

Lease Management

College Name: Tiruppur kumaran college for women

College Code: bru3j

TEAM ID: NM2025TMID28432

TEAM MEMBERS:

Team LeaderName: GAAYATHRI.D

Email: gaayathridurai06@gmail.com

Team Member1: MALAVIKA.B.R

Email: bchander789@gmail.com

Team Member2: JANANI.S

Email: jjananipandiyan@gmail.com

1.INTRODUCTION

1.1 Project Overview

The Lease Management System is a Salesforce-based application that simplifies real estate leasing operations by managing tenants, lease agreements, payments, and communications, while leveraging automation tools like flows, approval processes, and email alerts.



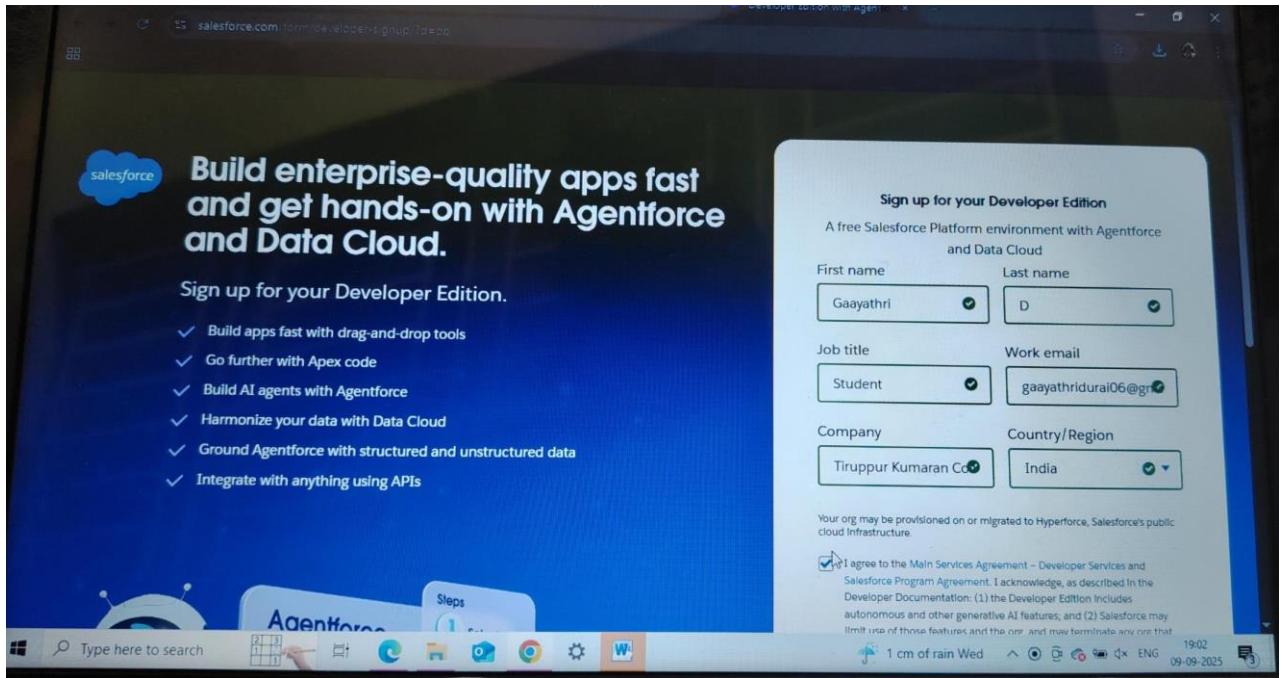
1.2 Purpose

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

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000z2TfI/Details/view

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 property'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' tab is selected. The main content area displays the 'property' object's details. The 'Description' section contains fields for API Name ('property_c'), Custom status ('Custom'), Singular Label ('property'), and Plural Label ('properties'). The 'Enable Reports' section includes checkboxes for Track Activities, Track Field History, and Deployment Status ('Deployed'). Help Settings and Standard salesforce.com Help Window are also listed.

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTbN/Details/view

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 Tenant'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' tab is selected. The main content area displays the 'Tenant' object's details. The 'Description' section contains fields for API Name ('Tenant_c'), Custom status ('Custom'), Singular Label ('Tenant'), and Plural Label ('Tenants'). The 'Enable Reports' section includes checkboxes for Track Activities, Track Field History, and Deployment Status ('Deployed'). Help Settings and Standard salesforce.com Help Window are also listed.

SETUP > OBJECT MANAGER

lease

Details

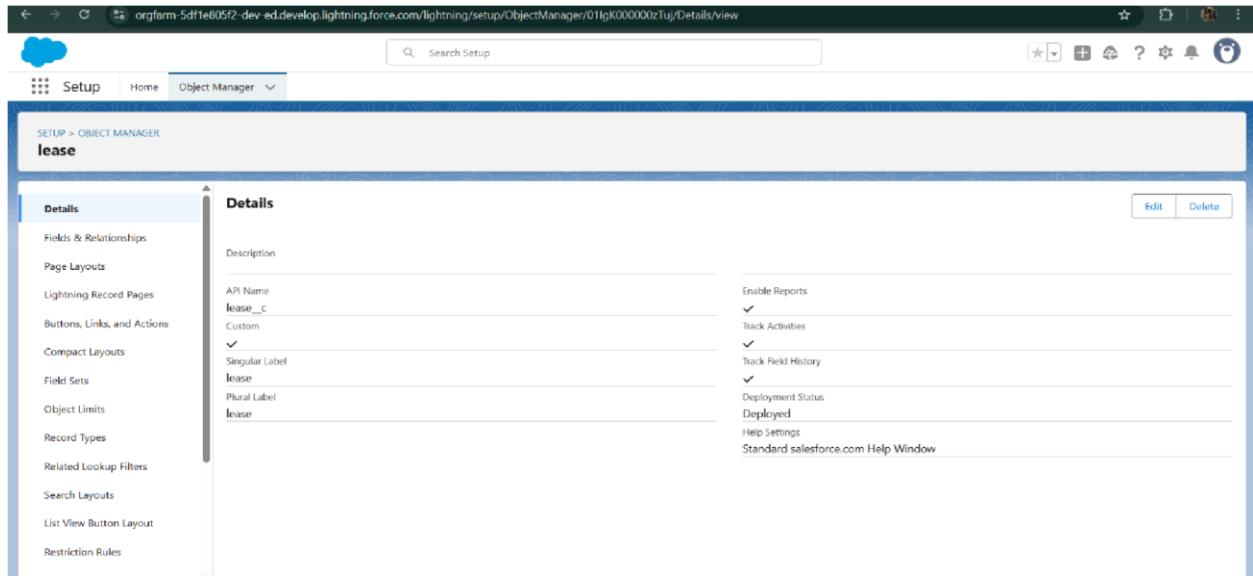
Description

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



SETUP > OBJECT MANAGER

Payment for tenant

Details

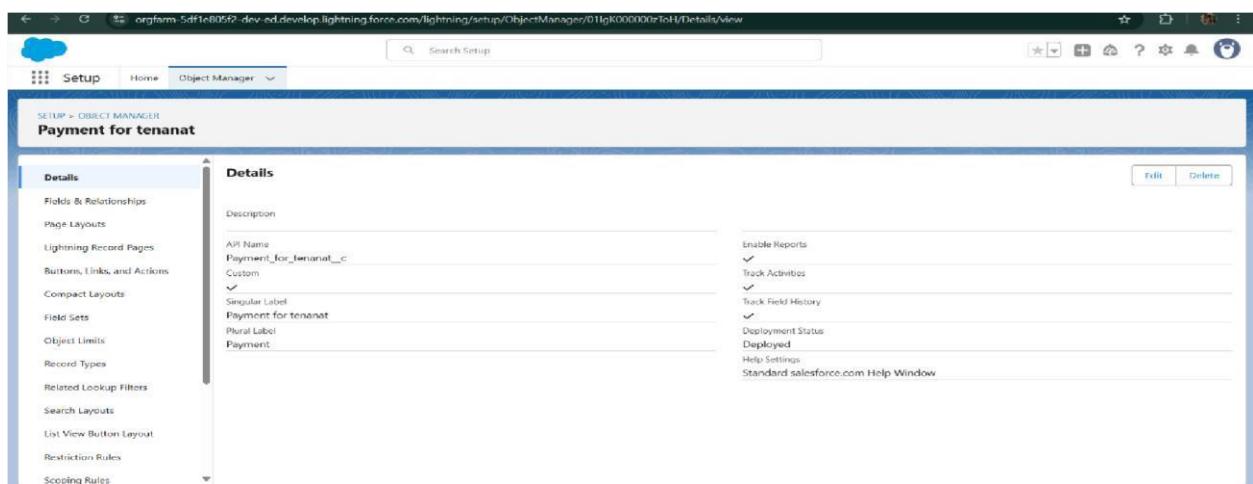
Description

API Name: Payment_for_tenant_c
Custom: ✓
Singular Label: Payment for tenant
Plural Label: Payment

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

Fields & Relationships
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



- 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)	✓	
sfqI	sfqI__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

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

- Developed Lightning App with relevant tabs

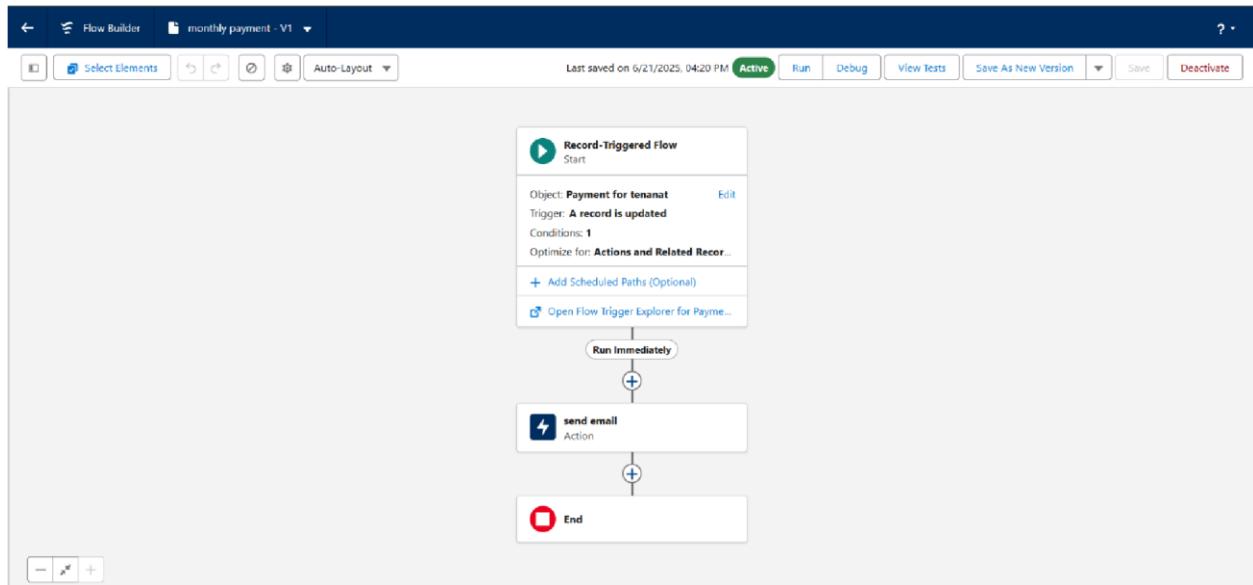
The screenshot shows the 'App Details & Branding' tab in the Lightning App Builder. On the left, a sidebar lists 'App Settings' sections: App Options, Utility Items (Desktop Only), Navigation Items, and User Profiles. The main area contains fields for 'App Name' (Lease Management), 'Developer Name' (Lease_Management), and a 'Description' box (Application to efficiently handle the processes related to leasing real estate properties.). It also includes an 'Image' section with a preview thumbnail, a color picker for 'Primary Color Hex Value' (#0070D2), and an 'Org Theme Options' checkbox. Below this is an 'App Launcher Preview' showing a card with the app's name and description.

The screenshot shows the 'Navigation Items' tab in the Lightning App Builder. The sidebar shows 'App Settings' sections: App Details & Branding, App Options, Utility Items (Desktop Only), and 'Navigation Items' (which is selected). Under 'Navigation Items', there are two columns: 'Available Items' (Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests) and 'Selected Items' (Payment, Tenants, property, lease). Navigation arrows between the columns allow items to be moved.

The screenshot shows the 'User Profiles' section within the 'App Settings' tab of the Lightning App Builder. On the left, there's a sidebar with tabs: 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', and 'Navigation Items'. The 'User Profiles' tab is selected and highlighted in blue. The main area is titled 'User Profiles' and contains the instruction: 'Choose the user profiles that can access this app.' Below this is a list of 'Available Profiles' which includes: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, and Customer Community Login User. To the right, under 'Selected Profiles', 'System Administrator' is listed. There are arrows for moving profiles between the two lists.

The screenshot shows the 'Payment' section of the 'Lease Management' tab. At the top, there are navigation links: 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. Below this is a search bar with the placeholder 'Search...'. The main area is titled 'Recently Viewed' and shows a list of 5 items, all updated a few seconds ago. The list includes: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, and 5. Lahari. Each item has a checkbox next to it. To the right of the list are buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'. Below these buttons is another search bar with the placeholder 'Search this list...' and various filter and sort icons.

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

The screenshot shows the Salesforce Setup > Object Manager page for the 'lease' object. The 'Validation Rule Edit' screen is open, displaying the following information:

- Validation Rule Edit** (Title)
- Rule Name:** lease_end_date
- Active:** checked
- Description:** (empty)
- Error Condition Formula:**

```
Example: Discount_Percent__c>0.30 More Examples...
Display an error if Discount is more than 30%.
If this formula expression is true, display the text defined in the Error Message area.
```
- Functions:**
 - Insert Field
 - Insert Operator
 - ABS
 - ACOS
 - ADDMONTHS
 - AND
 - ASCII
 - ASIN
- Quick Tips:**
 - Operators & Functions

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'lease Validation Rule'. The validation rule details are as follows:

- Validation Rule Detail**
- 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

The left sidebar lists various configuration options for the lease object, such as Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the Lease Management application. A modal window titled 'New Tenant' is open, displaying an error message: 'We hit a snag.' with the sub-instruction 'Review the errors on this page.' The error details are: 'A tenant can have only one property'. The form fields shown are Tenant Name (chinnu), Email (chinnu@gmail.com), and property (Parkside).

- 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

Search Setup

Home Object Manager

Setup

Q_email template

SETUP Classic Email Templates

Leave approved

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unified Public Classic Email Templates
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
	Edit Delete Close

Email Template

Send Test and Verify Merge Fields

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 confirmed. You can leave now.

```

The screenshot shows the Salesforce Setup interface. In the top navigation bar, there are links for 'Setup', 'Home', and 'Object Manager'. A search bar labeled 'Search Setup' is located at the top right. The main content area has a title 'SETUP Classic Email Templates'. On the left sidebar under 'Email', 'Classic Email Templates' is selected. The main panel displays a 'Text Email Template' titled 'tenant leaving'. The 'Email Template Detail' section shows the following information:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
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

Buttons for 'Edit', 'Delete', and 'Clone' are available at the top right of the detail section. Below the detail section is a preview area titled 'Email Template' with a 'Send Test and Verify Merge Fields' button. The preview 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, similar to the previous one but with a different email template. The main content area has a title 'SETUP Classic Email Templates'. On the left sidebar under 'Email', 'Classic Email Templates' is selected. The main panel displays a 'Text Email Template' titled 'Leave rejected'. The 'Email Template Detail' section shows the following information:

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 [Changed]
Description	
Created By	Sowmya Team, 6/20/2025, 1:11 AM
Modified By	Sowmya Team, 6/20/2025, 1:11 AM

Buttons for 'Edit', 'Delete', and 'Clone' are available at the top right of the detail section. Below the detail section is a preview area titled 'Email Template' with a 'Send Test and Verify Merge Fields' button. The preview shows the following content:

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

- Approval Process creation

For Tenant Leaving:

Process Definition Detail

Process Name	TenantApproval	Active
Unique Name	TenantApproval	Next Automated Approver Determined By
Description		
Entry Criteria	tenant:r status EQUALS Stay	
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template		
Initial Submitters	Tenant Owner	Modified By
Created By	Sowmya Team, 6/23/2025, 3:41 AM	Modified On

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:Sowmya Team	Final Rejection

For Check for Vacant:

Process Definition Detail

Process Name	check for vacant	Active
Unique Name	check_for_vacant	Next Automated Approver Determined By
Description		
Entry Criteria	Tenant:r status EQUALS Leaving	
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template	Leave approved	
Initial Submitters	Tenant Owner	Modified By
Created By	Sowmya Team, 6/20/2025, 3:18 AM	Modified On

Initial Submission Actions

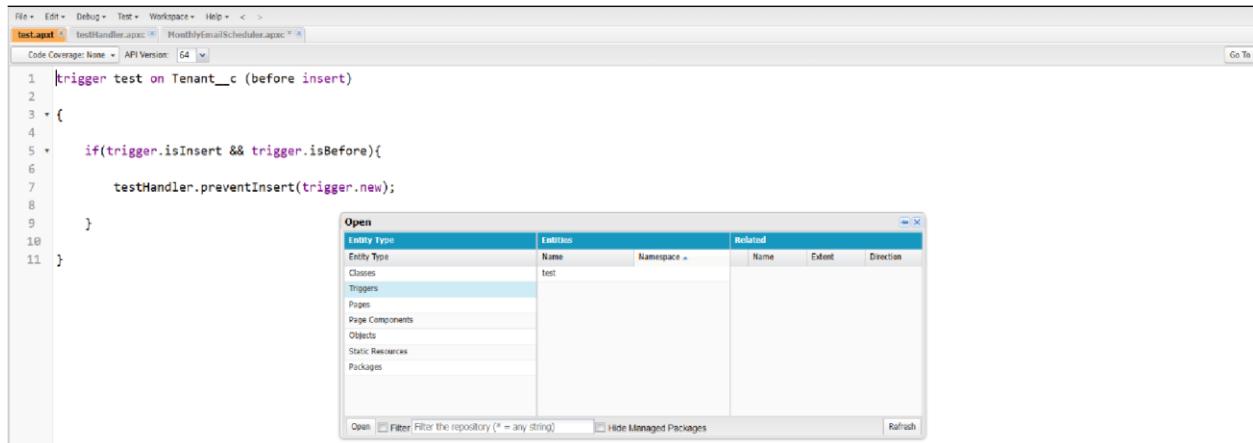
Action Type	Description
Record Lock	Lock the record from being edited
Edit Remove	Please approve my leave

Approval Steps

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

- Apex Trigger

Create an Apex Trigger

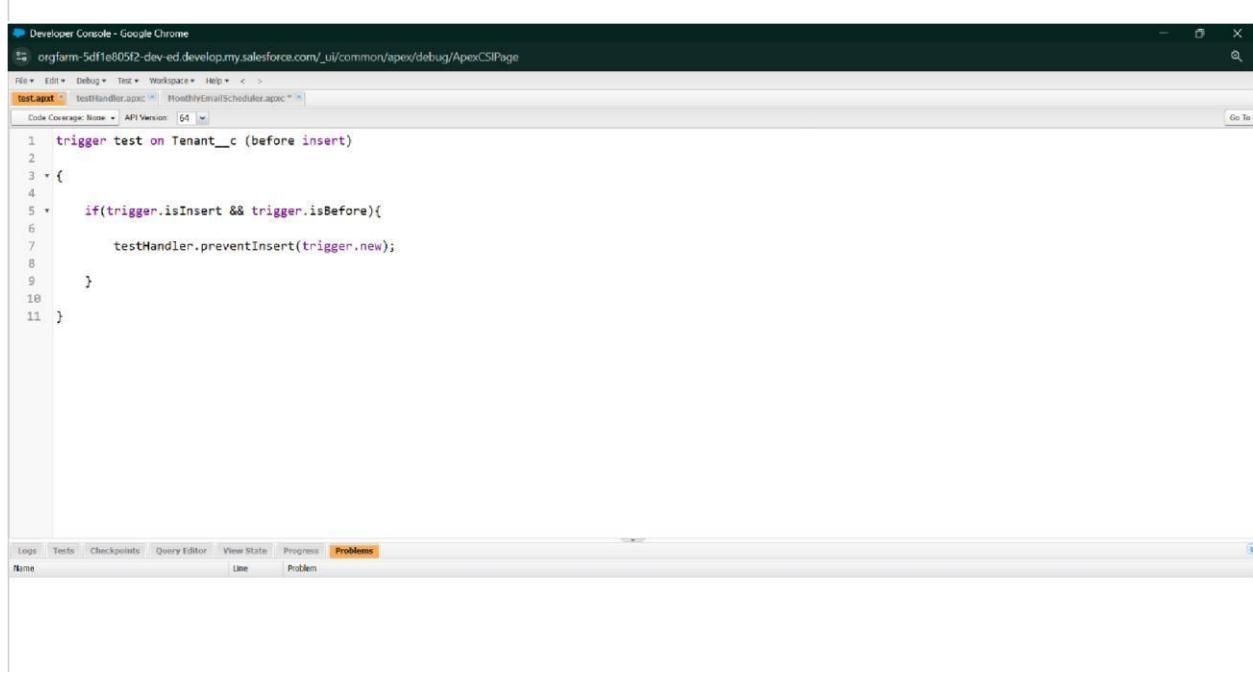


The screenshot shows the Salesforce IDE interface. At the top, there are tabs for 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below the tabs, the current file is 'test.apex' with the title 'testHandler.apex'. The code editor contains the following Apex trigger:

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

A modal window titled 'Open' is displayed, listing entities by type: Classes, Triggers, Pages, Page Components, Objects, Static Resources, and Packages. The 'Triggers' item is selected. A table shows one entry: 'test' under 'Name' and 'Namespace'.

Below the code editor, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is selected, showing a blank list.



The screenshot shows the Salesforce Developer Console in a web browser window. The URL is 'orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage'. The page title is 'Developer Console - Google Chrome'. The interface is identical to the IDE, showing the 'test.apex' file with the same trigger code and a blank 'Problems' list.

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 < >
testHandler.apc MonthlyEmailScheduler.apc
Code Coverage: None API Version: 64 Go To
1 * public class testHandler {
2
3 *   public static void preventInsert(List<Tenant__c> newList) {
4
5     Set<Id> existingPropertyIds = new Set<Id>();
6
7     for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9       existingPropertyIds.add(existingTenant.Id);
10
11     }
12
13     for (Tenant__c newTenant : newList) {
14
15       if (newTenant.Property__c != null) {
16
17         newTenantaddError('A tenant can have only one property');
18
19       }
20
21     }
22
23   }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Entity Type

Entry Type	Name	Namespace	Related
Classes	testHandler	MonthlyEmailScheduler	↪ test ApexTrigger ↪ property CustomField ↪ Tenant__c SObject ↪ Tenant__c SObject
Triggers			
Pages			
Components			
Objects			
Static Resources			
Package			

Open Filter Hide Managed Packages Refresh

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

```

File Edit Debug Test Workspace Help < >
testHandler.apc MonthlyEmailScheduler.apc
Code Coverage: None API Version: 64 Go To
1 * public class testHandler {
2
3 *   public static void preventInsert(List<Tenant__c> newList) {
4
5     Set<Id> existingPropertyIds = new Set<Id>();
6
7     for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9       existingPropertyIds.add(existingTenant.Property__c);
10
11     }
12
13     for (Tenant__c newTenant : newList) {
14
15       if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
16
17         newTenantaddError('A tenant can have only one property');
18
19       }
20
21     }
22
23   }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

- FLOWS

Flow Builder - monthly payment - V1

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

Configure Start

Field: check for payment Operator: Equals Value: Paid

When to Run the Flow for Updated Records:

- Every time a record is updated and meets the condition requirements
- Only when a record is updated to meet the condition requirements

Optimize Flow

Optimize the Flow for:

- Fast Field Updates**: Update fields on the record that triggers the flow to run. This high-performance flow runs **before the record is saved** to the database.
- Actions and Related Records**: Update any record and perform actions, like send an email. This more flexible flow runs **after the record is saved** to the database.

Is this flow making an external callout or connecting to an external system? An asynchronous path is required for flows that involve external systems.

Add Asynchronous Path

Configure Start

Select Object: Payment for tenantat

Configure Trigger

Trigger the Flow When:

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

Set Entry Conditions

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

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

Condition Requirements: All Conditions Are Met (AND)

- Schedule class:
Create an Apex Class

Developer Console - Google Chrome
orgfarm-5d1f1e005f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
testHandler.apex ▾ MonthlyEmailScheduler.apex ▾ MonthlyEmailScheduler.apex

Code Coverage: None API Version: 64 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             sendMonthlyEmails();  
9         }  
10    }  
11 }  
12  
13  
14  
15  
16 * public static void sendMonthlyEmail  
17  
18     List<Tenant__c> tenants = [SEL  
19  
20         for (Tenant__c tenant : tenants)  
21  
22             String recipientEmail = tenant.Email__c;  
23
```

Open Entity Type Utilities Related

Entity Type	Utilities	Related
Entity type	Name Namespace +	Name Extent Direction
Classes	testHandler	Email CronTrigger Referenced...
Triggers	MonthlyEmailScheduler	CustomField... References
Pages		← Tenant__c SObject References
Page Components		← Tenant__c SObject References
Objects		
Static Resources		
Packages		

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Developer Console - Google Chrome

orgfarm-5df1e0805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

50% Reset

```
1 * @ISODLL11000 methEmailScheduler 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 * }
```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has sections for Email, Custom Code, Environments, and a search bar. The main area displays the Apex Class MonthlyEmailScheduler. The Apex Class Detail section shows the class name, namespace prefix (Sowmya_Team), and creation date (6/23/2025, 2:46 AM). The status is Active, code coverage is 0% (0/15), and last modified by Sowmya Team on 6/23/2025, 2:47 AM. The Class Body tab is selected, showing the following Apex 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     public static void sendMonthlyEmails() {
9         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
10        for (Tenant__c tenant : tenants) {
11            ...
12        }
13    }
14}
15
16
17
18
19
20
```

The screenshot shows the Salesforce Lease Management Details page for tenant Aswini. The left sidebar has sections for Lease Management, Payment, Tenants, property, and lease. The main area shows the tenant details: Name (Aswini), Email (aswinimaraadi15@gmail.com), Phone ((905) 223-5567), Status (Leaving), and Property (Imran). The owner is Sowmya Team. The activity sidebar shows no upcoming or overdue activities. The bottom right corner has a modal for "Submit for Approval".

Screenshot of a Salesforce Lightning interface showing a tenant submission for approval.

The URL is https://orgfarm-5df1e805f2-dev-ed.lightning.force.com/lightning/r/Tenant__c/a01gK00000BAhdGQAT/view

Key elements:

- Tenant:** Aswini
- Status:** Tenant was submitted for approval.
- Details:**
 - Tenant Name: Aswini
 - Email: aswiniamaraadi15@gmail.com
 - Phone: (905) 223-5567
 - Status: Leaving
 - Property: Imran
- Activity:** No activities to show.

System status bar at the bottom shows: 30°C Cloudy, ENG IN, 12:46, 27-06-2025.

Screenshot of a Salesforce Lightning interface showing a process instance step for tenant approval.

The URL is <https://orgfarm-5df1e805f2-dev-ed.lightning.force.com/lightning/r/ProcessInstanceStep/04hgK00000CYNhQAO/view>

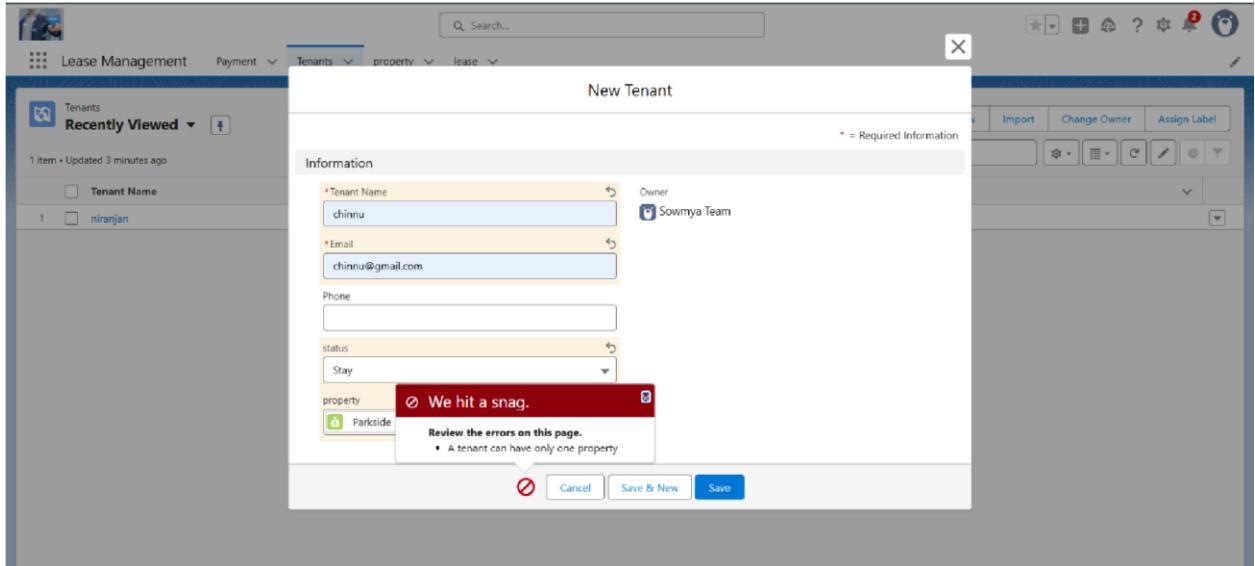
Key elements:

- Process Instance Step:** Tenant Approval (Approved)
- Submitter:** Sowmya Team
- Date Submitted:** Jun 27, 2025
- Actual Approver:** Sowmya Team
- Assigned To:** Sowmya Team
- Details:**
 - Approval Details:
 - Tenant Name: Aswini
 - Owner: Sowmya Team
 - Property: Imran
- Notifications:** A sidebar showing five notifications related to tenant approvals.

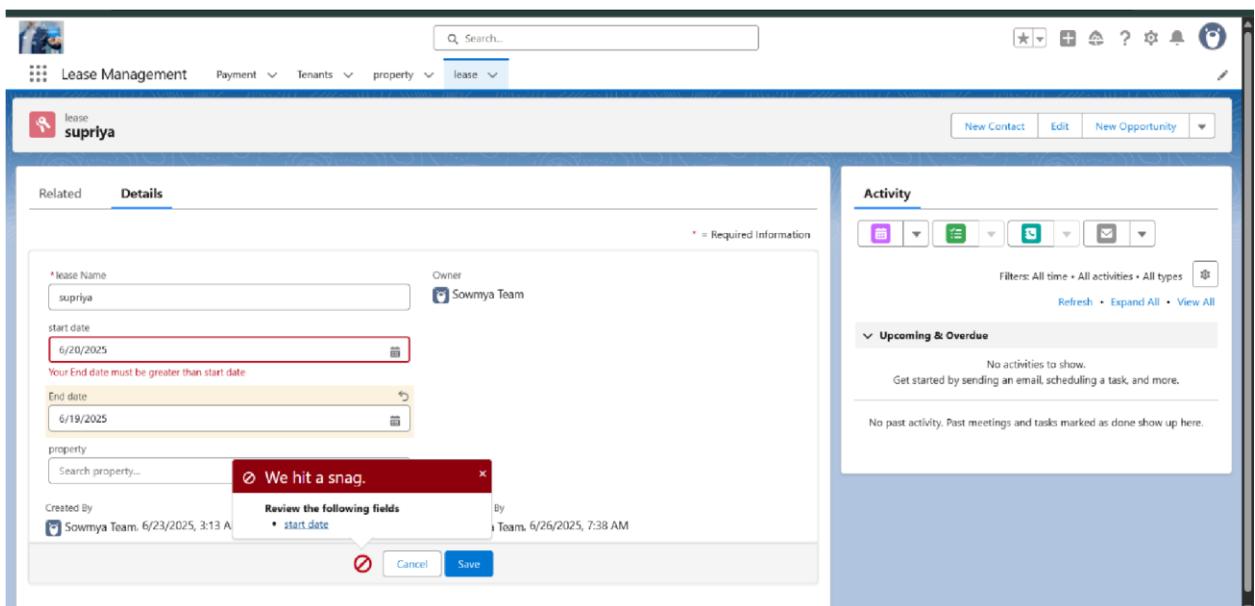
FUNCTIONAL AND PERFORMANCE TESTING

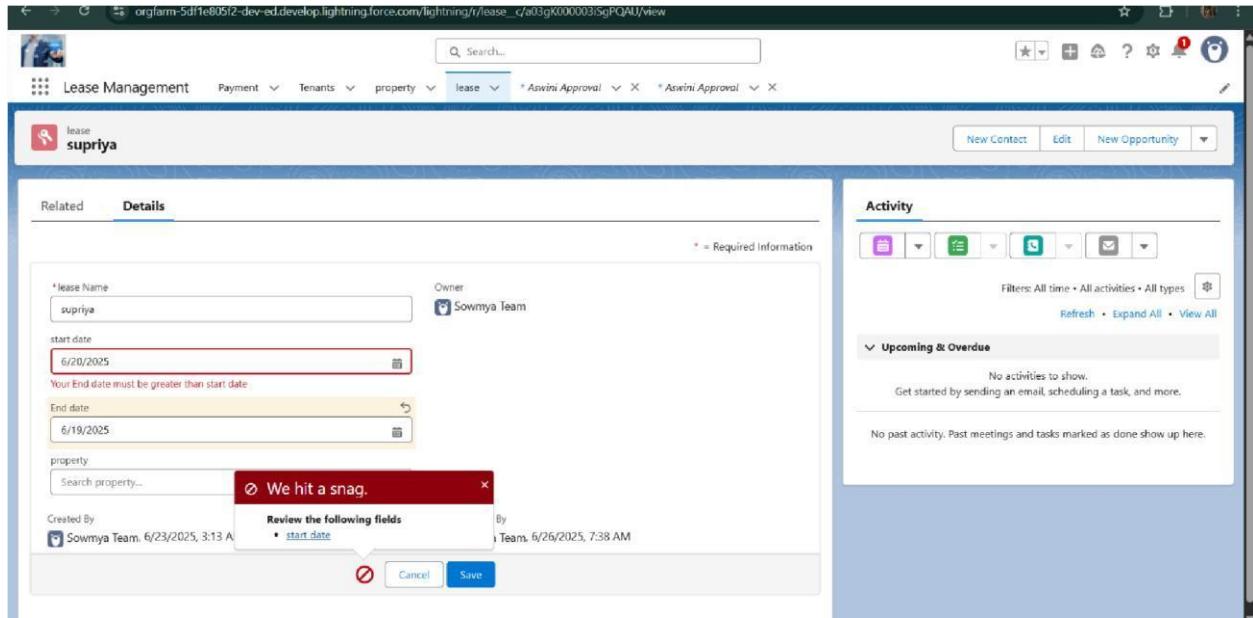
Performance Testing

- Trigger validation by entering duplicate tenant-property records

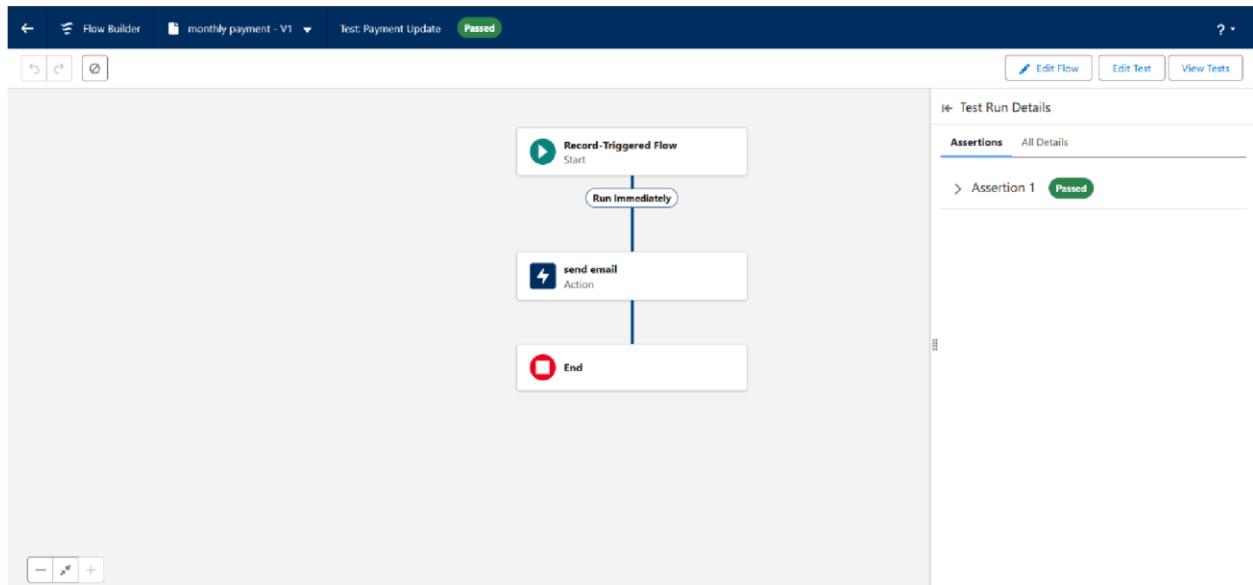


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the Microsoft Dynamics 365 interface for 'Lease Management'. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', 'lease', and 'niranjan Approval'. A search bar at the top right contains the placeholder 'Search...'. On the left, a sidebar titled 'Tenant niranjan' shows 'Related' and 'Details' tabs. The 'Details' tab displays tenant information: 'Tenant Name' (niranjan), 'Email' (niranjan1506@gmail.com), 'Phone' (empty), 'status' (Stay), and 'property' (Parkside Lofts). It also shows 'Created By' (Sowmya Team) and 'Last Modified By' (Sowmya Team). Below the form are 'Cancel' and 'Save' buttons. To the right, a 'Notifications' sidebar lists 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), and '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).

This screenshot continues from the previous one, showing the 'Approval History' section for 'niranjan'. The table lists six approval steps:

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, 5:59 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 the table are 'View All' and 'New' buttons. Further down, the 'Payment' section shows two entries: 'Jack' and 'Rahul', with a 'View All' button.

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

Action	Label	Tab Style	Description
Edit Del	lease	Keys	
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.

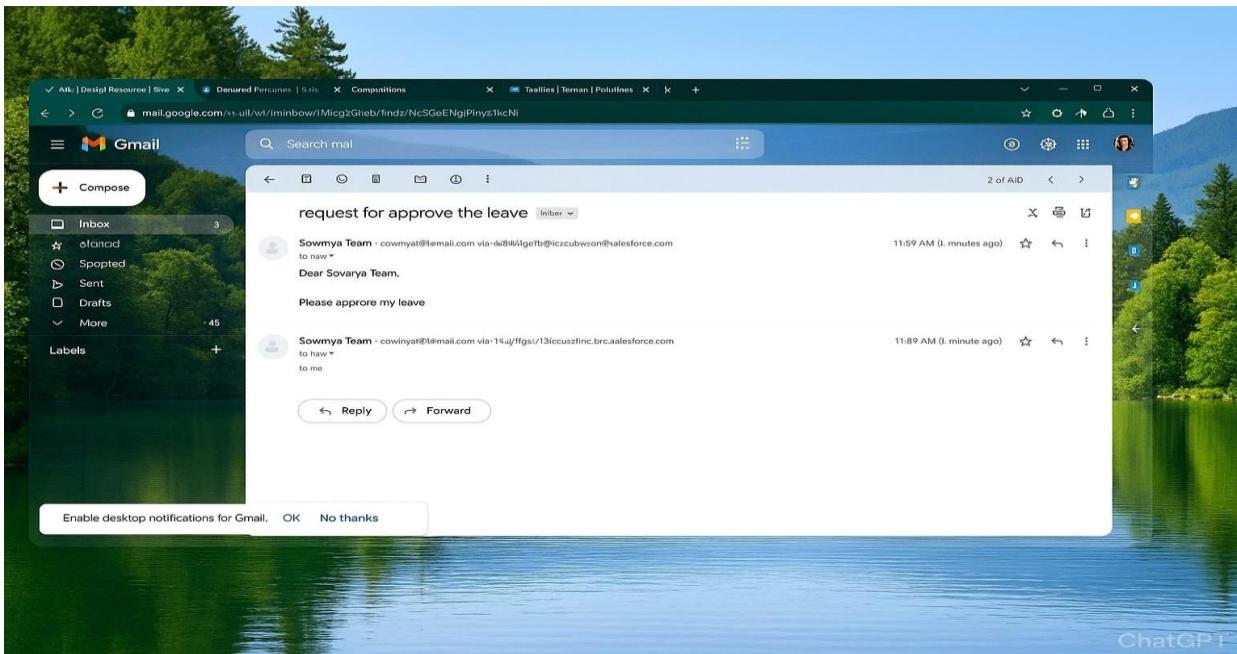
- Email alerts

Approval History

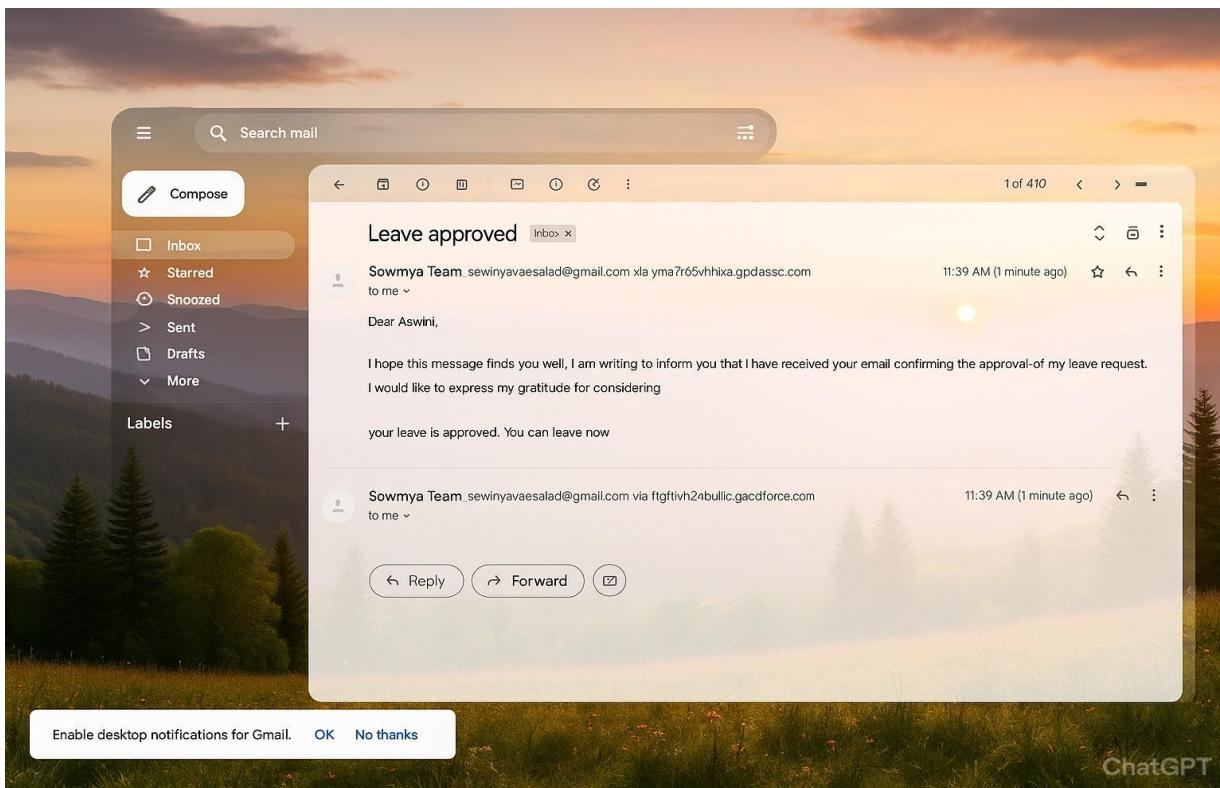
8 items • Sorted by Is Pending • Updated a few seconds ago

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

Search mail

Leave rejected Inbox ×

Sowmya Team sowmyassamelao@gmail.com via.kuvi?k3g@ahcchcu.3k-5pcduac.saeforce.com
to me 11:45 AM (10 minute

Dear Arvind,

I hope this email finds you well. Your contract has not ended. So we can't approve your leave
your leave was rejected

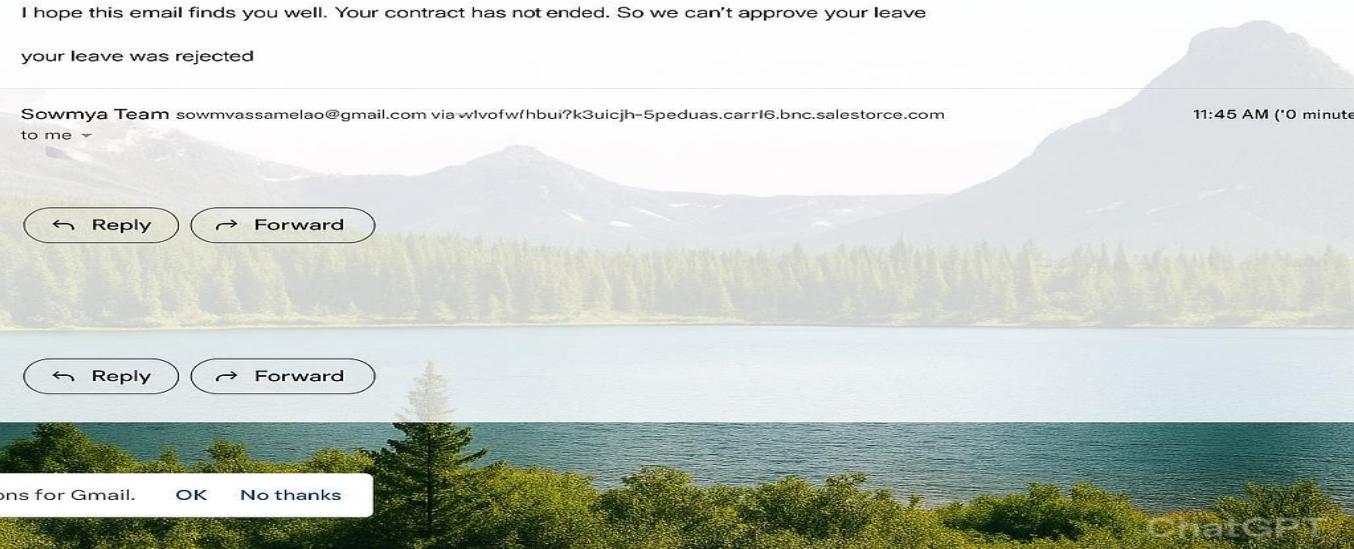
Sowmya Team sowmyassamelao@gmail.com via.wlvofwfhbu?k3uicjh-5peduas.carri6.bnc.salestorce.com
to me 11:45 AM ('0 minute

Reply Forward

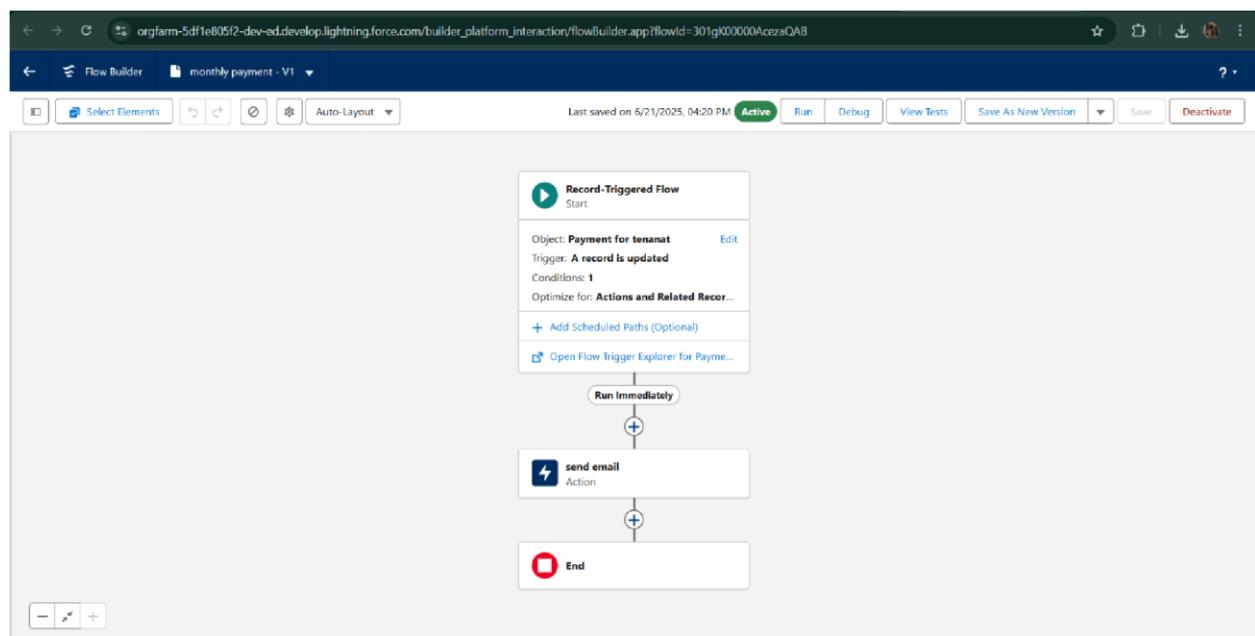
Reply Forward

on for Gmail. OK No thanks

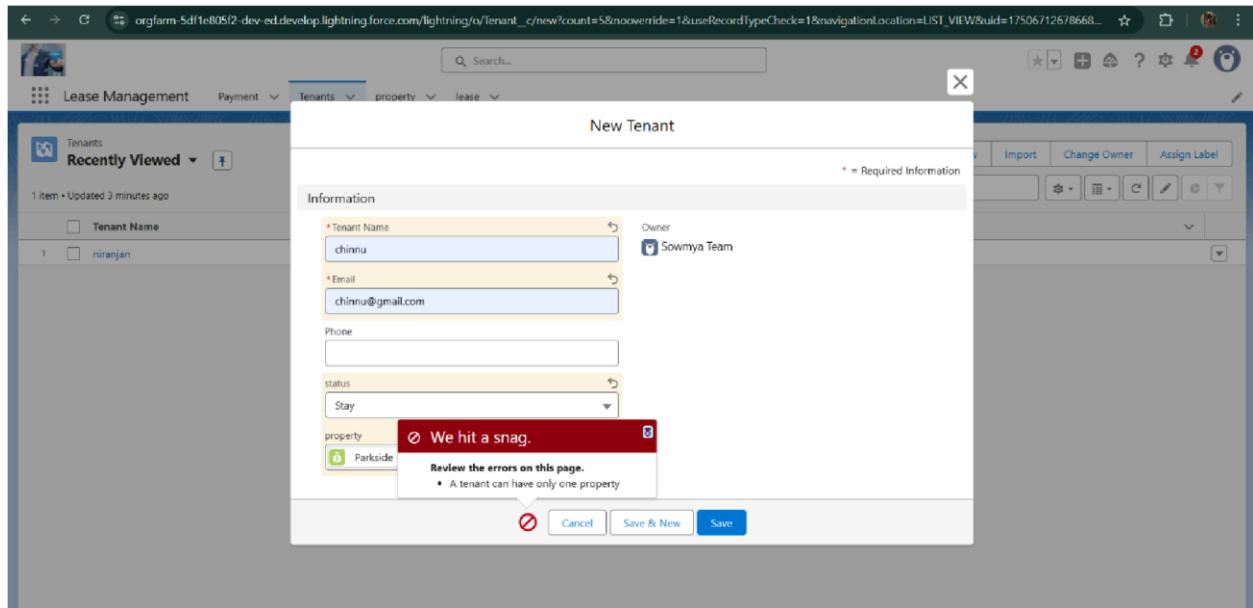
ChatGPT



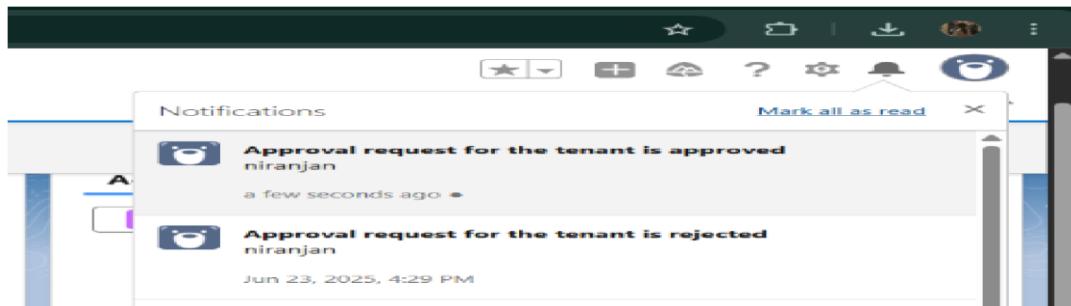
- Flow runs



- Trigger error messages



- Approval process notifications



ADVANTAGES & DISADVANTAGES

CONCLUSION

The Lease Management System enhances leasing operations by leveraging a structured Salesforce application with automation, ensuring better efficiency, seamless communication, and accurate data management for both administrators 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});

    }

}
```