

LEASE MANAGEMENT

College Name: KG College of Arts and Science

TEAM ID: NM2025TMID23566

TEAM MEMBERS

S.NO	NAME	ROLL.NO	MAIL ID
1	DHARSHANA S	2322JA09	2322ja09@kgcas.com
2	DHARSHINI PRIYA D	2322JA10	2322ja10@kgcas.com
3	GANGADHARAN J	2322JA11	2322ja11@kgcas.com
4	GOKULAKRISHNAN S	2322JA12	2322ja12@kgcas.com

1. INTRODUCTION

1.1 Project Overview

- The Lease Management System is a Salesforce-based application designed to streamline the processes associated with leasing real estate properties.
- It handles tenant management, lease contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



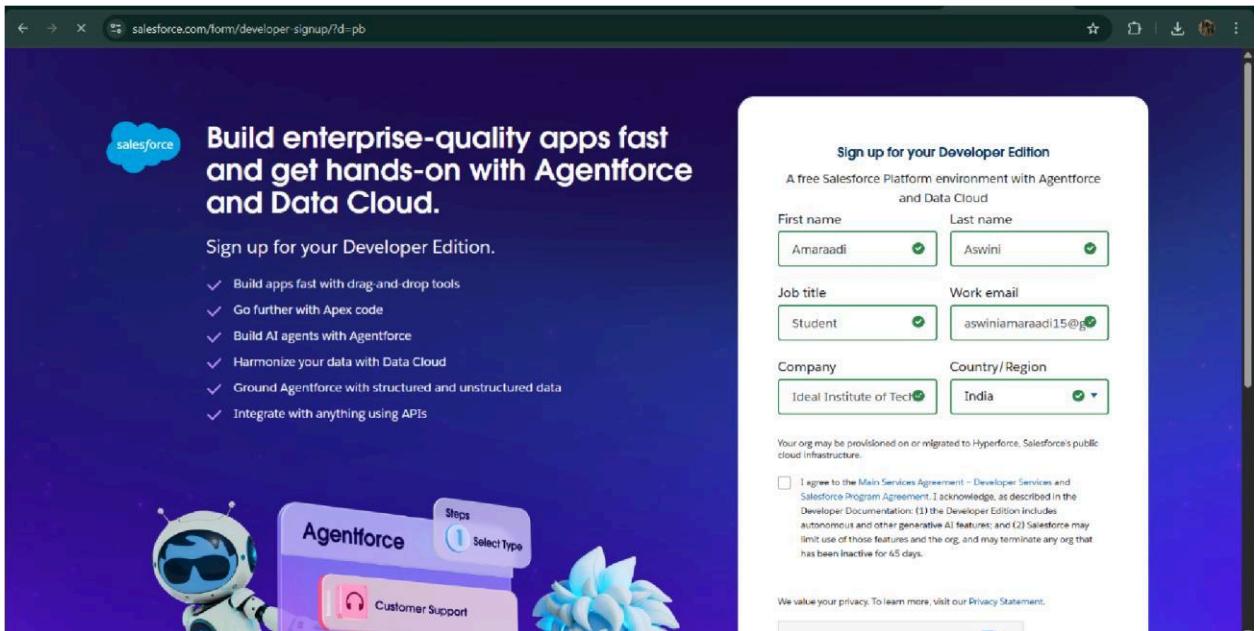
1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

DEVELOPMENT PHASE

Creating Developer Account:

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



- Created objects: Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area displays the details for the 'Tenant' object. The 'Details' tab is selected. The API Name is set to 'Tenant_c'. The Singular Label is 'Tenant' and the Plural Label is 'Tenants'. Under the 'Description' section, there are several checkboxes for setup features: Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (unchecked), Deployed (unchecked), Help Settings (unchecked), and Standard salesforce.com Help Window (unchecked). At the top right of the details page are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area displays the details for the 'lease' object. The 'Details' tab is selected. The API Name is set to 'lease_c'. The Singular Label is 'lease' and the Plural Label is 'lease'. Under the 'Description' section, there are several checkboxes for setup features: Enable Reports (checked), Track Activities (checked), Track Field History (checked), Deployment Status (unchecked), Deployed (unchecked), Help Settings (unchecked), and Standard salesforce.com Help Window (unchecked). At the top right of the details page are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details'. The main 'Details' section contains fields for 'Description', 'API Name' (set to 'Payment_for_tenantat_c'), 'Custom' (checked), 'Singular Label' (set to 'Payment for tenantat'), 'Plural Label' (set to 'Payment'), and 'Enable Reports' (checked). Other settings like 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window') are also visible.

- Configured fields and relationships

The screenshot shows the 'Fields & Relationships' section for the 'property' object. The sidebar on the left has 'Fields & Relationships' selected. The main area displays a table of fields, sorted by Field Label. The columns are 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The table rows include:

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)	<input checked="" type="checkbox"/>	
property	property__c	Lookup(property)	<input checked="" type="checkbox"/>	
property Name	Name	Text(80)	<input checked="" type="checkbox"/>	
sfqt	sfqt__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)	<input checked="" type="checkbox"/>	
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)	<input checked="" type="checkbox"/>	

Setup > Object Manager

lease

Fields & Relationships				
7 items, Sorted by Field Label				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)	<input checked="" type="checkbox"/>	
Owner	OwnerId	Lookup(User,Group)	<input checked="" type="checkbox"/>	
property	property_c	Lookup(property)	<input checked="" type="checkbox"/>	
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

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

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

*Developer Name

Description

App Branding

Image

Primary Color Hex Value

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

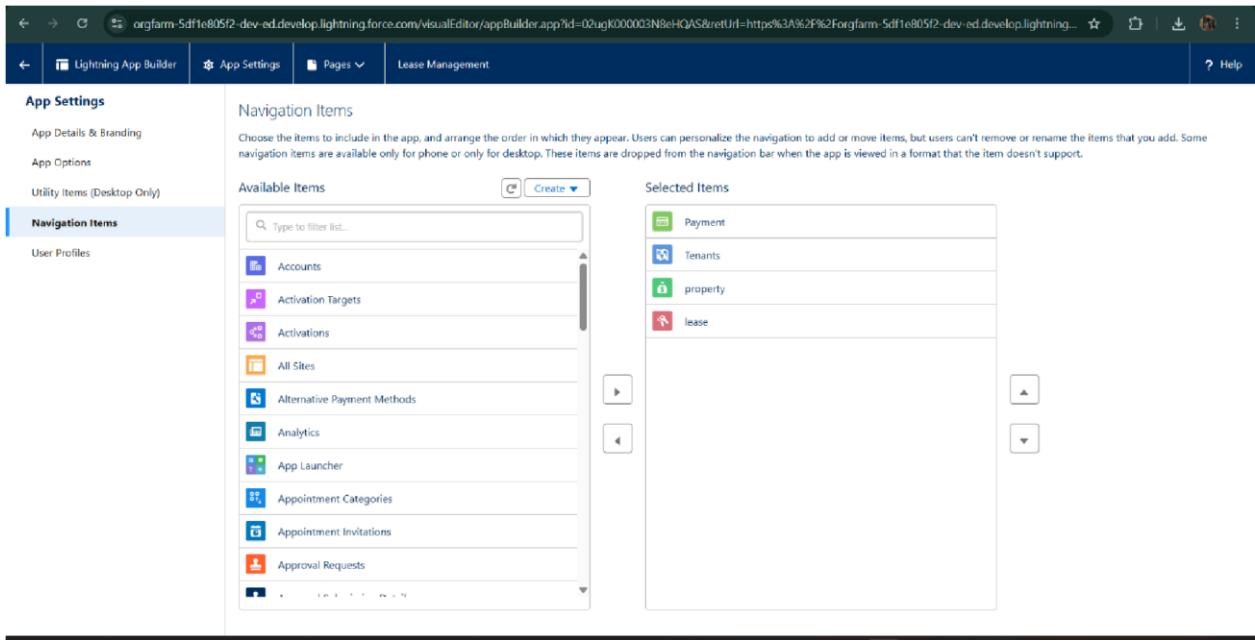
Type to filter list... Create ▾

- 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)
Navigation Items
User Profiles

User Profiles

Choose the user profiles that can access this app.

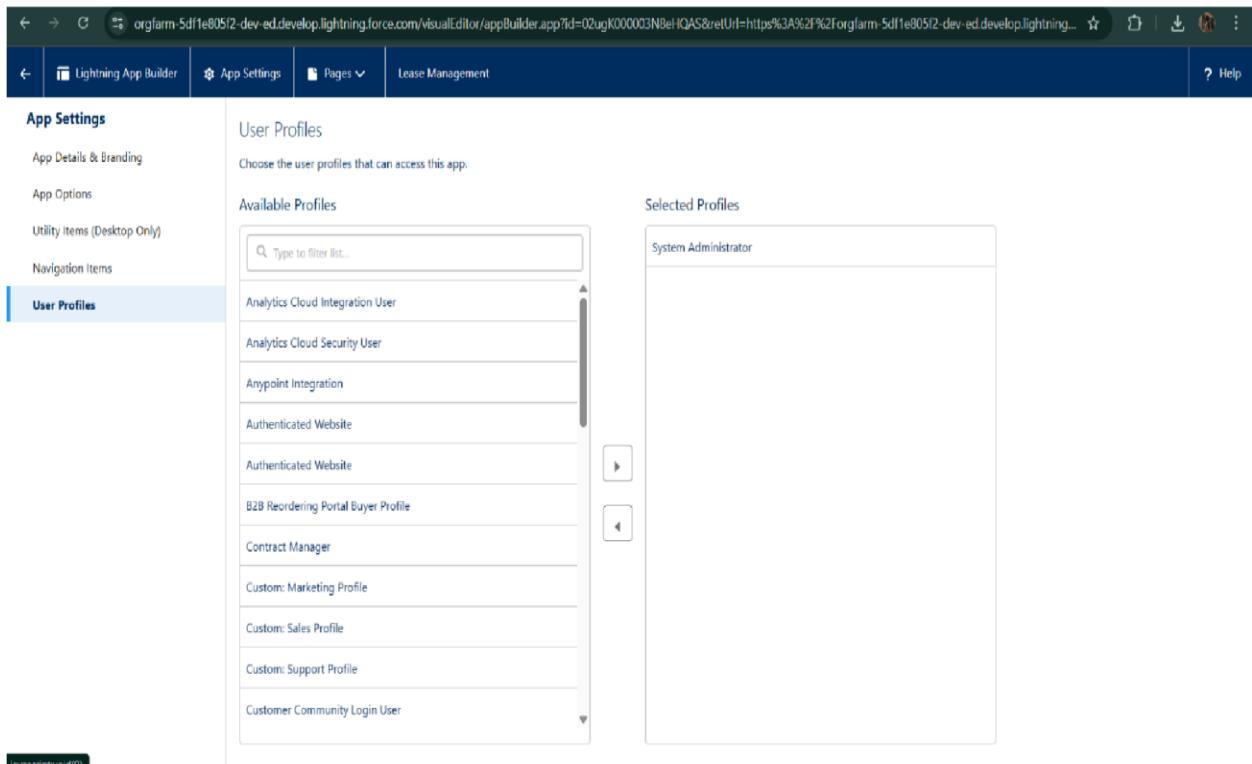
Available Profiles

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

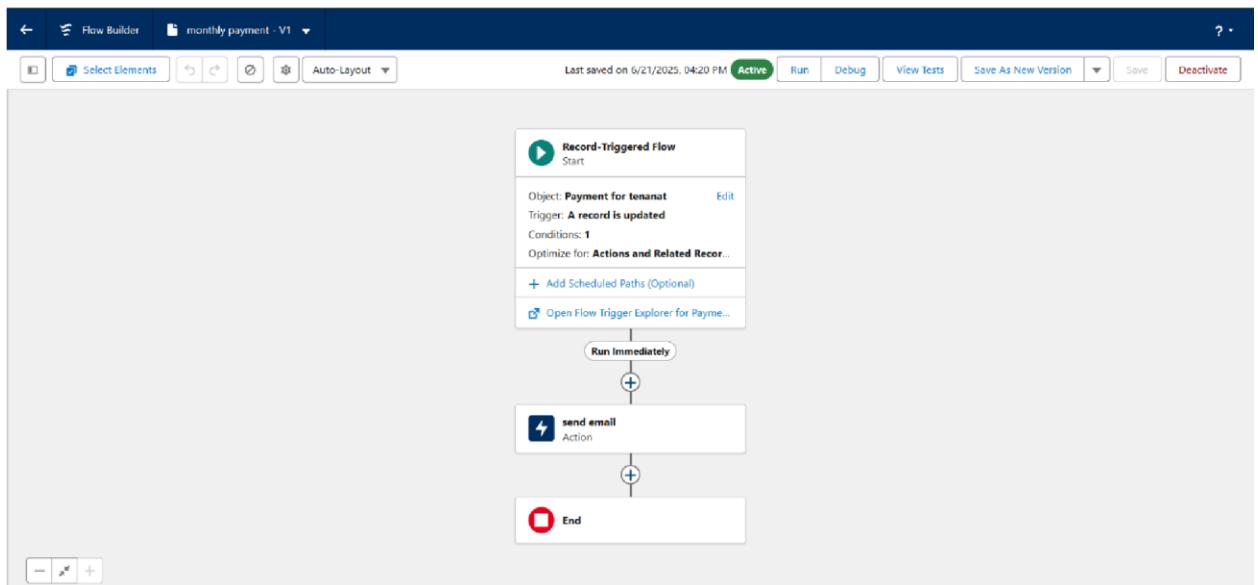
Selected Profiles

System Administrator



The screenshot shows a Salesforce Lightning interface for 'Lease Management'. The top navigation bar includes 'Payment', 'Tenants', 'property', and 'lease' dropdowns. The main area displays a list titled 'Recently Viewed' under the 'Payment' category. The list contains 5 items, all updated a few seconds ago, with columns for 'Payment Name' and a checkbox. The names listed are: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, and 5. Lahari. The interface features standard Salesforce navigation and search tools.

- Implemented Flows for monthly rent and payment success

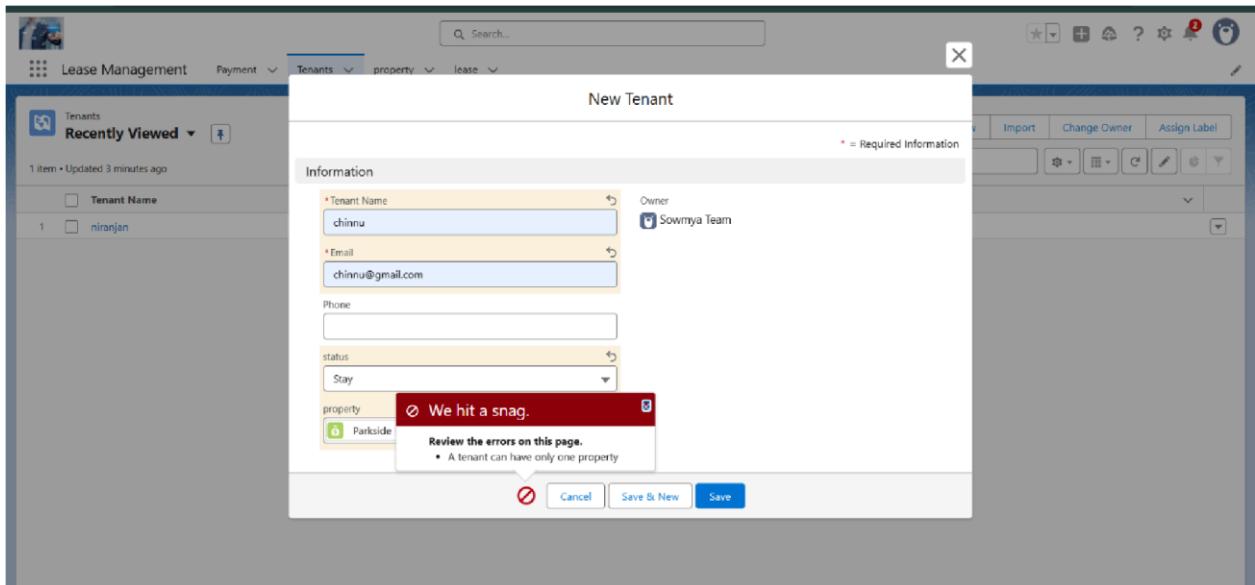


- To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' page for the 'lease' object. The 'Rule Name' is set to 'lease_end_date'. The 'Error Condition Formula' field contains the expression 'end_date_c <= start_date_c'. A tooltip for the 'ABS' function is visible, explaining it returns the absolute value of a number. The 'Active' checkbox is checked.

The screenshot shows the 'Validation Rule Detail' page for the 'lease_end_date' rule. The rule is active and has the formula 'end_date_c <= start_date_c'. The error message is 'Your end date must be greater than start date'. The rule was created by 'Sowmya Team' on 6/19/2025 at 5:37 AM and modified by 'Sowmya Team' on 6/26/2025 at 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' section, 'Classic Email Templates' is selected. A search result for 'Leave approved' is displayed in a modal window.

Email Template Detail:

Field	Value
Email Template Name	Leave approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team, 6/20/2025, 1:08 AM
Modified By	Sowmya Team, 6/20/2025, 1:08 AM

Email Template:

Subject: Leave approved

Plain Text Preview:

```
dear{!Tenant__c.Name},  
  
I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.  
  
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 in a modal window.

Email Template Detail:

Field	Value
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team, 6/20/2025, 1:06 AM
Modified By	Sowmya Team, 6/20/2025, 1:06 AM

Email Template:

Subject: request for approve the leave

Plain Text Preview:

```
Dear {Tenant__c.CreatedBy},  
  
Please approve my leave
```

Cloud icon

Search Setup

Setup Home Object Manager

Q_email template

v Email

Classic Email Templates

Lightning Email Templates

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

Text Email Template Leave rejected

Preview your email template below.

Email Template Detail

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

Available For Use ✓

Last Used Date

Times Used

Help for this Page

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.

Cloud icon

Search Setup

Setup Home Object Manager

Q_email template

v Email

Classic Email Templates

Lightning Email Templates

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

Text Email Template Tenant Email

Preview your email template below.

Email Template Detail

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

Available For Use ✓

Last Used Date

Times Used

Help for this Page

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.

This communication is a friendly reminder regarding your monthly rent payment, which is due on the 1st of each month. Please make sure to remit payment by the due date to avoid any late fees or disruption in services.

The screenshot shows the Salesforce Setup interface under the 'Email' category. A specific email template named 'tenant payment' is selected. The template's details are shown in the main pane, including its name, unique name, encoding, author, and creation date. Below the details, the template's content is previewed, showing a subject line 'Confirmation of Successful Monthly Payment' and a plain text body.

- Approval Process creation

For Tenant Leaving:

The screenshot shows the Salesforce Setup interface under the 'Data' category. An approval process named 'Tenant: TenantApproval' is selected. The process definition details include the process name 'TenantApproval', unique name 'TenantApproval', entry criteria 'Tenant__c.status equals Stay', and record editability 'Administrator ONLY'. The approval assignment email template is set to 'Initial Submitters' with 'Tenant Owner' as the initial submitter. The process is active. Below the definition, 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 named 'Step 1'.

For Check for Vacant:

- Apex Trigger

Create an Apex Trigger

```

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

```

Developer Console - Google Chrome
orgfarm-5dfe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

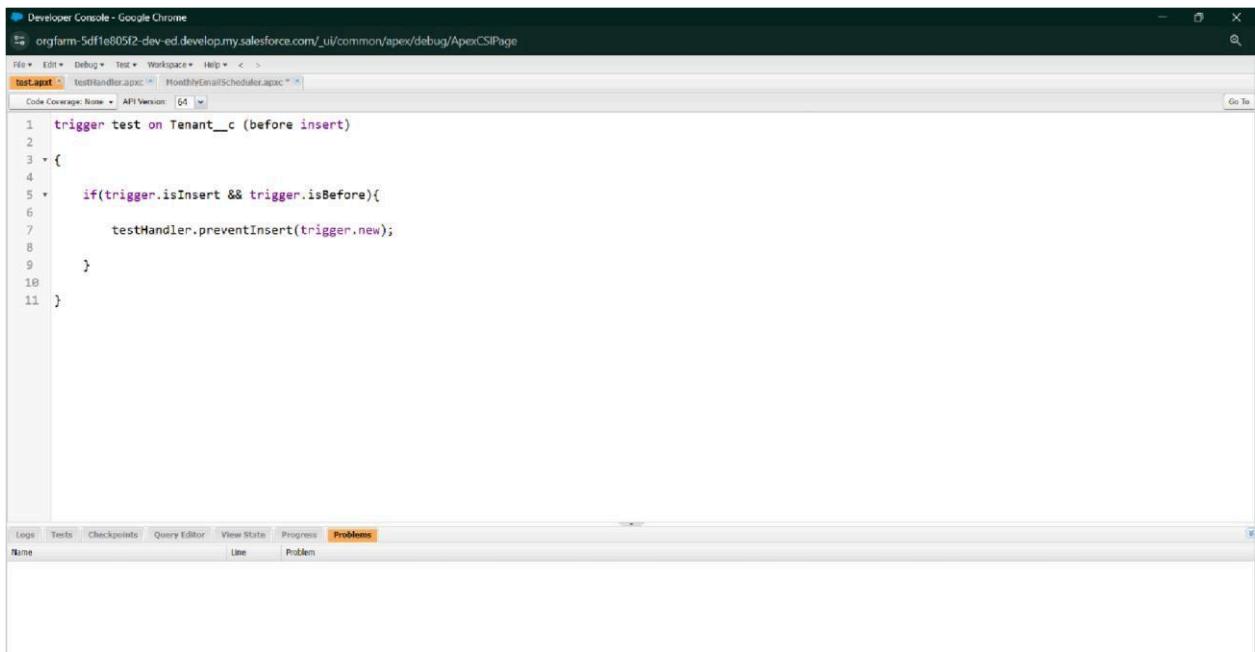
testApex [testHandler.apxc] MonthlyEmailScheduler.apxc

Code Coverage: None API Version: 64 Go To

```
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



Create an Apex Handler class

Developer Console - Google Chrome
orgfarm-5dfe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apex [TestHandler.apxc] MonthlyEmailScheduler.apxc

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(existin
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17             if (newTenant.Property__c != null) {
18
19                 newTenantaddError('A t
20
21             }
22
23     }
24 }
```

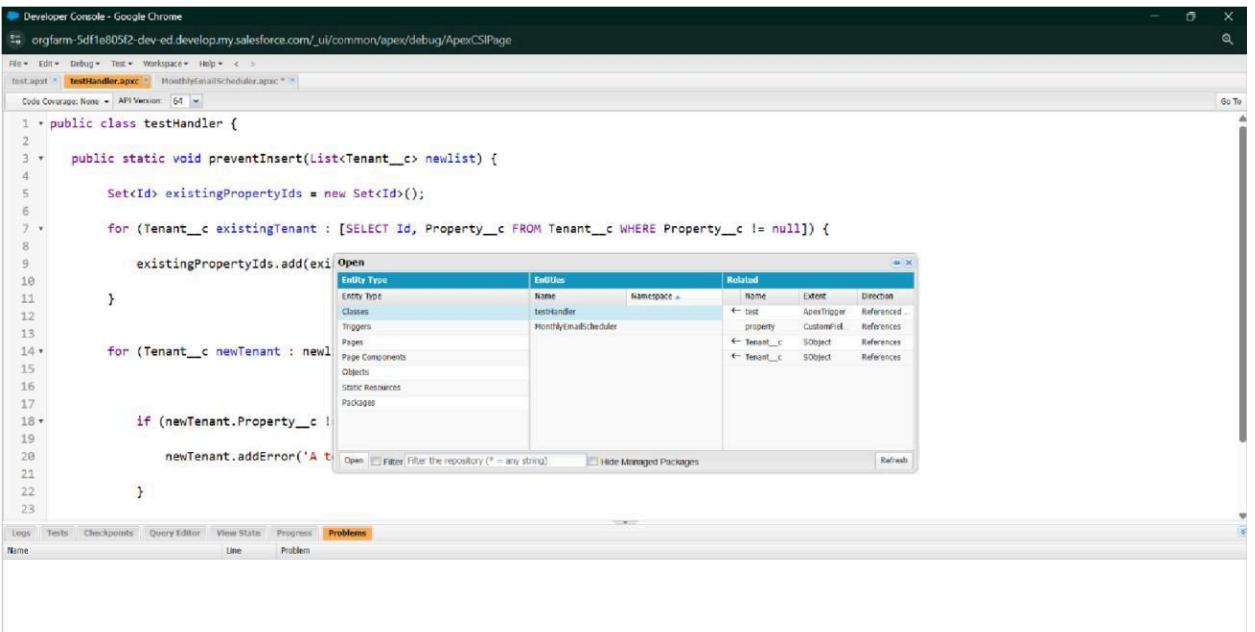
Entity Type Entity Related

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

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem



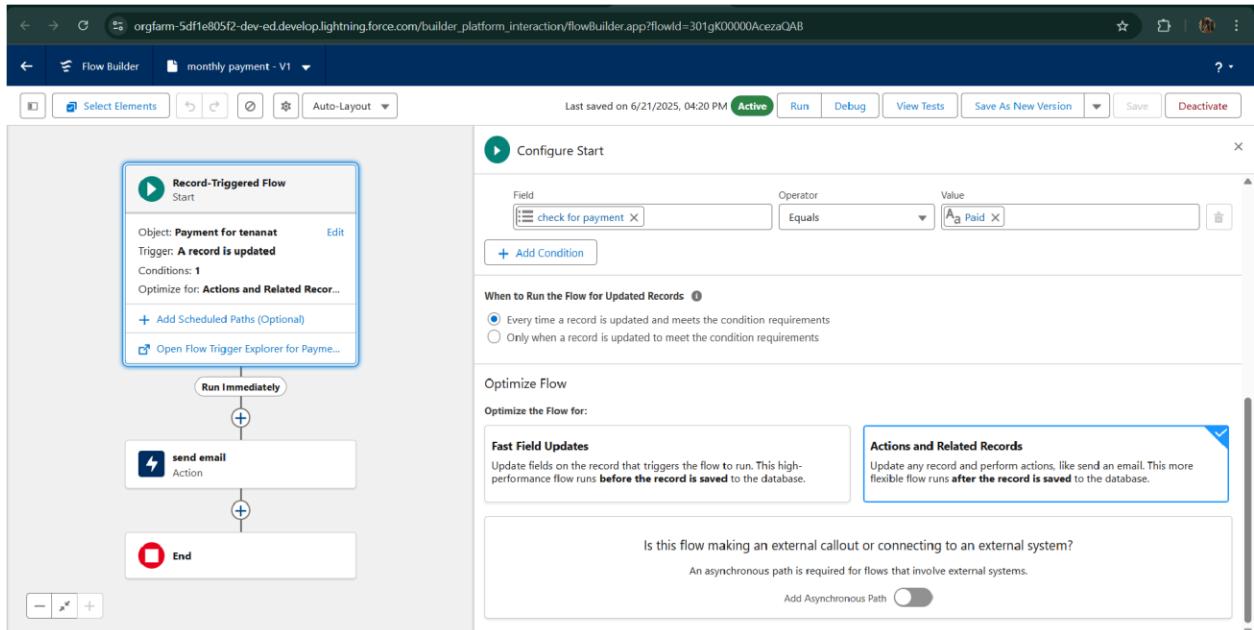
Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCIPage

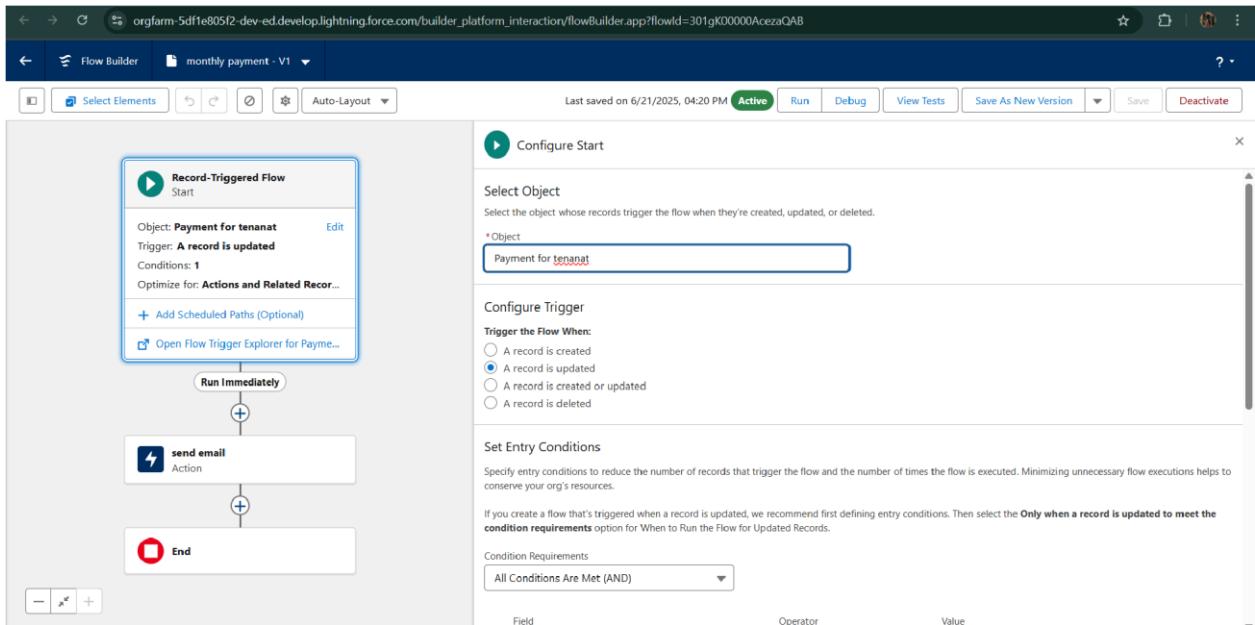
```

File Edit Debug Test Workspace Help < >
TestApex testHandler.apxc MonthlyEmailScheduler.apxc
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
18 *                 if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
19
20 *                     newTenantaddError('A tenant can have only one property');
21
22
23
}
  
```

Logs Tests Checkpoints Query Editor ViewState Progress Problems

● FLOWS





- Schedule class:
Create an Apex Class

```

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

```

Entity Type: Schedulable

Name	Namespace	Related
MonthlyEmailScheduler		Contingent References
Email		CustomField References
Tenant__c		Object References
Tenant__c		Object References

```

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             sendMonthlyEmails();
9         }
10    }
11 }
12
13
14 public static void sendMonthlyEmails() {
15
16     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
17
18     for (Tenant__c tenant : tenants) {
19
20         String recipientEmail = tenant.Email__c;
21
22         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
23
24         String emailSubject = 'Reminder: monthly rent payment due';
25
26         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
27
28         email.setToAddresses(new String[]{recipientEmail});
29
30         email.setSubject(emailSubject);
31
32         email.setPlainTextBody(emailContent);
33
34         messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
35
36     }
37
38 }
39
40 }
41
42

```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The search bar at the top contains "apex". The left sidebar has sections for Email (Apex Exception Email), Custom Code (Apex Classes selected), Apex Test Execution, Apex Test History, Apex Triggers, Environments (Jobs, Apex Flex Queue, Apex Jobs), and Global Search.

The main area displays the "Apex Classes" section for the "MonthlyEmailScheduler" class. The "Apex Class Detail" table shows:

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

The "Class Body" tab is selected, showing the Apex code:

```

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             sendMonthlyEmails();
9         }
10    }
11 }
12
13
14 public static void sendMonthlyEmails() {
15
16     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
17
18     for (Tenant__c tenant : tenants) {
19
20         String recipientEmail = tenant.Email__c;
21
22         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
23
24         String emailSubject = 'Reminder: monthly rent payment due';
25
26         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
27
28         email.setToAddresses(new String[]{recipientEmail});
29
30         email.setSubject(emailSubject);
31
32         email.setPlainTextBody(emailContent);
33
34         messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
35
36     }
37
38 }
39
40 }
41
42

```

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* = Required Information

Tenant Name: Aswini
Email: aswiniamaraadi15@gmail.com
Phone: (905) 223-5567
status: Leaving
property: Imran

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

New Contact Edit New Opportunity

Activity

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

Upcoming & Overdue

No activities to show. Get started by sending an email, scheduling a task, and more.

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

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* = Required Information

Tenant Name: Aswini
Email: aswiniamaraadi15@gmail.com
Phone: (905) 223-5567
status: Leaving
property: Imran

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

Tenant was submitted for approval.

New Contact Edit New Opportunity

Activity

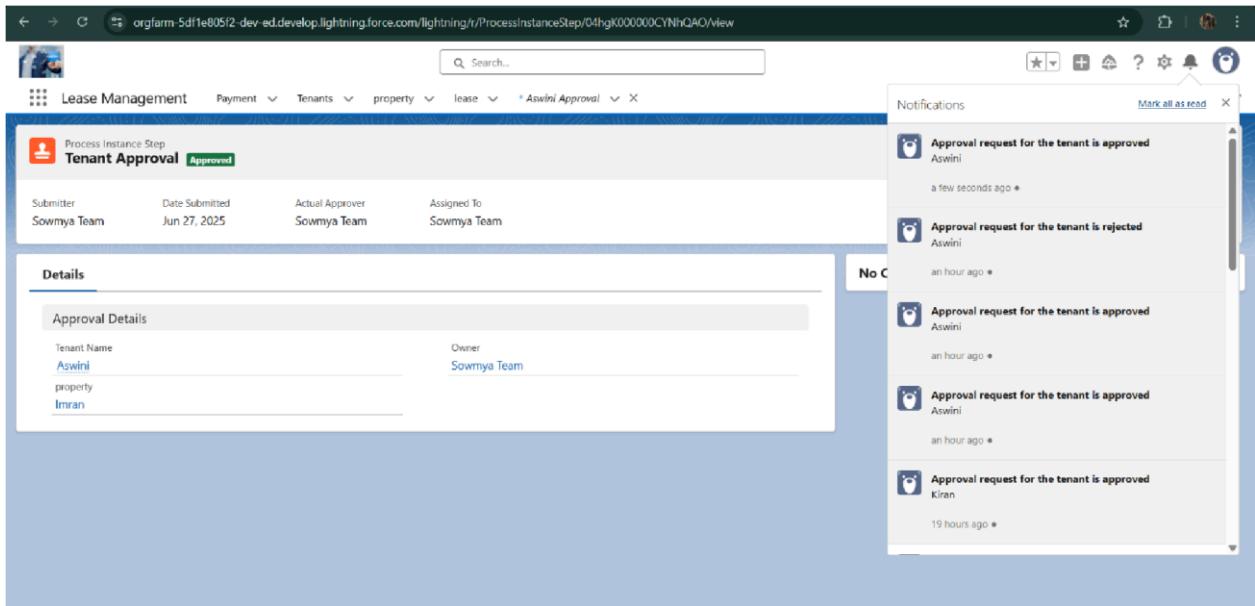
Filters: All time • All activities • All types Refresh • Expand All • View All

Upcoming & Overdue

No activities to show. Get started by sending an email, scheduling a task, and more.

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

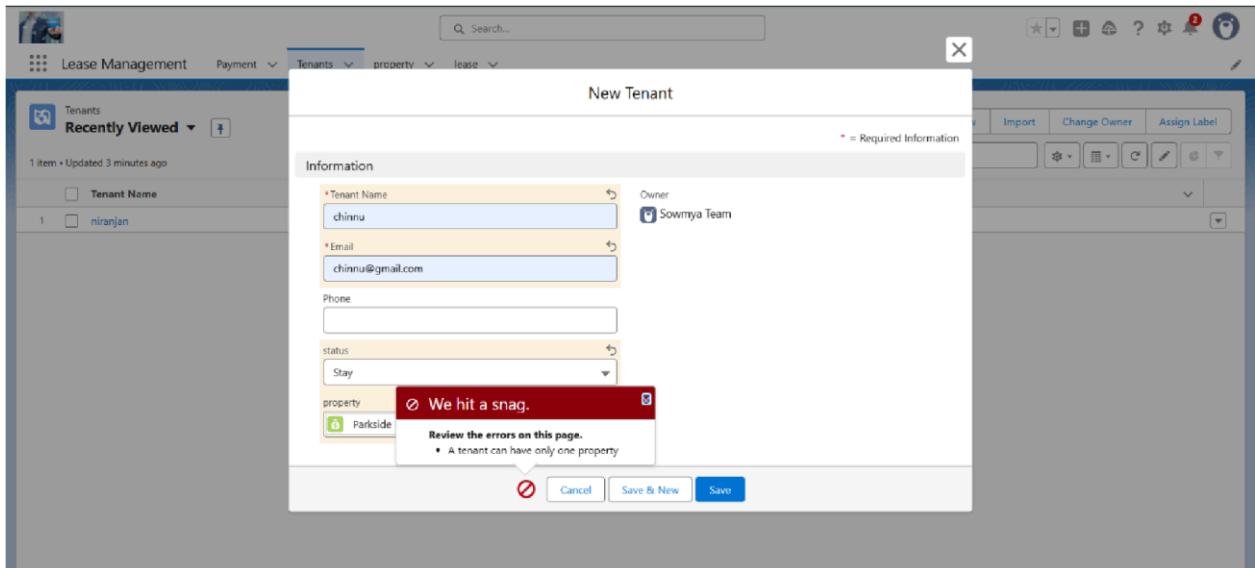
30°C Cloudy Search ENG IN 12:46 27-06-2025



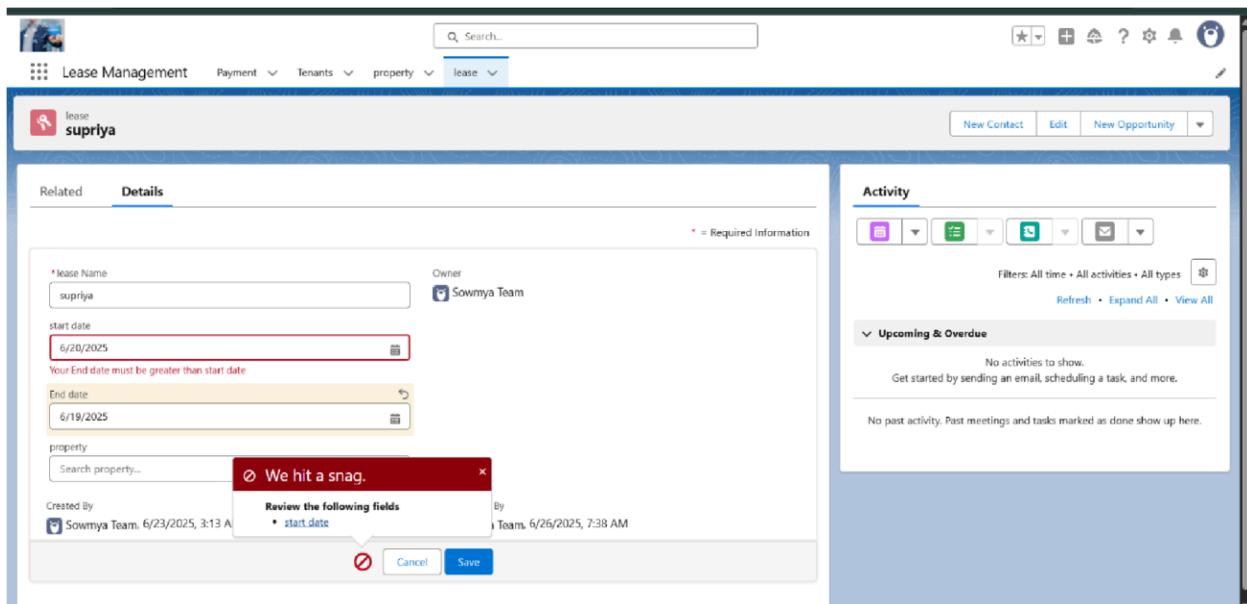
FUNCTIONAL AND PERFORMANCE TESTING

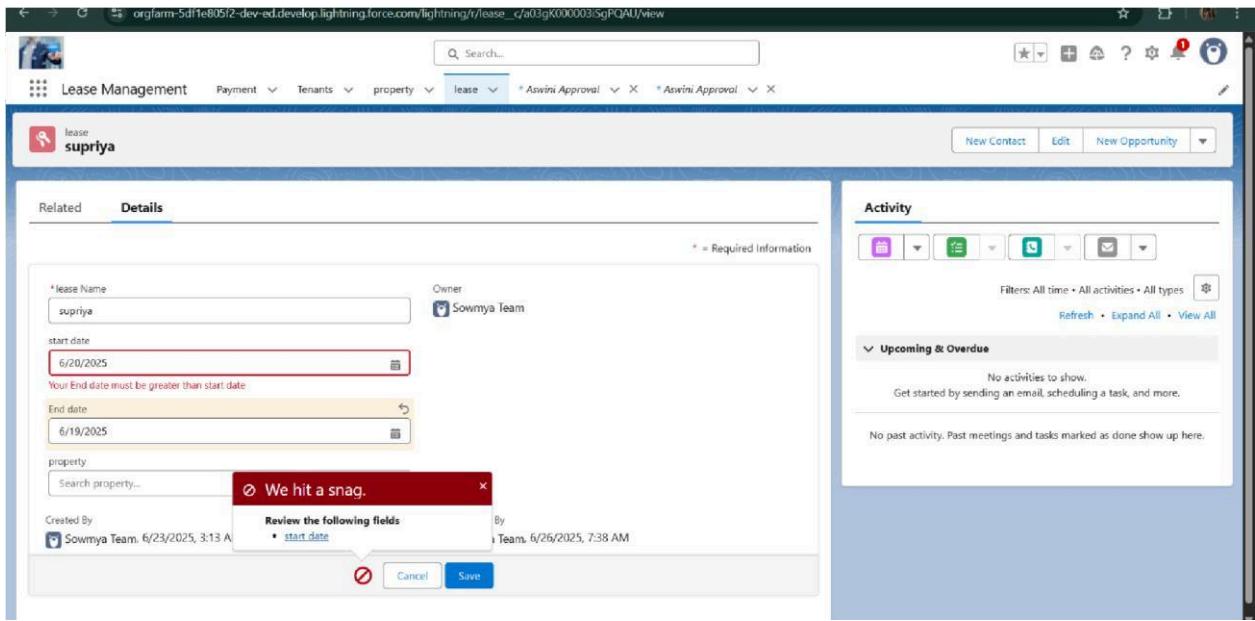
Performance Testing

- Trigger validation by entering duplicate tenant-property records

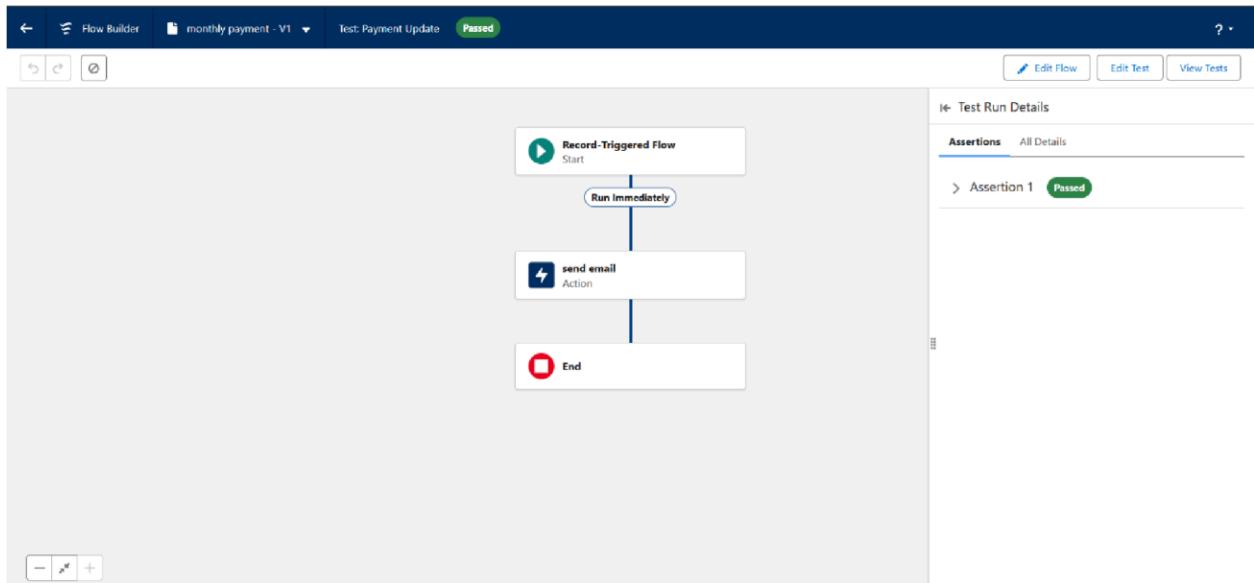


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows a tenant profile for "niranjan". The "Details" tab is selected. The form includes fields for Tenant Name (niranjan), Email (niranjan1506@gmail.com), Phone, status (Stay), property (Parkside Lofts), and owners (Sowmya Team). The "Created By" field shows Sowmya Team on 6/23/2025 at 2:33 AM, and the "Last Modified By" field shows Sowmya Team on 6/23/2025 at 3:58 AM. A sidebar on the right displays notifications:

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

The screenshot shows the same tenant profile for "niranjan". The "Approval History" section lists several steps with their dates, statuses, and assignees. The "Payment" section shows two entries for "Jack" and "Rahul". A sidebar on the right provides a summary of recent activity:

Get started by sending an email, scheduling a task, and more.
No past activity. Past meetings and tasks marked as done show up here.

Step Name	Date	Status	Assigned To
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team
Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team
Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team

Payment (2)

Payment Name
Jack
Rahul

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

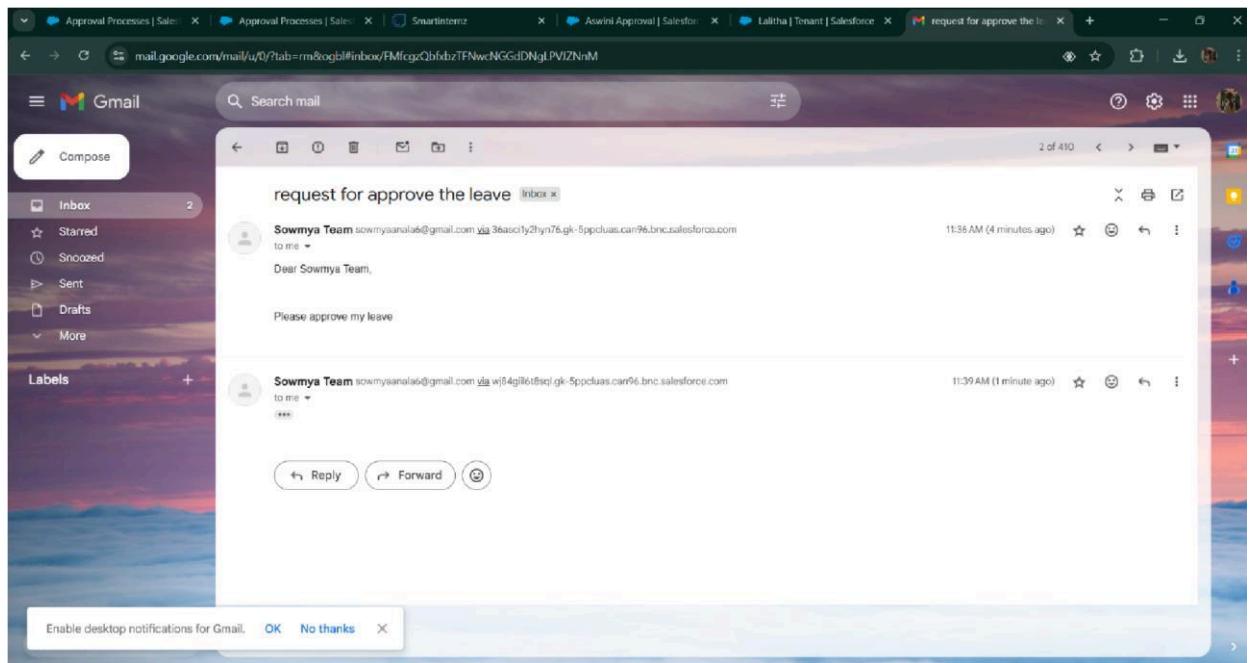
The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar has sections for 'User Interface' (selected), 'Rename Tabs and Labels', and 'Tabs'. A search bar at the top right says 'Search Setup'. The main content area is titled 'Custom Tabs' and contains three tabs: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' tab lists four custom tabs: 'lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'property' (Tab Style: Sack), and 'Tenants' (Tab Style: Map). Each entry has 'Edit | Del' buttons.

- Email alerts

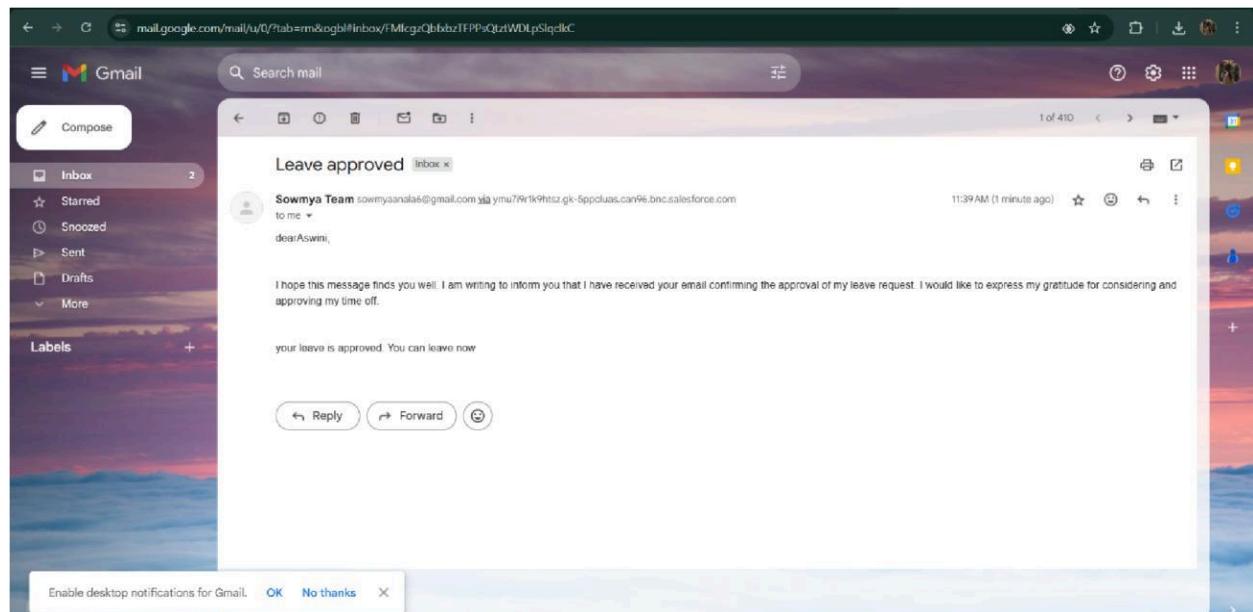
The screenshot shows the 'Lease Management' application interface with the 'Approval History' page open. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. The main content area displays a table titled 'Approval History' with 8 items. The table columns are: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data 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

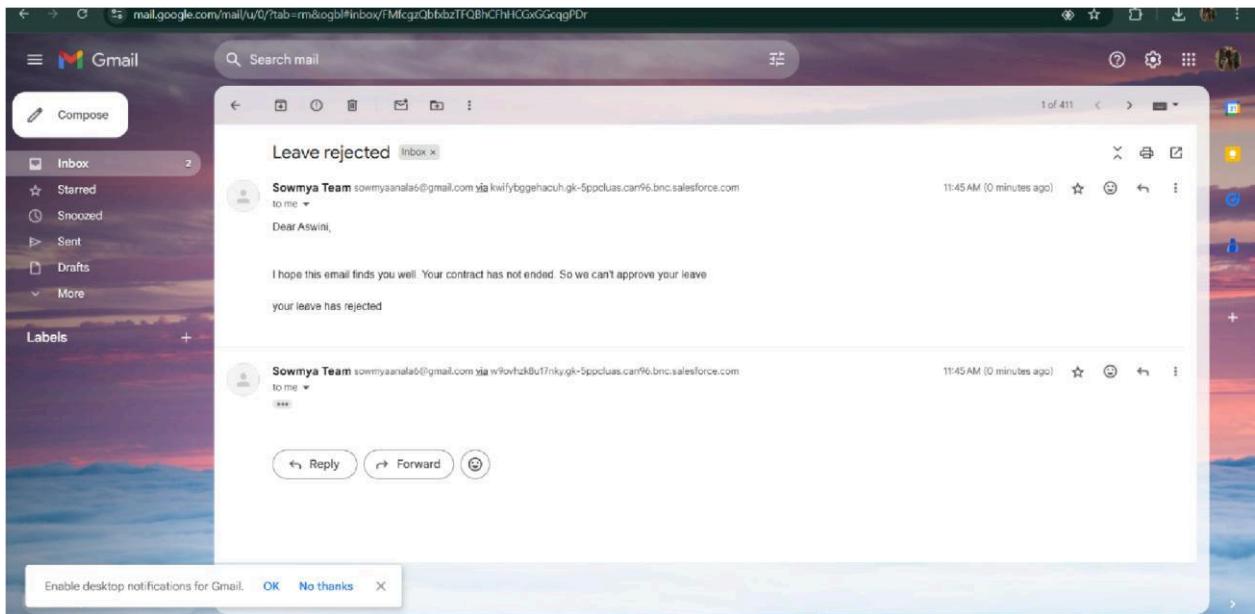
- Request for approve the leave



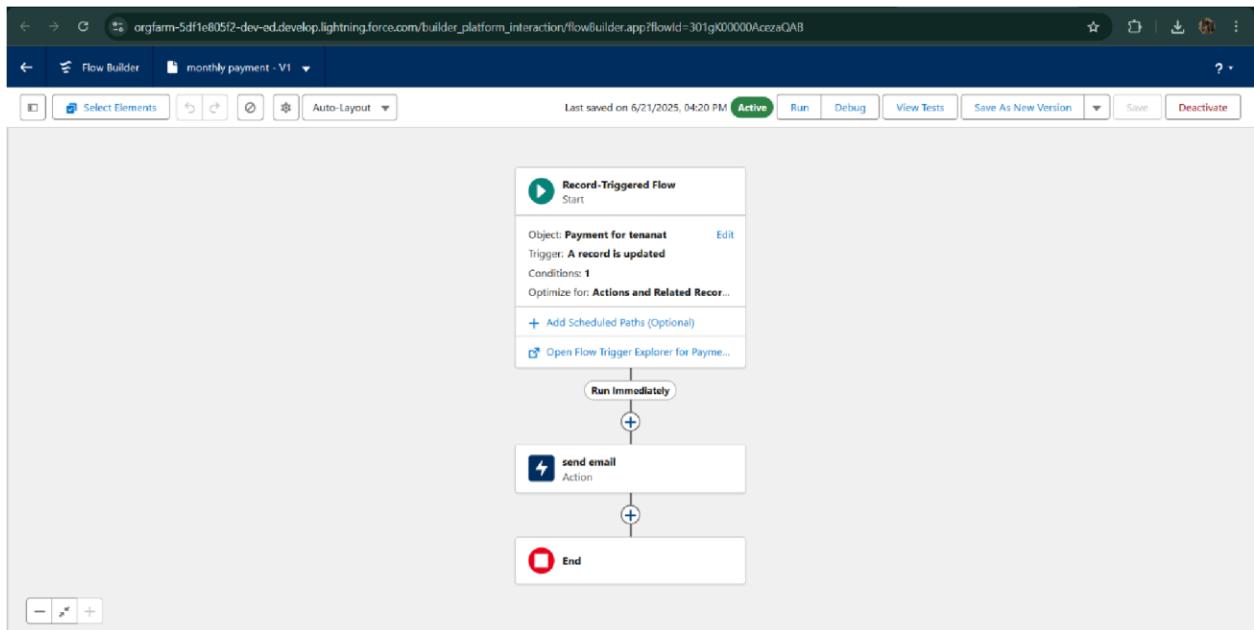
- Leave approved



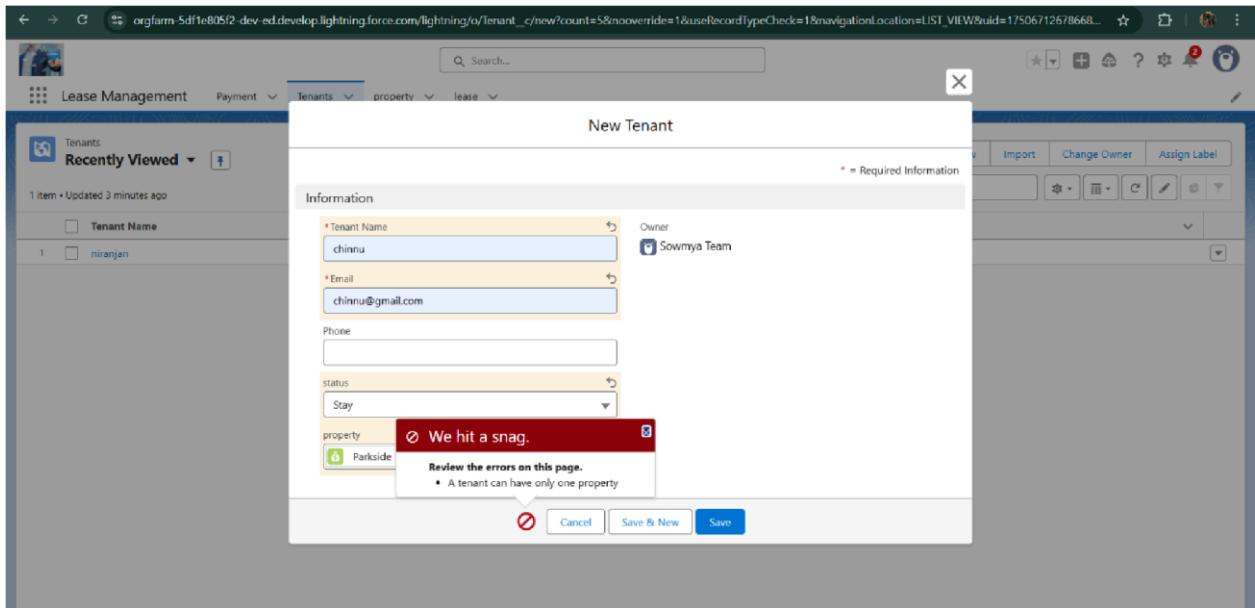
- Leave rejected



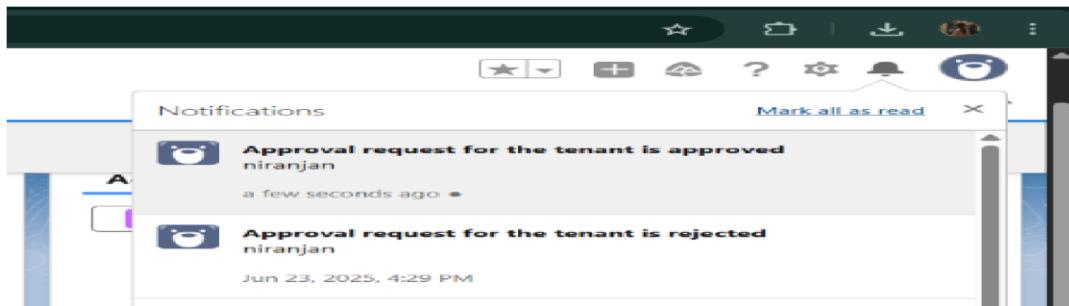
- Flow runs



- 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});  
    }  
}  
}
```