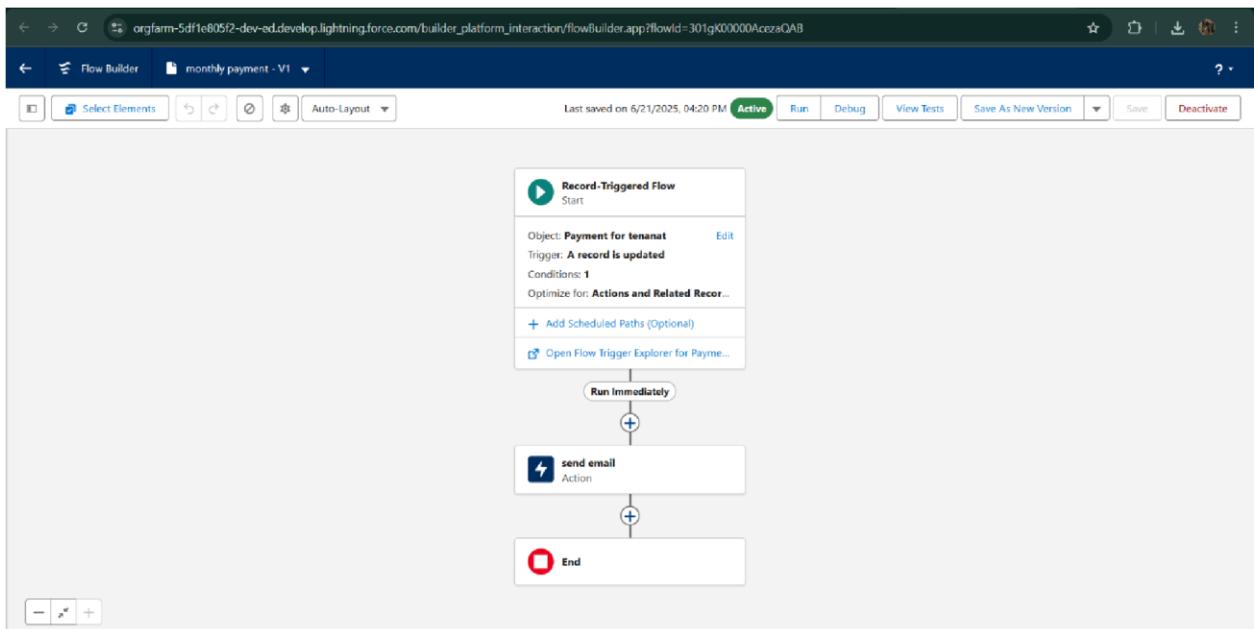
- Leave approved



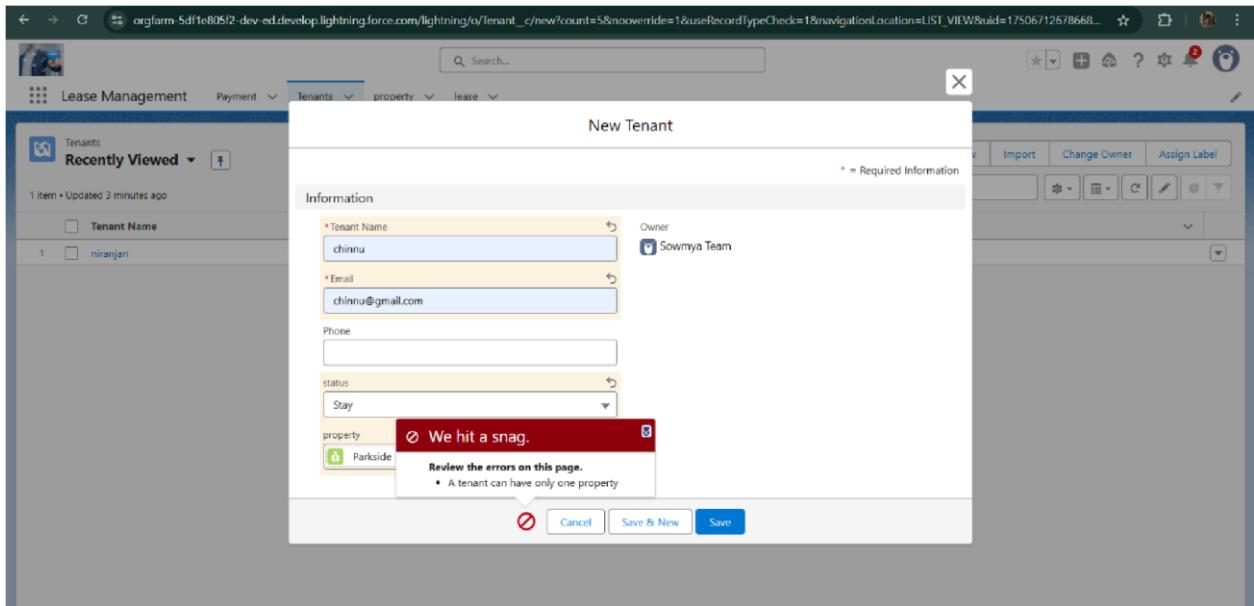
- Leave rejected



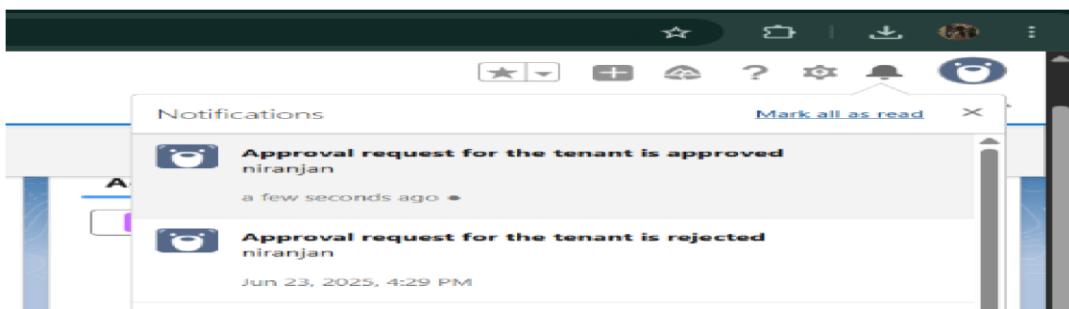
- Flowruns



- Triggererrormessages



- Approvalprocess 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

- **SourceCode:** Provided in Apex Classes and Triggers

### Test.apxt:

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

### testHandler.apxc:

```
public class
testHandler {
    public static void
    preventInsert(List<
        Tenantc> newList)
    {
        Set<Id>
        existingPropertyIds
        = new Set<Id>()
        for(Tenantc existingTenant:[SELECT Id, Propertyc FROM Tenantc WHERE
        Propertyc != null])
        {
            existingPropertyIds.add(existingTenant.Propertyc);
        }
    }
}
```

```

}for(TenantcnewTenant:

newlist) {

    if (newTenant.Propertyc != null &&
existingPropertyIds.contains(newTenant.Propertyc)){newTenant.addError('A

tenantcanhaveonlyoneproperty');

}

}

}

```

### **MothlyEmailScheduler.apxc:**

```

globalclassMonthlyEmailSchedulerimplementsSchedulable{ global

void execute(SchedulableContext sc) { Integer currentDay =

Date.today().day(); if (currentDay == 1) {

sendMonthlyEmails();

}

}public static void

sendMonthlyEmails(){List<Tenantc>

tenants=[SELECTId,EmailcFROM

Tenantc]; for (Tenantc tenant :

tenants){

StringrecipientEmail=tenant.Emailc;

StringemailContent='Itrustthisemailfindsyouwell.Iamwritingtoremindyou that the

monthly rentis dueYour timely payment ensures the smooth functioning of our rental

arrangement and helps maintain a positive living environment for all.';

StringemailSubject='Reminder:MonthlyRentPaymentDue';

```

```
        Messaging.SingleEmailMessage email = new  
        Messaging.SingleEmailMessage(); email.setToAddresses(new  
        String[]{recipientEmail}); email.setSubject(emailSubject);  
        email.setPlainTextBody(emailContent);  
        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});  
    }  
}  
}
```