

PROJECT TITLE

College Name: **pollachi college of arts and science**

College Code: **bruaf**

TEAM ID: **NM2025TMID21307**

TEAM MEMBERS:

Team LeaderName: **SRIDHARAN.A**

Email: 23nmcssridharan@gmail.com

Team Member: **SHANKAR.K**

Email: 23nmcsshankar@gmail.com

Team Member: **NAVANEETHAKRISHNAN.J**

Email: 23nmcsnavaneethakrishnan@gmail.com

Team Member: **ANANDAN.M**

Email: 23nmcsanandhan@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.



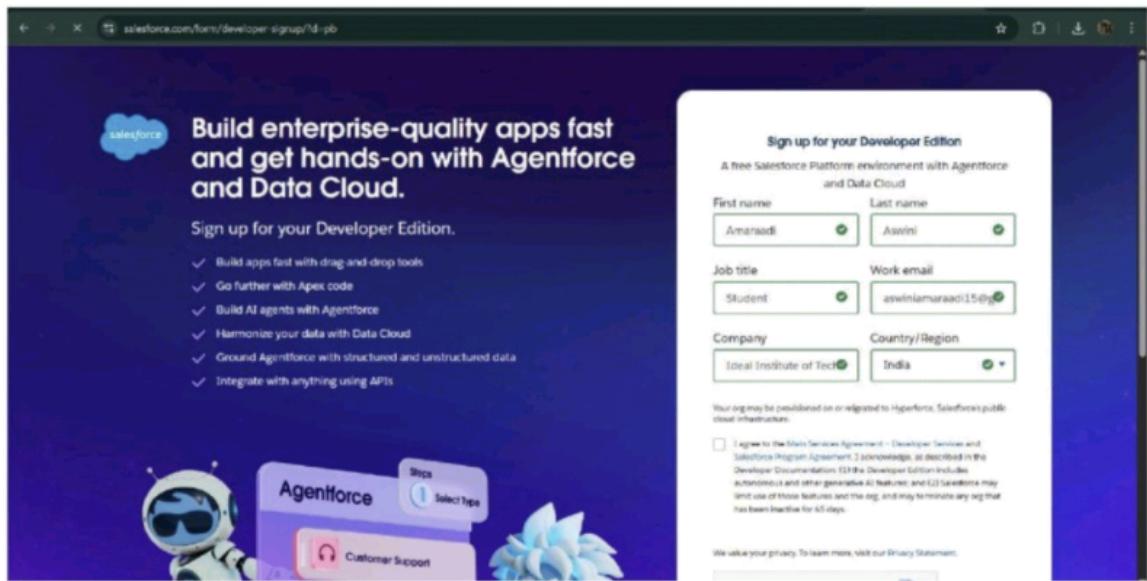
1. Purpose

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/sd=pb>



- Created objects: Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Object Manager interface for the "property" object. 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 List API Names, Search Layouts, List View Button Layout, Worklist Rules, and Standard Values.

The main panel displays the "Details" tab for the "property" object. It includes sections for Description, API Name (property__c), Custom (checked), Singular Label (property), Plural Label (properties), and a "Share" section with checkboxes for Track Activities, Track Read History, Deployment Status (Deployed), and Help Topics (Standard salesforce.com Help Window).

Setup > Object Manager

Tenant

Details

Description

API Name: Tenant_c
Custom
✓
Singular Label: Tenant
Plural Label: Tenants

Editable Reports:
✓ Stock Activities
✓ Stock 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

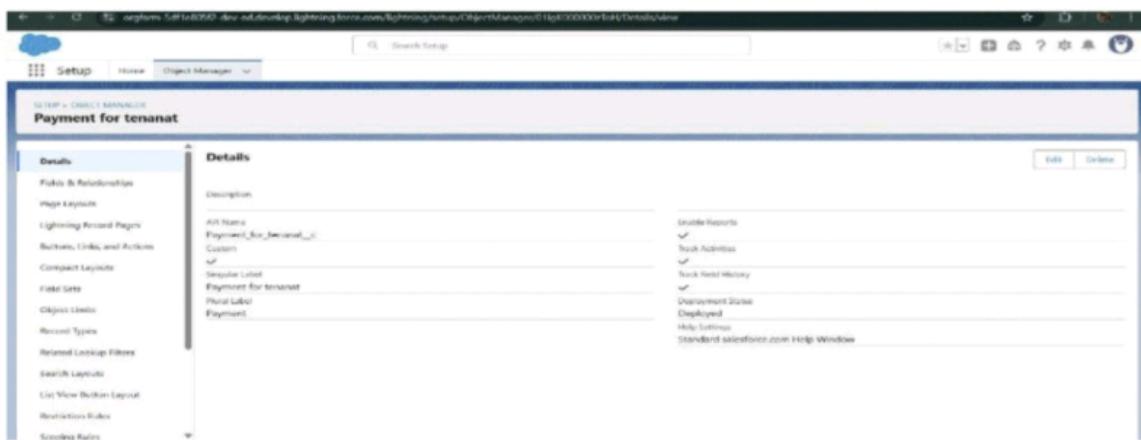
Details

Description

API Name: lease_c
Custom
✓
Singular Label: lease
Plural Label: lease

Editable Reports:
✓ Stock Activities
✓ Stock 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

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts:	Address:	Address_c	Long Text Area(32768)		
Lightning Record Pages:	Created By:	CreatedById	Lookup(User)		
Buttons, Links, and Actions:	Last Modified By:	LastModifiedById	Lookup(User)		
Compact Layouts:	Name:	Name_c	Text(25)		
Field Sets:	Owner:	OwnerId	Lookup(User,Group)		
Object Limits:	property:	property_c	Lookup(property)		
Record Types:	property Name:	Name	Text(50)		
Related Lookup Filters:	slip:	slip_c	Text(10)		
Search Layouts:	Type:	Type_c	Picklist		
List View Button Layout:					
Restriction Rules:					
Scoping Rules:					

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenantat

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules

Fields & Relationships
7 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Number(18, 0)		
check for payment	check_for_payment_c	Boolean		
Created By	CreatedBy	LookupUser		
Last Modified By	LastModifiedBy	LookupUser		
Owner	OwnerId	LookupUser,Group		
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)		

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules Scoping Rules

Fields & Relationships
7 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	LookupUser		
End date	End_date_c	Date		
Last Modified By	LastModifiedBy	LookupUser		
lease Name	Name	Text(80)		
Owner	OwnerId	LookupUser,Group		
property	property_c	Lookup(property)		
start date	start_date_c	Date		

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedBy	Lookup(User)		
Lightning Record Pages	Email	Email_c	Email		
Buttons, Links, and Actions	Last Modified By	LastModifiedBy	Lookup(User)		
Compact Layouts	Owner	OwnerId	Lookup(OwnerGroup)		
Field Sets	Phone	Phone_c	Phone		
Object Limits	Status	Status_c	Picklist		
Record Types	Insert Name	Name	Text(80)		
Related Lookup Filters:					
Search Layouts					
List View Button Layout					
Restriction Rules					
Scoping Rules					

- Developed Lightning App with relevant tabs

App Details & Branding

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

App Details	App Branding
*App Name: <input type="text" value="Lease Management"/>	Image: <input type="file"/>
*Developer Name: <input type="text" value="Lease Management"/>	Primary Color Hex Value: <input type="color" value="#0070C0"/>
Description: Application to efficiently handle the processes related to leasing real estate properties.	Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme

App Launcher Preview

Lease Management
Application to efficiently handle the processes related to leasing real estate properties.

App Settings

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

Selected Items

Available Items	Selected Items
Account	Payment
Activation Targets	Tenants
Activations	Property
All Sites	Issue
Alternative Payment Methods	
Analytics	
App Launcher	
Appointment Categories	
Appointment Invitations	
Approval Requests	
Case	

App Settings

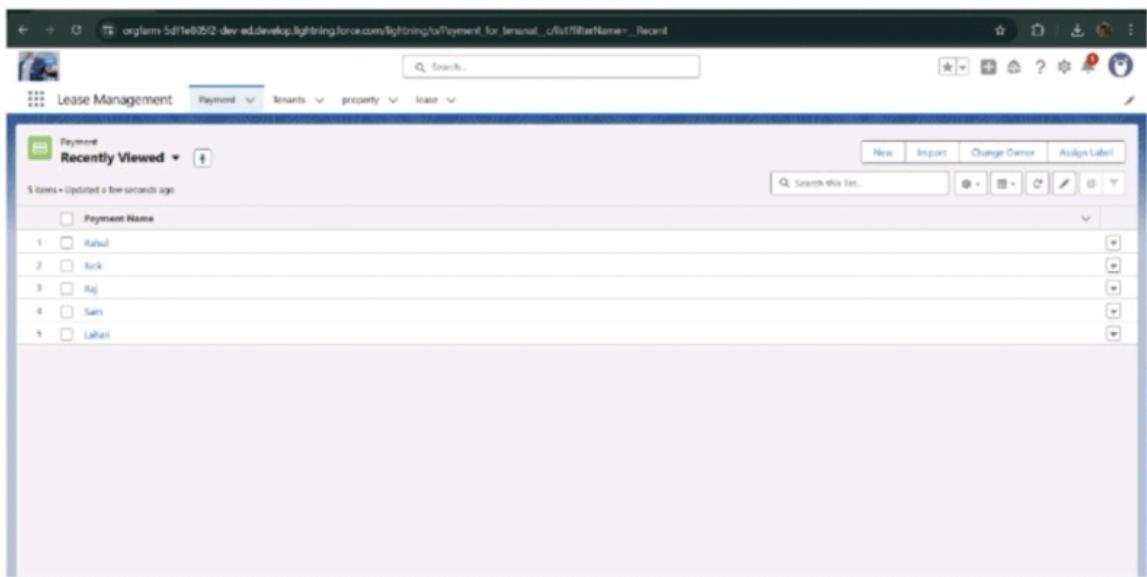
User Profiles

Choose the user profiles that can access this app.

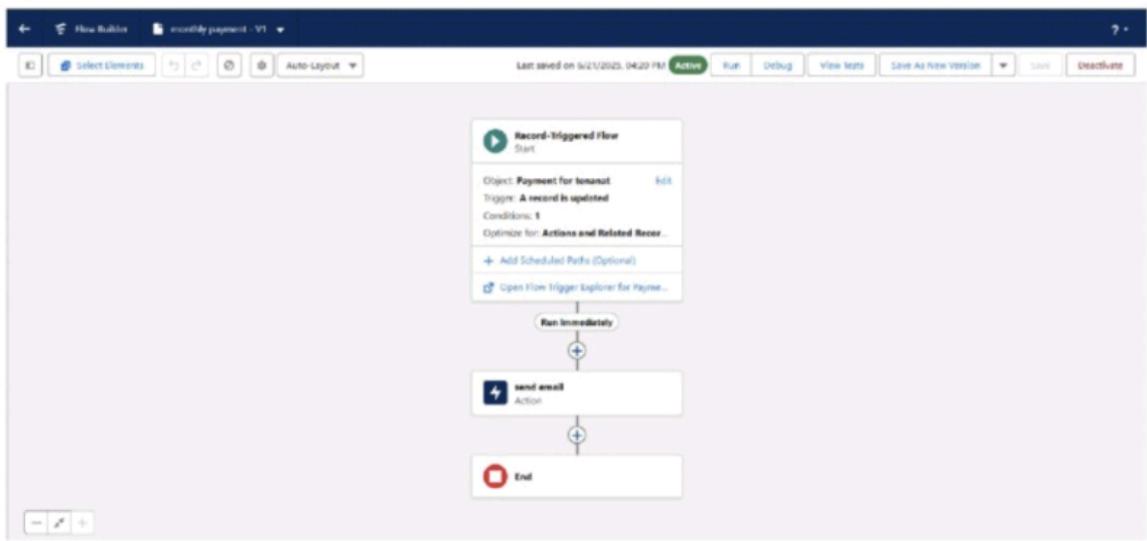
Available Profiles

Selected Profiles

Available Profiles	Selected Profiles
Analytics Cloud Integration User	System Administrator
Analytics Cloud Security User	
Anypoint Integration	
Authenticated Website	
Authenticated Website	
B2B Rendering Portal Buyer Profile	
Contract Manager	
Custom Marketing Profile	
Custom Sales Profile	
Custom Support Profile	
Customer Community Login User	



- 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 < 0.30)` More Examples...

If this formula expression is true, display the text defined in the Error Message area.

Insert Field: `End Date` Insert Operator: `<=` Insert Value: `Start Date`

Functions: All Function Categories

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Insert Selected Function: ABS(Argument) Returns the absolute value of a number, a number without its sign.

Help ID: 01n1T000000C9pQAS

Check Syntax

Save Save & New Cancel

Setup > Object Manager

lease

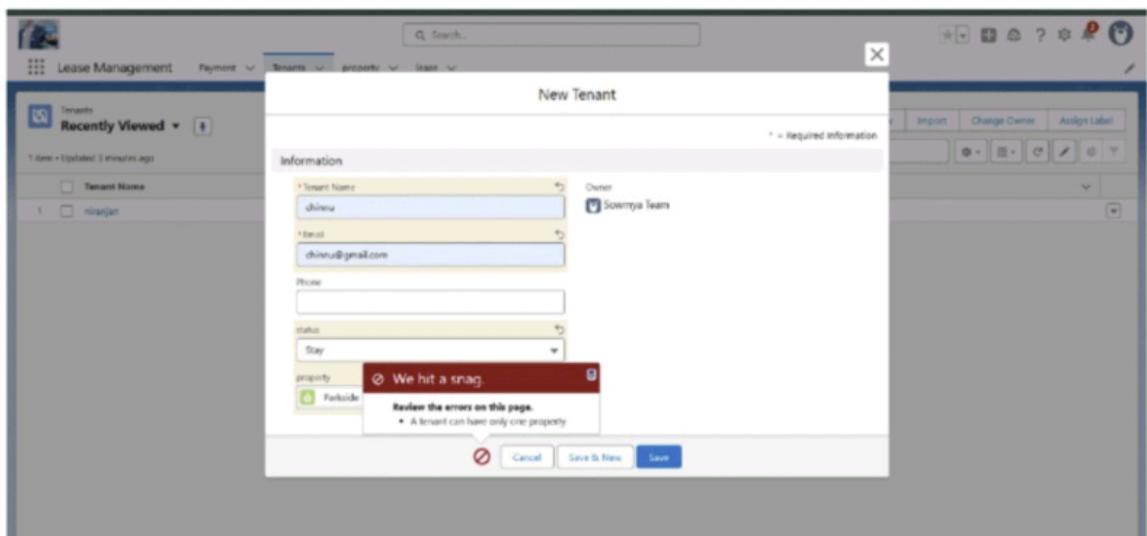
lease Validation Rule

Validation Rule Detail

Role Name:	lease_end_date	Action:	<input checked="" type="checkbox"/>
Error Condition Formula:	<code>End Date <= Start Date</code>	Error Location:	Start Date
Error Message:	Your End date must be greater than start date	Created By:	Soumya Datta 9/19/2015, 9:37 AM
Description:			
Created By:	Soumya Datta 9/19/2015, 9:37 AM		

Help for this Page

- 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+             sendEmail();
10+        }
11+
12+    }
13+
14+
15+ public static void sendEmail() {
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+
37+ }
38+
39+ 
```

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

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

- Email Template Name:** Leave approved
- Template Unique Name:** Leave_approved
- Encoding:** Unicode (UTF-8)
- Author:** Someone_2000 (Closed)
- Description:** Created By: Someone_2000, 6/19/2015, 1:00 AM
- Modified By:** Someone_2000, 6/19/2015, 1:00 AM

The preview pane shows the email content:

```

Subject: Leave approved
Plain Text Preview:
dear{From__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.

```

The screenshot shows the Salesforce Setup interface with the 'Email' tab selected. Under 'Classic Email Templates', a template named 'tenant leaving' is displayed. The template details are as follows:

- Email Template Name:** tenant leaving
- Template Unique Name:** tenant_leaving
- Encoding:** Unicode (UTF-8)
- Author:** Someone_2000 (Closed)
- Description:** Created By: Someone_2000, 6/19/2015, 1:00 AM
- Modified By:** Someone_2000, 6/19/2015, 1:00 AM

The preview pane shows the email content:

```

Subject: request for approve the leave
Plain Text Preview:
Dear {From__c.Creativity},  

Please approve my leave.

```

Setup Home Object Manager

Q_email template

SETUP Classic Email Templates

Leave rejected

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Salesforce Team (Shared)
Description	
Created By	Salesforce Team (Shared) 6/25/2025, 1:11 AM
Modified By	Salesforce Team (Shared) 6/25/2025, 1:11 AM

Available For Use

Last Used Date

Times Used

Edit Delete Close

Email Template

Send Text and Verify Merge Fields

Subject: Leave rejected

Please 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

Setup Home Object Manager

Q_email template

SETUP Classic Email Templates

Tenant Email

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Salesforce Team (Shared)
Description	
Created By	Salesforce Team (Shared) 6/25/2025, 1:12 AM
Modified By	Salesforce Team (Shared) 6/25/2025, 1:12 AM

Available For Use

Last Used Date

Times Used

Edit Delete Close

Email Template

Send Text and Verify Merge Fields

Subject: Urgent: Monthly Rent Payment Reminder

Please 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 message contains an automatic generated message and does not represent a personal communication. No liability is assumed for its contents.

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

- Page Header:** Search Setup, Help for This Page
- Left Navigation:** Setup, Home, Object Manager, Email (selected), Classic Email Templates, Lightning Email Templates.
- Section:** Test Email Template
- Title:** Classic Email Templates
- Email Template Detail:**
 - Email Template Name: tenant payment
 - Template Unique Name: tenant_payment
 - Encoding: Unicode (UTF-8)
 - Author: Sumeeta.Jebar (Changed)
 - Description: Created by Sumeeta.Jebar, 6/25/2025, 1:13 AM
 - Modified By: Sumeeta.Jebar, 6/25/2025, 1:13 AM
- Email Template Body:**
 - Subject: Confirmation of Successful Monthly Payment
 - Plain Text Preview: Dear {Recipient__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 following details:

- Page Header:** Search Setup, Help for This Page
- Left Navigation:** Setup, Home, Object Manager, Approval (selected), Approval Processes.
- Section:** Approval Processes
- Process Definition Detail:**
 - Process Name: TenantApproval
 - Unique Name: TenantApproval
 - Description: *Automatically review records flagged as Adminstrator ONLY*
 - Last Edited By: Sumeeta.Jebar, 6/25/2025, 1:11 AM
 - Action: Active
 - Next Automated Approver Determined By: *Allow Submitters to Recall Approval Requests*
- Approval Assignment Email Template:**
 - Initial Submitter: Tenant Owner
 - Created By: Sumeeta.Jebar, 6/25/2025, 1:11 AM
 - Modified By: Sumeeta.Jebar, 6/25/2025, 1:11 PM
- Initial Submission Actions:**
 - Action Type: Record Lock
 - Description: Lock the record from being edited
- Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Initial Actions	1	Step 1			User:Income Team	Final Rejection

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page selected. The process is titled 'check for vacant'. The 'Process Definition Detail' section includes fields for Process Name (check_for_vacant), Description (Tenant check for vacant), Entry Criteria (Tenant__c status = 100000000000000000), and Initial Submitter (Tenant Owner). The 'Initial Submission Actions' section contains a single entry: 'Record Lock' with a description 'Lock the record from being edited'.

● Apex Trigger

Create an Apex Trigger

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'test.apl' open. The trigger code is as follows:

```

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

```

A modal window titled 'Open' is displayed, showing a list of entities such as Entities, Triggers, Pages, and Components. The 'Triggers' item is selected in the list.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `https://orgname-5d1fe0c92-dev-ed-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCIPage`. The tab bar shows 'test.apex' is selected. The code editor contains the following Apex trigger code:

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

Create an Apex Handler class

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `https://orgname-5d1fe0c92-dev-ed-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCIPage`. The tab bar shows 'test.apex' is selected. The code editor contains the following Apex class code:

```
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.Id);
        }
        for (Tenant__c newTenant : newList) {
            if (newTenant.Property__c != null) {
                newList.addError('A');
            }
        }
    }
}
```

A dependency browser window is open over the code editor, showing the relationships between classes. The browser has three tabs: 'Entity Types', 'Editline', and 'Related'. The 'Entity Types' tab lists:

Entity Type	Entity	Relationship
Entity Sets	None	None
Classes	testHandler	None
Triggers	None	None
Pages	None	None
Page Components	None	None
Objects	None	None
Static Resources	None	None
Packages	None	None

The 'Editline' tab shows the current file 'test.apex'.

The 'Related' tab lists:

Name	Current	Depends
test	Apex trigger	Referenced
Property	Custom field	Referenced
Tenant__c	Object	Referenced
Tenant__c	Object	Referenced

Developer Console - Google Chrome

org: Sdf1e052-dev-ed.my.salesforce.com/u/common/types/debug/ApexCSPage

test.apex [1] testHandler.apex [2] InsertApexBatched.apex [3]

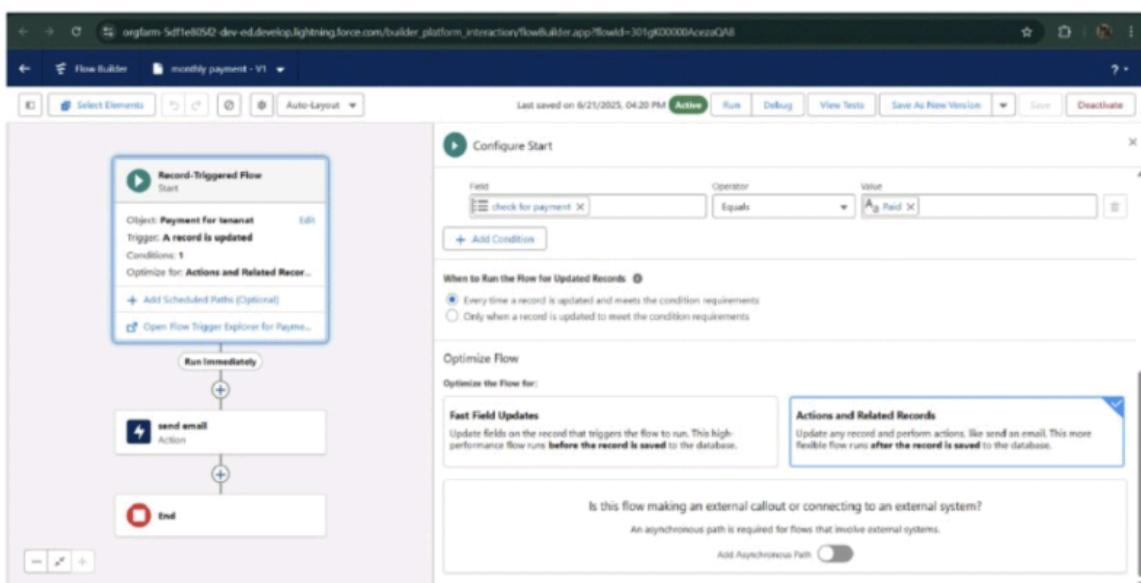
Data Coverage: 100% API Version: 54.0

```

1 * public class testHandler {
2
3 *     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
18
19                 newTenantaddError('A tenant can have only one property');
20
21             }
22
23         }
24
25     }
26
27 }
```

Log Tests Checkstyle Query Editor View Notes Progress Problems

● FLOWS



The screenshot shows the Flow Builder interface for a 'Record-Triggered Flow' named 'Payment for tenant'. The flow starts with a trigger 'A record is updated' for the object 'Payment for tenant'. It then branches to two actions: 'Run Immediately' and 'send_email'. Finally, it ends. The configuration details for the trigger and object selection are visible on the right side of the screen.

- Schedule class:
Create an Apex Class

```

1 global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendEmail();
6         }
7     }
8
9     public static void sendEmail() {
10        List<Tenant__c> tenants = [SELECT id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            String recipientEmail = tenant.Email__c;
13            String messageText = "I trust this email finds you well. I am writing to remind you that the monthly rent is due now. Kindly please ensure the smooth functioning of our rental arrangement and help maintain a positive living environment for all.";
14            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
15            email.setToRecipients(recipientEmail);
16            email.setSubject('Monthly Rent Payment');
17            email.setPlainTextMessage(messageText);
18        }
19    }
20
21 }

```

Schedule Apex class

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' section selected. The class 'MonthlyEmailScheduler' is displayed with its details:

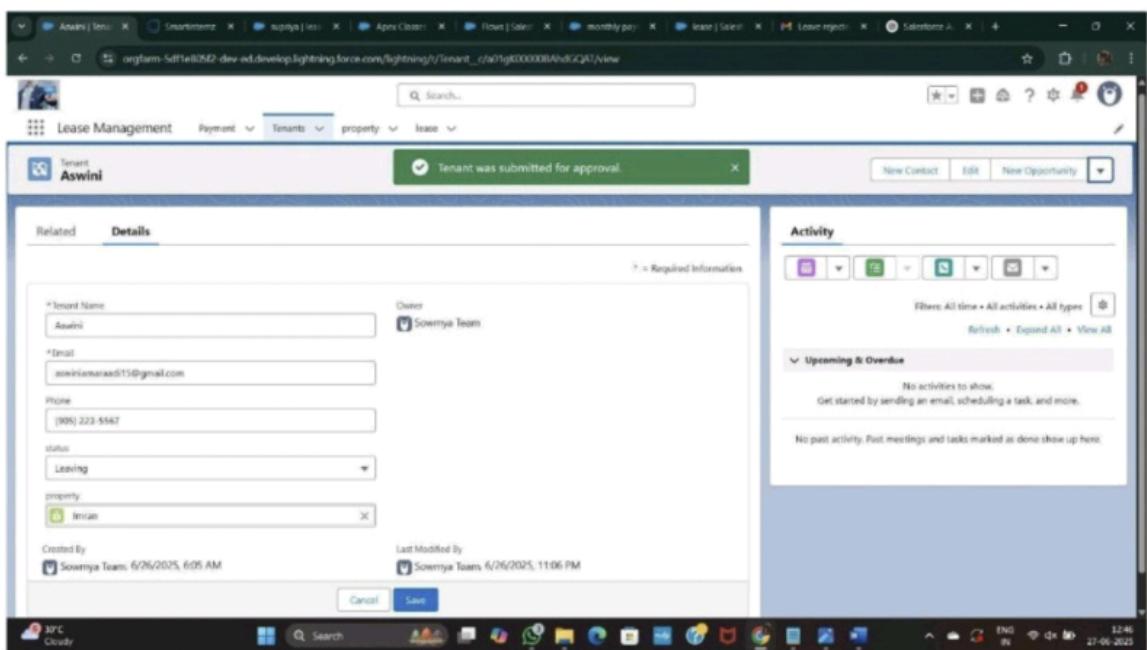
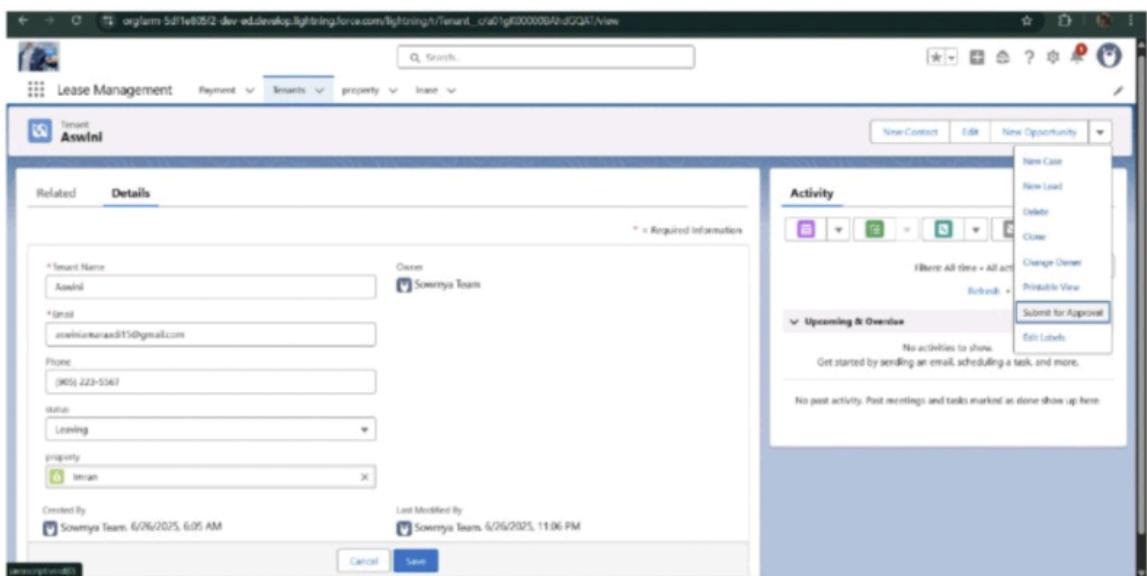
- Name:** MonthlyEmailScheduler
- Namespace Prefix:** None
- Created By:** Sonika Teja - 6/23/2025, 2:46 AM
- Status:** Active
- Code Coverage:** 0% (0/10)
- Last Modified By:** Sonika Teja - 6/23/2025, 2:47 AM

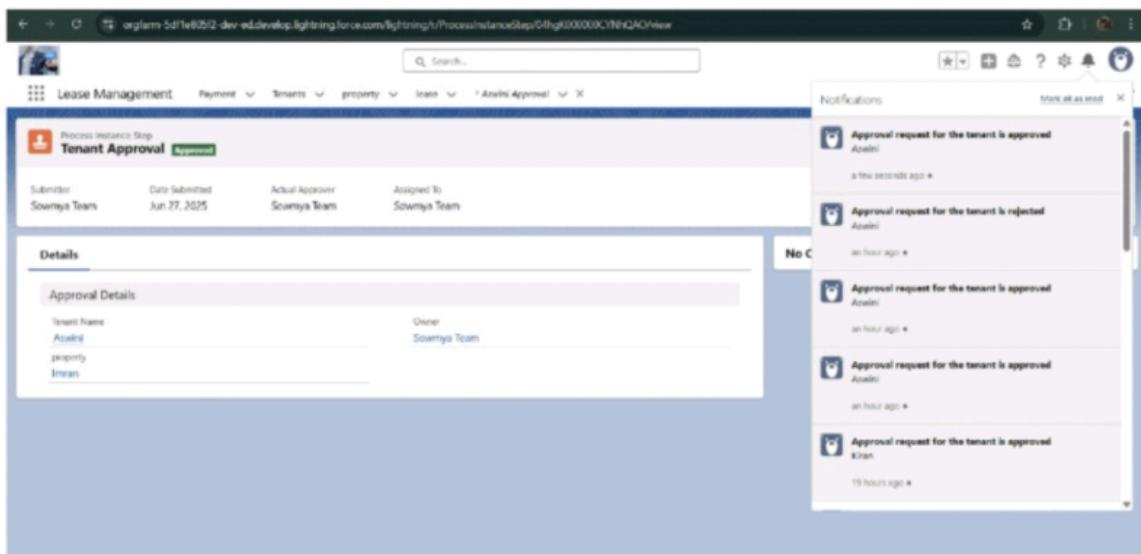
The 'Class Body' tab is active, showing the Apex code:

```

1 global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendEmail();
6         }
7     }
8
9     public static void sendEmail() {
10        List<Tenant__c> tenants = [SELECT id, Email__c FROM Tenant__c];
11        for (Tenant__c tenant : tenants) {
12            String recipientEmail = tenant.Email__c;
13            String messageText = "I trust this email finds you well. I am writing to remind you that the monthly rent is due now. Kindly please ensure the smooth functioning of our rental arrangement and help maintain a positive living environment for all.";
14            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
15            email.setToRecipients(recipientEmail);
16            email.setSubject('Monthly Rent Payment');
17            email.setPlainTextMessage(messageText);
18        }
19    }
20
21 }

```

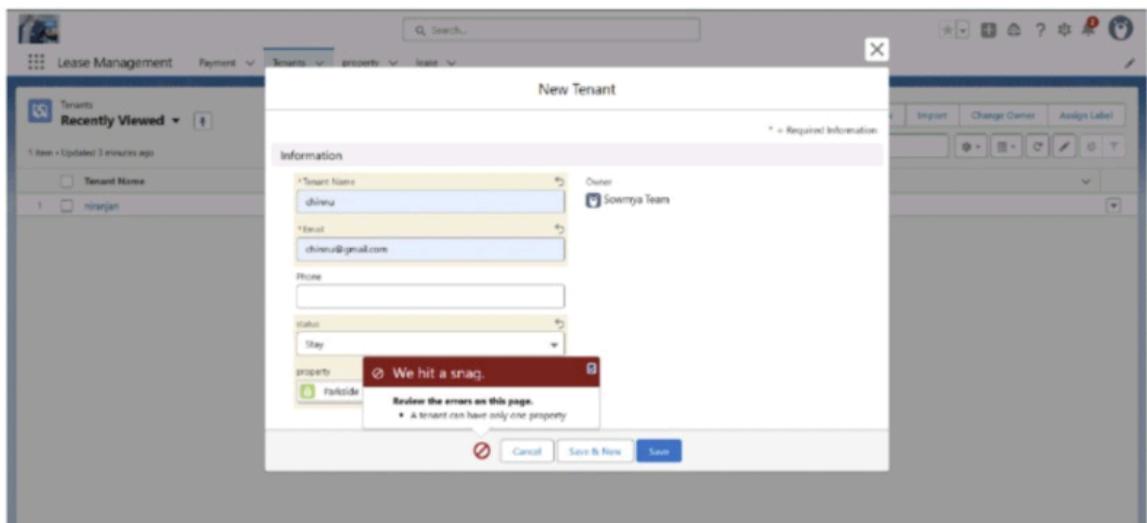




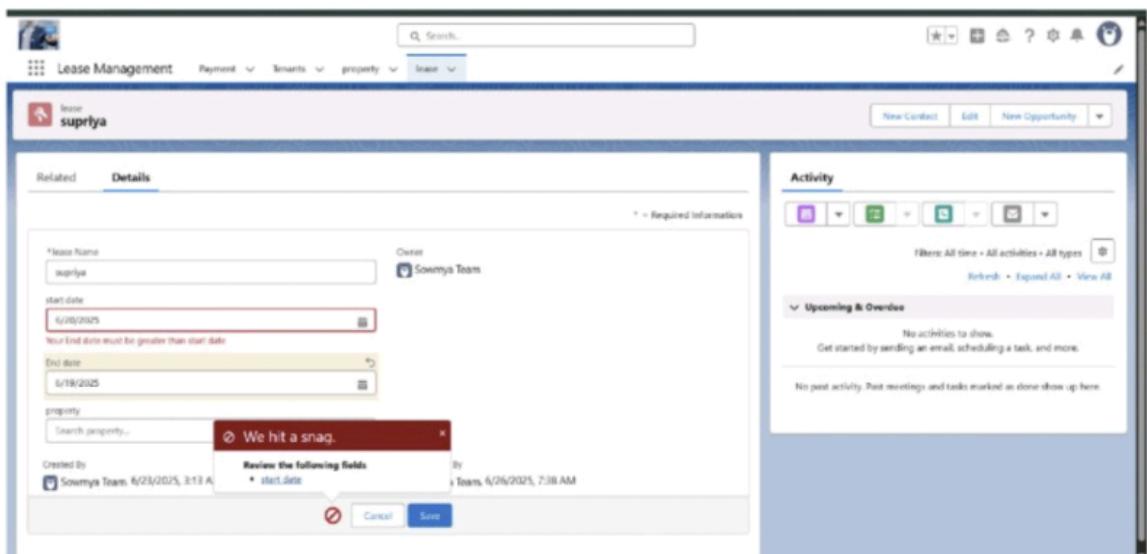
FUNCTIONAL AND PERFORMANCE TESTING

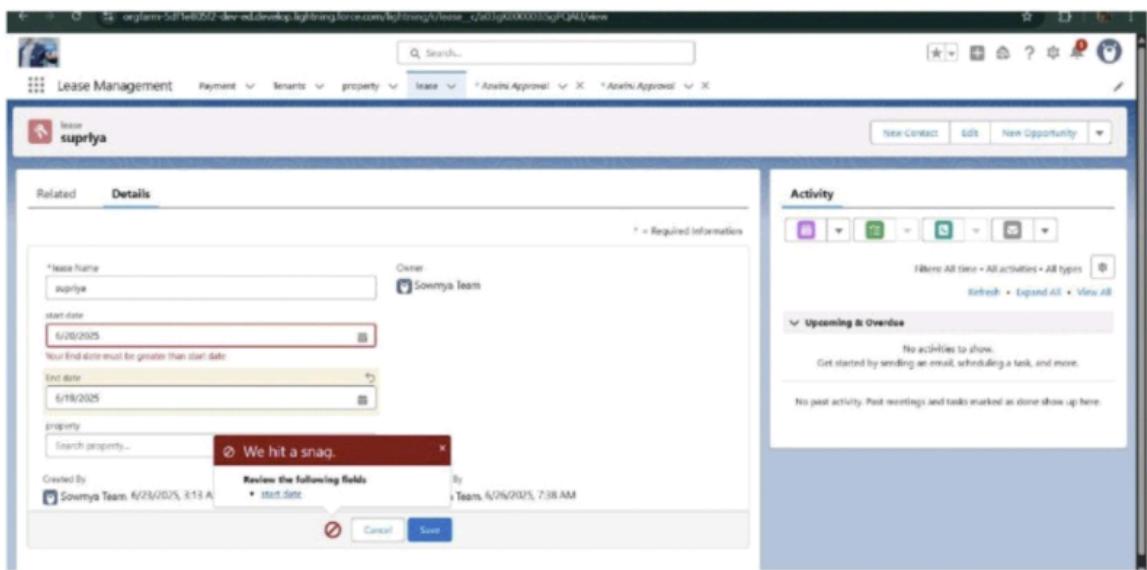
Performance Testing

- Trigger validation by entering duplicate tenant-property records

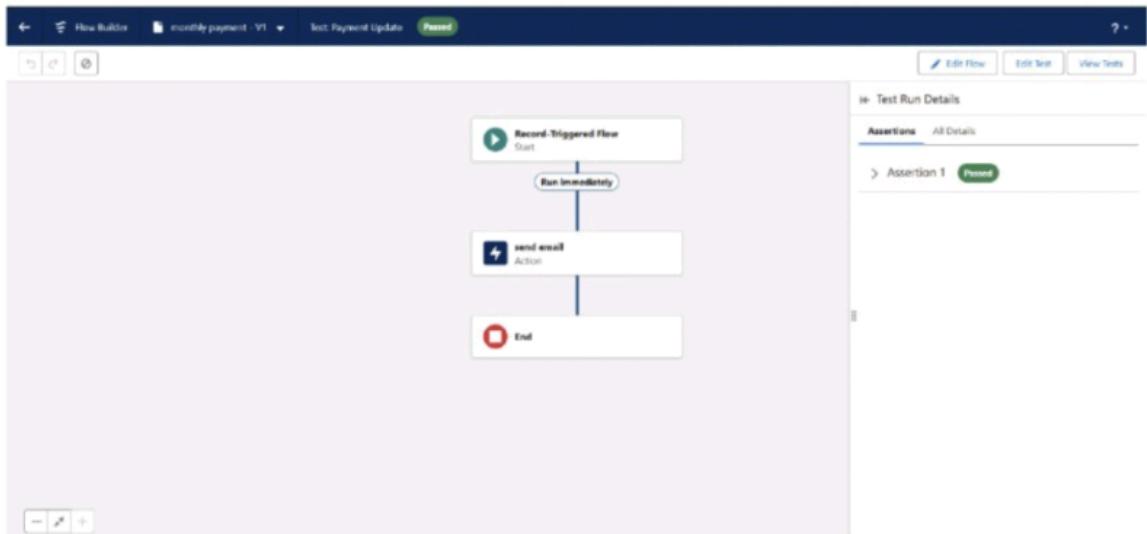


● Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the Dynamics 365 Lease Management interface. On the left, the 'Tenants' section displays a list of tenants, with 'niranjan' selected. The main area shows the 'Details' tab for 'niranjan'. The form includes fields for Tenant Name (niranjan), Email (niranjan106@gmail.com), Phone, Status (Stay), and Property (Parkside Lofts). It also shows the 'Created By' field (Sowmya Team) and 'Last Modified By' field (Sowmya Team, 6/23/2025, 3:58 AM). A 'Save' button is visible at the bottom right. To the right of the main form is a sidebar titled 'Notifications' which lists several recent activity items:

- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:29 PM)
- 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, 4:29 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, 1:25 PM)

This screenshot shows the Dynamics 365 Lease Management interface for the 'niranjan' tenant. The 'Details' tab is active. Below it, the 'Approval History (6+)' section displays a table of approval steps:

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

A 'View All' link is located below the table. The 'Payment (2)' section shows two entries: 'JAIK' and 'Rahul'. A 'New' button is available to add more payments. To the right, a sidebar provides a summary of recent activity:

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

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

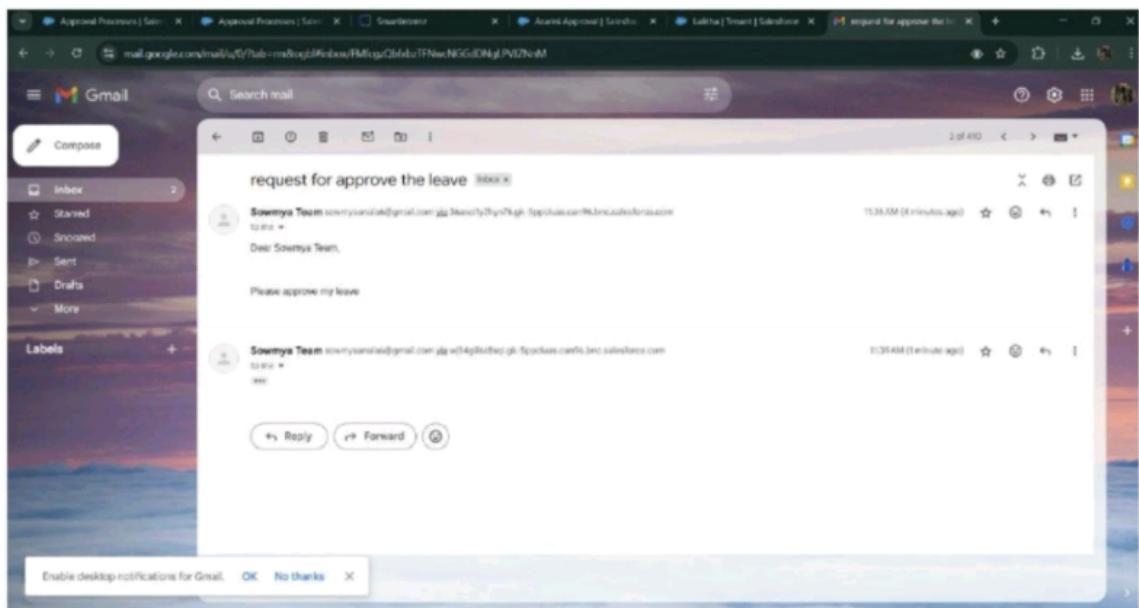
The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The page title is 'Custom Tabs'. It includes sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Lease', 'Payment', 'Property', and 'Tenant' with their respective 'Label' and 'Tab Style' (e.g., My tab, Credit card, Back, Help). A note at the bottom states: 'Custom Object tabs look and behave like the standard tabs provided by 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.'

- Email alerts

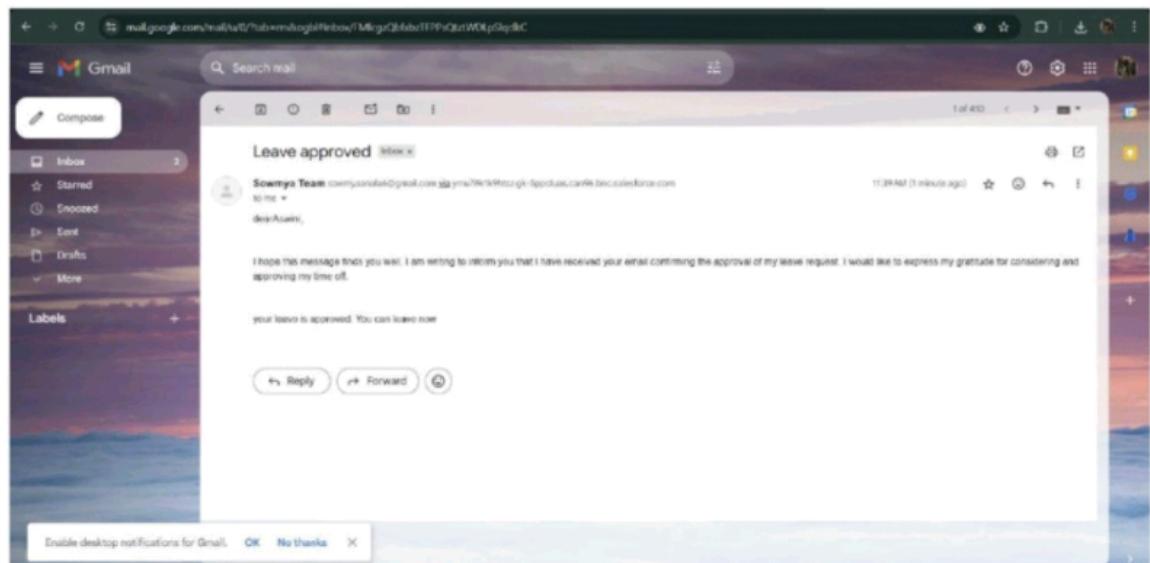
The screenshot shows the Salesforce interface with the 'Lease Management' tab selected. In the top navigation bar, there are dropdown menus for 'Payment', 'Tenants', 'Property', and 'Leave'. Below the navigation bar, the page title is 'Approval History'. The page displays a table of approval steps with columns for 'Step Name', 'Date', 'Status', 'Assigned To', 'Actual Approver', and 'Comments'. The table shows several steps for different users, with some comments indicating 'Approved' or 'Leaving'.

Step Name	Date	Status	Assigned To	Actual Approver	Comments
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
Approval Request Submitted	6/23/2025, 3:59 AM	Submitted	Sowmya Team	Sowmya Team	leaving
Step 1	6/23/2025, 3:51 AM	Approved	Sowmya Team	Sowmya Team	Approved
Approval Request Submitted	6/23/2025, 3:51 AM	Submitted	Sowmya Team	Sowmya Team	leaving
Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	leaving

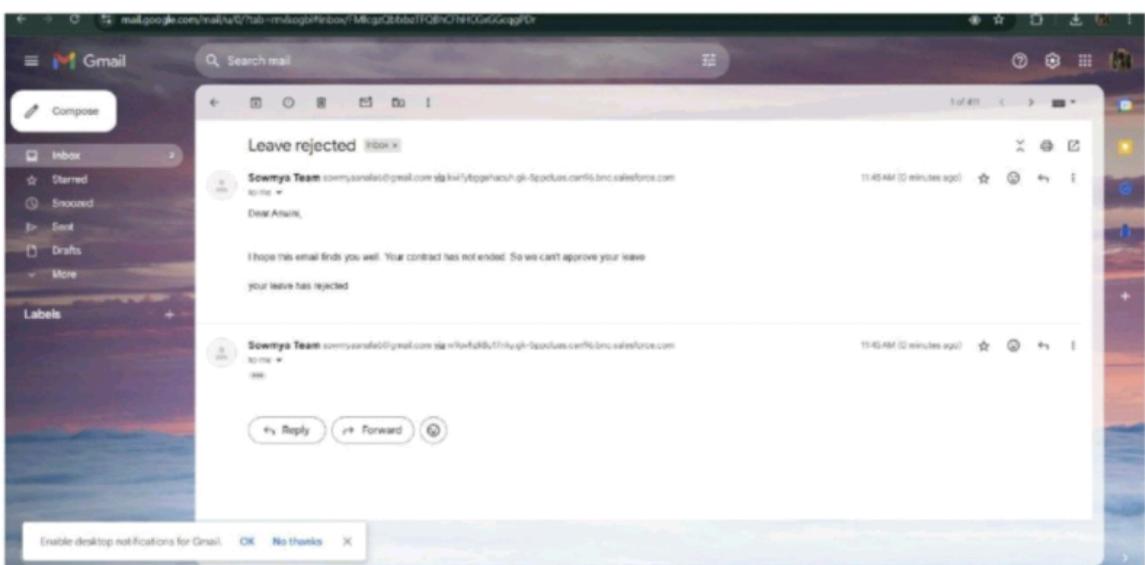
- Request for approve the leave



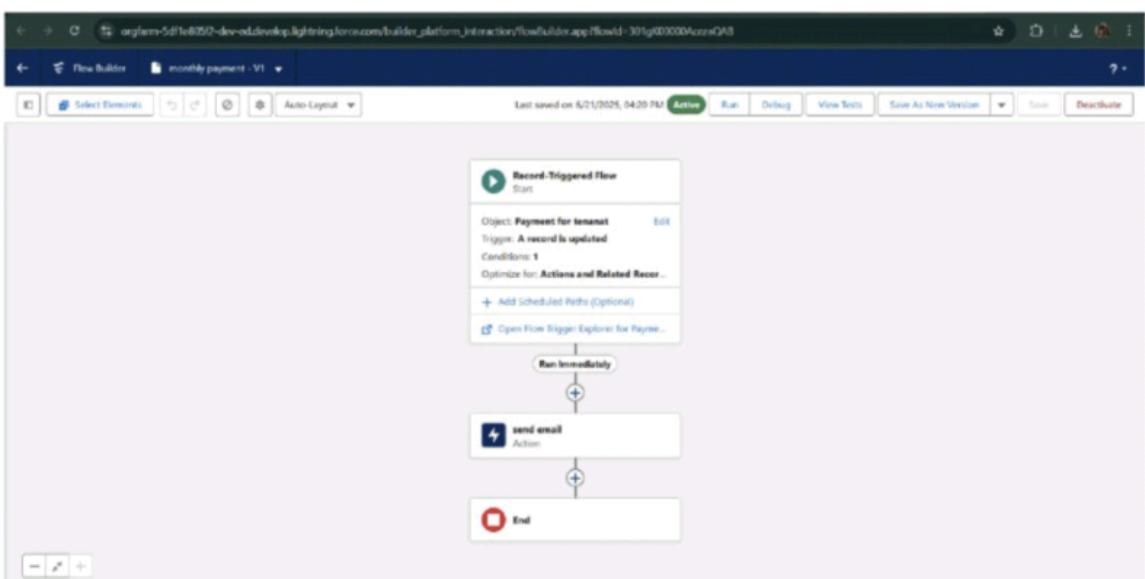
- Leave approved



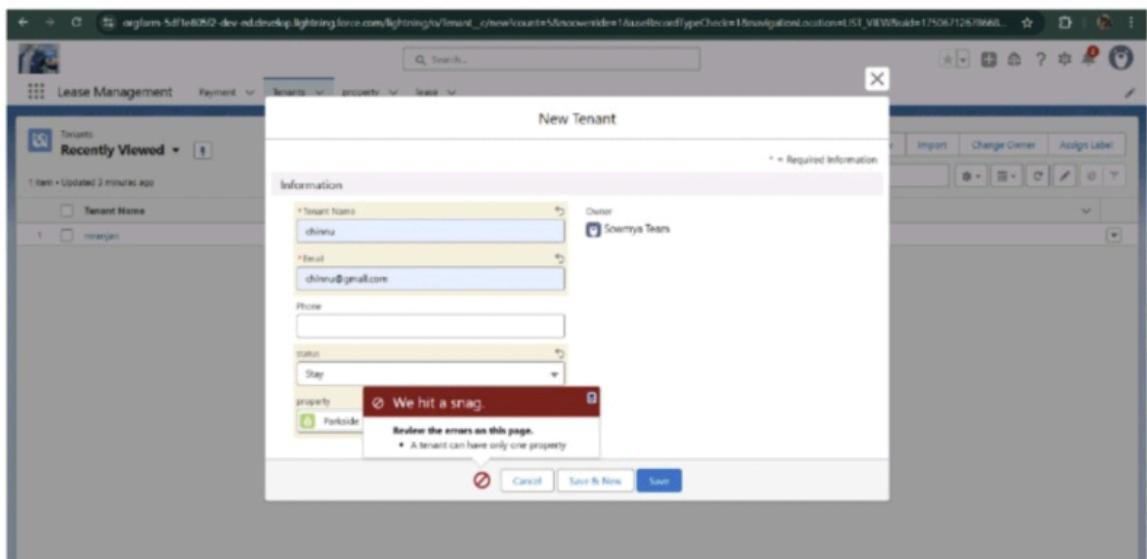
- Leave rejected



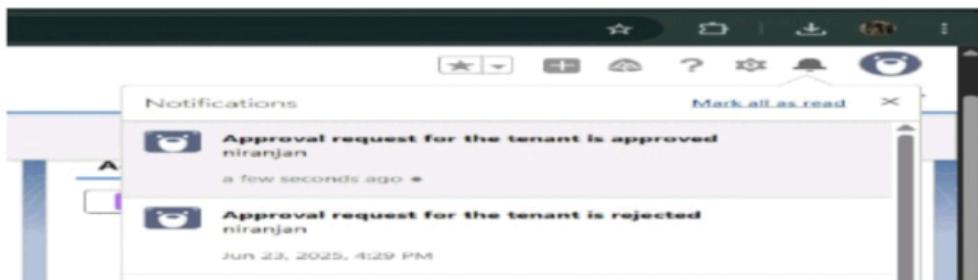
- Flow runs



- 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});
```

↳
↳
↳