

# **LEASE MANAGEMENT**

**College Name: KALLAM HARANADHAREEDY INSTITUTE OF TECHNOLOGY**

**TEAM ID:** LTVIP2025TMID30830

**TEAM MEMBERS:**

**Team Leader: Harischandra vara prasad velaga Email:**

[Harivelaga8@gmail.com](mailto:Harivelaga8@gmail.com)

**Team Member: Areti chandrayas**

**Email: [Chandrayasareti14@gmail.com](mailto:Chandrayasareti14@gmail.com) Team**

**Member: Kodali murali krishna**

**Email: [Kodalimuralikrishna2@gmail.com](mailto:Kodalimuralikrishna2@gmail.com)**

**Team Member: [Pamulapati Ravindra](mailto:Pamulapati Ravindra)**

**Email:[228X1A4559@khitguntur.ac.in](mailto:228X1A4559@khitguntur.ac.in)**

# 1. INTRODUCTION

## 1.1 Project Overview

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



## 1.2 Purpose

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

## 2. IDEATION PHASE

### 2.1 Problem Statement

Managing lease information manually can lead to errors, missed payments, and communication delays. There is a need for a centralized, automated system to maintain lease records, track payments, and manage tenants efficiently.

### 2.2 Empathy Map Canvas

- **Think & Feel:** Needs confidence in lease management.
- **Hear:** Concerns from tenants about payment reminders.
- **See:** Missed lease renewals and tracking issues.
- **Say & Do:** Wants a smooth system to manage leases and communications.
- **Pain:** Manual tracking, miscommunication.
- **Gain:** Streamlined processes, reduced errors.

### 2.3 Brainstorming

Ideas included automated alerts, integrated payment tracking, custom objects for modular data, and use of approval processes to manage tenant transitions.

---

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

Tenants interact with the system by applying for leases, receiving emails for payment and status updates, and submitting leave requests. Admins handle object creation, approval processes, and maintain oversight through dashboards and flows.

### 3.2 Solution Requirement

- Salesforce platform
- Custom Objects: Property, Tenant, Lease, Payment
- Email Templates
- Apex Triggers & Classes

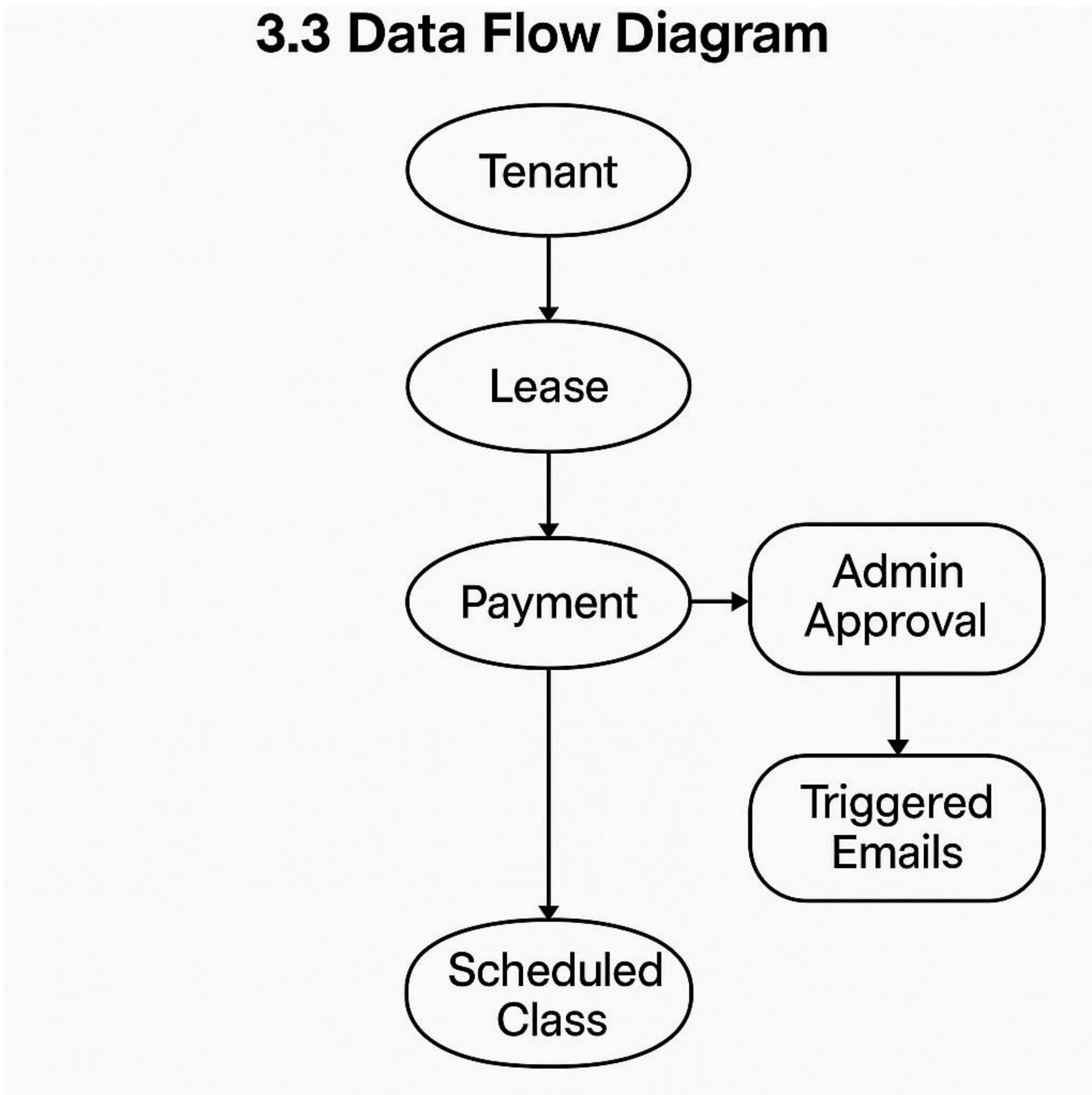
- Flow automation
- Approval Processes

### 3.3 Data Flow Diagram

Tenant → Lease → Payment

Admin approval → Triggered Emails

↳ ↳ Scheduled class → Monthly Reminders



## 3.4 Technology Stack

- **Platform:** Salesforce Lightning
  - **Language:** Apex
  - **Automation:** Salesforce Flow
  - **Database:** Salesforce Custom Objects
  - **Communication:** Classic Email Templates
- 

## 4. PROJECT DESIGN

### 4.1 Problem-Solution Fit

The system solves lease tracking, tenant management, and payment notification problems using Salesforce's automation and customization features.

### 4.2 Proposed Solution

Create custom Salesforce objects and implement workflows, email templates, and Apex triggers to automate all critical lease management functions.

### 4.3 Solution Architecture

- **Frontend:** Salesforce UI via Lightning Apps
  - **Backend:** Custom Objects and Apex
  - **Automation:** Flows, Validation Rules, Scheduled Jobs
  - **Integration:** Email alerts via templates
- 

## 5. PROJECT PLANNING & SCHEDULING

## **5.1 Project Planning**

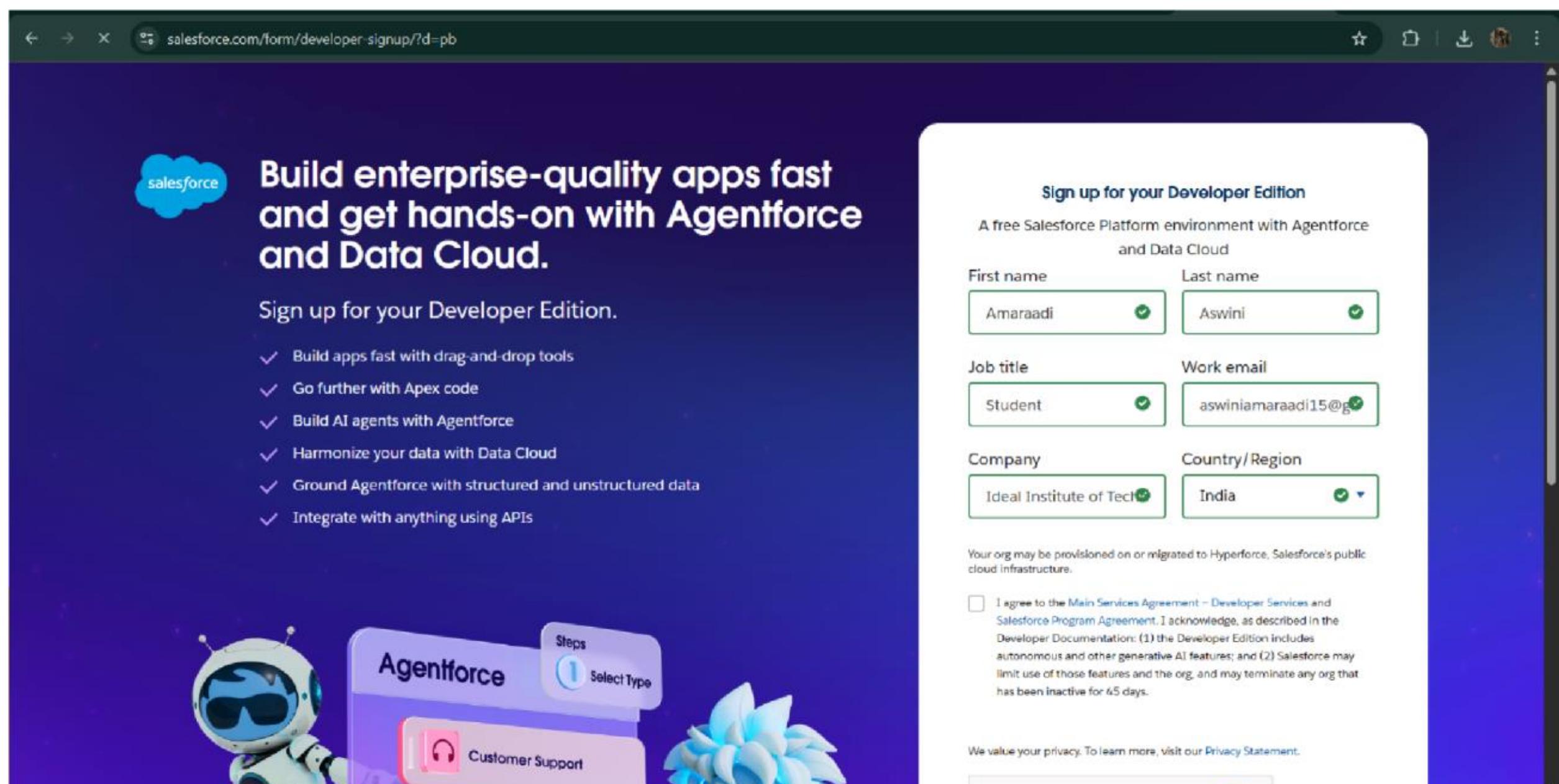
- Week 1: Object and Field Creation
- Week 2: Tab and App Setup
- Week 3: Flows and Email Templates

- Week 4: Apex Triggers & Approval Process
  - Week 5: Testing & Validation
- 

## 6.DEVELOPMENT PHASE

### Creating Developer Account:

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



The screenshot shows a web browser window with the URL [salesforce.com/form/developer-signup/?d=pb](https://www.salesforce.com/form/developer-signup/?d=pb) in the address bar. The page has a dark blue background with a central white sign-up form.

**Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.**

Sign up for your Developer Edition.

- ✓ 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

**Sign up for your Developer Edition**

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Amaraadi      Last name: Aswini

Job title: Student      Work email: aswiniamaraadi15@gmail.com

Company: Ideal Institute of Tech      Country/Region: India

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

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

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

- Created objects: Property, Tenant, Lease, Payment

SETUP > OBJECT MANAGER

### property

**Details**

Fields & Relationships	Description
Page Layouts	API Name
Lightning Record Pages	property_c
Buttons, Links, and Actions	Custom
Compact Layouts	Singular Label
Field Sets	property
Object Limits	Plural Label
Record Types	property
Related Lookup Filters	
Search Layouts	
List View Button Layout	
Restriction Rules	
Scoping Rules	

**Details**

Description	
API Name	property_c
Custom	✓
Singular Label	property
Plural Label	property
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

**Edit | Delete**

SETUP > OBJECT MANAGER

### Tenant

**Details**

Fields & Relationships	Description
Page Layouts	API Name
Lightning Record Pages	Tenant_c
Buttons, Links, and Actions	Custom
Compact Layouts	Singular Label
Field Sets	Tenant
Object Limits	Plural Label
Record Types	Tenants
Related Lookup Filters	
Search Layouts	
List View Button Layout	
Restriction Rules	
Scoping Rules	

**Details**

Description	
API Name	Tenant_c
Custom	✓
Singular Label	Tenant
Plural Label	Tenants
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

**Edit | Delete**

SETUP > OBJECT MANAGER

## lease

**Details**

Description

API Name: lease\_c

Custom: ✓

Singular Label: lease

Plural Label: lease

**Details**

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

**Setup** **Home** **Object Manager**

Search Setup

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'lease' object. The left sidebar lists various configuration options such as Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area displays the 'Details' tab for the 'lease' object. It includes fields for Description, API Name (lease\_c), Custom status (✓), Singular Label (lease), and Plural Label (lease). On the right, there are sections for Enable Reports, Track Activities, Track Field History, Deployment Status (set to Deployed), and Help Settings (Standard salesforce.com Help Window). Navigation links at the bottom include Setup, Home, and Object Manager, along with a search bar and standard browser controls.

SETUP > OBJECT MANAGER

## Payment for tenant

**Details**

Description

API Name: Payment\_for\_tenant\_c

Custom: ✓

Singular Label: Payment for tenant

Plural Label: Payment

**Details**

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

**Setup** **Home** **Object Manager**

Search Setup

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'Payment for tenant' object. The left sidebar lists various configuration options. The main content area displays the 'Details' tab for the 'Payment for tenant' object. It includes fields for Description, API Name (Payment\_for\_tenant\_c), Custom status (✓), Singular Label (Payment for tenant), and Plural Label (Payment). On the right, there are sections for Enable Reports, Track Activities, Track Field History, Deployment Status (set to Deployed), and Help Settings (Standard salesforce.com Help Window). Navigation links at the bottom include Setup, Home, and Object Manager, along with a search bar and standard browser controls.

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area displays a table titled 'Fields & Relationships' with 9 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User,Group)		✓
property	property__c	Lookup(property)		✓
property Name	Name	Text(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

The screenshot shows the Salesforce Object Manager interface for the 'Payment for tenant' object. The left sidebar lists various setup options. The main area displays a table titled 'Fields & Relationships' with 7 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

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 > 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

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

### 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

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

- Developed Lightning App with relevant tabs

The screenshot shows the Lightning App Builder interface with the 'Lease Management' app open. The top navigation bar includes 'Lightning App Builder', 'App Settings', 'Pages', and 'Lease Management'. The left sidebar lists 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main content area is titled 'App Details & Branding' and contains sections for 'App Details' and 'App Branding'. In 'App Details', fields include 'App Name' (Lease Management), 'Developer Name' (Lease\_Management), and 'Description' (Application to efficiently handle the processes related to leasing real estate properties.). In 'App Branding', there is an 'Image' section with a preview of a logo featuring a hand and geometric shapes, and a 'Primary Color Hex Value' set to #0070D2. An 'Org Theme Options' checkbox is also present.

The screenshot shows the Lightning App Builder interface with the 'Navigation Items' tab selected. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'Navigation Items' and includes sections for 'Available Items' and 'Selected Items'. The 'Available Items' list contains various Salesforce objects like Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, and more. The 'Selected Items' list contains four items: 'Payment' (with a green icon), 'Tenants' (with a blue icon), 'property' (with a green icon), and 'lease' (with a red icon). Navigation arrows between the two lists allow users to move items between them.

Lightning App Builder | App Settings | Pages | Lease Management | Help

**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

Lease creation (0)

Lease Management | Payment | Tenants | property | lease | Search... | ⚙️ ? 🔍

**Payment**

Recently Viewed ▾

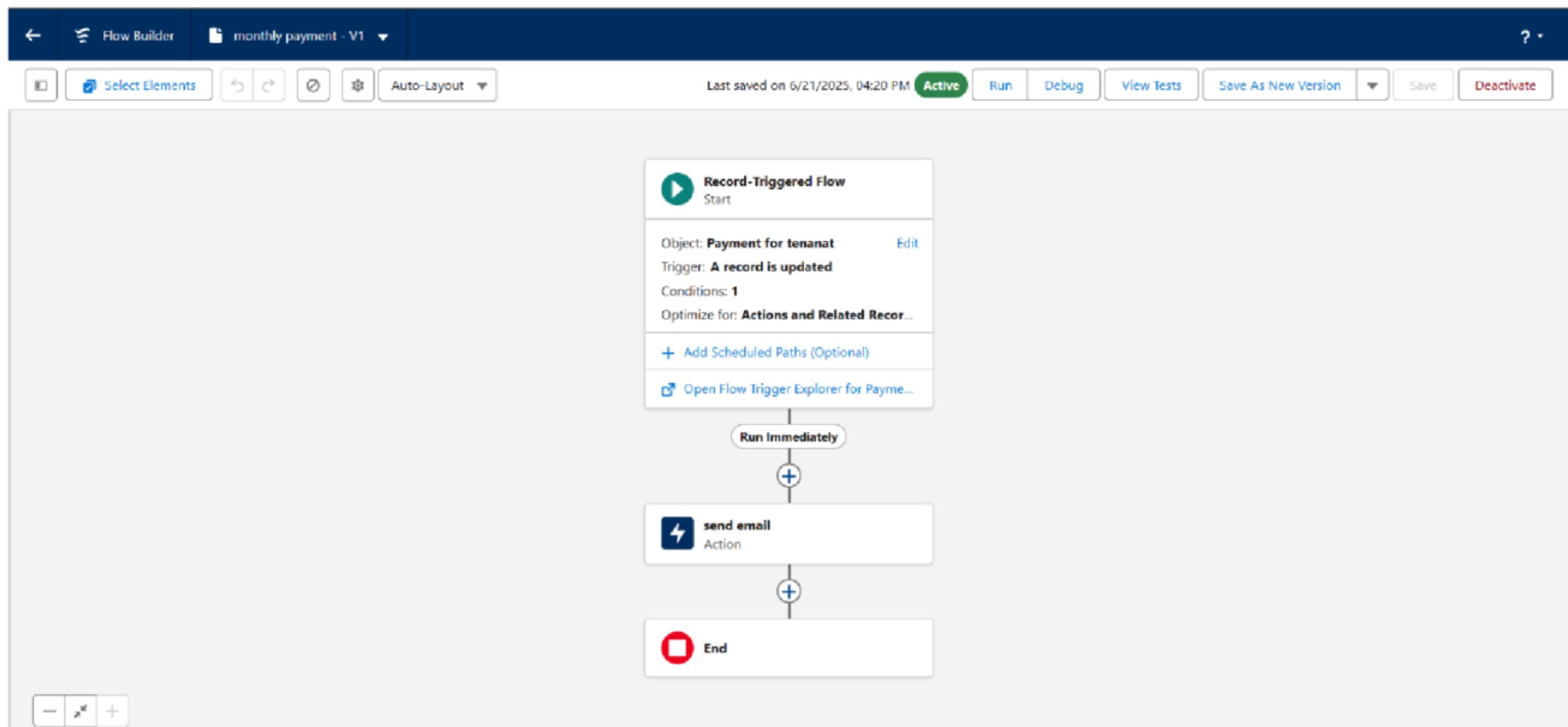
5 items • Updated a few seconds ago

	<input type="checkbox"/> Payment Name	<input type="checkbox"/>
1	<input type="checkbox"/> Rahul	<input type="checkbox"/>
2	<input type="checkbox"/> Jack	<input type="checkbox"/>
3	<input type="checkbox"/> Raj	<input type="checkbox"/>
4	<input type="checkbox"/> Sam	<input type="checkbox"/>
5	<input type="checkbox"/> Lahari	<input type="checkbox"/>

New Import Change Owner Assign Label

Search this list... ⚙️ 🔍

- 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 in the Object Manager.

- Validation Rule Edit**
- Rule Name:** lease\_end\_date
- Active:** checked
- Description:** (empty)
- Error Condition Formula:**

```
End date c <= start date d
```

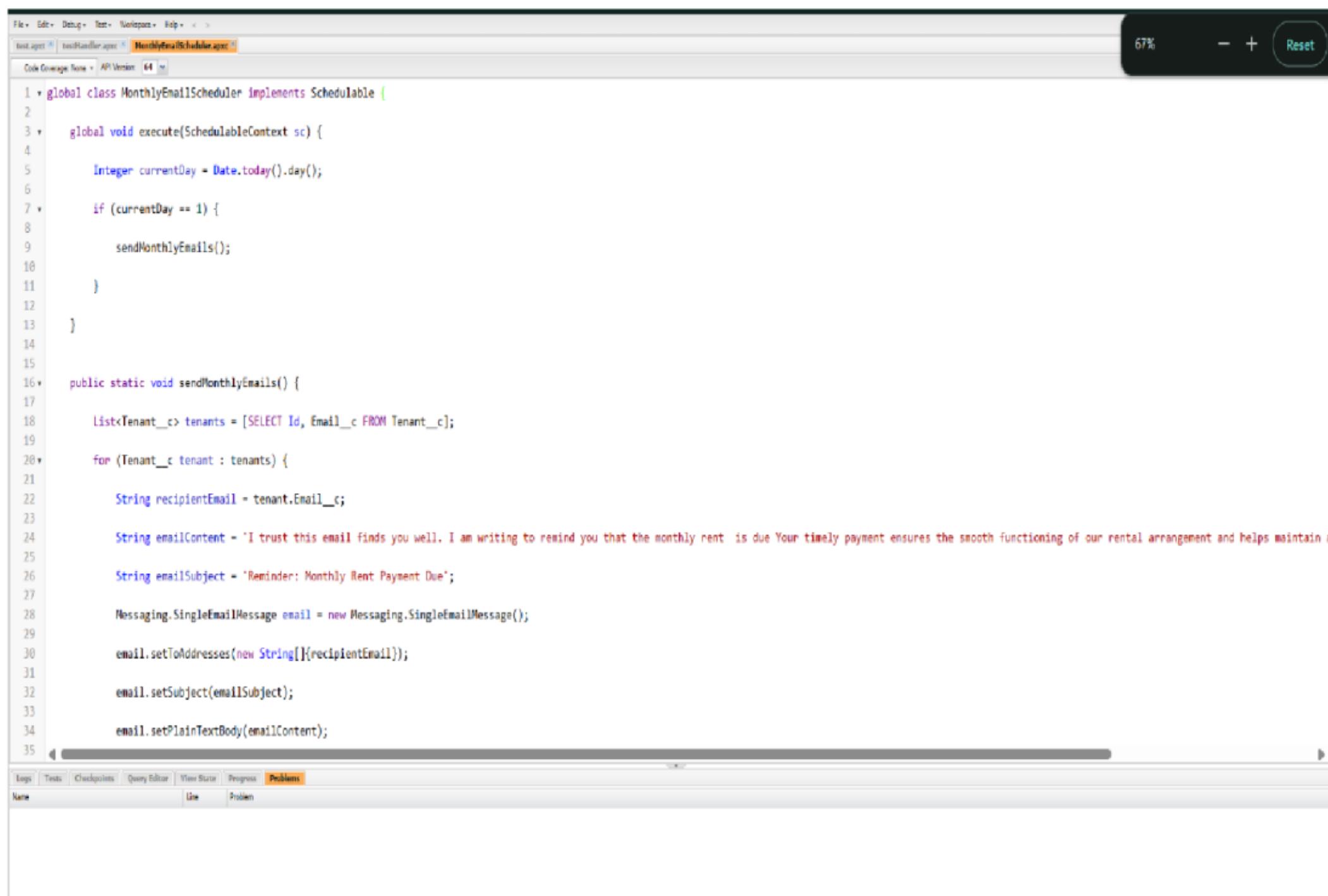
Example: `Discount_Percent_c > 30` More Examples...  
If this formula expression is true, display the text defined in the Error Message area
- Functions:**
  - ABS
  - ACOS
  - ADDMONTHS
  - AND
  - ASCII
  - ASIN

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Validation Rules. The main content area displays the 'Validation Rule Detail' for a rule named 'lease\_end\_date'. The rule checks if the end date is less than or equal to the start date. It has an error message: 'Your End date must be greater than start date'. The rule is active and was created by 'Sowmya Team' on 6/19/2025 at 5:37 AM, last modified by 'Sowmya Team' on 6/26/2025 at 7:47 AM.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the 'Lease Management' application's 'Tenants' screen. A modal window titled 'New Tenant' is open, showing fields for Tenant Name (chinnu), Email (chinnu@gmail.com), Phone, status (Stay), and property (Parkside). A validation error message box appears, stating: 'We hit a snag.' and 'Review the errors on this page.' with the note: '• A tenant can have only one property'. The 'Save' button is disabled due to the error.

- Scheduled monthly reminder emails using Apex class

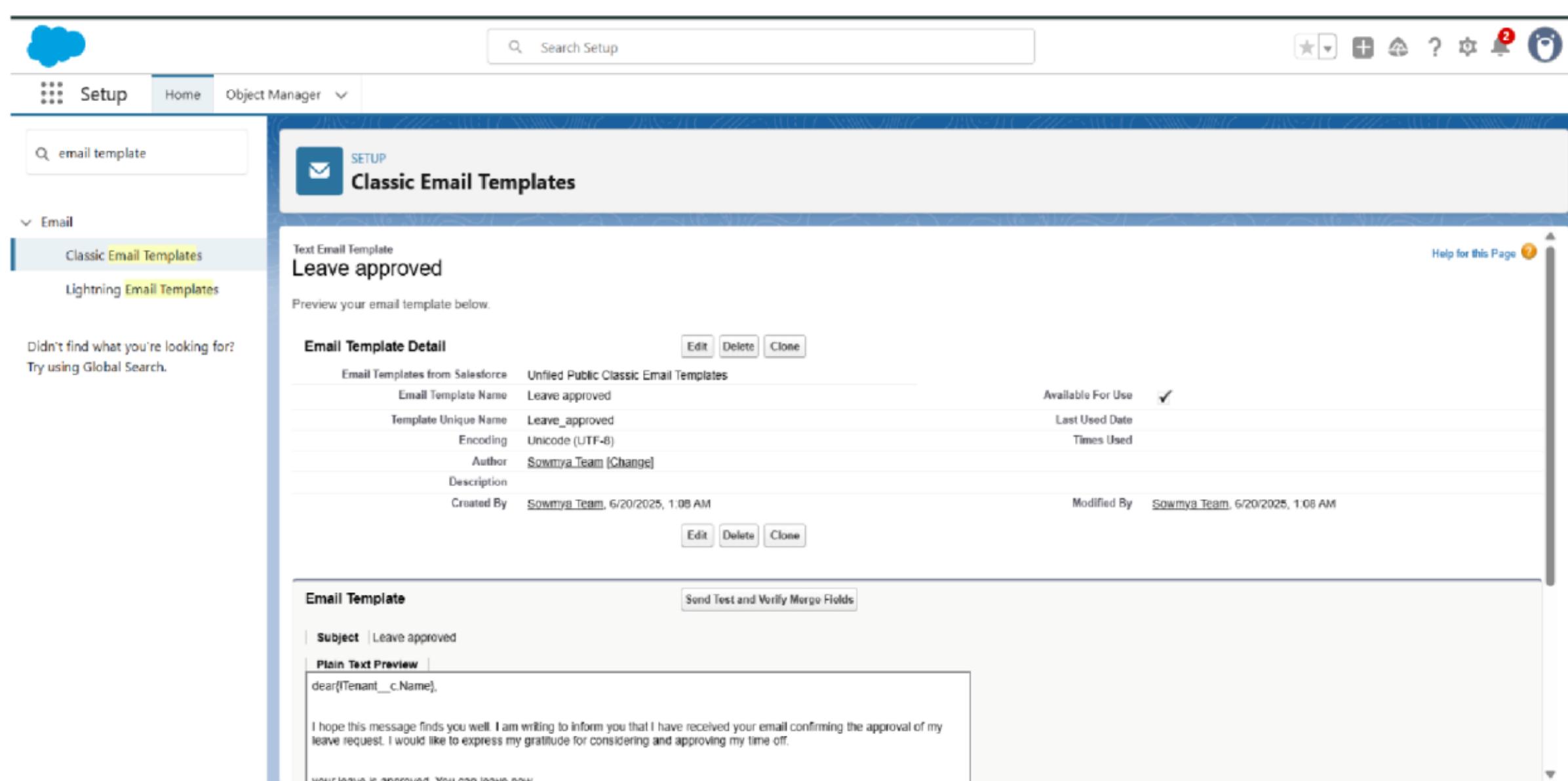


```

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 a positive relationship. Thank you for your cooperation.'; // Truncated
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setToAddresses(new String[]{recipientEmail});
31
32         email.setSubject(emailSubject);
33
34         email.setPlainTextBody(emailContent);
35

```

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



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

- Page Header:** Search bar with "Search Setup" placeholder.
- Top Navigation:** Icons for Home, Object Manager, and a bell notification with a red badge.
- Left Sidebar:**
  - Section: Email
  - Sub-section: Classic Email Templates (highlighted)
  - Sub-section: Lightning Email Templates
  - Text: Didn't find what you're looking for? Try using Global Search.
- Main Content Area:**

### SETUP Classic Email Templates

**Leave approved**

Text Email Template

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unfiled 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
<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Clone</a>	

**Email Template**

**Subject:** Leave approved

**Plain Text Preview:**

```

dear{Tenant__c.Name}.

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

Your leave is approved. You can leave now.

```

The screenshot shows the Salesforce Setup interface with the search bar set to "email template". The left sidebar under "Email" is expanded, showing "Classic Email Templates" selected. The main content area displays the "Classic Email Templates" page for a "Text Email Template" named "tenant leaving".

**Email Template Detail**

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 [Change]
Description	
Created By	Sowmya Team, 6/20/2025, 1:06 AM
Modified By	Sowmya Team, 6/20/2025, 1:06 AM

**Email Template**

Send Test and Verify Merge Fields

**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 search bar set to "email template". The left sidebar under "Email" is expanded, showing "Classic Email Templates" selected. The main content area displays the "Classic Email Templates" page for a "Text Email Template" named "Leave rejected".

**Email Template Detail**

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team, 6/20/2025, 1:11 AM
Modified By	Sowmya Team, 6/20/2025, 1:11 AM

**Email Template**

Send Test and Verify Merge Fields

**Subject:** Leave rejected

**Plain Text Preview:**

```
Dear {!Tenant__c.Name}.

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

**Email Template Detail**

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

**Email Template**

**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.
```

**Email Template Detail**

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Change]
Description	
Created By	Sowmya Team 6/20/2025, 1:13 AM
Modified By	Sowmya Team 6/20/2025, 1:13 AM

**Email Template**

**Subject:** Confirmation of Successful Monthly Payment

**Plain Text Preview:**

```
Dear {!Tenant__c.Email__c}.

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.
```

- Approval Process creation

For Tenant Leaving:

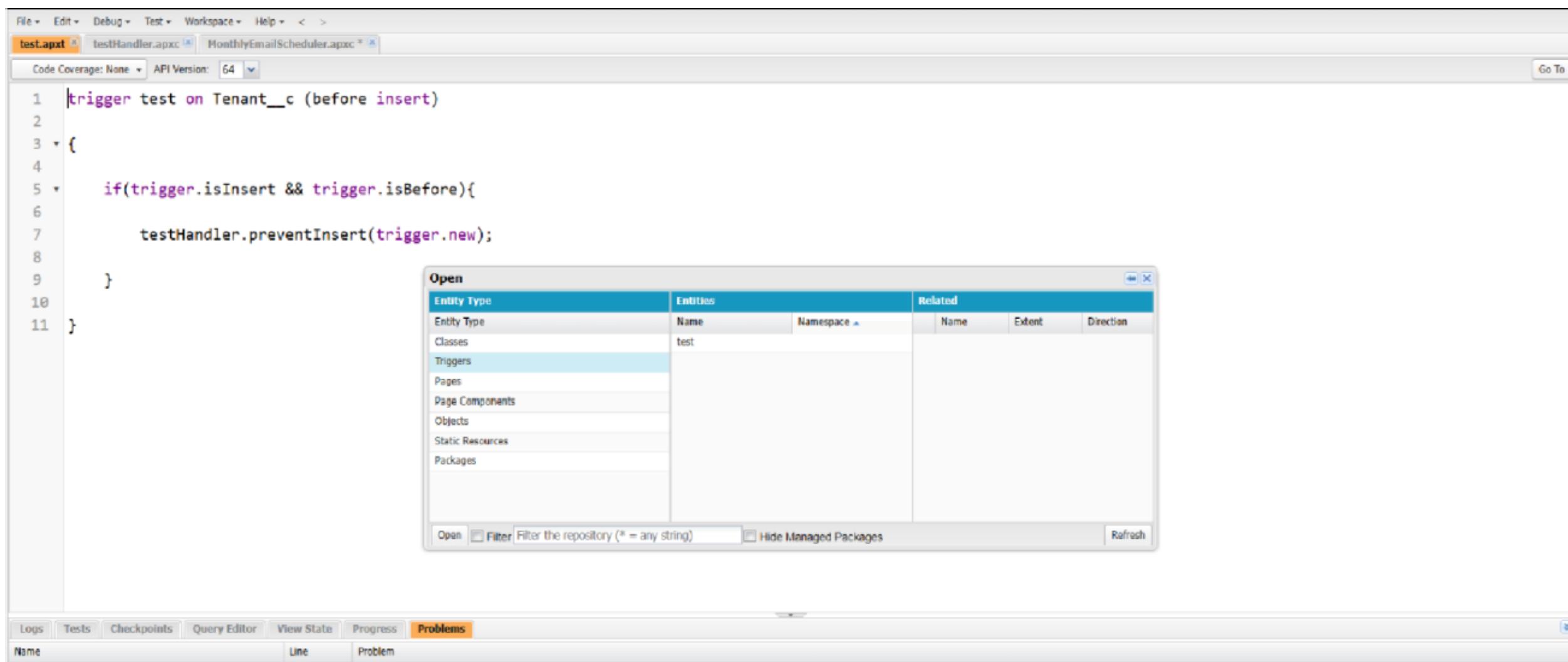
The screenshot shows the Salesforce Setup interface for 'Approval Processes'. The process is named 'TenantApproval' with a unique name of 'TenantApproval'. The entry criteria is set to 'Tenant: status EQUALS Stay'. The record editability is restricted to 'Administrator ONLY'. The approval assignment email template is 'Initial Submitters' and the initial submitter is 'Tenant Owner'. The process was created by 'Sowmya Team' on 6/23/2025 at 3:41 AM and modified by 'Sowmya Team' on 6/26/2025 at 11:57 PM. There is one initial submission action: 'Record Lock' which locks the record from being edited. The approval step is labeled 'Step 1' with the name 'Step 1'.

### For Check for Vacant:

The screenshot shows the Salesforce Setup interface for a new approval process named 'check for vacant' with a unique name of 'check\_for\_vacant'. The entry criteria is set to 'Tenant: status EQUALS Leaving'. The record editability is restricted to 'Administrator ONLY'. The approval assignment email template is 'Leave approved' and the initial submitter is 'Tenant Owner'. The process was created by 'Sowmya Team' on 6/20/2025 at 3:18 AM and modified by 'Sowmya Team' on 6/26/2025 at 11:02 PM. There are two initial submission actions: 'Record Lock' (locks the record from being edited) and 'Email Alert' (with the message 'please approve my leave'). The approval step is labeled 'Step 1' with the name 'step1'.

- Apex Trigger

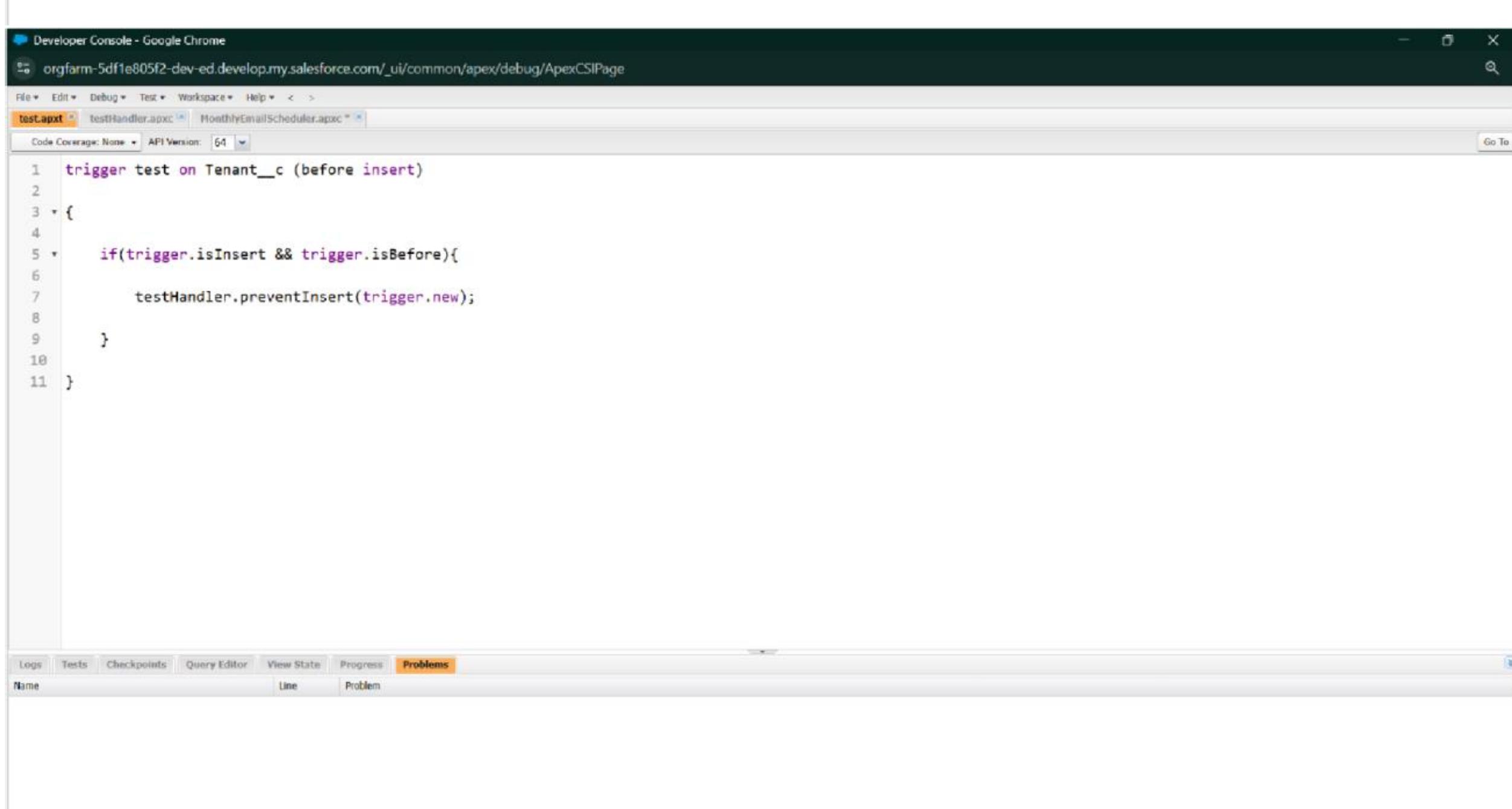
Create an Apex Trigger



The screenshot shows the Salesforce IDE interface. In the top-left corner, there are tabs for 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below these are three specific tabs: 'test.apx' (highlighted in orange), 'testHandler.apxc', and 'MonthlyEmailScheduler.apxc'. A status bar at the bottom indicates 'Code Coverage: None', 'API Version: 64', and a 'Go To' button. The main area contains the following Apex trigger code:

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

A modal dialog titled 'Open' is displayed, listing various entity types: Entity Type (Classes, Triggers, Pages, Page Components, Objects, Static Resources, Packages), Entities (Name, Namespace), and Related (Name, Extent, Direction). The 'Triggers' item is selected. A search bar at the bottom of the dialog says 'Filter the repository (\* = any string)' and a 'Refresh' button is also present.

The screenshot shows the Salesforce Developer Console in a Google Chrome browser window. The URL is 'orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage'. The interface is identical to the IDE, with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems' (which is currently selected). The 'Problems' section is empty. The trigger code is identical to the one shown in the IDE.

Create an Apex Handler class

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex testHandler.apxc MonthlyEmailScheduler.apxc

Code Coverage: None API Version: 64 Go To

```

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

```

Open Entity Type Entity Use Namespace Related

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

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex testHandler.apxc MonthlyEmailScheduler.apxc

Code Coverage: None API Version: 64 Go To

```

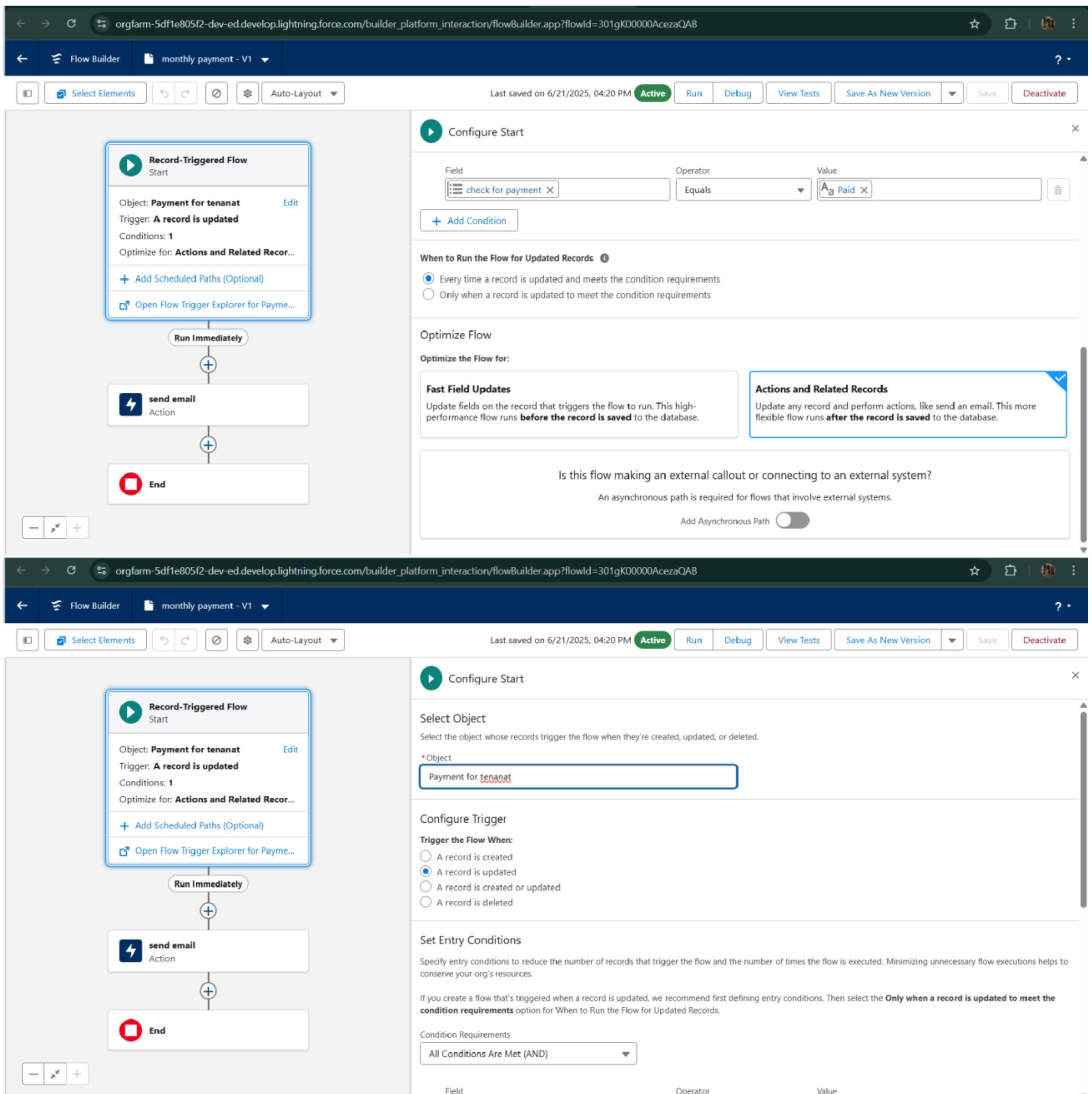
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

- FLOWS



## ● Schedule class:

Create an Apex Class

Developer Console - Google Chrome

orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apex testHandler.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
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

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Open Entity Type Entities Related

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

Open Filter Hide Managed Packages Refresh

Developer Console - Google Chrome

orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apex testHandler.apex MonthlyEmailScheduler.apex

Code Coverage: None API Version: 64 Reset

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10
11     }
12
13 }
14
15
16 * public static void sendMonthlyEmails() {
17
18     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20     for (Tenant__c tenant : tenants) {
21
22         String recipientEmail = tenant.Email__c;
23
24         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setToAddresses(new String[]{recipientEmail});
31
32         email.setSubject(emailSubject);
33
34         email.setPlainTextBody(emailContent);
35
36         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38     }
39
40 }
41
42

```

Logs Tests Checkpoints Query Editor View State Progress Problems

## Schedule Apex class

Screenshot of the Salesforce Setup Apex Classes page:

**Apex Class Detail**

**MonthlyEmailScheduler**

**Class Body**

```
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            // Implementation details...  
12        }  
13    }  
14}  
15  
16
```

**Apex Class Detail Fields:**

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

Screenshot of the Salesforce Lease Management Tenant record detail page:

**Tenant Aswini**

**Details**

**Related**

**Activity**

**Upcoming & Overdue**

No activities to show.

Get started by sending an email, scheduling a task, and more.

**Filters:** All time • All act

**Buttons:** New Case, Edit, New Opportunity, Delete, Clone, Change Owner, Printable View, Submit for Approval, Edit Labels

**Fields:**

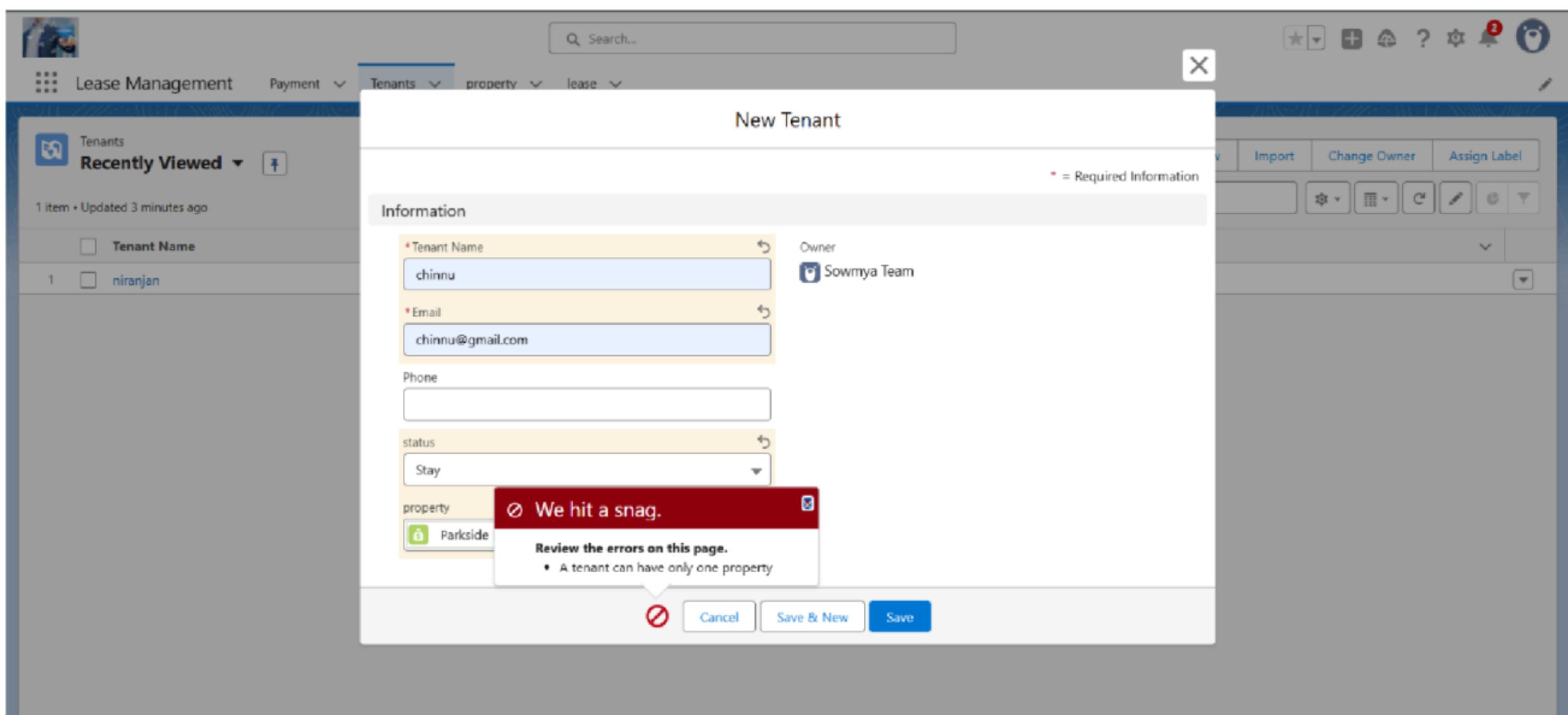
- \*Tenant Name: Aswini
- \*Email: aswiniamaraadi15@gmail.com
- Phone: (905) 223-5567
- status: Leaving
- property: Imran
- Created By: Sowmya Team, 6/26/2025, 6:05 AM
- Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

The screenshot shows a Salesforce Lightning page for a tenant named Aswini. The top navigation bar includes tabs for Lease Management, Payment, Tenants, property, and lease. A green banner at the top right indicates "Tenant was submitted for approval." The main form contains fields for Tenant Name (Aswini), Email (aswiniamaraadi15@gmail.com), Phone ((905) 223-5567), status (Leaving), and property (Imran). The owner is listed as Sowmya Team. The "Details" tab is selected. At the bottom, there are "Cancel" and "Save" buttons. To the right, an "Activity" sidebar shows no upcoming or overdue activities. The system status bar at the bottom right shows it's 12:46 on June 27, 2025.

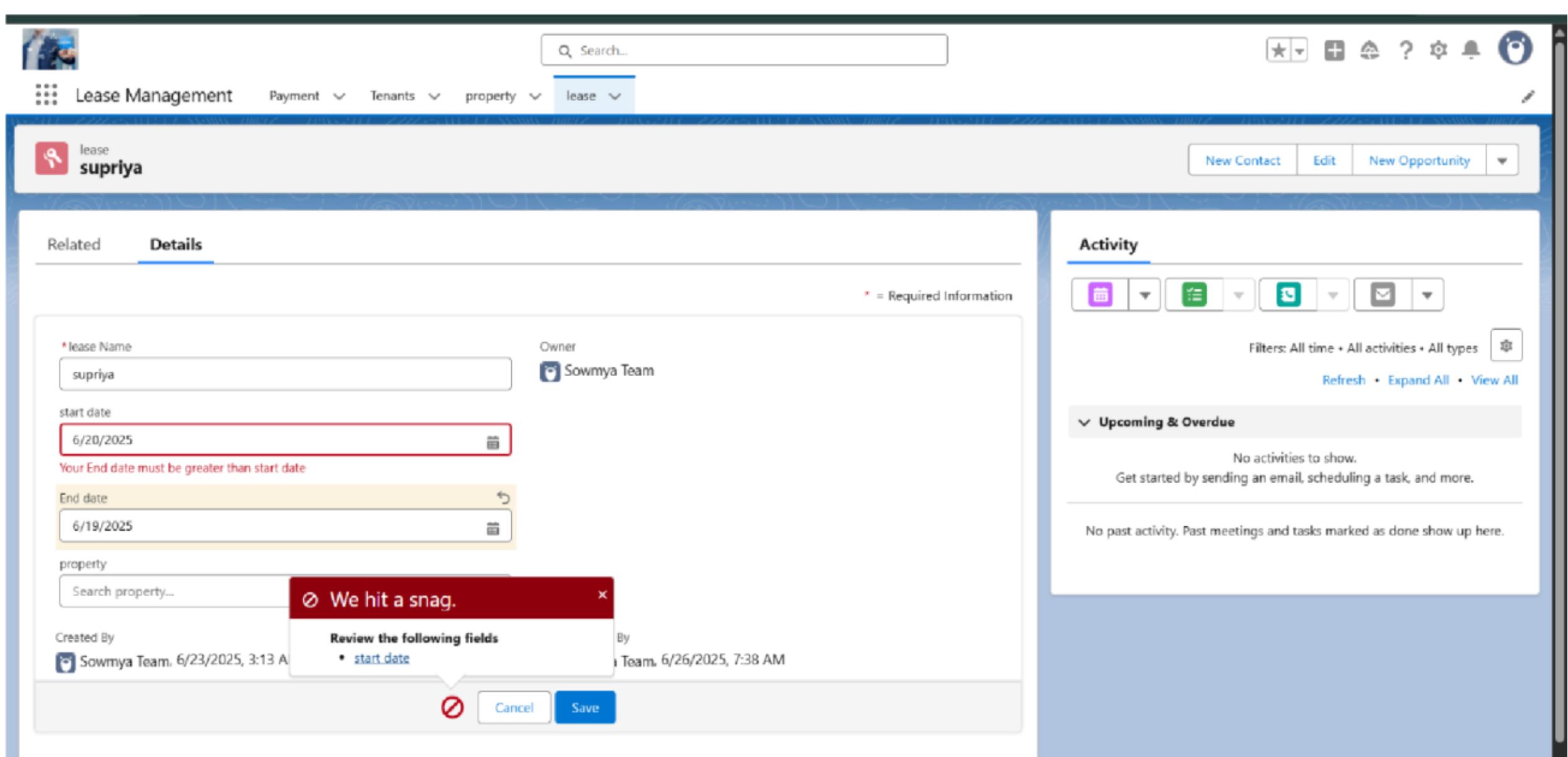
The screenshot shows a Salesforce Lightning page for a process instance step named "Tenant Approval". The status is "Approved". The details section shows the submitter as Sowmya Team, the date submitted as Jun 27, 2025, the actual approver as Sowmya Team, and the assigned to field also as Sowmya Team. The "Details" tab is selected. On the right, a "Notifications" sidebar lists several notifications from Aswini and Kiran, all related to tenant approvals. The notifications are: "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), and "Approval request for the tenant is approved Kiran" (19 hours ago).

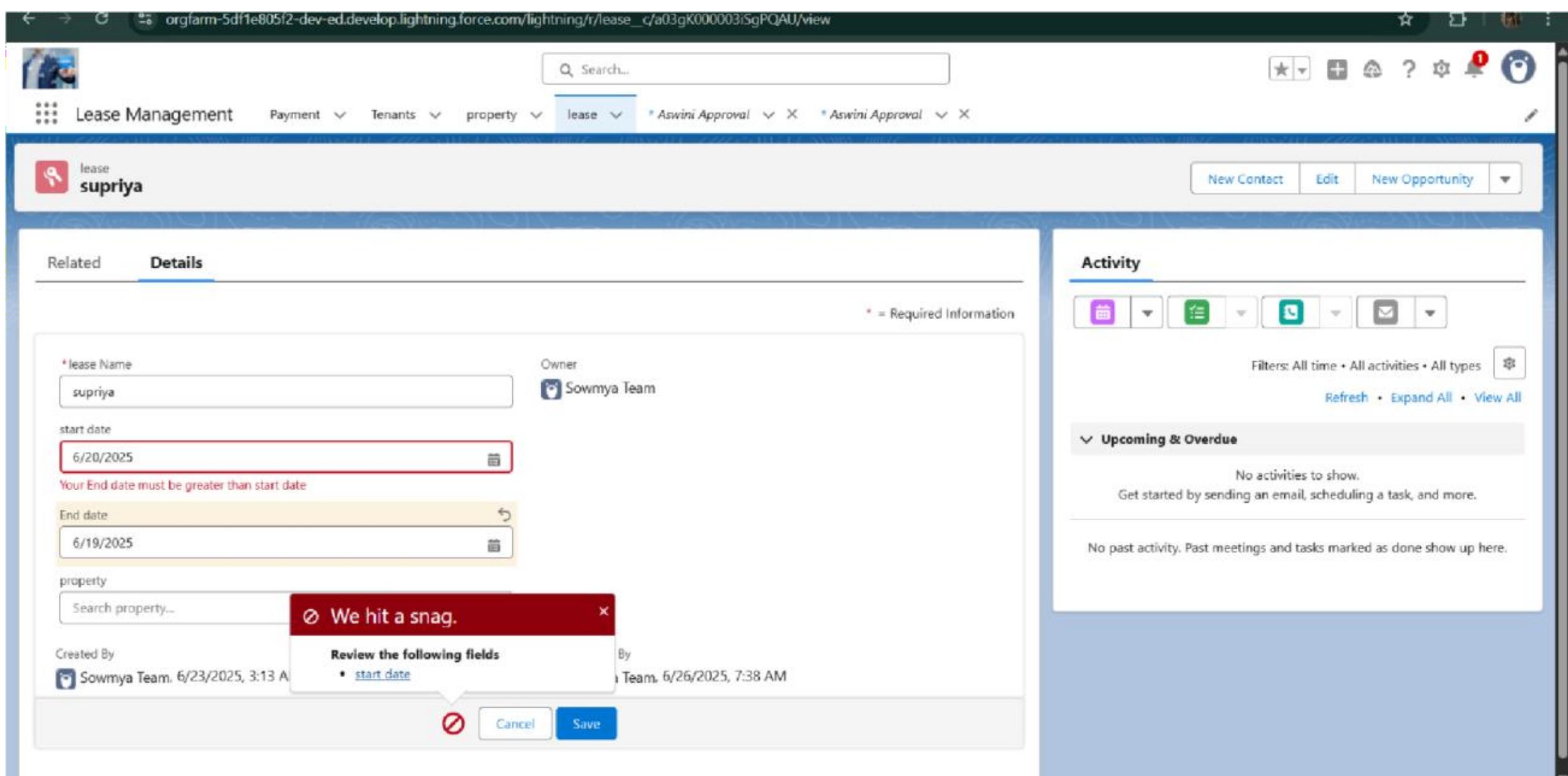
# 7. FUNCTIONAL AND PERFORMANCE TESTING

## 7.1 Performance Testing

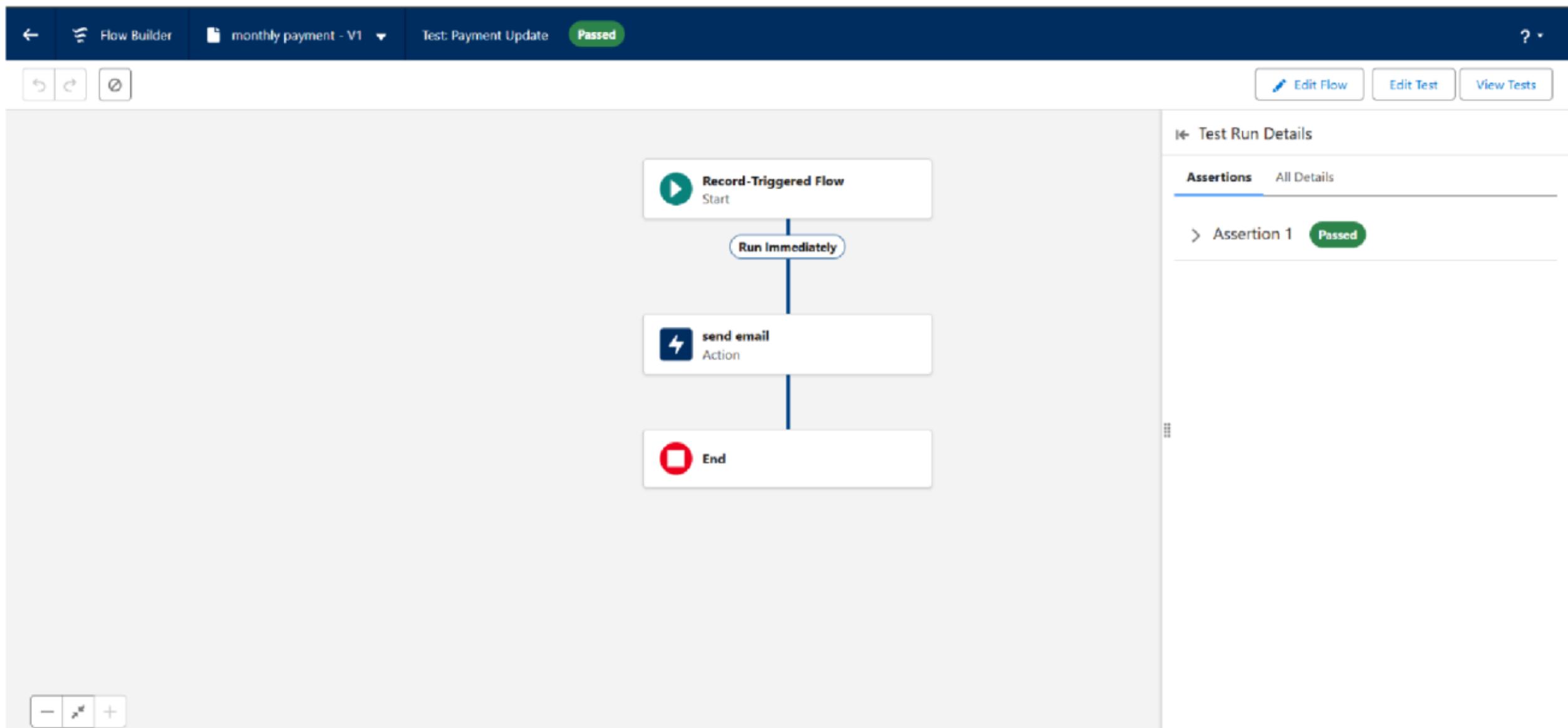


- Trigger validation by entering duplicate tenant-property records
- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the Microsoft Dynamics 365 interface. On the left, the 'Lease Management' ribbon is visible with 'Tenants' selected. The main area displays the 'Tenant Details' for 'niranjan'. The 'Details' tab is active, showing fields such as 'Tenant Name' (niranjan), 'Email' (niranjan1506@gmail.com), 'Phone' (empty), 'status' (Stay), 'property' (Parkside Lofts), 'Created By' (Sowmya Team, 6/23/2025, 2:33 AM), and 'Last Modified By' (Sowmya Team, 6/23/2025, 3:58 AM). A 'Save' button is at the bottom right. On the right, the 'Notifications' sidebar is open, showing several notifications: 'Approval request for the tenant is approved niranjan' (a few seconds ago), 'Approval request for the tenant is rejected niranjan' (Jun 23, 2025, 4:29 PM), 'Approval request for the tenant is approved niranjan' (Jun 23, 2025, 4:25 PM), 'Approval request for the tenant is approved niranjan' (Jun 23, 2025, 4:14 PM), and a general message about 'New Guidance Center learning resource available' (Jun 20, 2025, 1:28 PM).

The screenshot shows the Microsoft Dynamics 365 interface. The 'Lease Management' ribbon is visible with 'Tenants' selected. The main area displays the 'Approval History' section for 'niranjan', showing a list of steps with details like date, status, and assigned to. The steps include 'Step 1' (Approved, 6/25/2025, 5:39 AM), 'Approval Request Submitted' (Submitted, 6/25/2025, 5:39 AM), 'Step 1' (Rejected, 6/23/2025, 3:59 AM), 'Approval Request Submitted' (Submitted, 6/23/2025, 3:58 AM), 'Step 1' (Approved, 6/23/2025, 3:55 AM), and 'Approval Request Submitted' (Submitted, 6/23/2025, 3:55 AM). Below this is a 'Payment' section with two entries: 'Jack' and 'Rahul'. A 'New' button is at the top right of the payment section. To the right, there is a sidebar with buttons for 'New Contact', 'Edit', and 'New Opportunity', and a message stating '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.'

# 8.RESULTS

## 8.1 Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. On the left, there's a sidebar with 'User Interface' and 'Tabs' selected. The main area displays 'Custom Object Tabs' with four entries:

Action	Label	Tab Style	Description
Edit   Del	lease	Keys	
Edit   Del	Payment	Credit card	
Edit   Del	property	Sack	
Edit   Del	Tenants	Map	

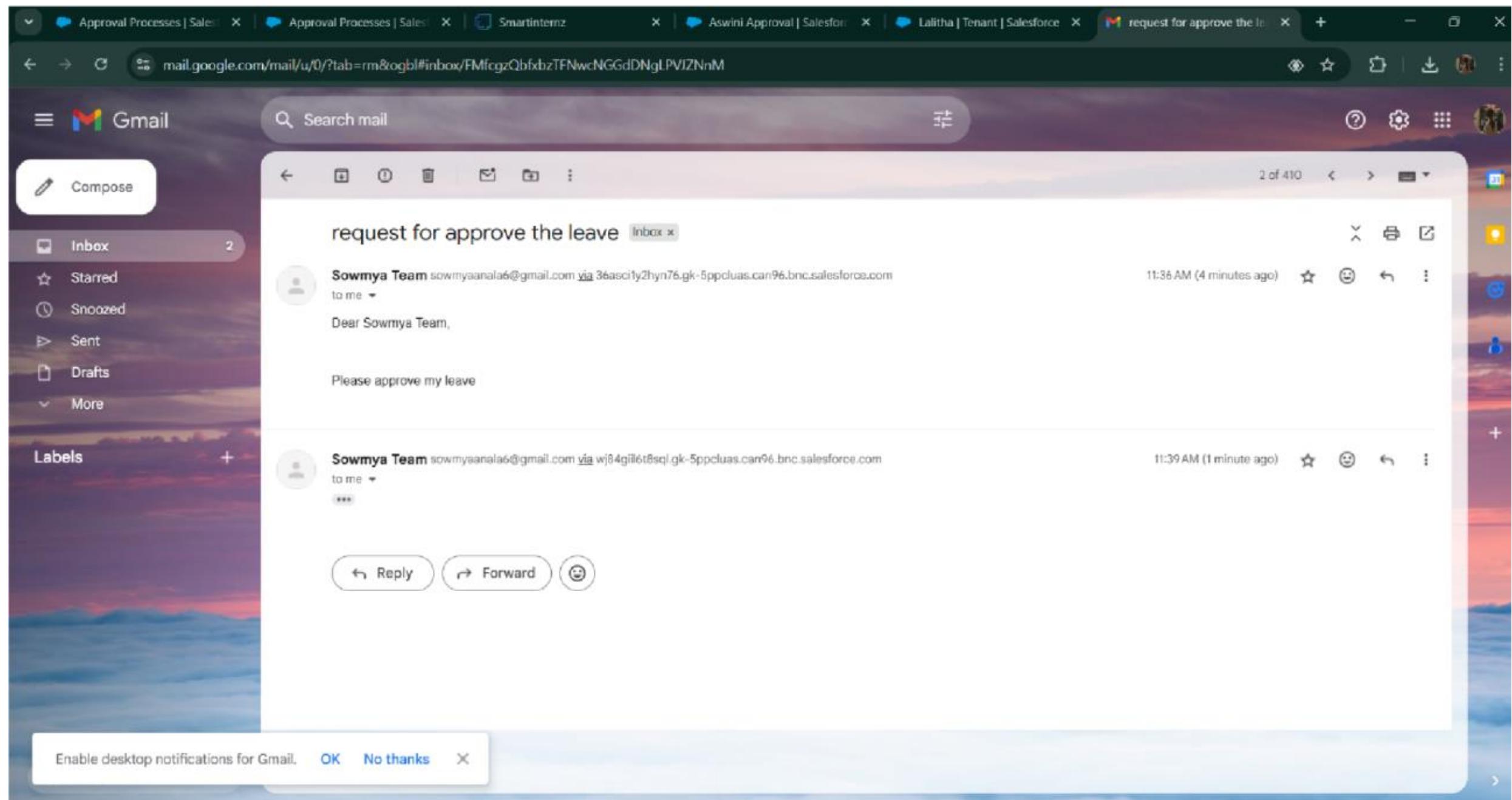
Below this are sections for 'Web Tabs' (No Web Tabs have been defined) and 'Visualforce Tabs' (No Visualforce Tabs have been defined).

- Email alerts

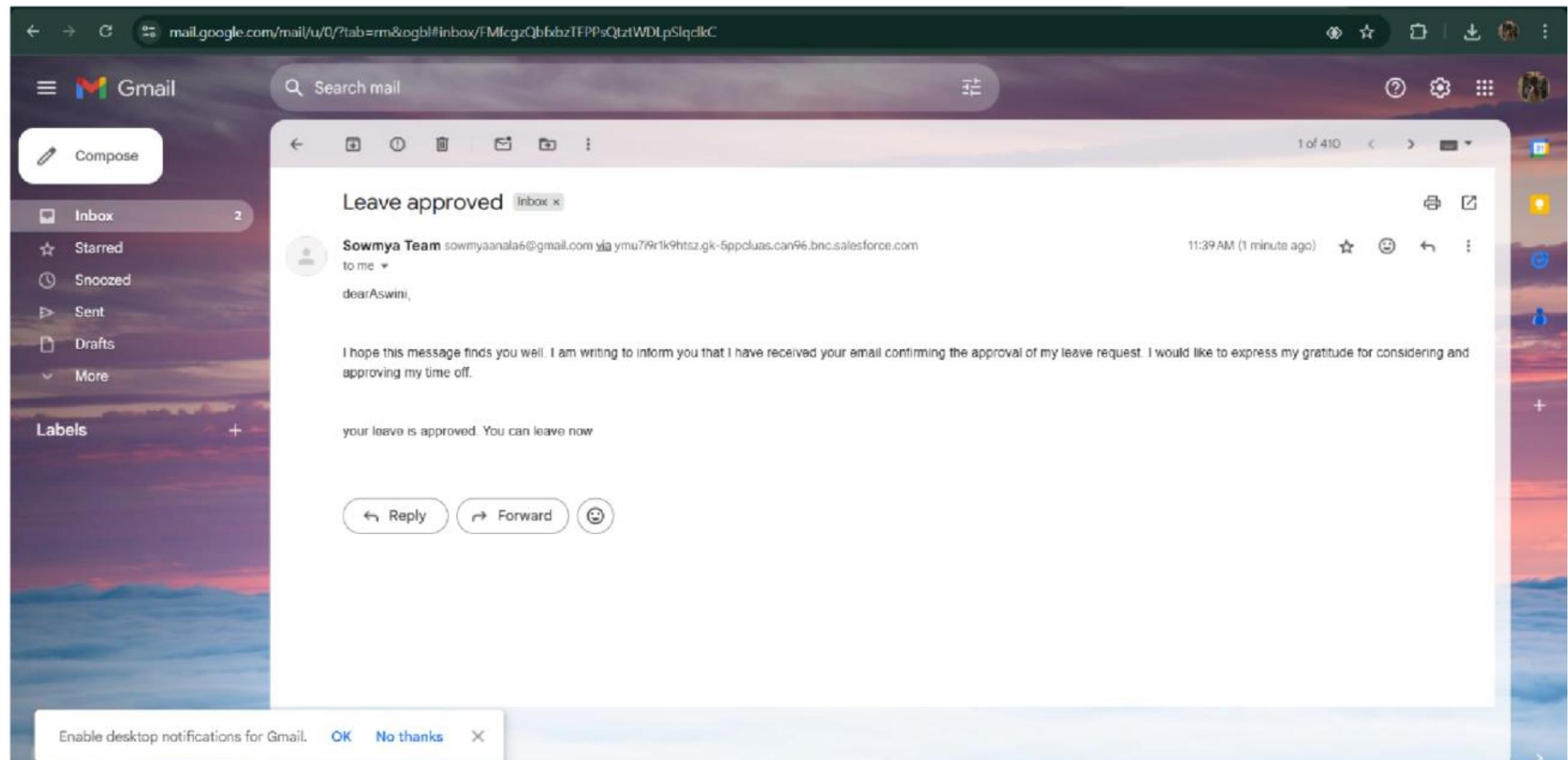
The screenshot shows the 'Approval History' page for a specific tenant. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. The main content area is titled 'Approval History' and shows a table of 8 items:

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
3 Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
4 Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team	Sowmya Team	Leaving
5 Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team	Sowmya Team	Approved
6 Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team	Sowmya Team	leaving
7 Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
8 Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	Leaving

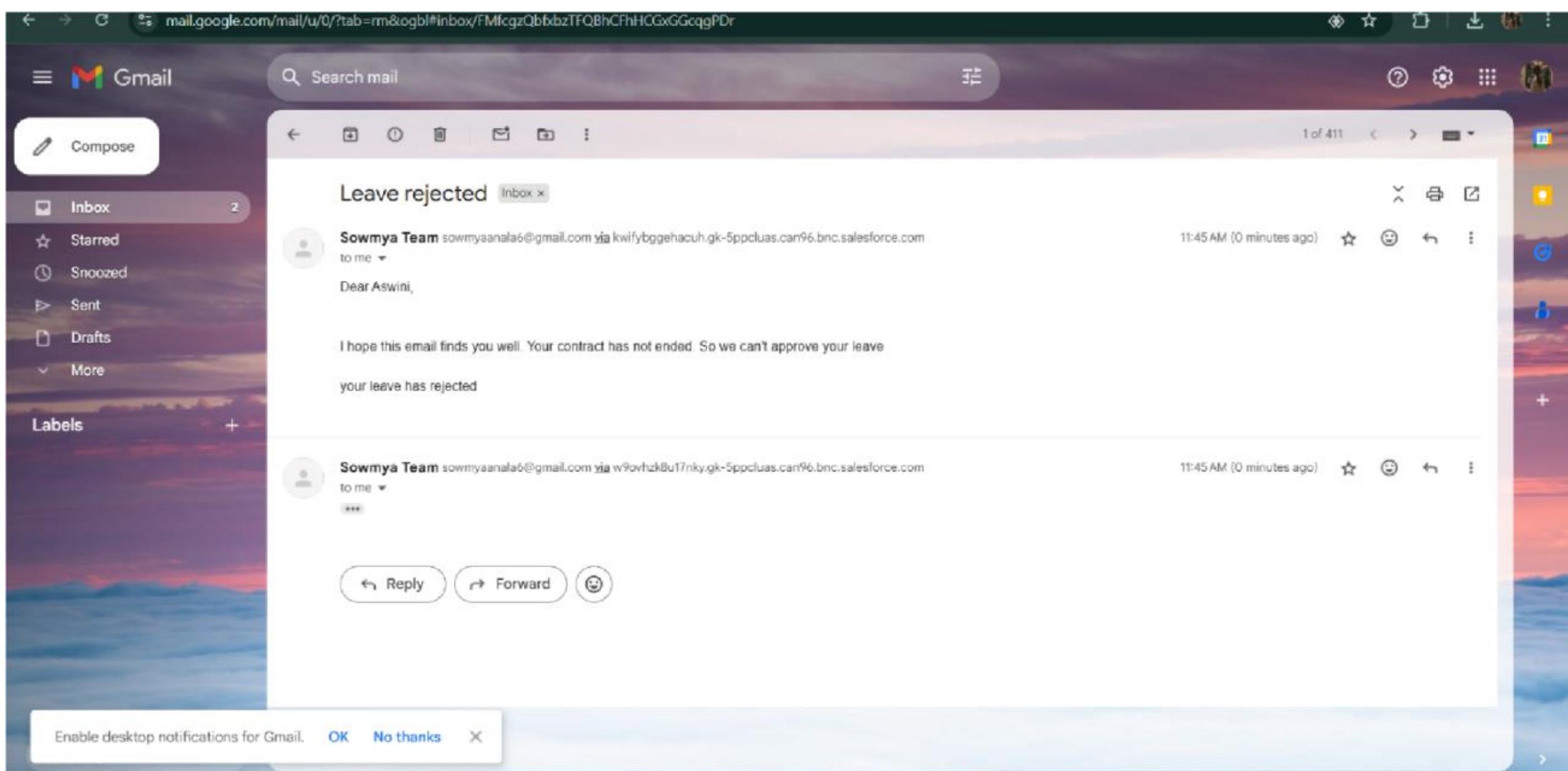
- Request for approve the leave



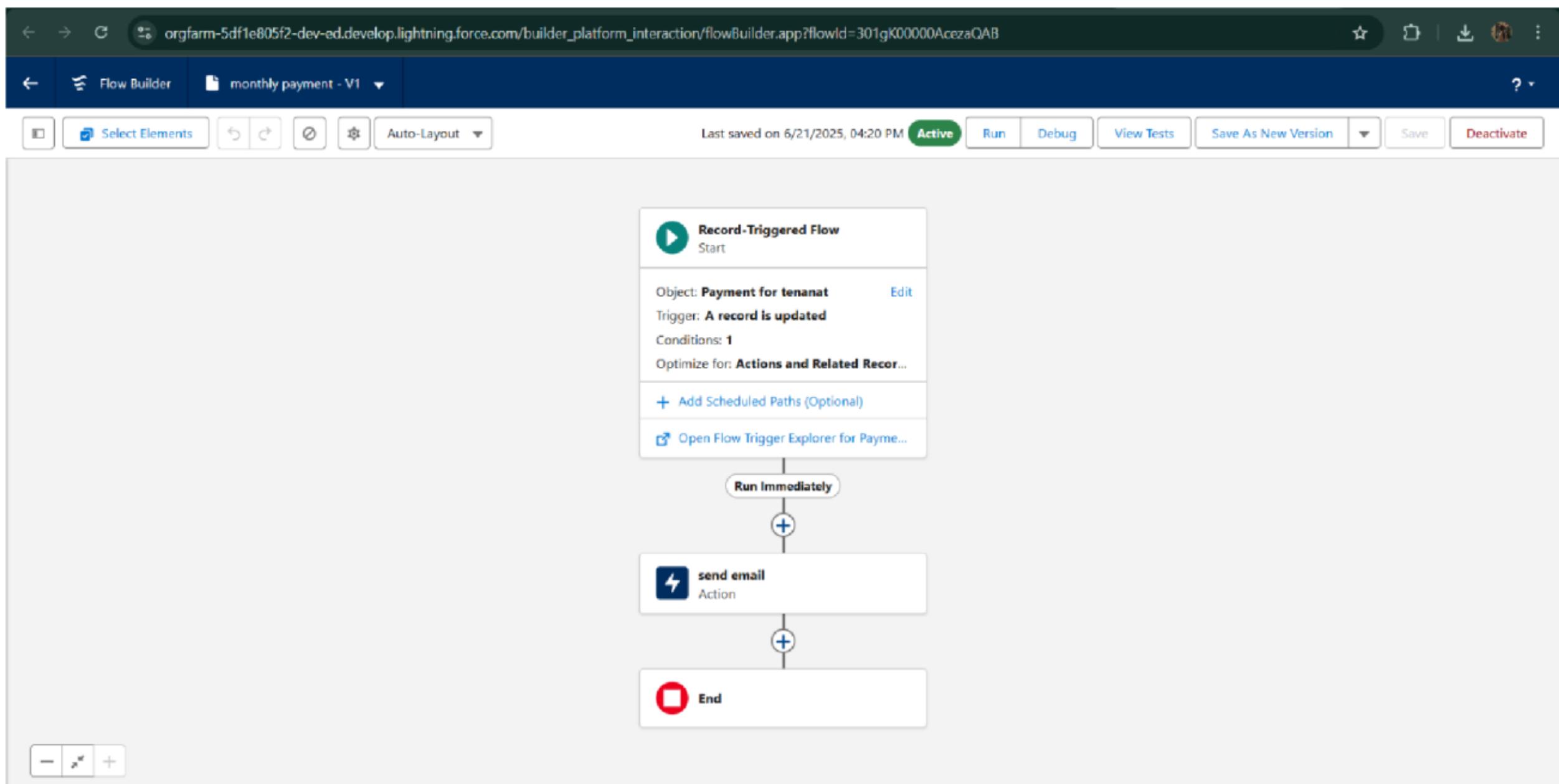
- Leave approved



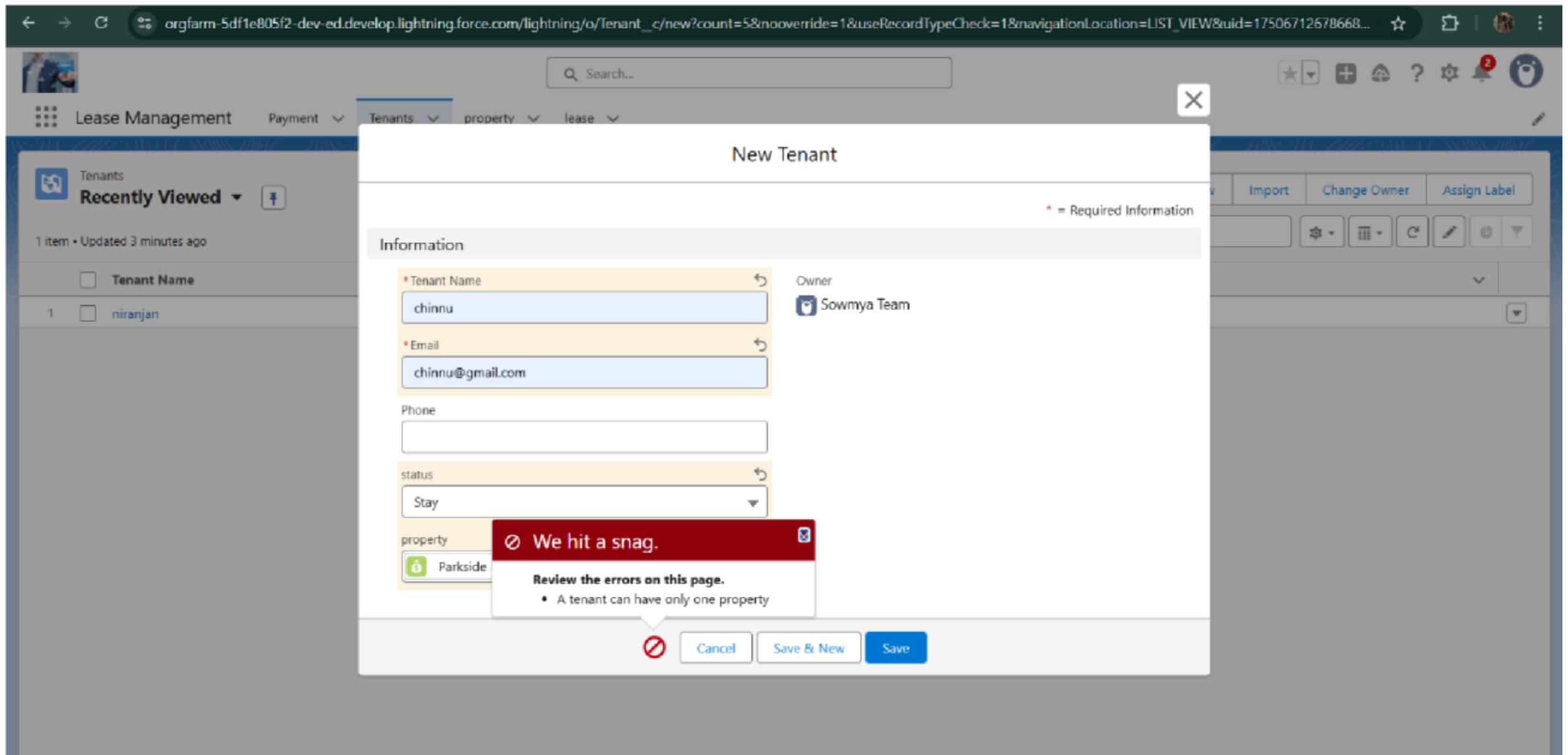
- Leave rejected



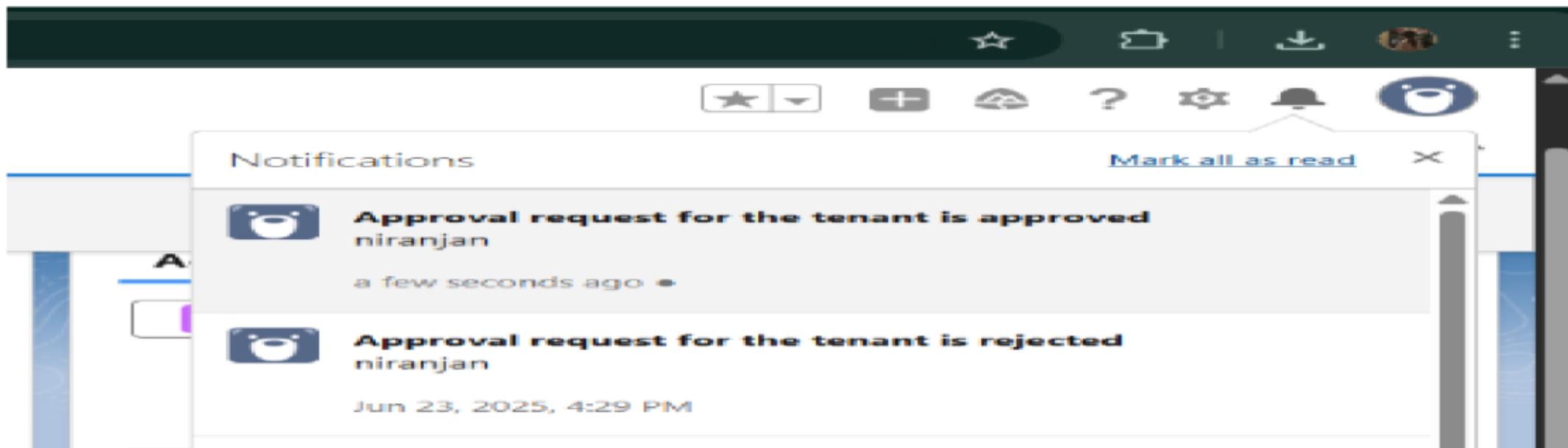
## ● Flow runs



## ● Trigger error messages



- Approval process notifications



## 9. ADVANTAGES & DISADVANTAGES

### Advantages:

#### 1. Centralized lease data

All lease agreements, documents, and tenant information are stored in one system, improving organization and accessibility.

#### 2. Automated lease tracking

System-generated reminders for lease renewals, rent due dates, and compliance deadlines reduce missed tasks and penalties.

### **3. Improved reporting and analytics**

Real-time dashboards and reports provide insights into occupancy, lease expirations, and revenue, supporting better decision-making.

### **4. Enhanced tenant communication**

Automated emails and notifications streamline communication with tenants about payments, renewals, or maintenance updates.

### **5. Audit readiness and compliance**

Accurate records and change tracking help ensure compliance with lease accounting standards and audit requirements.

---

## **Disadvantages:**

### **1. Initial setup complexity**

Implementing a lease management system can be time-consuming, requiring detailed data migration and configuration.

### **2. User training required**

Property managers and staff may need training to fully understand and use the system effectively.

### **3. High implementation costs**

Custom development, integration, and licensing fees can be expensive, especially for smaller firms.

### **4. Dependence on digital access**

Lease data is typically cloud-based, so system access requires internet connectivity and appropriate user permissions.

### **5. Data security concerns**

Storing sensitive lease and financial data digitally introduces the risk of unauthorized access or data breaches if not properly secured.

---

## **10. 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.

---

# 11. FUTURE SCOPE

## Future Scope of Lease Management System

### 1. Integration with Online Payment Gateways

Integrating secure online payment gateways (like Stripe, Razorpay, or PayPal) will allow tenants to pay rent, deposits, and other charges directly through the system. This ensures faster transactions, automated receipts, and real-time payment tracking—reducing manual reconciliation for finance teams.

### 2. Visual Dashboards for Better Insights

Advanced dashboards using tools like Salesforce Reports or third-party analytics platforms can provide real-time insights into lease status, occupancy rates, rent collection, pending renewals, and more. These visualizations help property managers make data-driven decisions and forecast trends effectively.

### 3. SMS Notification Features

Alongside emails, SMS alerts can notify tenants and staff about payment due dates, lease renewals, maintenance schedules, or compliance reminders. SMS communication ensures broader reach and immediate attention, especially when tenants are offline or prefer mobile updates.

### 4. Role-Based Access and Analytics

Implementing detailed role-based permissions allows specific users (e.g., admin, property manager, tenant) to access only relevant features and data. Coupling this with role-specific analytics helps each user view KPIs tailored to their responsibilities, improving operational efficiency and security.

### 5. Mobile App Support

A dedicated mobile app or mobile-responsive interface can empower tenants and managers to manage leases, payments, documents, and communications on the go. This improves accessibility and user engagement.

### 6. AI-Powered Predictive Maintenance

Future versions could use AI to analyse maintenance patterns and predict when certain property assets (like HVAC or plumbing systems) may require servicing—helping reduce downtime and costs.

### 7. E-signature and Document Automation

Integration with e-signature platforms (e.g., DocuSign, Adobe Sign) will allow digital signing of lease agreements and auto-generation of documents, streamlining onboarding and renewals.

## **8. Compliance and Legal Tracking**

Future features could include automated compliance tracking based on local real estate laws, lease accounting standards (like IFRS 16), and tax implications—minimizing legal risks.

## **9. IoT Integration for Smart Leasing**

Long-term, the system can integrate with IoT devices (e.g., smart locks, energy meters) for automated check-ins/out, usage-based billing, and enhanced security or environmental monitoring.

## **10. Multi-property and Franchise Support**

As property portfolios grow, the system can evolve to support multiple properties across regions, including different currencies, tax rules, and regulatory compliance—making it suitable for franchises or large real estate chains.

---

# **12. 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});  
  
}  
}  
}
```