

# **LEASE MANAGEMENT**

**College Name:** Karuppanan mariappan college

**College Code:** Bru3s

**TEAM ID:**

**TEAM MEMBERS:** 4

**Team LeaderName:** AARTHI G

**Email:** [aarthiprasanth2006@gmail.com](mailto:aarthiprasanth2006@gmail.com)

**Team Member1:** ARAVIND M

**Email:** [aravinddarmaraj34@gmail.com](mailto:aravinddarmaraj34@gmail.com)

**Team Member:** BOOPALAN K

**Email:** [boopalanboopalan719@gmail.com](mailto:boopalanboopalan719@gmail.com)

**Team Member:** PRAVEEN P

**Email:** [pp9829690@gmail.com](mailto:pp9829690@gmail.com)

# 1. INTRODUCTION

## 1.1 Project Overview

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



## 1.2 Purpose

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

# DEVELOPMENT PHASE

## Creating Developer Account:

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

The screenshot shows the Salesforce Developer Edition sign-up page. The background features a dark blue gradient with a white sidebar on the left containing the Salesforce logo and promotional text about Agentforce and Data Cloud. The main form is titled "Sign up for your Developer Edition". It includes fields for First name (Aarthi), Last name (G), Job title (student), Work email (aarthiprasanth2006@gmail.com), Company (Karuppanan Mariap), and Country/Region (India). A checkbox for agreeing to the Main Services Agreement is checked. The page also includes a note about org provisioning and a link to the Privacy Statement.

**Sign up for your Developer Edition**

A free Salesforce Platform environment with Agentforce and Data Cloud

First name  Last name

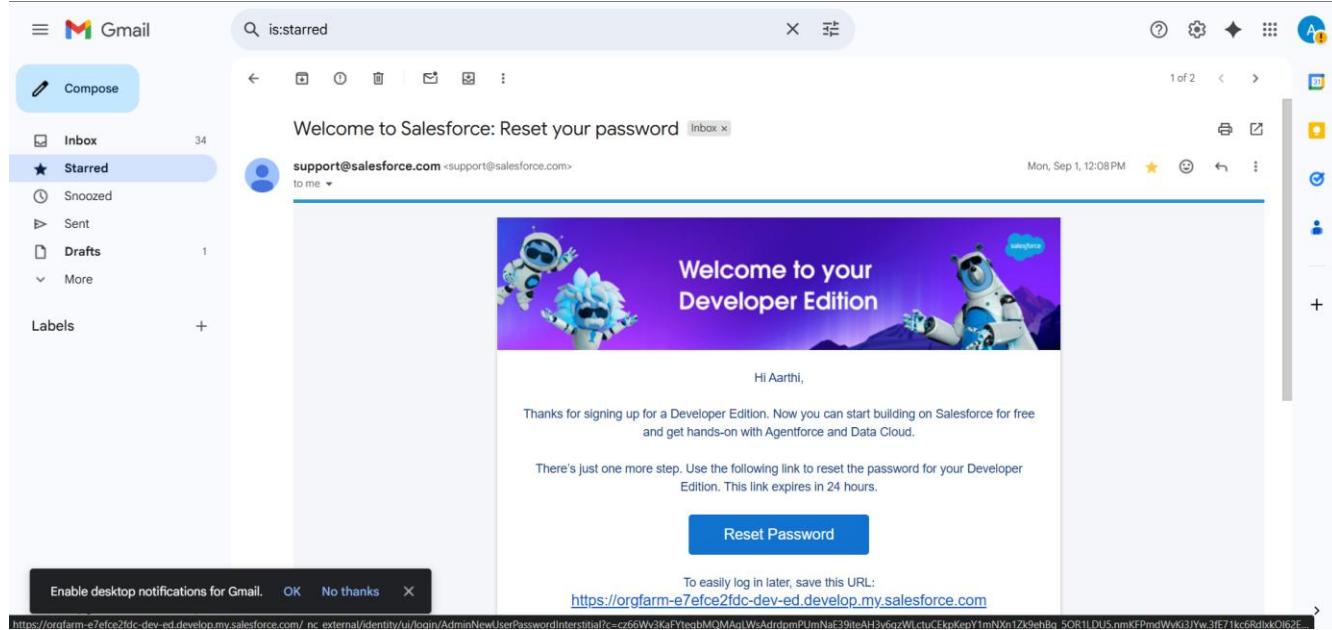
Job title  Work email

Company  Country/Region

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

I agree to the [Main Services Agreement](#) – Developer Services and [Salesforce Program Agreement](#). I acknowledge, as described in the [Developer Documentation](#): (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our [Privacy Statement](#).



- Created objects: Property, Tenant, Lease, Payment

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

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

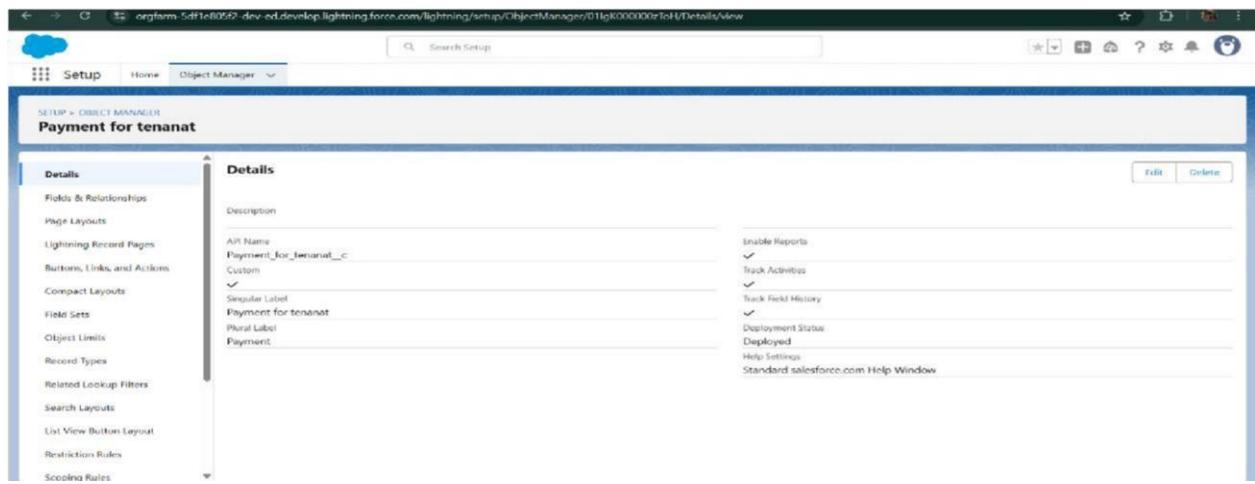
Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules



- Configured fields and relationships

SETUP > OBJECT MANAGER

## property

Fields & Relationships				
9 Items, Sorted by Field Label				
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(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

SETUP > OBJECT MANAGER

## Payment for tenant

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**

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		

Details

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

Quick Find New Deleted Fields Field Dependencies Set History Tracking

Setup Home Object Manager

SETUP > OBJECT MANAGER  
**Tenant**

Fields & Relationships

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)		✓

Details

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

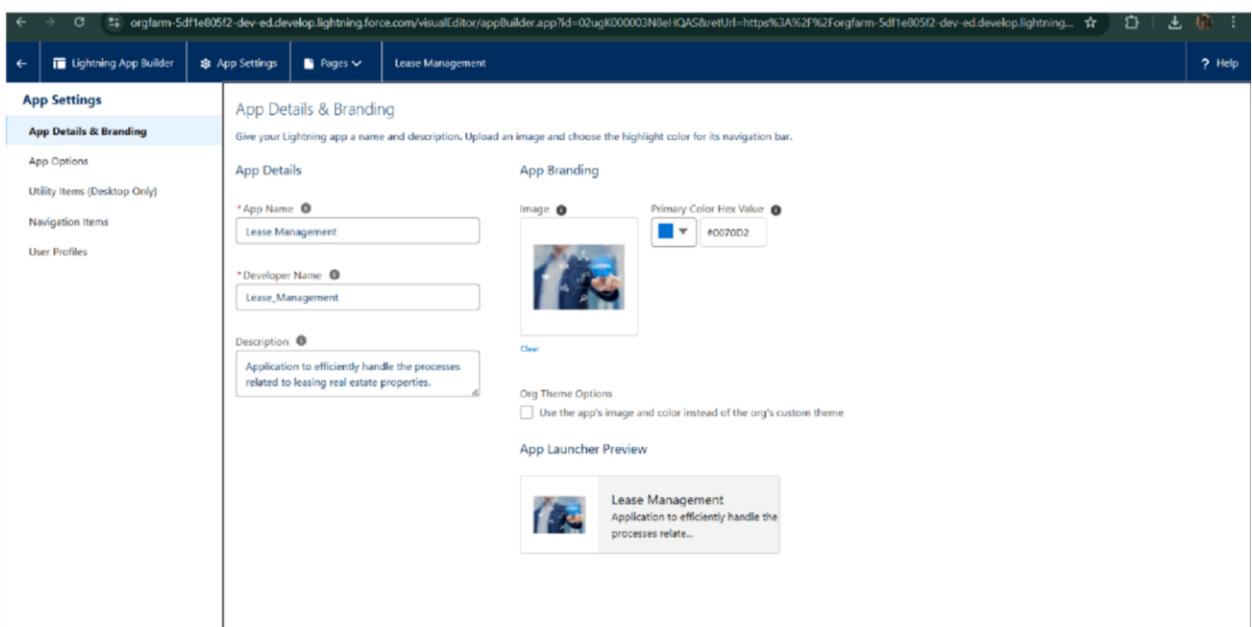
List View Button Layout

Restriction Rules

Scoping Rules

Quick Find New Deleted Fields Field Dependencies Set History Tracking

- Developed Lightning App with relevant tabs



The screenshot shows the "Navigation Items" section of the Lightning App Builder. On the left, a sidebar lists "App Settings", "App Options", and "Utility Items (Desktop Only)". The "Navigation Items" tab is selected, highlighted with a blue border. Below the tabs, there are two main sections: "Available Items" and "Selected Items".

**Available Items:**

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

**Selected Items:**

- Payment
- Tenants
- property
- lease

Below the "Selected Items" list are up and down arrow buttons for reordering.

The screenshot shows the "User Profiles" section of the Lightning App Builder. On the left, a sidebar lists "App Settings", "App Options", and "Utility Items (Desktop Only)". The "User Profiles" tab is selected, highlighted with a blue border. Below the tabs, there are two main sections: "Available Profiles" and "Selected Profiles".

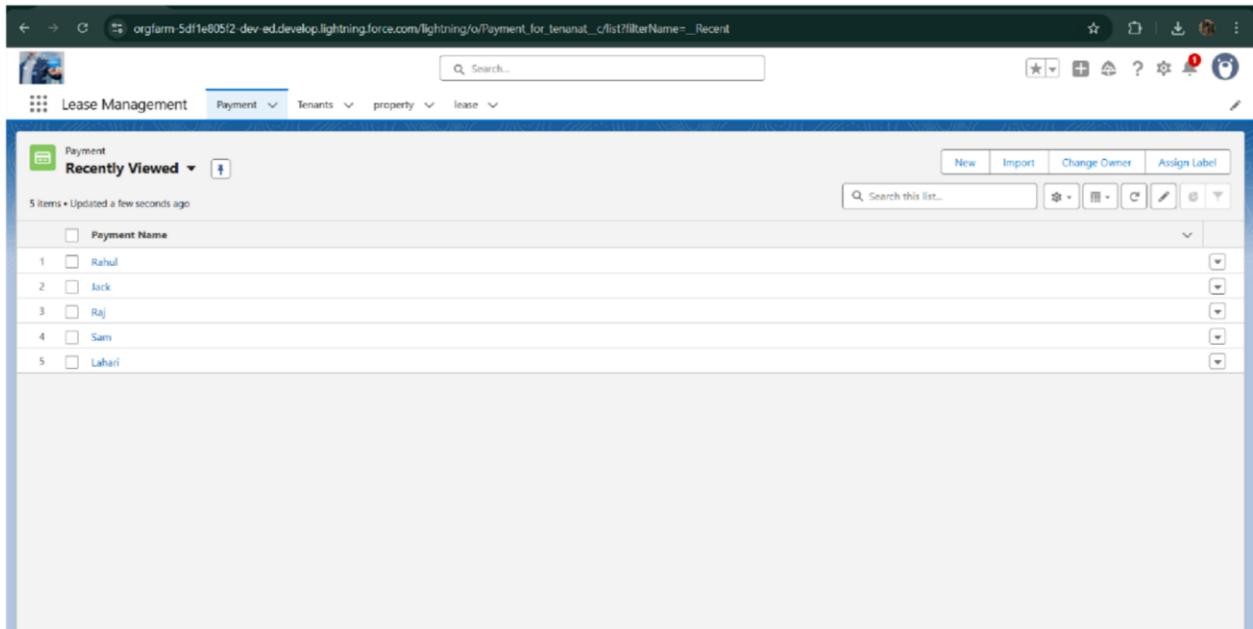
**Available Profiles:**

- 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

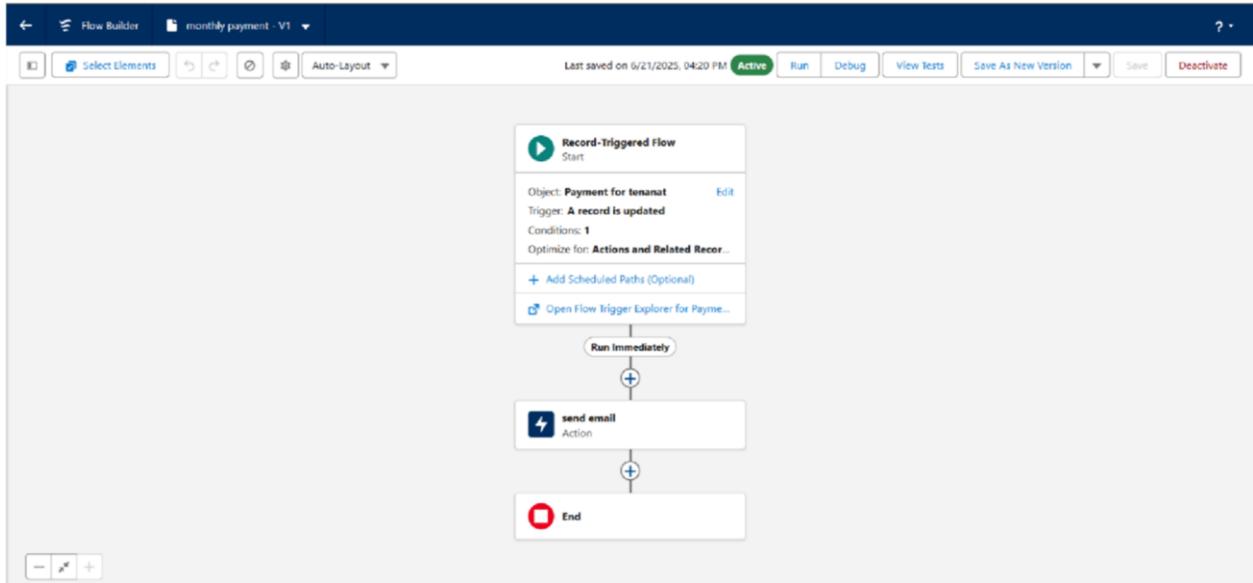
**Selected Profiles:**

- System Administrator

Below the "Selected Profiles" list are up and down arrow buttons for reordering.



- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

Setup > Object Manager lease

**Validation Rule Edit**

Role Name: lease\_end\_date Active:

Description:

Error Condition Formula:

Example: `(Discount_Percent__c > 30)` More Examples...  
If this formula expression is true, display the text defined in the Error Message area

Insert Field Insert Operator `End_date__c <= start_date__c`

Functions: ABS ACOS ADDMONTHS AND ASCII ASIN

Check Syntax

Quick Tips: Operators & Functions

F = Required Information

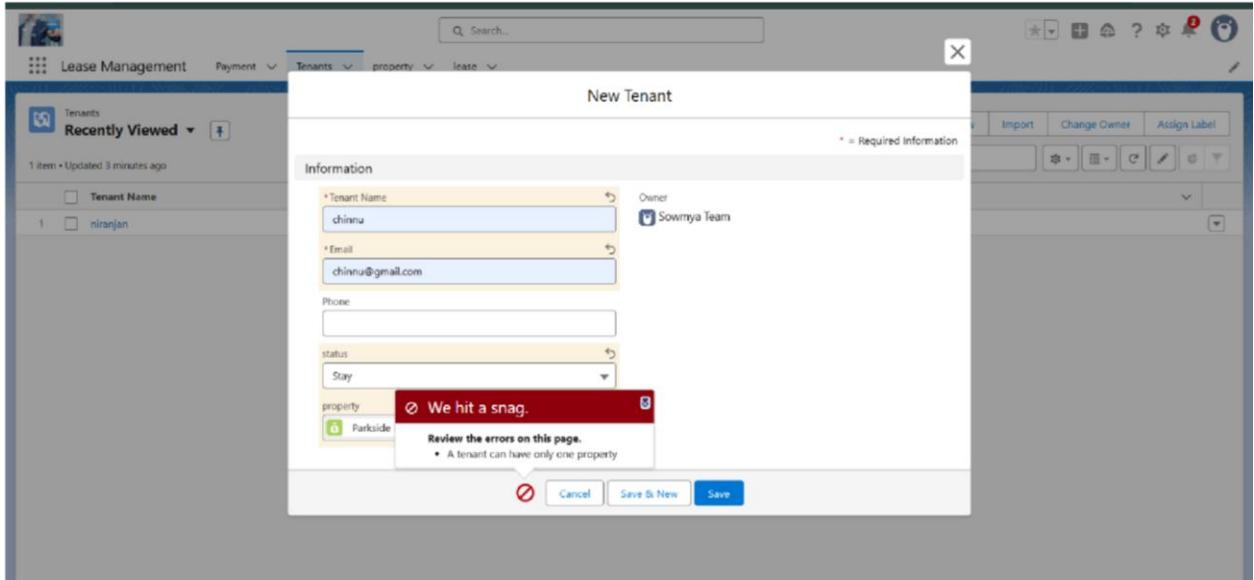
Setup > Object Manager lease

**lease Validation Rule**

Validation Rule Detail

Rule Name	lease_end_date	Active	<input checked="" type="checkbox"/>
Error Condition Formula	<code>End_date__c &lt;= start_date__c</code>	Error Location	start date
Error Message	Your End date must be greater than start date	Created By	Seamya Team 6/19/2025, 5:37 AM
Description		Modified By	Seamya 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 search bar set to 'email template'. The main window displays the 'Classic Email Templates' page for a template named 'Leave approved'. The template details are as follows:

Email Template Detail	Value
Email Template Name	Leave approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Changed]
Created By	Sowmya Team
Modified By	Sowmya Team

The 'Email Template' section shows the following content:

```

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 status is announced. You can leave now.
  
```

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. The main window displays the 'Classic Email Templates' page for a template named 'tenant leaving'. The template details are as follows:

Email Template Detail	Value
Email Template Name	Tenant leaving
Template Unique Name	Tenant_leaving
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Changed]
Created By	Sowmya Team
Modified By	Sowmya Team

The 'Email Template' section shows the following content:

```

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

Please approve my leave
  
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', the 'Leave rejected' template is displayed. The template details are as follows:

Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Saamya.Team [Change]
Description	
Created By	Saamya.Team 6/20/2025, 1:11 AM
Modified By	Saamya.Team 6/20/2025, 1:11 AM

The 'Available For Use' checkbox is checked. The 'Email Template' preview shows the subject 'Leave rejected' and the 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 'Email' section selected. Under 'Classic Email Templates', the 'Tenant Email' template is displayed. The template details are as follows:

Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Saamya.Team [Change]
Description	
Created By	Saamya.Team 6/20/2025, 1:12 AM
Modified By	Saamya.Team 6/20/2025, 1:12 AM

The 'Available For Use' checkbox is checked. The 'Email Template' preview shows the subject 'Urgent: Monthly Rent Payment Reminder' and the 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. The main content area displays the 'Classic Email Templates' page for 'tenant payment'. The template details include:

- Email Template Detail**
  - Email Templates from Salesforce: Untitled Public Classic Email Templates
  - Email Template Name: tenant payment
  - Template Unique Name: tenant\_payment
  - Encoding: Unicode (UTF-8)
  - Author: Scamya Team [Change]
  - Description: Created By: Scamya Team, 6/20/2025, 1:13 AM
  - Available For Use: checked
  - Last Used Date: Times Used: 0
  - Modified By: Scamya Team, 6/20/2025, 1:13 AM
- Email Template**
  - Subject: Confirmation of Successful Monthly Payment
  - Plain Text Preview: Dear {Tenant\_c.Email\_c},  
We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

- Approval Process creation

### For Tenant Leaving:

The screenshot shows the Salesforce Setup interface with the search bar set to 'approval'. The left sidebar under 'Data' has 'Approval Processes' selected. The main content area displays the 'Approval Processes' page for 'Tenant: TenantApproval'. The process details include:

- Process Definition Detail**
  - Process Name: TenantApproval
  - Unique Name: TenantApproval
  - Description:
  - Entry Criteria: Tenant\_\_r.status equals Stay
  - Record Editability: Administrator ONLY
  - Active: checked
  - Next Automated Approver Determined By:
  - Allow Submitters to Recall Approval Requests:
- Approval Assignment Email Template**
  - Initial Submitter: Tenant Owner
  - Created By: Scamya Team, 6/23/2025, 3:41 AM
  - Modified By: Scamya Team, 6/26/2025, 11:57 PM
- Initial Submission Actions**

Action Type	Description
Record Lock	Lock the record from being edited
- Approval Steps**

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

## For Check for Vacant:

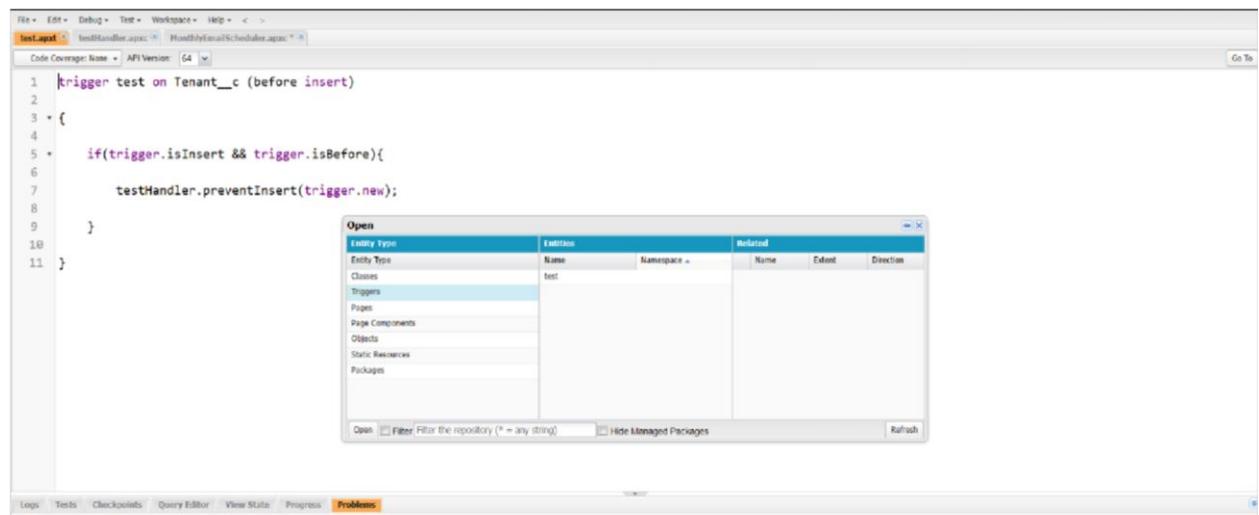
The screenshot shows the Salesforce Setup interface for 'Approval Processes'. The process is named 'check for vacant' and is set to 'Active'. It has one approval step named 'step1' assigned to 'User: Sowmya Team'. The initial submission action 'Record Lock' is described as locking the record from being edited.

Action	Type	Description
Record Lock		Lock the record from being edited
Email Alert		PLEASE APPROVE MY LEAVE

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

- Apex Trigger

Create an Apex Trigger



The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for test.apex, testHandler.apex, and MonthlyEmailScheduler.apex. Below the tabs is a dropdown for Code Coverage Name and API Version (54). The main area contains the following Apex code:

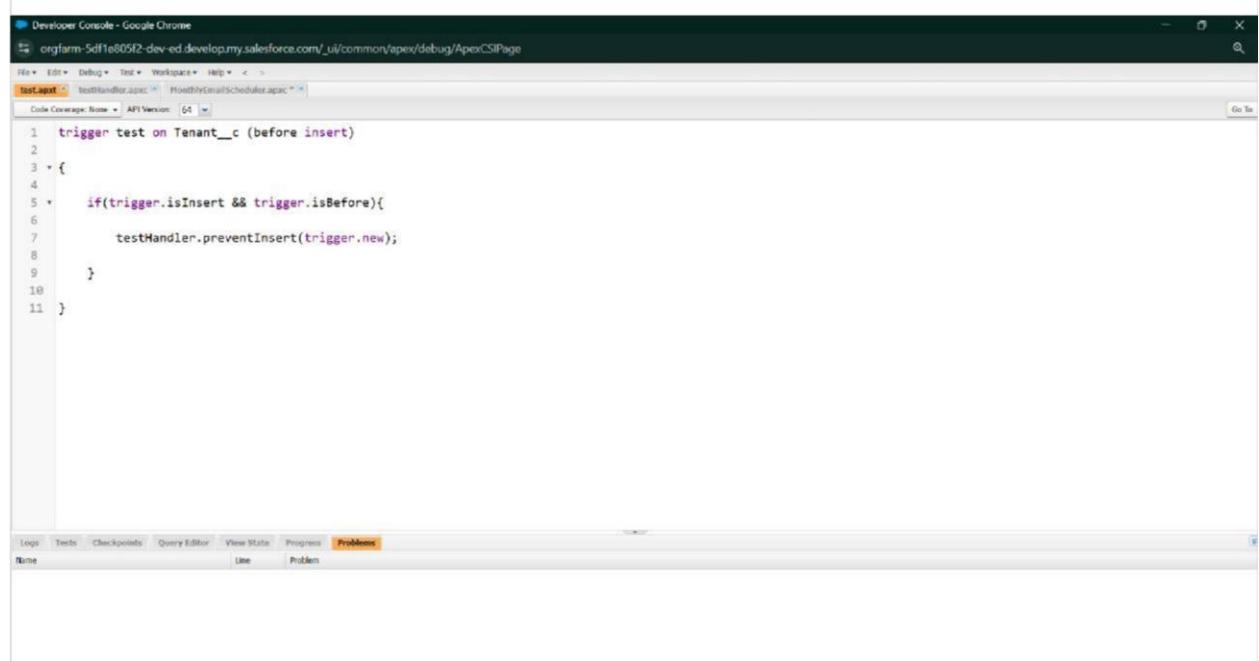
```

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

```

A modal dialog titled "Open" is displayed, listing various entity types: Entity Type (Classes, Triggers, Pages, Page Components, Objects, Static Resources, Packages), Entity (Name: test), and Related (Name, Extent, Direction). The "triggers" row is selected. At the bottom of the modal are Open, Filter, Hide Managed Packages, and Refresh buttons.

Below the code editor, a toolbar with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems is visible. The Problems tab is currently selected. The status bar at the bottom shows the URL orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage.



The second screenshot shows the Developer Console window titled "Developer Console - Google Chrome". It has a similar header and toolbar. The code editor contains the same Apex code as the first screenshot. The results pane below the code editor is completely blank.

Create an Apex Handler class

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 54 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

None Line Problem

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex testHandler.apc MonthlyEmailScheduler.apc

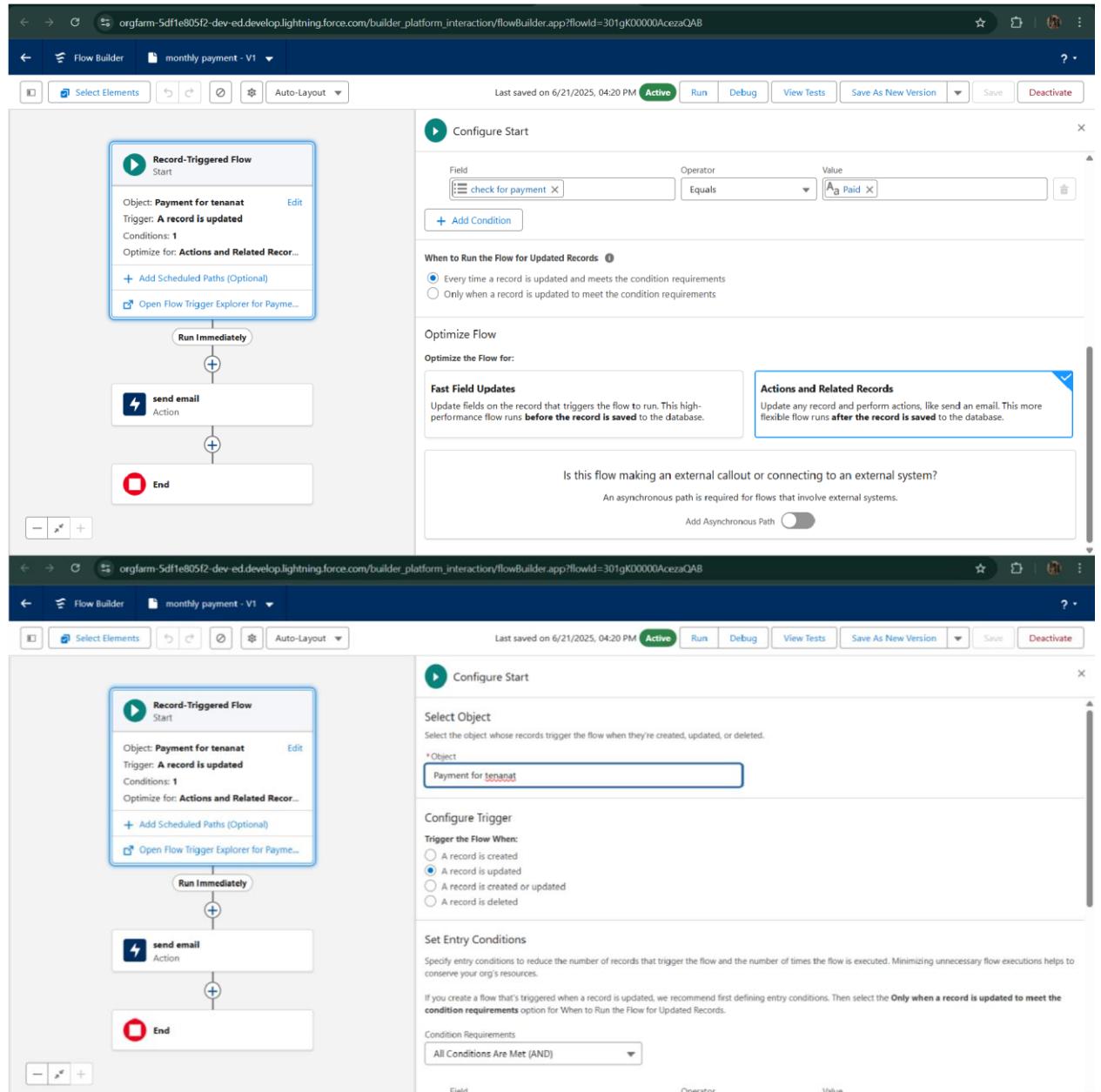
Code Coverage: None API Version: 54 Go To

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

Logs Tests Checkpoints Query Editor View State Progress Problems

None Line Problem

- FLOWS



- Schedule class:  
Create an Apex Class

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apexp testHandler.apexp MonthlyEmailScheduler.apc

Code Coverage: None API Version: 54 Go To

```

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

```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Open Entity Type Edition Related

Name	Namespace	Name	Parent	CustomField	References
MonthlyEmailScheduler		Email			
		← Tenant__c	Object		References
		← Tenant__c	Object		References

Open Filter Hide Managed Packages Refresh

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apexp testHandler.apexp MonthlyEmailScheduler.apc

Code Coverage: None API Version: 54 Reset 50%

```

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 = "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
36         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38     }
39
40 }
41
42

```

Logs Tests Checkpoints Query Editor View Status Progress Problems

## 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), Environments (Jobs), and Didn't find what you're looking for? Try using Global Search. The main content area displays the Apex Class Detail for MonthlyEmailScheduler. The class body contains the following code:

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

The screenshot shows the Lease Management detail page for Tenant Aswini. The top navigation bar includes tabs for Payment, Tenants (selected), property, and lease. The main content area has tabs for Related and Details. The Details tab displays the following information:

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

Below this, it shows Created By: Sowmya Team, 6/26/2025, 6:05 AM and Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM. On the right side, there is an Activity sidebar with buttons for New Contact, Edit, New Opportunity, New Case, New Lead, Delete, Clone, Change Owner, Printable View, Refresh, Submit for Approval, and Edit Labels. It also displays a message: "No activities to show. Get started by sending an email, scheduling a task, and more." and "No past activity. Past meetings and tasks marked as done show up here."

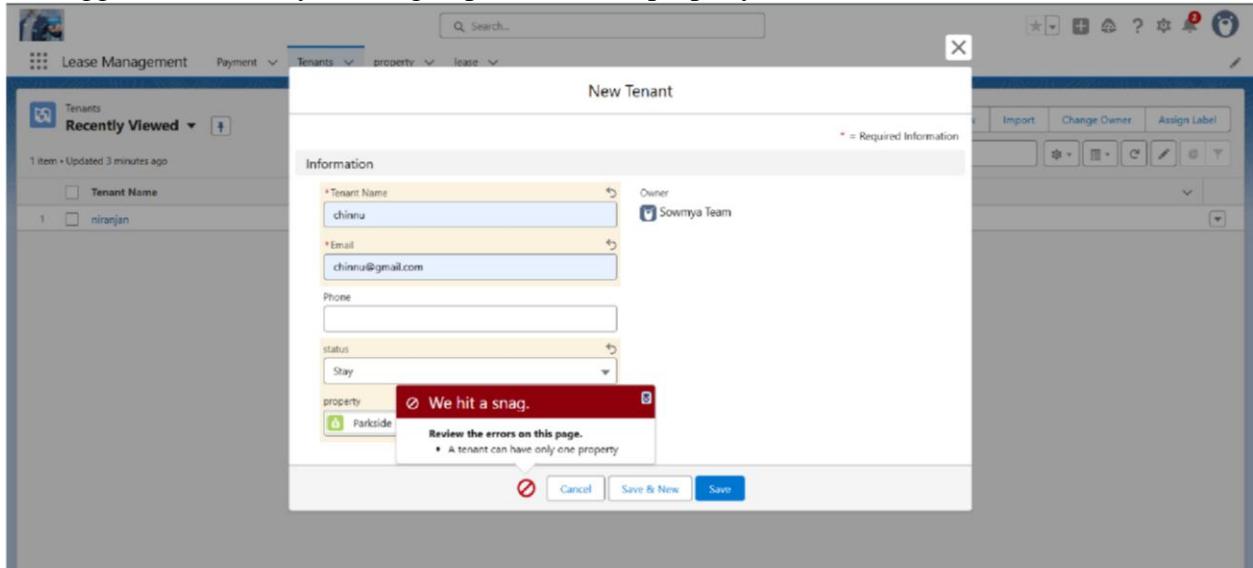
The screenshot shows a Salesforce Lightning page for 'Lease Management'. The top navigation bar includes tabs for 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A green success message at the top right states 'Tenant was submitted for approval.' The main area displays a form for a tenant named 'Aswini' with fields for 'Email' (aswinimaraadi15@gmail.com), 'Phone' ((905) 223-5567), 'status' (Leaving), and 'property' (Imran). The owner is listed as 'Sowmya Team'. A note indicates that the last modified by 'Sowmya Team' on June 26, 2025, at 6:05 AM. On the right side, there is an 'Activity' section with a button to 'New Contact', 'Edit', and 'New Opportunity'. The bottom of the screen shows a Windows taskbar with various icons and the system tray indicating the date and time as 27-06-2025.

The screenshot shows a Salesforce Lightning page for a 'Process Instance Step' titled 'Tenant Approval'. The status is 'Approved'. The details section shows the 'Submitter' as 'Sowmya Team', 'Date Submitted' as 'Jun 27, 2025', 'Actual Approver' as 'Sowmya Team', and 'Assigned To' as 'Sowmya Team'. The 'Approval Details' section shows the tenant name 'Aswini' and the owner 'Sowmya Team'. On the right, a 'Notifications' sidebar lists five recent notifications all from 'Aswini': 'Approval request for the tenant is approved' (a few seconds ago), 'Approval request for the tenant is rejected' (an hour ago), 'Approval request for the tenant is approved' (an hour ago), 'Approval request for the tenant is approved' (an hour ago), and 'Approval request for the tenant is approved' (19 hours ago).

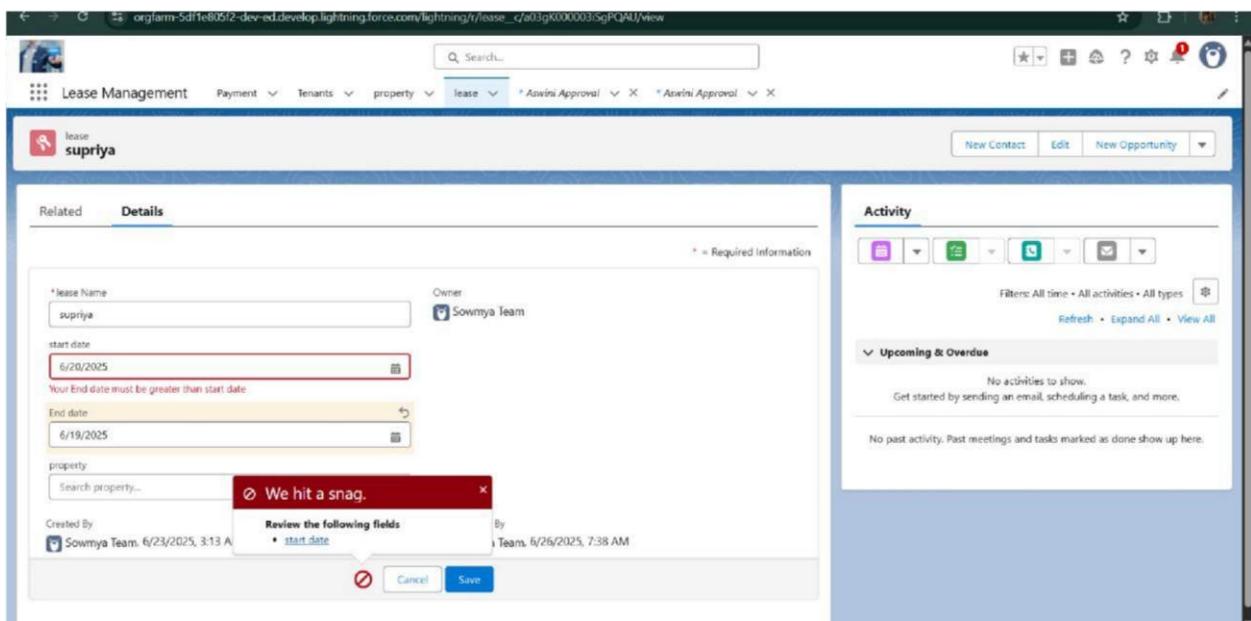
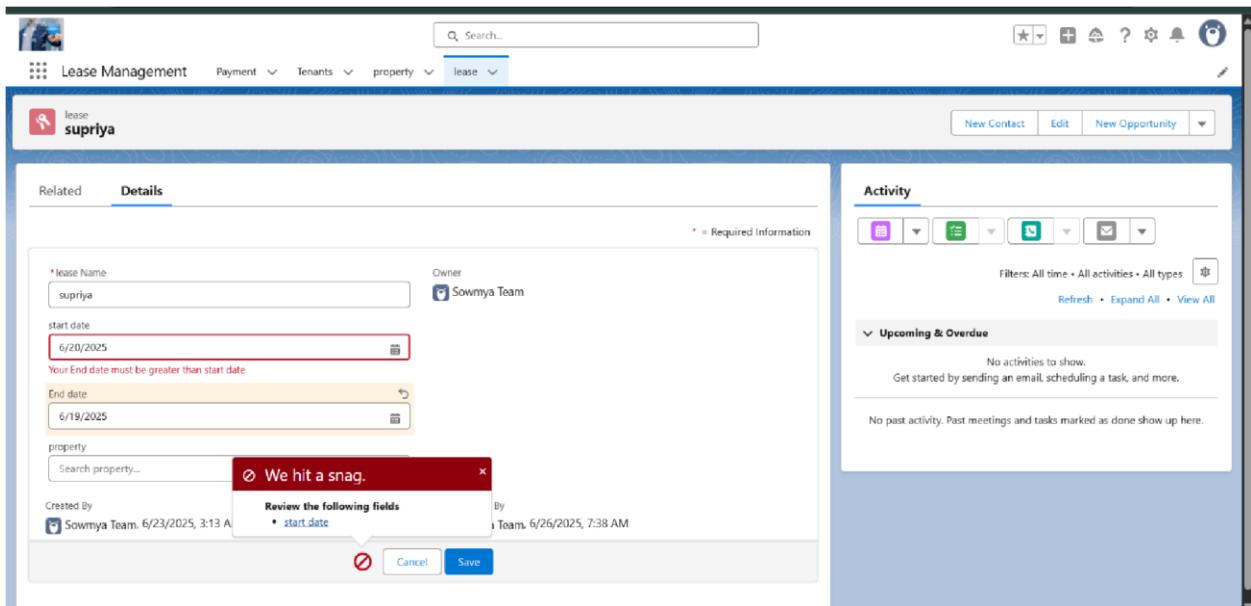
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

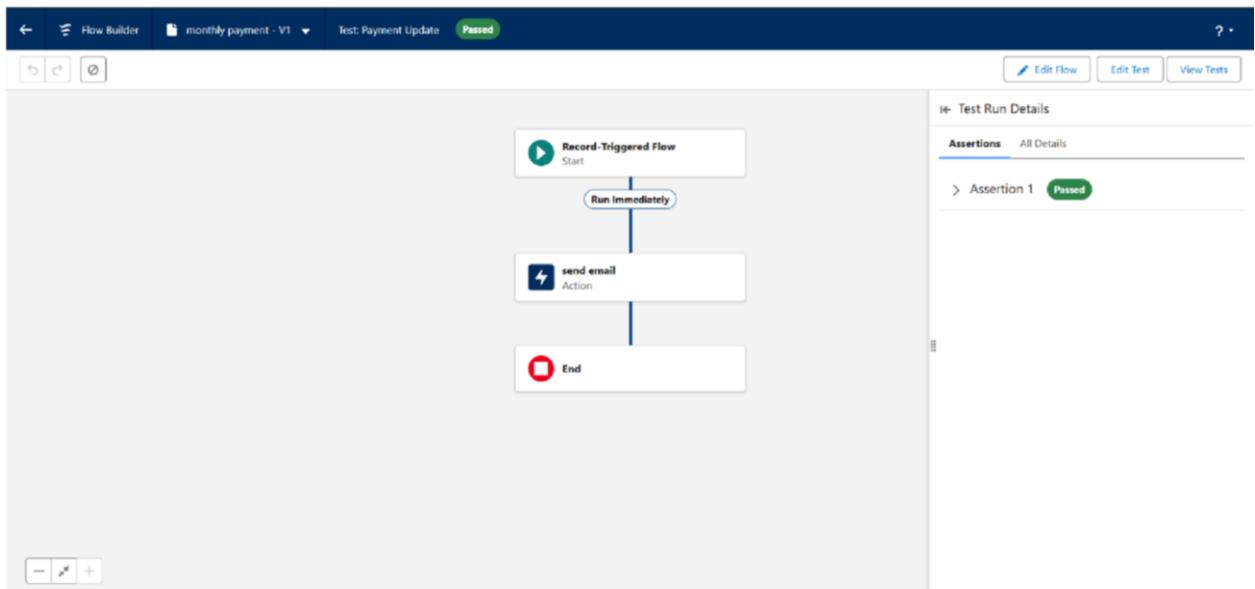
- Trigger validation by entering duplicate tenant-property records



- Validation Rule checking



- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the Lease Management system interface with the following details:

- Page:** Lease Management > Payment > Tenants > property > lease > niranjan Approval
- Tenant Profile:**
  - Related:** Details
  - Details:**
    - Tenant Name: niranjan
    - Email: niranjan1506@gmail.com
    - Phone:
    - Status: Stay
    - Property: Parkside Lofts
    - Created By: Sowmya Team, 6/23/2025, 2:33 AM
    - Last Modified By: Sowmya Team, 6/23/2025, 3:58 AM
- Notifications:**
  - Approval request for the tenant is approved niranjan (a few seconds ago)
  - Approval request for the tenant is rejected niranjan (Jun 23, 2025, 4:29 PM)
  - Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:25 PM)
  - Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:14 PM)
  - New Guidance Center learning resource available Define Your Sales Process Learn how to guide reps through the sales process. (Jun 20, 2025, 1:28 PM)

The screenshot shows a software application window titled "Lease Management". The top navigation bar includes tabs for "Lease Management", "Payment", "Tenants", "property", and "lease". A search bar is located at the top right. The main content area displays two sections: "Approval History (6+)" and "Payment (2)".

**Approval History (6+)**

Step Name	Date	Status	Assigned To
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 2:59 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

A sidebar on the right side of the interface includes buttons for "New Contact", "Edit", and "New Opportunity", along with a message: "Get started by sending an email, scheduling a task, and more." Below this, it says "No past activity. Past meetings and tasks marked as done show up here."

---

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

**Custom Tabs**

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Action	Label	Tab Style	Description
Edit   Del	Lease	Key	
Edit   Del	Payment	Credit card	
Edit   Del	Property	Sack	
Edit   Del	Tenants	Map	

**Web Tabs**

No Web Tabs have been defined.

**Visualforce Tabs**

No Visualforce Tabs have been defined.

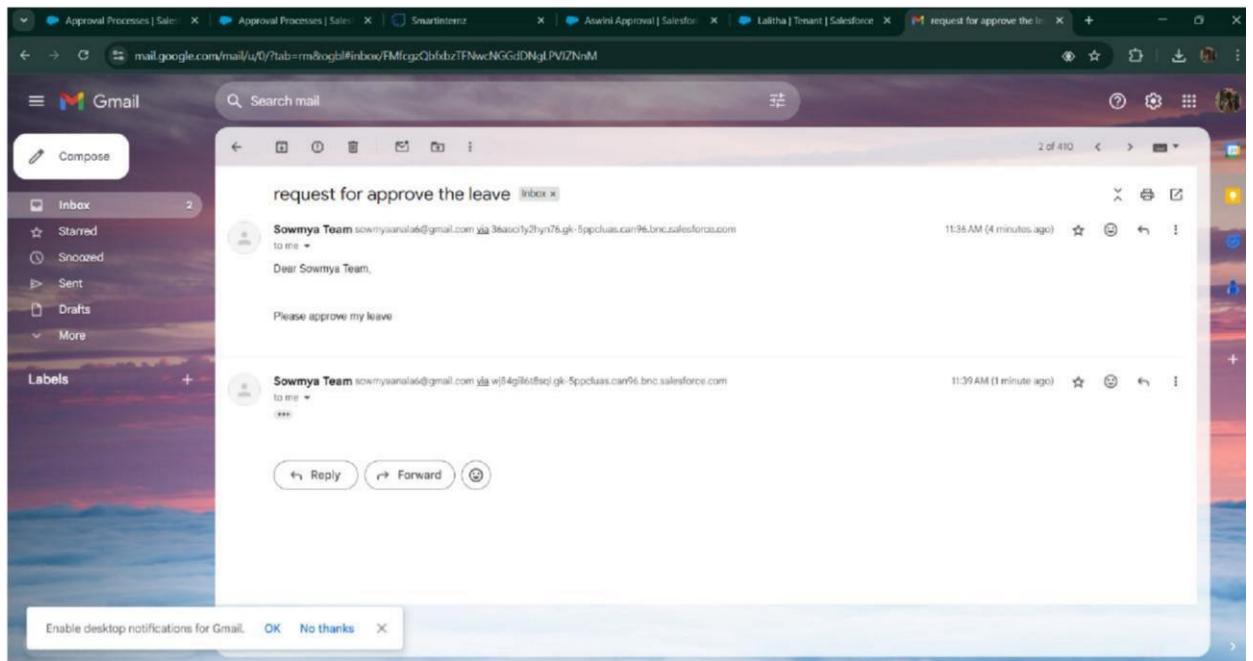
- Email alerts

**Approval History**

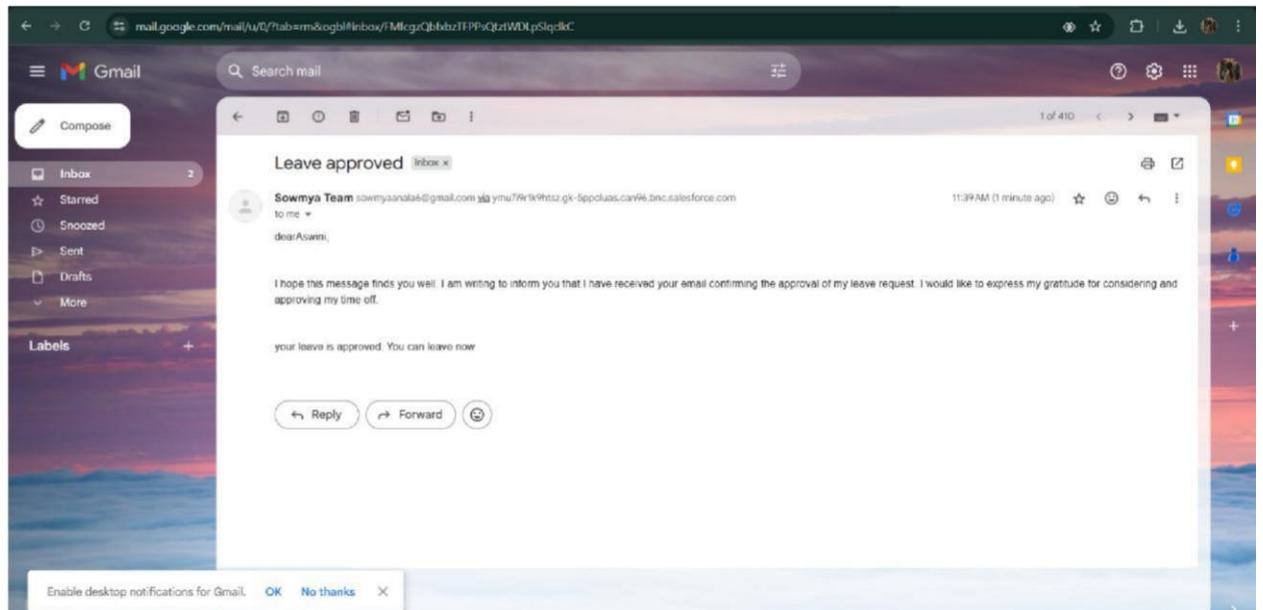
8 items • Sorted by Is Pending • Updated a few seconds ago

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

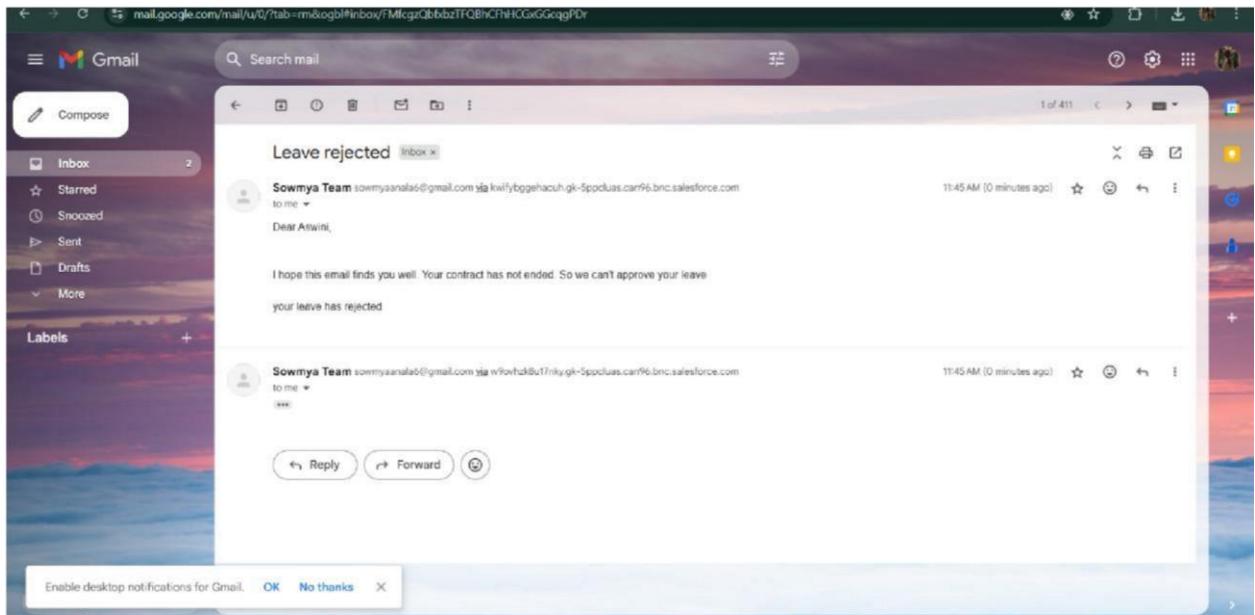
- Request for approve the leave



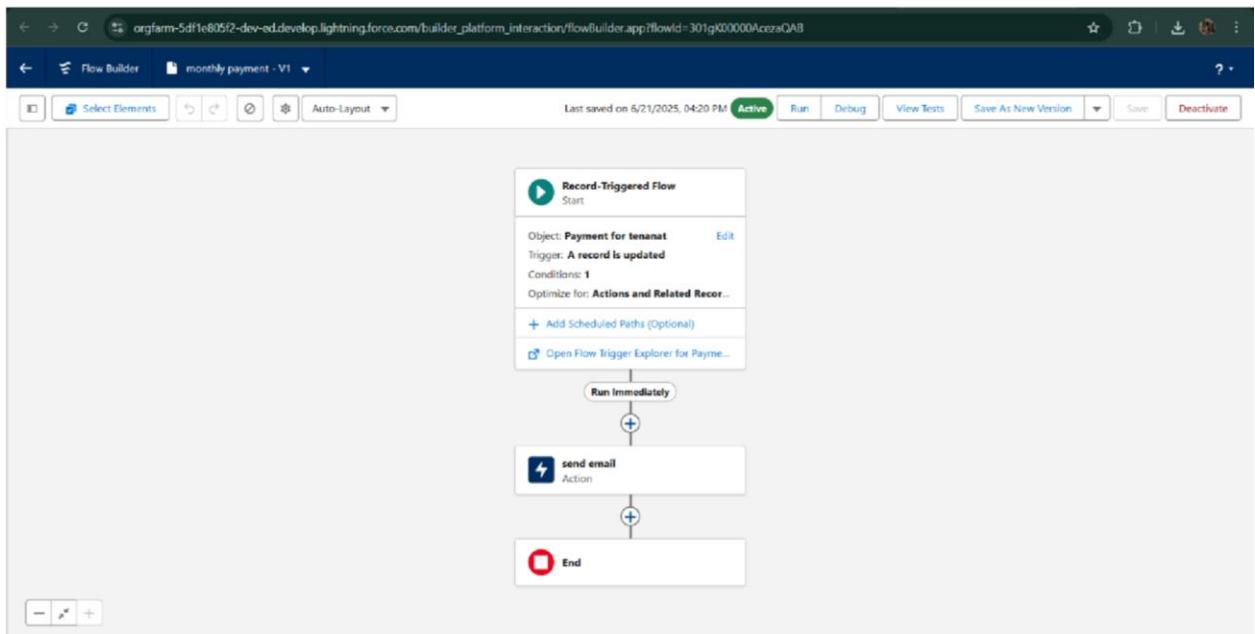
- Leave approved



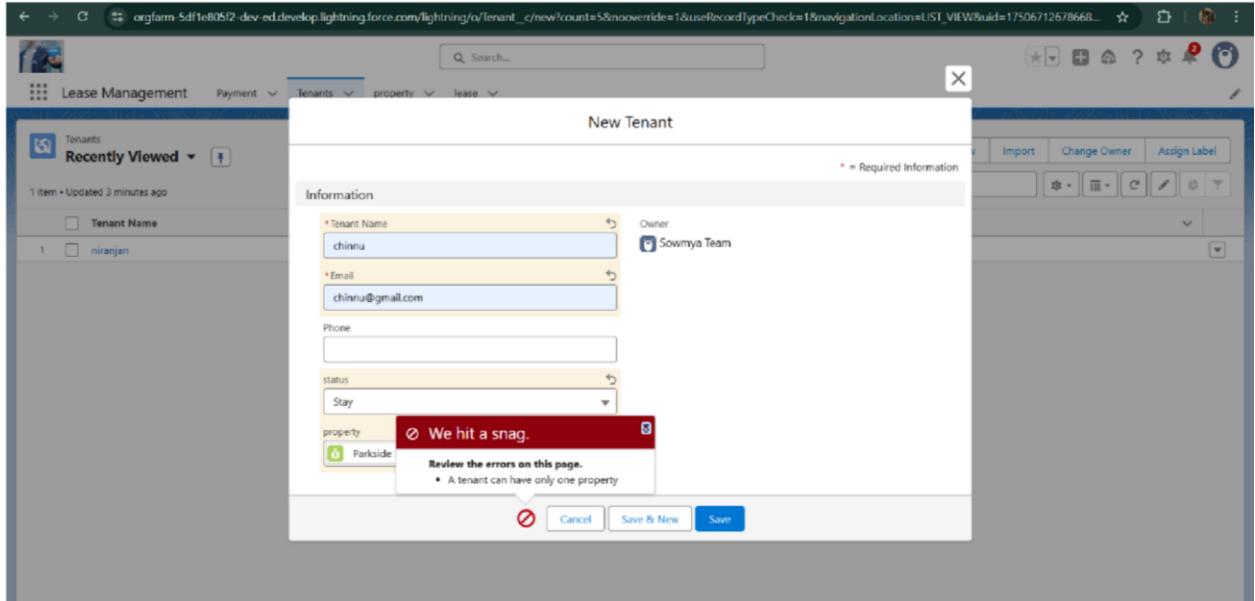
- Leave rejected



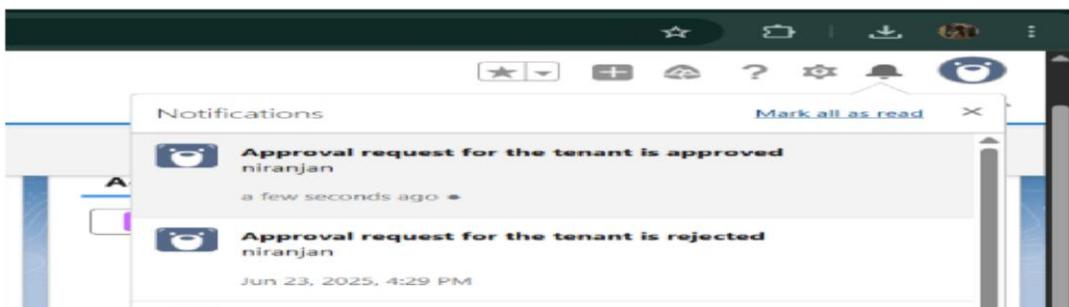
- Flow runs



- 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)) { newTenantaddError('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});
        }
    }
}

```

}

}