

LEASE MANAGEMENT

COLLEGE NAME: KG COLLEGE OF ARTS AND SCIENCE

COLLEGE CODE: bru4y

TEAM ID: NM2025TMID23882

TEAM MEMBERS:

- **DEIVA KUMAR T** - 2326JB09@kgcas.com
- **DHANYA V** – 2326JB10@kgcas.com
- **DHANYATHA S** – 2326JB11@kgcas.com
- **GOKUL V** – 2326JB12@kgcas.com

INTRODUCTION

The Salesforce Lease Management project is an initiative to replace fragmented, manual lease administration processes with a unified, automated system on the Salesforce platform. Traditional methods, reliant on spreadsheets, emails, and physical files, lead to critical data silos, compliance risks from missed deadlines, and a lack of financial visibility. This project addresses these challenges by centralizing all lease data—from financial terms and critical dates to tenant communications and documents—into a single source of truth. The core purpose is to gain control over the entire lease lifecycle, from origination and negotiation to ongoing management and renewal, thereby ensuring compliance with modern accounting standards like ASC 842 and IFRS 16, mitigating financial risk, and enabling data-driven strategic decisions for the real estate portfolio.



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.

ADVANTAGES:

- **Operational Efficiency:** Automates key processes like rent escalation calculations, invoice generation, and critical date alerts, drastically reducing manual effort and errors.

- **Enhanced Compliance & Risk Mitigation:** System-enforced workflows ensure no key deadline is missed and that all lease data is accurately tracked for financial reporting, significantly reducing legal and financial risk.
- **360-Degree Portfolio Visibility:** Provides real-time dashboards and reports on portfolio performance, occupancy rates, and financial obligations, empowering strategic decision-making.
- **Improved Stakeholder Experience:** Offers tenants and internal teams a transparent, communicative, and efficient process for managing all lease-related interactions and requests.

DISADVANTAGES:

- **Implementation Complexity & Cost:** Requires a significant initial investment in Salesforce licensing, development, and data migration services.
- **Organizational Change Management:** Success depends on user adoption; training is essential to transition teams away from familiar legacy systems like spreadsheets.
- **Ongoing Maintenance:** The system requires dedicated administrators to manage updates, complex lease amendments, and potential integrations with other enterprise systems.

DEVELOPMENT PHASE

Creation of Developer Account

A Salesforce Developer account was created using the signup link:

<https://www.salesforce.com/form/developer-signup>

The screenshot shows the Salesforce Developer Edition sign-up page. The header features the Salesforce logo and the text "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this, there's a section titled "Sign up for your Developer Edition." with a list of benefits:

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

On the right side, there's a "Sign up for your Developer Edition" form with fields for First name (Amaraadi), Last name (Aswini), Job title (Student), Work email (aswiniamaraadi15@gmail.com), Company (Ideal Institute of Tech), and Country/Region (India). A note below the form states: "Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure." There's also a checkbox for agreeing to the Main Services Agreement and a link to the Privacy Statement.

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 Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scrolling Rules. The main panel displays the 'Details' tab for the 'property' object. The 'Details' section includes fields for API Name (property__c), Singular Label (property), Plural Label (property), and several checkboxes for enabling reports, tracking activities, and field history. Deployment status is set to 'Deployed' and help settings point to the standard Salesforce Help Window.

The screenshot shows the Salesforce Setup Object Manager interface. The left sidebar lists various object configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area displays the 'Details' tab for the 'Tenant' object. The 'Details' section includes fields for Description, API Name (set to 'Tenant__c'), Singular Label (set to 'Tenant'), and Plural Label (set to 'Tenants'). On the right, there are checkboxes for enabling Reports, Activities, and Field History, all of which are checked. Deployment status is set to 'Deployed'. Help settings and a standard help window link are also present. At the top right, there are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Setup Object Manager interface. The left sidebar lists various object configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area displays the 'Details' tab for the 'lease' object. The 'Details' section includes fields for Description, API Name (set to 'lease__c'), Singular Label (set to 'lease'), and Plural Label (set to 'lease'). On the right, there are checkboxes for enabling Reports, Activities, and Field History, all of which are checked. Deployment status is set to 'Deployed'. Help settings and a standard help window link are also present. At the top right, there are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details' and 'Fields & Relationships'. The main content area is titled 'Details' and contains fields for 'Description', 'API Name' (Payment_for_tenantat_c), 'Custom' (checked), 'Singular Label' (Payment for tenantat), 'Plural Label' (Payment), and 'Enable Reports' (checkboxes for Track Activities, Track Field History, Deployment Status, Help Settings). Buttons for 'Edit' and 'Delete' are at the bottom right.

Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. 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 options under 'Details' and 'Fields & Relationships'. The main content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Text(25)		
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>
property	property_c	Lookup(property)		<input checked="" type="checkbox"/>
property Name	Name	Text(80)		<input checked="" type="checkbox"/>
sfqt	sfqt_c	Text(18)		
Type	Type_c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenantat

Fields & Relationships					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Number(18, 0)			
check for payment	check_for_payment_c	Picklist			
Created By	CreatedById	Lookup(User)			
Last Modified By	LastModifiedById	Lookup(User)			
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>	
Payment date	Payment_date_c	Date			
Payment Name	Name	Text(90)		<input checked="" type="checkbox"/>	

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)		<input checked="" type="checkbox"/>	
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>	
property	property_c	Lookup(property)		<input checked="" type="checkbox"/>	
start date	start_date_c	Date			

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone__c	Phone		
status	status__c	Picklist		
Tenant Name	Name	Text(80)		

Developed Lightning App with relevant tabs

App Details & Branding

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

App Details

- *App Name: Lease Management
- *Developer Name: Lease_Management
- Description: Application to efficiently handle the processes related to leasing real estate properties.

App Branding

- Image:
- Primary Color Hex Value: #0070D2

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

App Launcher Preview

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

App Settings

App Details & Branding
App Options
Utility Items (Desktop Only)

Navigation Items

User Profiles

Available Items

Type to filter list... [Create](#)

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

Selected Items

- Payment
- Tenants
- property
- lease

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

App Settings

App Details & Branding
App Options
Utility Items (Desktop Only)

User Profiles

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

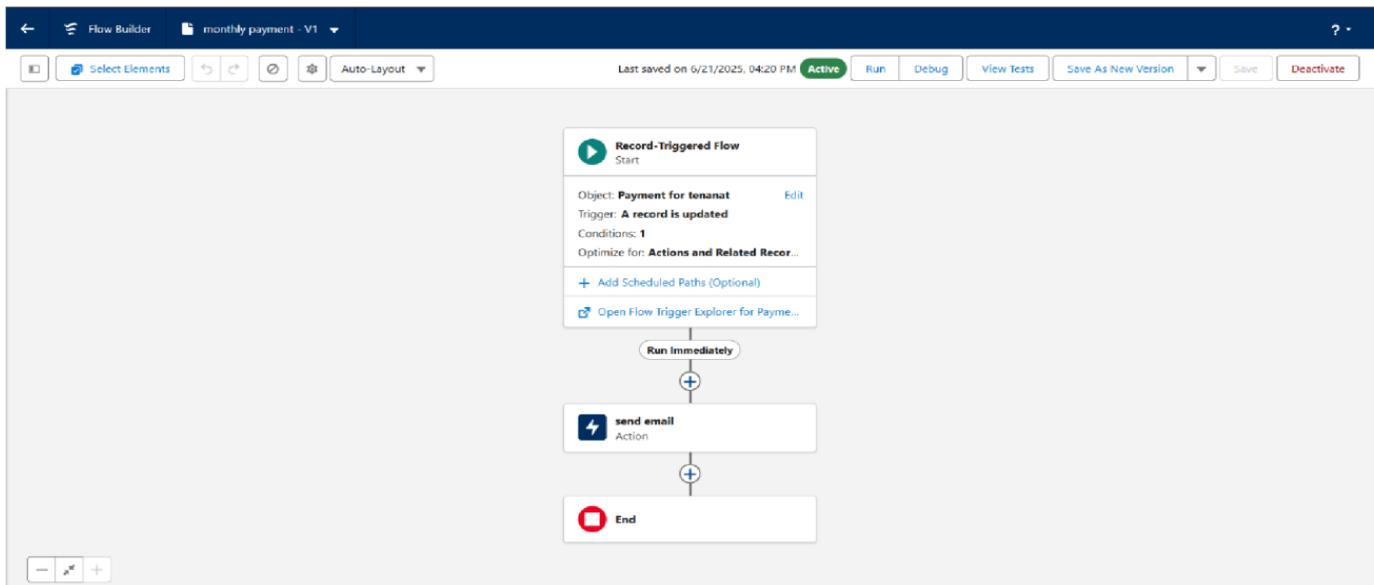
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator

The screenshot shows a Salesforce Lightning interface for 'Lease Management'. The top navigation bar includes tabs for 'Payment', 'Tenants', 'property', and 'lease'. A search bar at the top right contains the placeholder 'Search...'. Below the navigation is a header for 'Payment' with a 'Recently Viewed' dropdown and a 'New' button. The main content area displays a list titled 'Payment Name' with 5 items, updated a few seconds ago. The list includes entries: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, and 5. Lahari. To the right of the list are various filter and search tools.

Implemented Flows for monthly rent and payment success



To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' screen for the 'lease' object. The 'Rule Name' is set to 'lease_end_date'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'End_date_c <= start_date_c'. A tooltip for the 'ABS' function is displayed, stating: 'Returns the absolute value of a number, a number without its sign'. The 'Error Message' area is empty.

Validation Rule Edit

Rule Name: lease_end_date

Active:

Description:

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

Insert Field Insert Operator

Functions

- All Function Categories -

ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

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

Help on this function

Check Syntax

Error Message

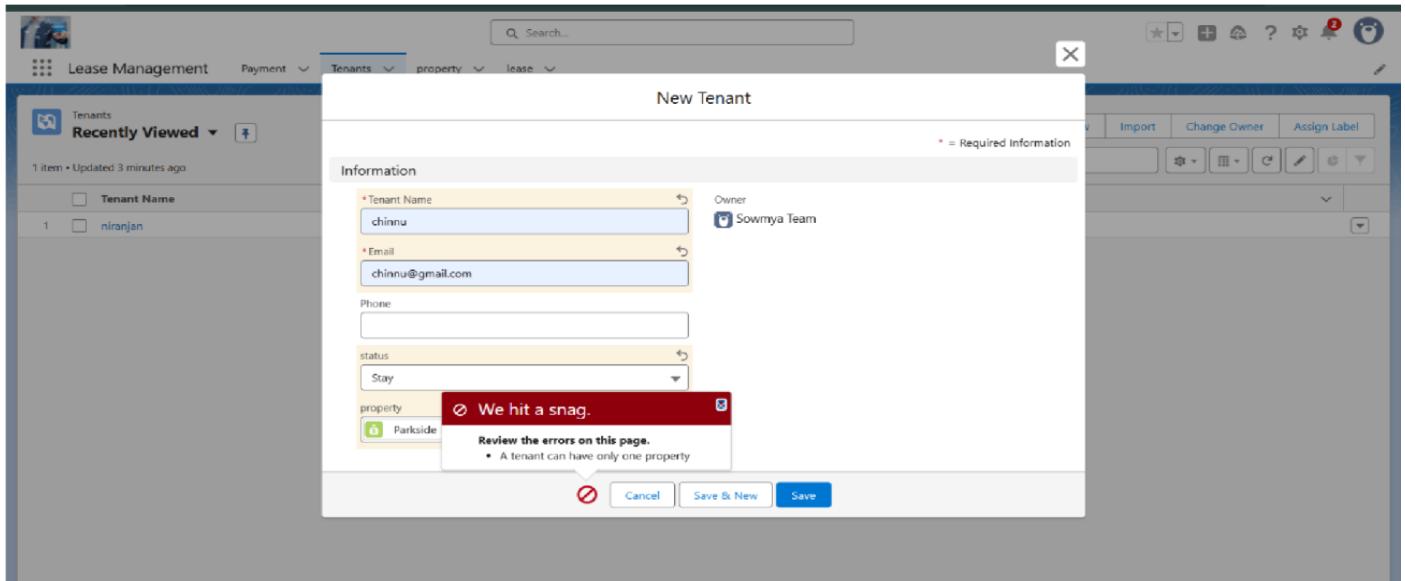
The screenshot shows the 'Validation Rule Detail' screen for the 'lease_end_date' rule. The rule is active and has the formula 'End_date_c <= start_date_c'. The error message is 'Your End date must be greater than start date'. The rule was created by 'Sowmya Team' on 6/19/2025, 5:37 AM and modified by 'Sowmya Team' on 6/26/2025, 7:47 AM.

lease Validation Rule

Validation Rule Detail

Rule Name	lease_end_date	Active
Error Condition Formula	End_date_c <= start_date_c	<input checked="" type="checkbox"/>
Error Message	Your End date must be greater than start date	Error Location: start date
Description		
Created By	Sowmya Team, 6/19/2025, 5:37 AM	Modified By: Sowmya Team, 6/26/2025, 7:47 AM

Added Apex trigger to restrict multiple tenants per property



Scheduled monthly reminder emails using Apex class

A screenshot of the Salesforce Developer Console. The top navigation bar shows "File", "Edit", "Design", "Test", "Workspaces", "Help", "test.apex", "testHandler.apex", "MonthlyEmailScheduler.apex", "Code Coverage", "API Version 44", and "67%". The main area displays the Apex code for the "MonthlyEmailScheduler" class. The code implements the "Schedulable" interface and defines an "execute" method that checks if the current day is the 1st. If it is, it calls the "sendMonthlyEmails" static method. The "sendMonthlyEmails" method queries all "Tenant__c" records and sends a monthly reminder email to each tenant's email address. The email content, subject, and recipient are defined. The developer console also shows tabs for "Logs", "Tests", "Checkpoints", "Query Editor", "View Status", "Progress", and "Problems", with the "Problems" tab currently selected.

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

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

- 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
- Email Template Preview:**
 - 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.

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

- Email Template Detail:**
 - Email Templates from Salesforce: Unified Public Classic Email Templates
 - Email Template Name: tenant_leaving
 - Template Unique Name: tenant_leaving
 - Encoding: Unicode (UTF-8)
 - Author: Sowmya_Team [Change]
 - Description:
 - Created By: Sowmya_Team 6/20/2025, 1:06 AM
 - Modified By: Sowmya_Team 6/20/2025, 1:06 AM
- Email Template Preview:**
 - 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 'Classic Email Templates' page open. The template name is '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 [Change]
Description	
Created By	Sowmya_Team, 6/20/2025, 1:11 AM

On the right side, there are status fields: 'Available For Use' (checked), 'Last Used Date', and 'Times Used'. Below the detail section are 'Edit', 'Delete', and 'Clone' buttons.

The 'Email Template' preview section shows the subject 'Leave rejected' and the plain text preview:

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

At the bottom of the preview section, there is a note: "This communication is a friendly reminder concerning your monthly rent payment. What is currently authorized. So nothing to worry about."

The screenshot shows the Salesforce Setup interface with the 'Classic Email Templates' page open. The template name is 'Tenant Email'. The 'Email Template Detail' section shows the following information:

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

On the right side, there are status fields: 'Available For Use' (checked), 'Last Used Date', and 'Times Used'. Below the detail section are 'Edit', 'Delete', and 'Clone' buttons.

The 'Email Template' preview section shows the subject 'Urgent: Monthly Rent Payment Reminder' and the plain text preview:

Subject : Urgent: Monthly Rent Payment Reminder

Plain Text Preview

Dear {Tenant__c.Name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

At the bottom of the preview section, there is a note: "This communication is a friendly reminder concerning your monthly rent payment. What is currently authorized. So nothing to worry about."

The screenshot shows the Salesforce Setup interface under the 'Email' section. A search bar at the top right contains 'Search Setup'. The left sidebar has sections for 'Email' (selected), 'Classic Email Templates' (selected), and 'Lightning Email Templates'. A message at the bottom left says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Classic Email Templates' and shows a 'Text Email Template' named 'tenant payment'. It includes fields for 'Email Template Detail' such as 'Email Template Name' (tenant payment), 'Template Unique Name' (tenant_payment), 'Encoding' (Unicode (UTF-8)), 'Author' (Sowmya_Team [Change]), and 'Description' (Created By Sowmya_Team, 6/20/2025, 1:13 AM). On the right, it shows 'Available For Use' checked, 'Last Used Date' (Times Used), and 'Modified By' (Sowmya_Team, 6/20/2025, 1:13 AM). Below this is a preview window showing the email template's subject ('Confirmation of Successful Monthly Payment') and 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 under the 'Data' section. A search bar at the top right contains 'Search Setup'. The left sidebar has sections for 'Data' (selected), 'Feature Settings', 'Process Automation', and 'Approval Processes' (selected). A message at the bottom left says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Approval Processes' and shows a process named 'Tenant: TenantApproval'. It includes fields for 'Process Definition Detail' such as 'Process Name' (TenantApproval), 'Unique Name' (TenantApproval), 'Description' (tenant: status equals Stay), 'Entry Criteria' (tenant: status equals Stay), 'Record Editability' (Administrator ONLY), and 'Approval Assignment Email Template' (Initial Submitters: Tenant Owner). On the right, it shows 'Active' checked, 'Next Automated Approver Determined By' (Allow Submitters to Recall Approval Requests checked), and 'Modified By' (Sowmya_Team, 6/26/2025, 11:57 PM). Below this are sections for 'Initial Submission Actions' (Add Existing, Add New) and 'Approval Steps' (Step Number 1, Name Step 1, Description Lock the record from being edited, Criteria, Assigned Approver User:Sowmya_Team, Reject Behavior Final Rejection).

For Check for Vacant:

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

- Page Title:** Approval Processes
- Process Name:** check for vacant
- Unique Name:** check_for_vacant
- Description:** Tenant: status EQUALS Leaving
- Entry Criteria:** Administrator ONLY
- Record Editability:** Leave approved
- Approval Assignment Email Template:** Initial Submitters
- Initial Submitters:** Tenant Owner
- Created By:** Sowmya_Team
- Modified By:** Sowmya_Team
- Status:** Active (checked)
- Next Automated Approver Determined By:** None
- Allow Submitters to Recall Approval Requests:** Unchecked
- Initial Submission Actions:**
 - Action: Record Lock, Type: Email Alert, Description: Lock the record from being edited, please approve my leave.
- Approval Steps:**
 - Action: Step Number 1, Name: step1, Description: None, Assigned Approver: User:Sowmya_Team, Reject Behavior: Final Rejection.

Apex Trigger

Create an Apex Trigger

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

- Trigger Name:** test
- Object:** Tenant__c
- Event:** before insert
- Code:**

```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```
- Code Coverage:** None
- API Version:** 64
- Repository:** A modal window titled "Open" shows the repository structure:

Entity Type	Entities	Related
Classes	test	Name Extent Direction
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar shows `test.apex` (highlighted in orange), `testHandler.apxc`, and `MonthlyEmailScheduler.apxc`. The main area displays 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 }
8
9 }
10
11 }

```

Below the code editor, there is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing a table with one row:

Name	Line	Problem

Create an Apex Handler class

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar shows `test.apex` (highlighted in orange), `testHandler.apxc`, and `MonthlyEmailScheduler.apxc`. The main area displays the following Apex code:

```

1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(exi
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17             if (newTenant.Property__c != null) {
18                 newTenantaddError('A t
19
20             }
21
22         }
23     }
24
25 }

```

A modal dialog titled "Open" is displayed over the code editor, showing the repository browser. The Entity Type is set to "Classes". The results table shows the following entries:

Entity Type	Entity Name	Namespace	Name	Extent	Direction
Classes	testHandler	MonthlyEmailScheduler	test	ApexTrigger	Referenced...
			property	CustomField	References
			Tenant__c	SObject	References
			Tenant__c	SObject	References

Below the table, there are buttons for "Open", "Filter", "Hide Managed Packages", and "Refresh".

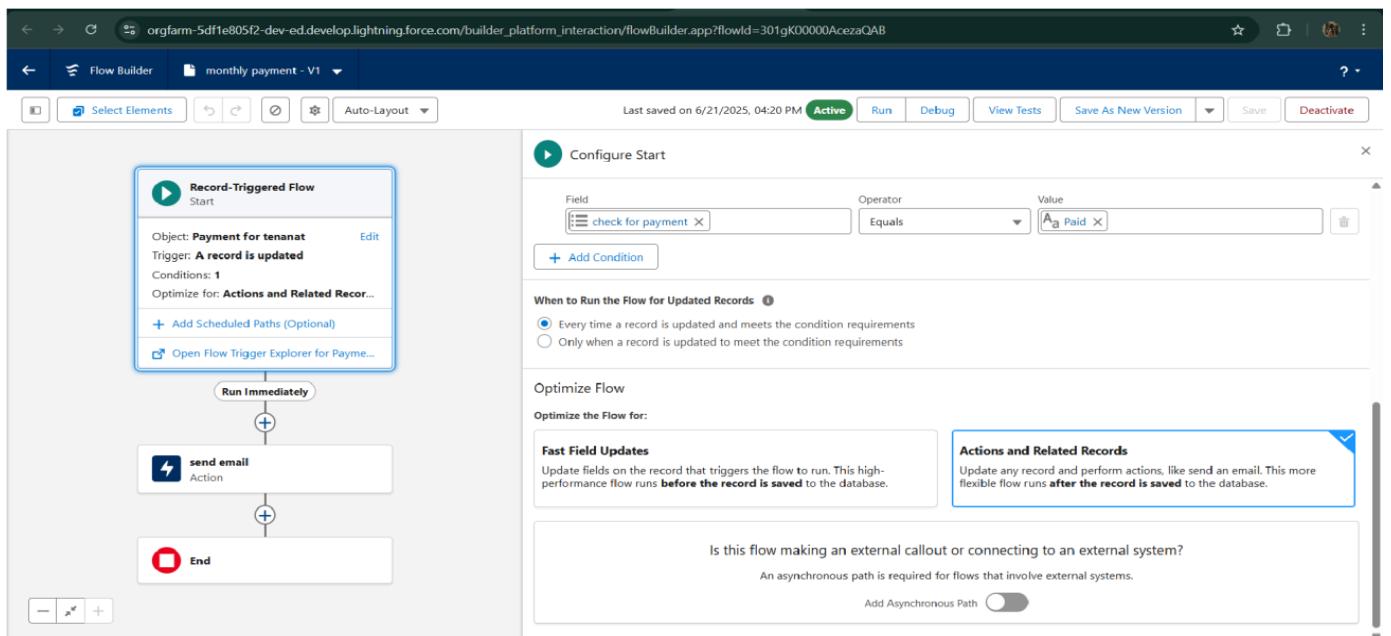
Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

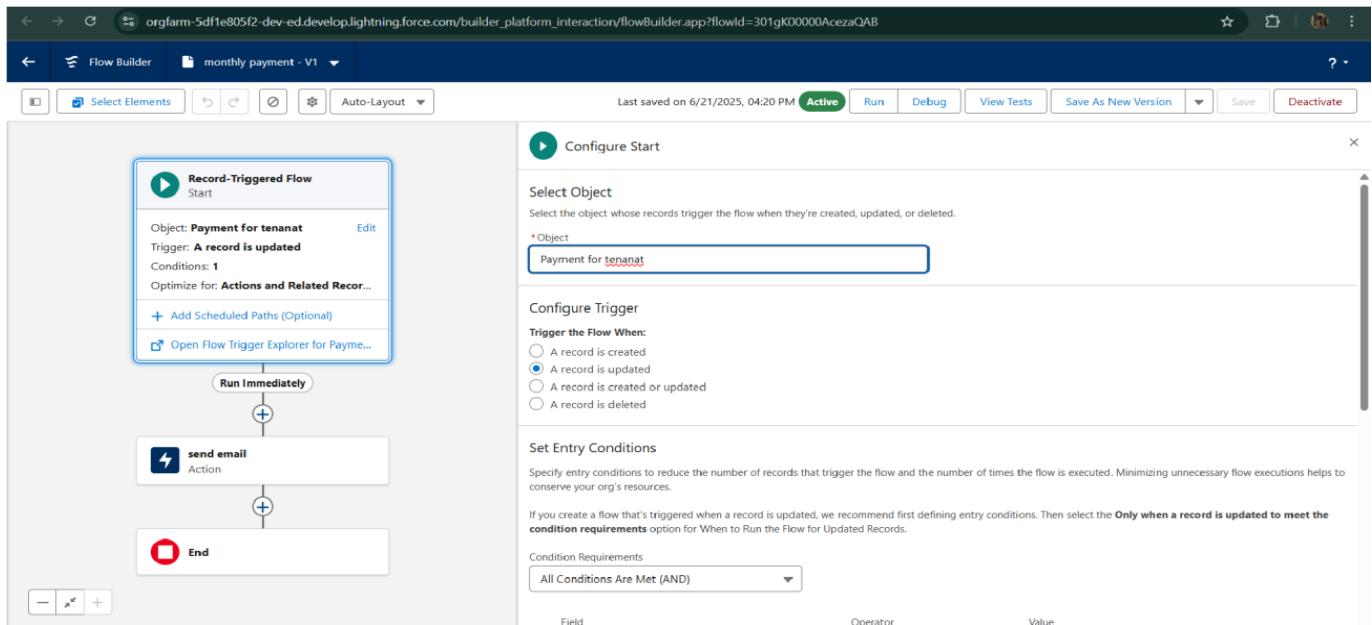
```

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
28 }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

FLOW





Schedule class: Create an Apex Class

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10        }
11    }
12
13 }
14
15
16 * public static void sendMonthlyEmail
17
18     List<Tenant__c> tenants = [SEL
19
20     for (Tenant__c tenant : tenants
21
22         String recipientEmail = tenant.Email__c;
23

```

Open

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

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

```

1 global class MonthlyEmailScheduler implements schedulable {
2 
3     global void execute(SchedulableContext sc) {
4         Integer currentDay = Date.today().day();
5 
6         if (currentDay == 1) {
7             sendMonthlyEmails();
8         }
9     }
10 
11 }
12 
13 
14 public static void sendMonthlyEmails() {
15     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
16 
17     for (Tenant__c tenant : tenants) {
18         String recipientEmail = tenant.Email__c;
19 
20         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.';
21 
22         String emailSubject = 'Monthly Rent Payment Due';
23 
24         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
25 
26         email.setToExternal(true);
27         email.setSubject(emailSubject);
28         email.setPlainTextBody(emailContent);
29 
30         messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
31     }
32 }
33 
34 }
35 
```

File Edit View Insert Home Project Profile

Schedule Apex class

Setup Home Object Manager ▾

Search Setup

Apex Classes

Apex Class MonthlyEmailScheduler

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

Class Body Class Summary Version Settings Trace Flags

```

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

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* = Required Information

* Tenant Name: Aswini
Owner: Sowmya Team

* Email: aswiniamaraadi15@gmail.com

Phone: (905) 223-5567

Status: Leaving

Property: Imran

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

New Contact Edit New Opportunity

Activity

Upcoming & Overdue

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

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

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

Filters: All time • All activities • All types Refresh

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* = Required Information

* Tenant Name: Aswini
Owner: Sowmya Team

* Email: aswiniamaraadi15@gmail.com

Phone: (905) 223-5567

Status: Leaving

Property: Imran

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

✓ Tenant was submitted for approval.

New Contact Edit New Opportunity

Activity

Upcoming & Overdue

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

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

30°C Cloudy

Cloudy

Search

12:46 27-06-2025

The screenshot shows a Salesforce Lightning interface for 'Lease Management'. A process instance step for 'Tenant Approval' has been completed and marked as 'Approved'. The approval details indicate the tenant name is Aswini and the owner is Sowmya Team. A notifications sidebar on the right lists several approval requests for the tenant, all of which have been approved by different users (Aswini, Imran, Kiran) at various times.

Submitter	Date Submitted	Actual Approver	Assigned To
Sowmya Team	Jun 27, 2025	Sowmya Team	Sowmya Team

Approval Details

Tenant Name	Owner
Aswini	Sowmya Team
Imran	

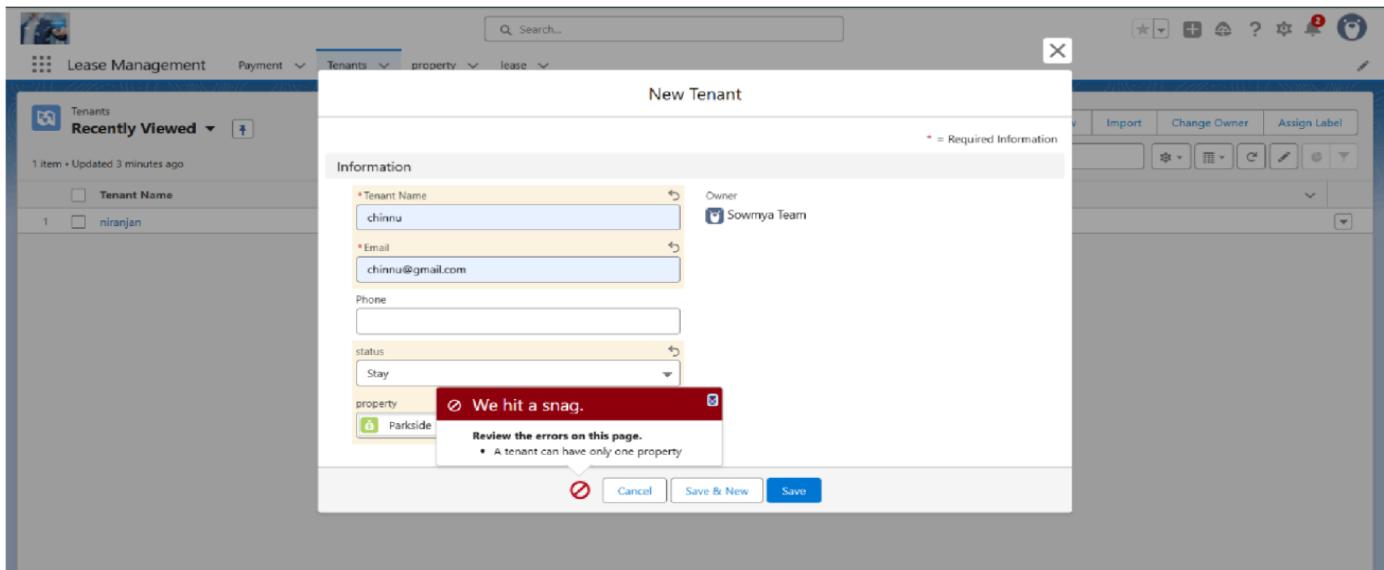
Notifications

- Approval request for the tenant is approved Aswini (a few seconds ago)
- Approval request for the tenant is rejected Aswini (an hour ago)
- Approval request for the tenant is approved Aswini (an hour ago)
- Approval request for the tenant is approved Aswini (an hour ago)
- Approval request for the tenant is approved Kiran (19 hours ago)

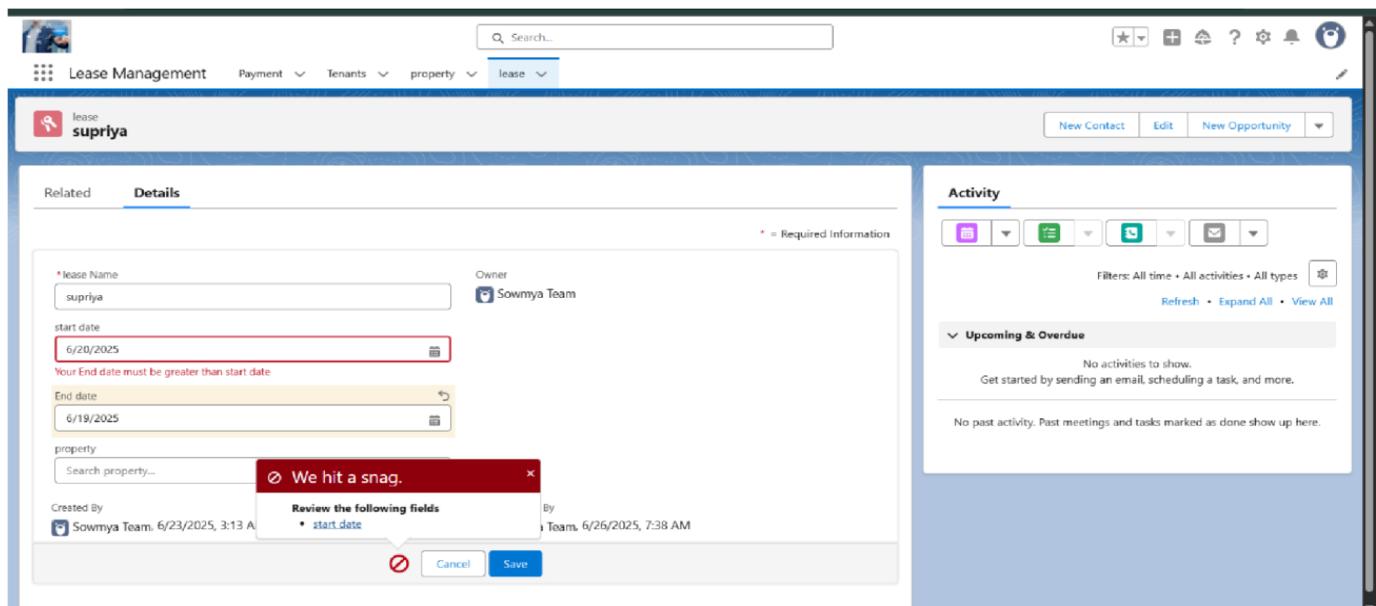
FUNCTIONAL AND PERFORMANCE TESTING

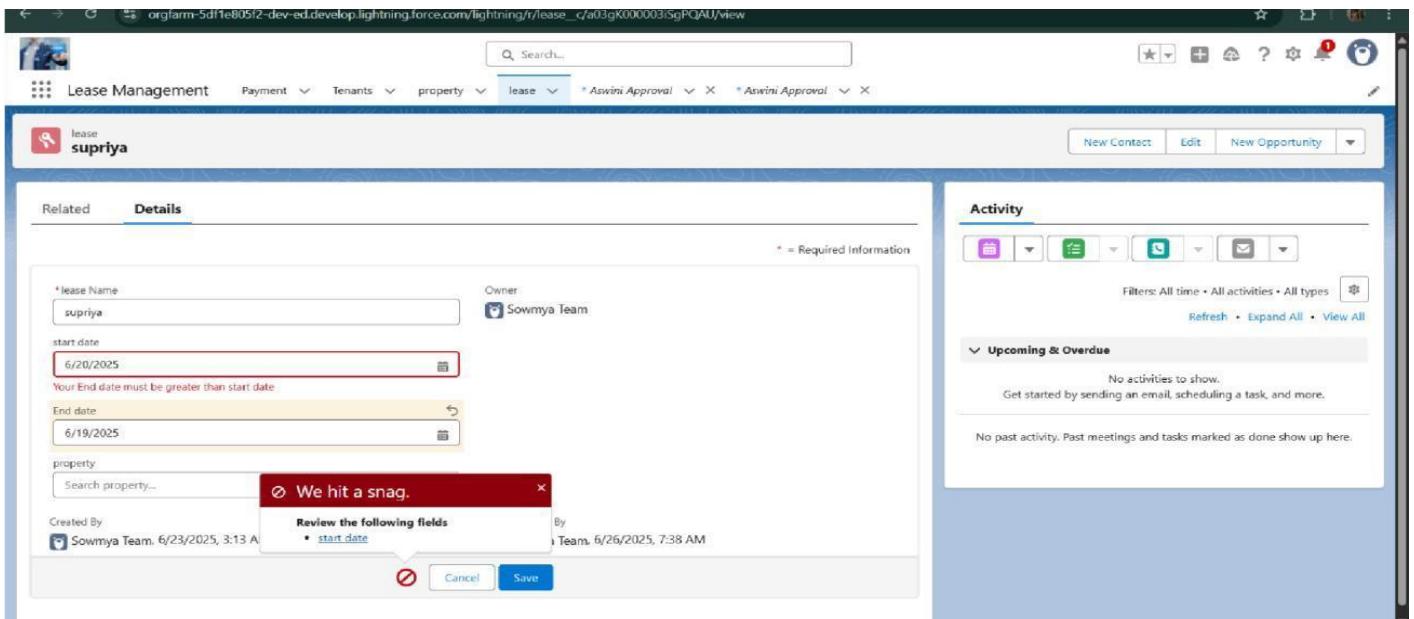
Performance Testing

Trigger validation by entering duplicate tenant-property records

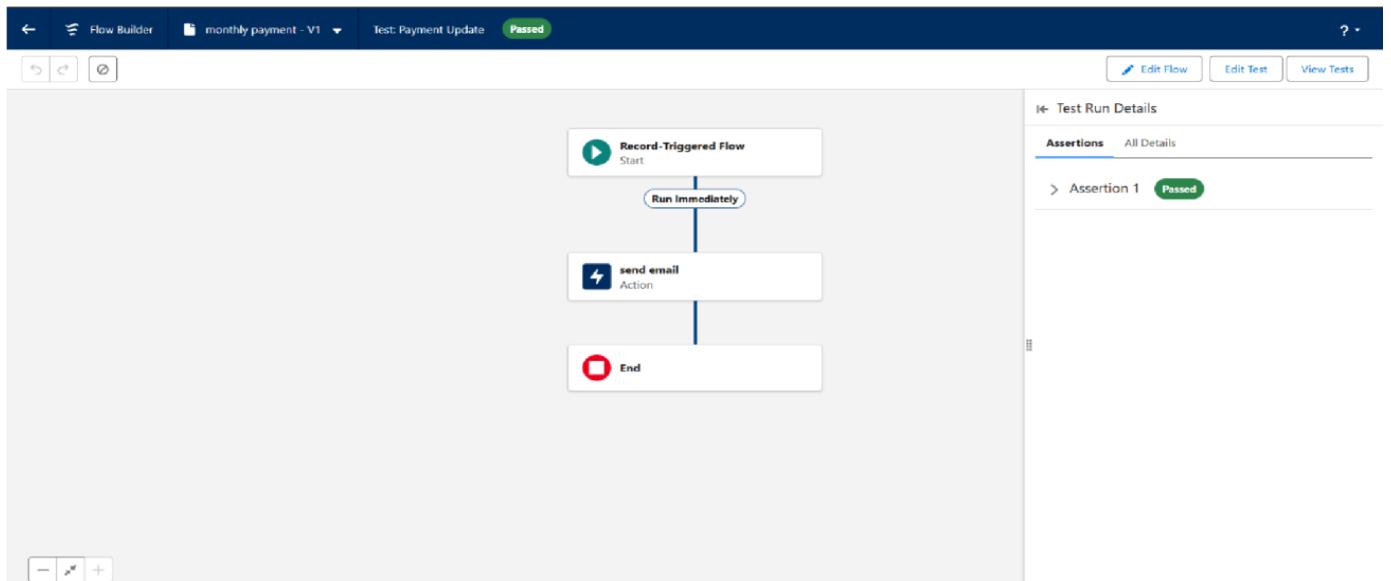


Validation Rule checking





Test flows on payment update



Approval process validated through email alerts and status updates

The screenshot shows a CRM interface for 'Lease Management'. A search bar at the top has the query 'niranjan Approval'. Below it, a card for 'Tenant niranjan' displays details: Tenant Name (niranjan), Email (niranjan1506@gmail.com), Phone (empty), Status (Stay), and Property (Parkside Lofts). The card also shows 'Created By' (Sowmya Team) and 'Last Modified By' (Sowmya Team). Buttons for 'Cancel' and 'Save' are at the bottom.

To the right, a 'Notifications' sidebar lists four 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

A promotional message for 'New Guidance Center learning resource available' is also shown.

The screenshot shows a CRM interface for 'Lease Management'. A search bar at the top has the query 'niranjan'. Below it, a card for 'Tenant niranjan' displays the 'Approval History' section, which lists six steps: Step 1 (Approved), Approval Request Submitted (Submitted), Step 1 (Rejected), Approval Request Submitted (Submitted), Step 1 (Approved), and Approval Request Submitted (Submitted). A 'View All' button is below this section.

Below the approval history is a 'Payment (2)' section, showing two entries: 'Jack' and 'Rahul'. A 'View All' button is at the bottom of this section.

The right side of the screen features a sidebar with buttons for 'New Contact', 'Edit', and 'New Opportunity'. A message says '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.'

RESULTS

Output Screenshots

Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The page title is 'Custom Tabs'. It includes sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. In the 'Custom Object Tabs' section, there are four tabs labeled 'Lease', 'Payment', 'Property', and 'Tenants', each with a corresponding icon and tab style.

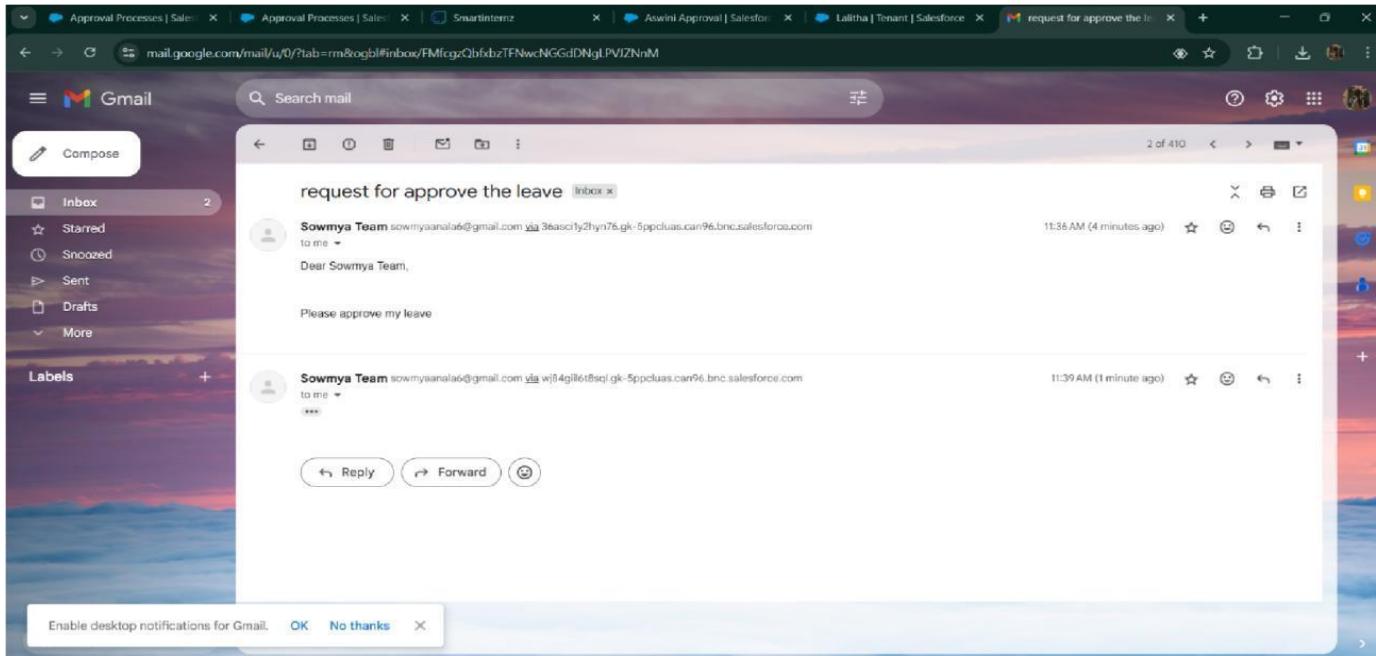
Action	Label	Tab Style	Description
Edit Del	Lease	Keys	
Edit Del	Payment	Credit card	
Edit Del	Property	Sack	
Edit Del	Tenants	Map	

Email alerts

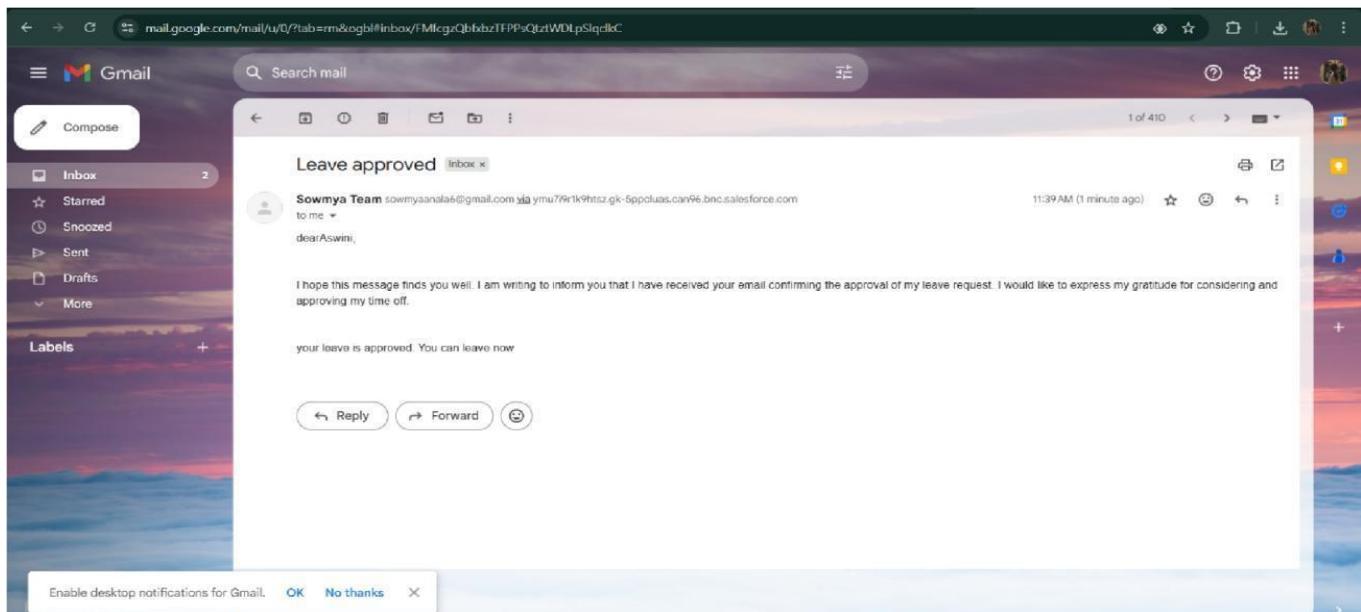
The screenshot shows the 'Lease Management' application with the 'Approval History' page selected. The page title is 'Approval History'. It displays a table of approval steps for a specific tenant. The columns include Step Name, Date, Status, Assigned To, Actual Approver, and Comments.

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

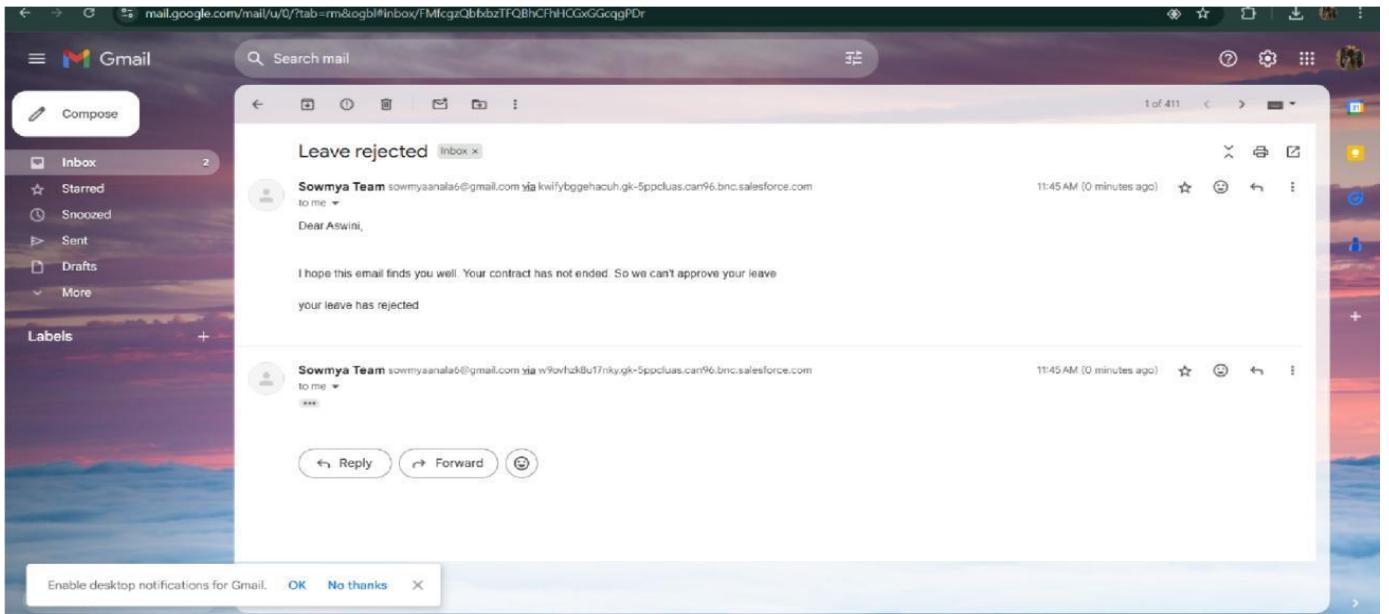
Request or approve the leave



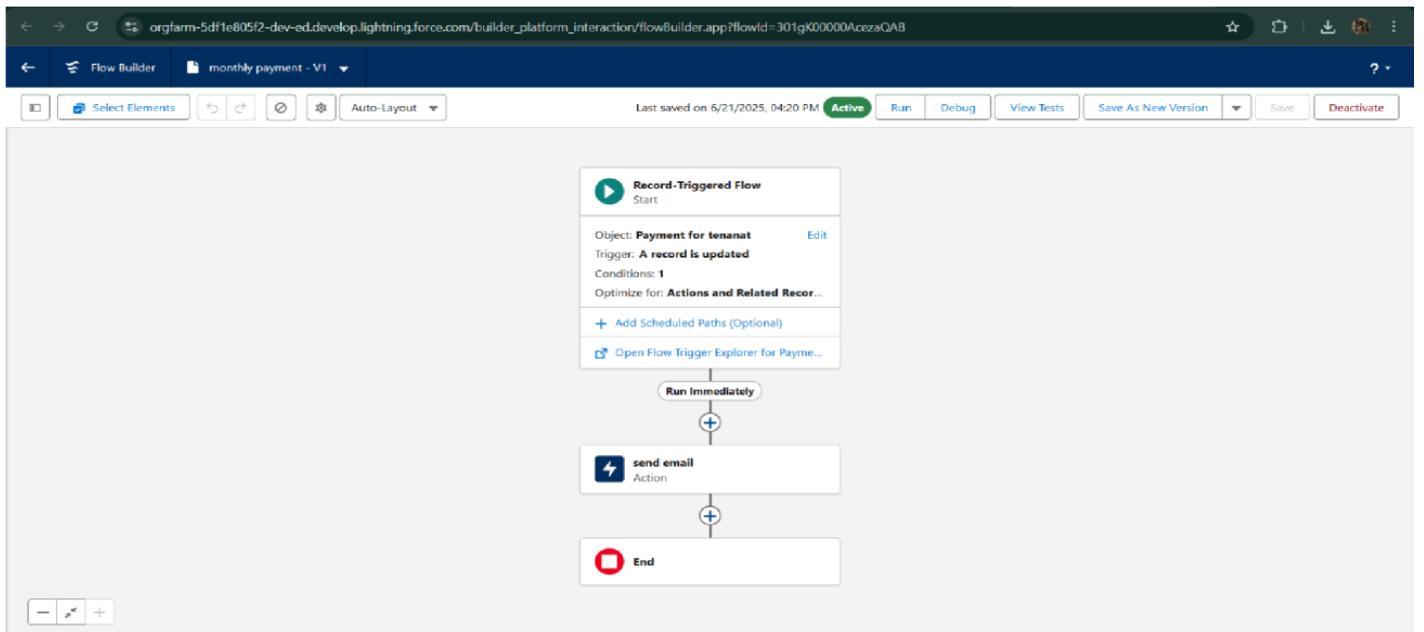
Leave approved



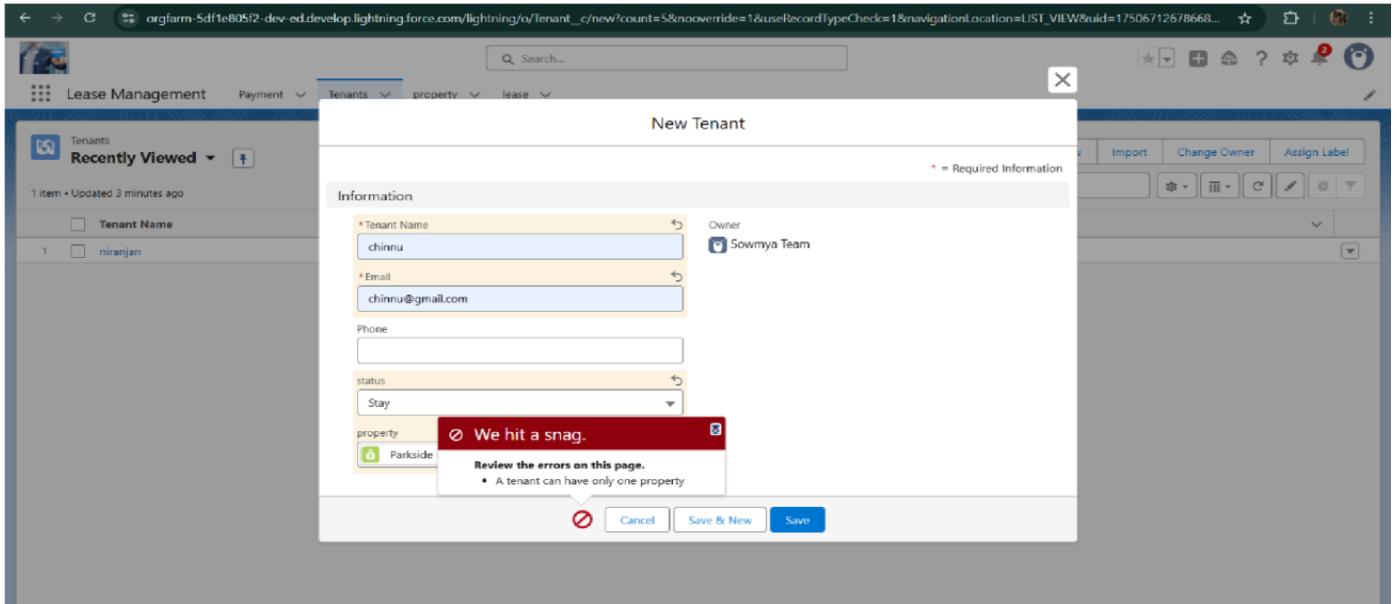
Leave rejected



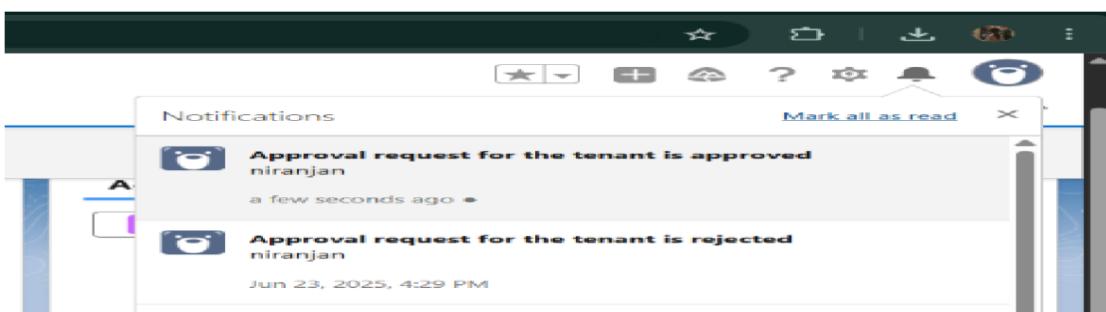
Flow runs



Trigger error messages



Approval process notifications



APPENDIX

Source Code: Provided in Apex Classes and Triggers

Test.apxt: trigger test on Tenant__c (beforeinsert)

```
{  
    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});

    }
}
```

RESULT

Achieved Operational Excellence: The automation of critical processes has eliminated manual errors and freed up valuable team time, allowing them to focus on strategic tasks rather than administrative tracking.

Assured Compliance and Mitigated Risk: The system acts as a safeguard against costly oversights by providing unwavering oversight of all financial obligations and critical deadlines, ensuring full adherence to lease accounting standards.

Data-Driven Decision Making: Executives and portfolio managers now have immediate access to accurate, real-time data and analytics, enabling them to optimize portfolio performance, negotiate better terms, and plan for the future with confidence.

Scalable Platform for Growth: The solution provides a flexible and scalable architecture that can easily accommodate portfolio growth, new property acquisitions, and evolving business requirements without requiring a fundamental overhaul.

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.