

# LEAST MANAGEMENT

**College Name:** Sri Ramalinga Sowdambigai College of Science and Commerce

**College Code:** bru3y

**TEAM ID:** NM2025TMID25501

**TEAM MEMBERS:** 4

**Team LeaderName:** Meenatchisundaram S

**Email:** meenatchis40@gmail.com

**Team Member1:** Manikandan C

**Email:** manikandanmani35847@gmail.com

**Team Member:** Ramesh G

**Email:** ramesh0929rk@gmail.com

**Team Member:** Saravanan S

**Email:** saravanansaro915@gmail.com

# 1. INTRODUCTION

## 1.1 Project Overview

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



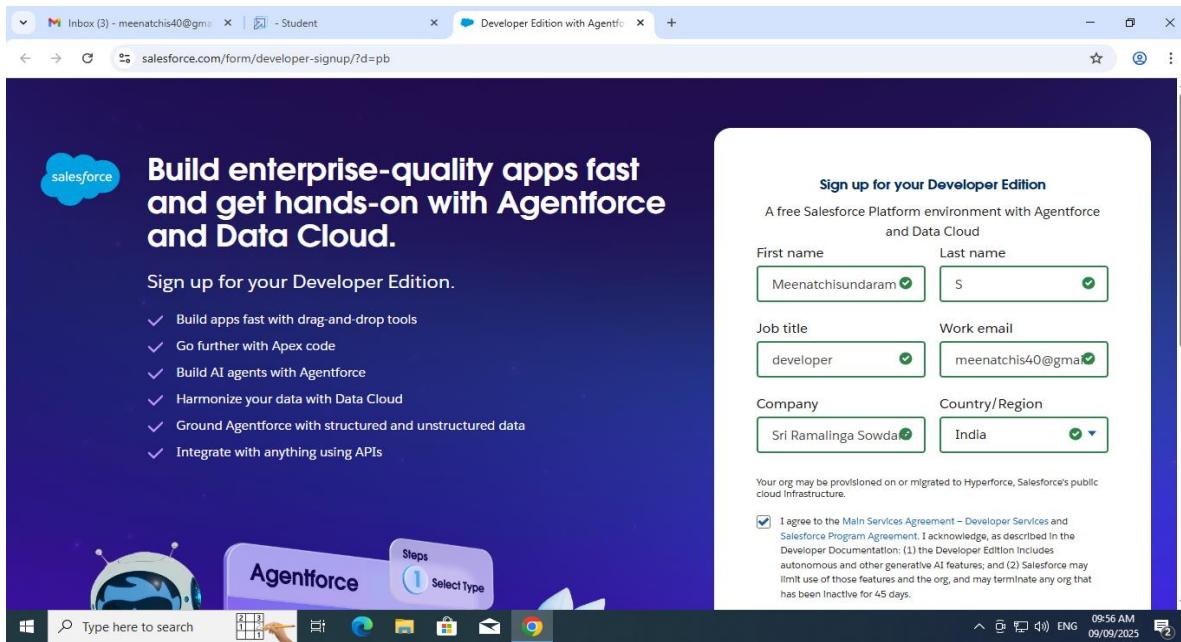
## 1.2 Purpose

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

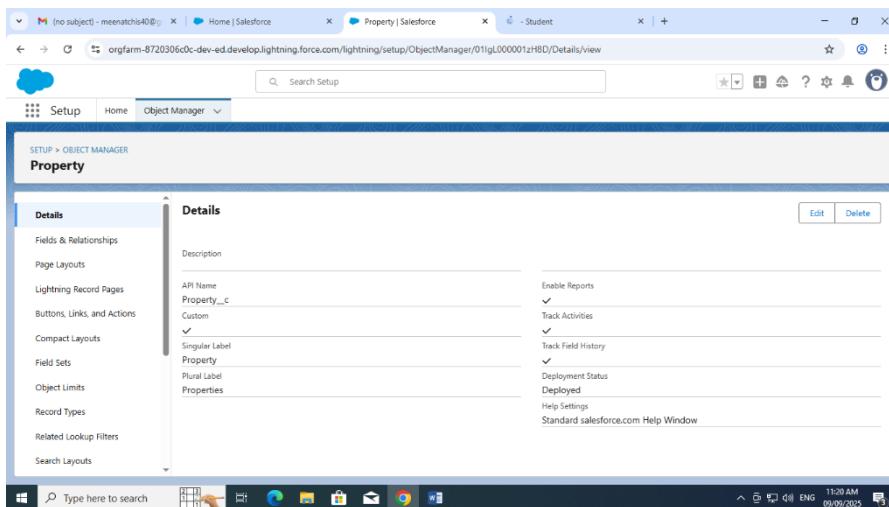
# DEVELOPMENT PHASE

## Creating Developer Account:

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



- Created objects: Property, Tenant, Lease, Payment



Screenshot of the Salesforce Object Manager for the Tenant object.

**Object Details:**

- API Name:** Tenant\_c
- Custom:** ✓
- Singular Label:** Tenant
- Plural Label:** Tenants

**Object Settings:**

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

**Navigation:** SETUP > OBJECT MANAGER  
Tenant

**Left Sidebar:**

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts

**System Information:** Type here to search, Windows taskbar, 11:23 AM 09/09/2025

Screenshot of the Salesforce Object Manager for the lease object.

**Object Details:**

- API Name:** lease\_c
- Custom:** ✓
- Singular Label:** lease
- Plural Label:** lease

**Object Settings:**

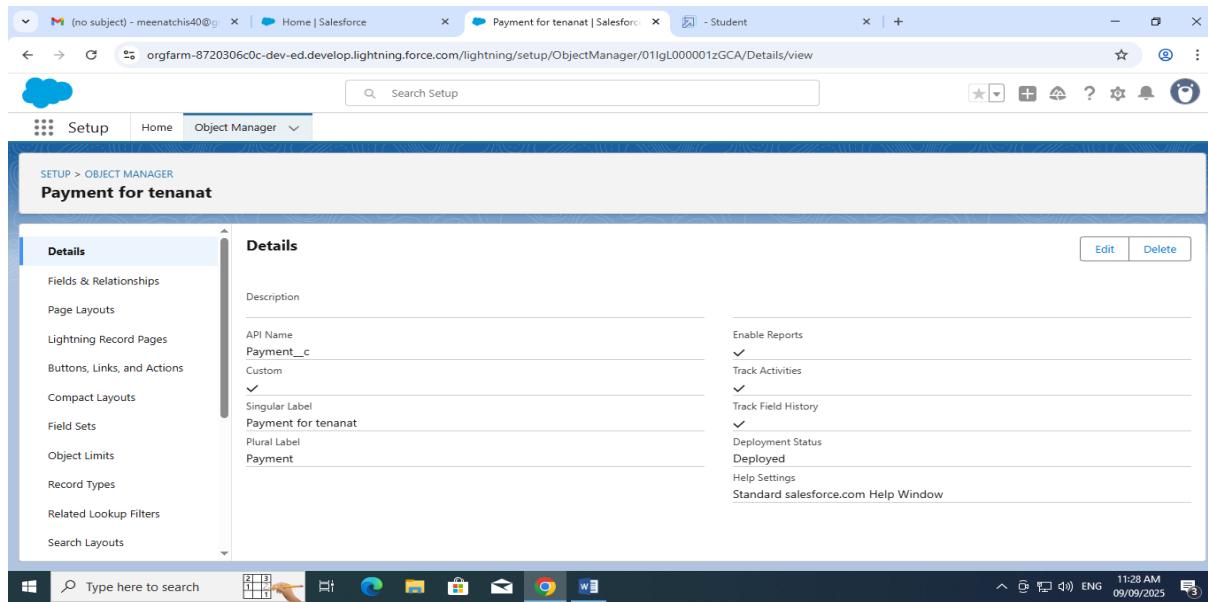
- Enable Reports: ✓
- Track Activities: ✓
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

**Navigation:** SETUP > OBJECT MANAGER  
lease

**Left Sidebar:**

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts

**System Information:** Type here to search, Windows taskbar, 11:25 AM 09/09/2025



- Configured fields and relationships

Screenshot of the Salesforce Setup interface showing the Object Manager for the 'Property' object.

The 'Fields & Relationships' section displays 8 items, sorted by Field Label:

| FIELD LABEL      | FIELD NAME       | DATA TYPE             | CONTROLLING FIELD | INDEXED |
|------------------|------------------|-----------------------|-------------------|---------|
| Address          | Address_c        | Long Text Area(32768) |                   |         |
| Created By       | CreatedById      | Lookup(User)          |                   |         |
| Last Modified By | LastModifiedById | Lookup(User)          |                   |         |
| Name             | Name_c           | Text(25)              |                   |         |
| Owner            | OwnerId          | Lookup(User,Group)    |                   | ✓       |
| Property Name    | Name             | Text(80)              |                   | ✓       |
| sfqt             | sfqt_c           | Text(18)              |                   |         |

Screenshot of the Salesforce Setup interface showing the Object Manager for the 'Payment for tenantat' object.

The 'Fields & Relationships' section displays 9 items, sorted by Field Label:

| FIELD LABEL       | FIELD NAME          | DATA TYPE          | CONTROLLING FIELD | INDEXED |
|-------------------|---------------------|--------------------|-------------------|---------|
| Amount            | Amount_c            | Number(18, 0)      |                   |         |
| check for payment | check_for_payment_c | Picklist           |                   |         |
| Created By        | CreatedById         | Lookup(User)       |                   |         |
| Last Modified By  | LastModifiedById    | Lookup(User)       |                   |         |
| Owner             | OwnerId             | Lookup(User,Group) |                   | ✓       |
| Payment date      | Payment_date_c      | Date               |                   |         |
| Payment Name      | Name                | Text(80)           |                   | ✓       |
| Property          | Property_c          | Lookup(Property)   |                   | ✓       |

(no subject) - meenatchis40@gmail.com | Home | Salesforce | lease | Salesforce | - Student | +

orgfarm-8720306c0c-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001zIUS/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

**lease**

Details

**Fields & Relationships**

7 Items, Sorted by Field Label

| 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               |                   |         |

Type here to search

11:33 AM 09/09/2025

(no subject) - meenatchis40@gmail.com | Home | Salesforce | Tenant | Salesforce | - Student | +

orgfarm-8720306c0c-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001zHUn/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

**Tenant**

Details

**Fields & Relationships**

8 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              |                   |         |
| Property         | Property_c       | Lookup(Property)   |                   | ✓       |
| status           | status_c         | Picklist           |                   |         |
| Tenant Name      | Name             | Text(80)           |                   | ✓       |

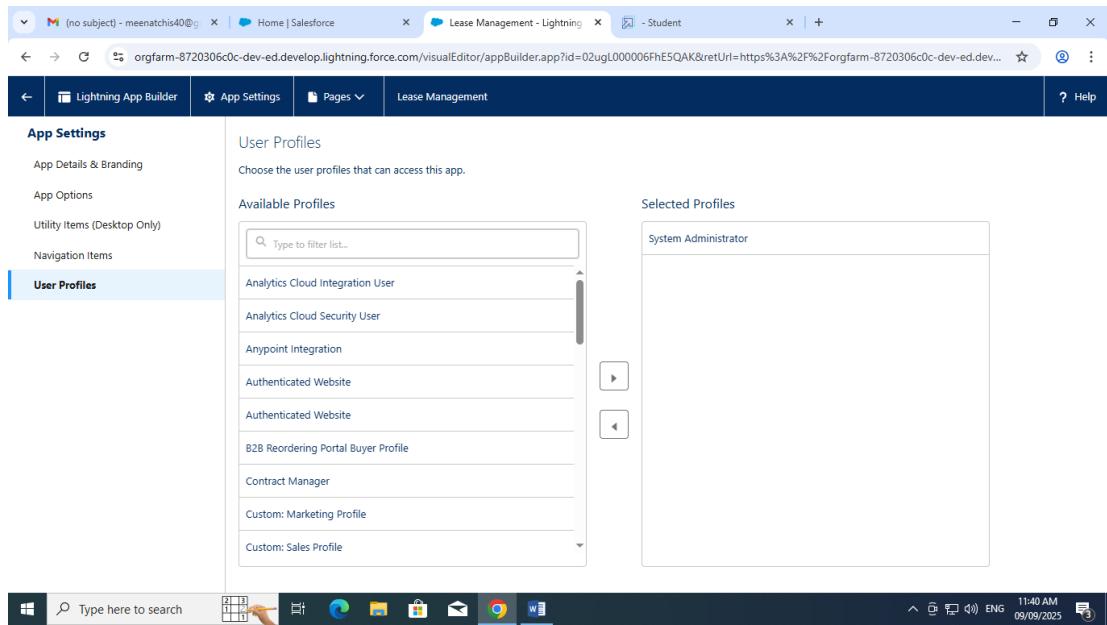
Type here to search

11:34 AM 09/09/2025

- Developed Lightning App with relevant tabs

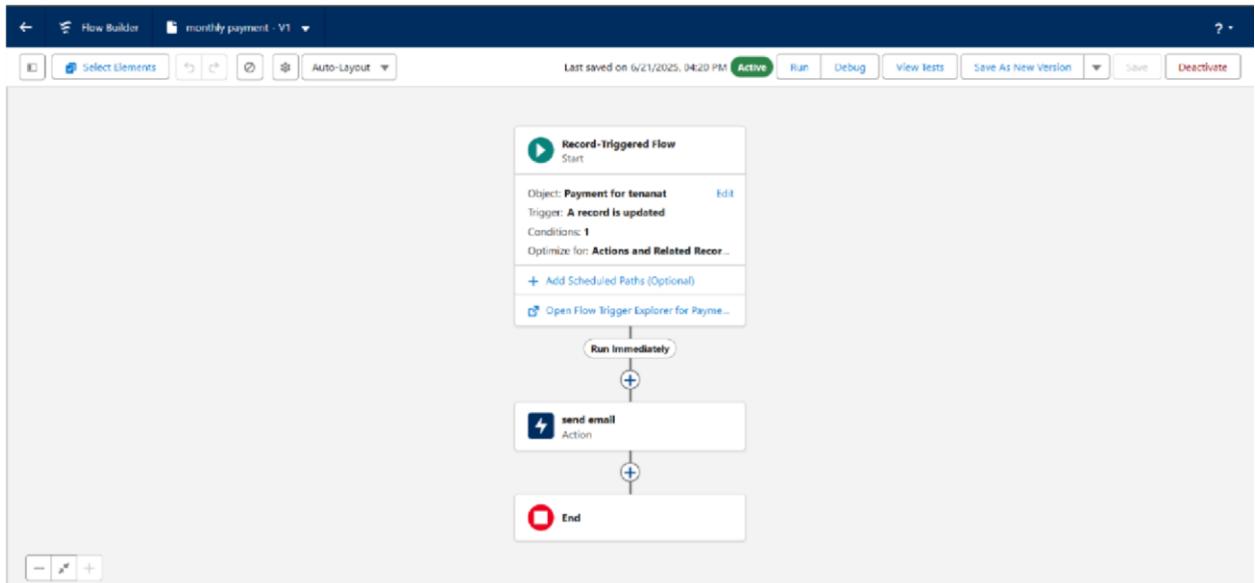
The screenshot shows the 'Lease Management - Lightning' app in the Salesforce Lightning App Builder. The left sidebar has 'App Settings' selected. The main area is titled 'App Details & Branding' with the sub-section 'App Details & Branding'. It contains fields for 'App Name' (Lease Management), 'Developer Name' (Lease\_Management), 'Description' (Enter a description...), 'Image' (a blue square icon with 'LM'), and 'Primary Color Hex Value' (#0070D2). Below these are 'Org Theme Options' and an 'App Launcher Preview' showing the 'LM' icon and 'Lease Management' label.

The screenshot shows the 'Lease Management - Lightning' app in the Salesforce Lightning App Builder. The left sidebar has 'Navigation Items' selected. The main area is titled 'Navigation Items' with the sub-section 'Available Items'. It lists items like Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, and Appointment Categories. To the right, under 'Selected Items', are Home, Properties, Tenants, Lease, and Payment. Navigation arrows between the two columns allow items to be moved. A message at the bottom right says 'Activate Windows Go to Settings to activate Windows.' The status bar at the bottom shows 'Type here to search' and system icons.



The screenshot shows the 'Lease Management' application interface. The top navigation bar includes 'Payment', 'Tenants', 'property', 'lease', and a search bar. Below the navigation is a 'Recently Viewed' section for 'Payment' with 5 items. The main area displays a list of payments with columns for 'Payment Name' and other details. A toolbar at the top of the list view includes 'New', 'Import', 'Change Owner', and 'Assign Label' buttons. The bottom of the screen shows a standard Windows taskbar with various application icons.

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

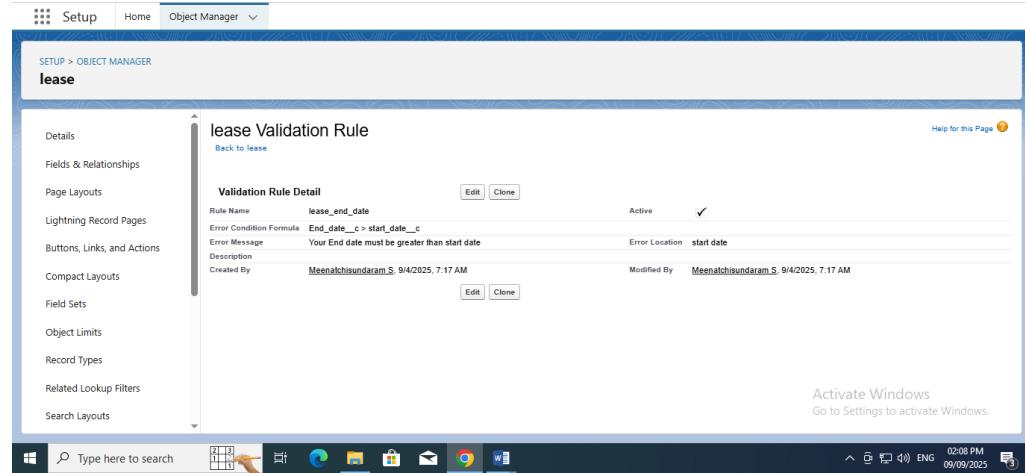
The screenshot shows the Validation Rule Edit screen for the 'lease' object:

- Validation Rule Edit** screen with 'Save', 'Save & New', and 'Cancel' buttons.
- Rule Name:** lease\_end\_date
- Active:** checked
- Description:** (empty)
- Error Condition Formula:**

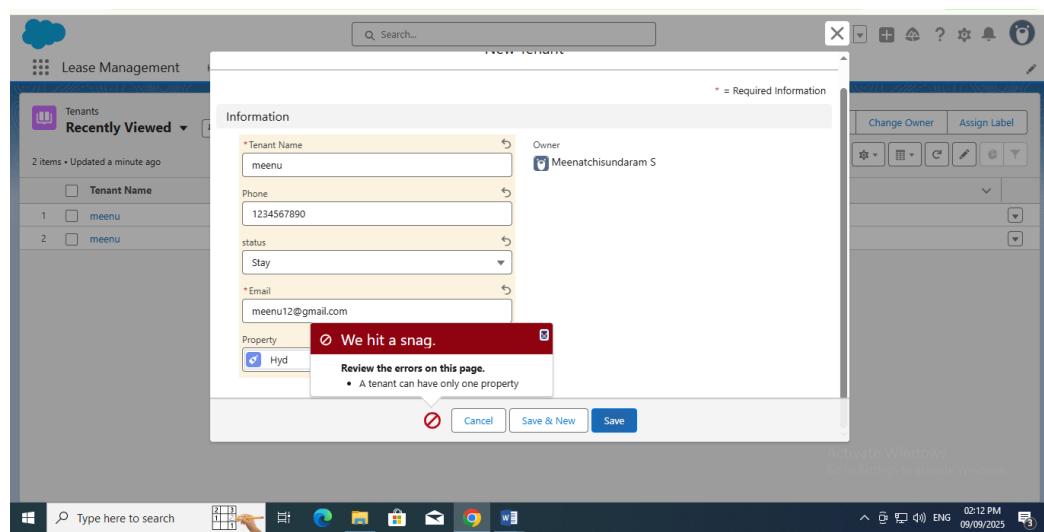
Example: `Discount_Percent < 0.30` More Examples...

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

`End_date__c < start_date__c`
- Functions:**
  - All Function Categories
  - ABS
  - ACOS
  - ADMONTHS
  - AND
  - ASCII
  - ASIN
  - Insert Selected Function
  - ABS(Number)
  - Returns the absolute value of a number, a number without its sign
  - Help on this function
- Quick Tips:**
  - Operators & Functions



- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

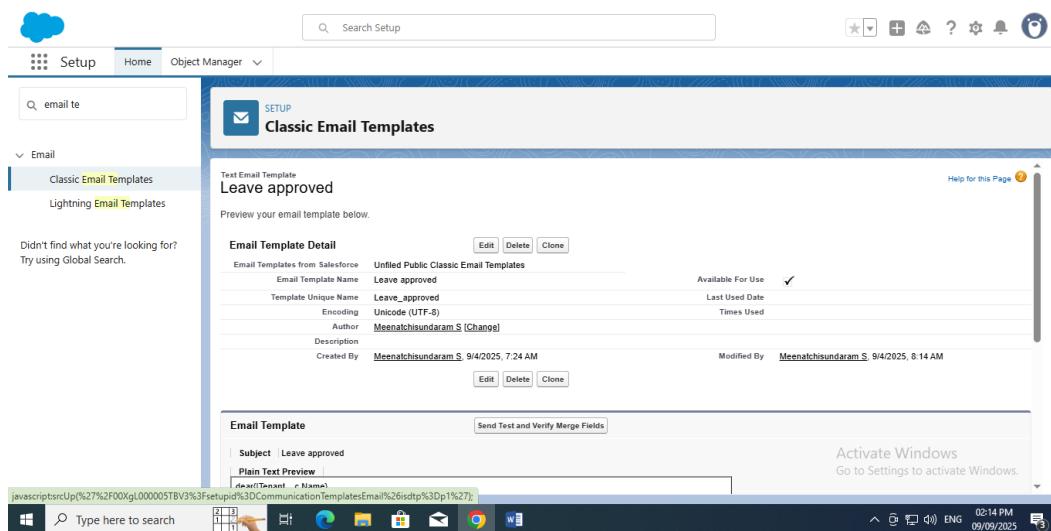
The screenshot shows a Java code editor with the following details:

- Title Bar:** File > New > Design > Run > Save > Help > ...
- Toolbar:** Undo (ctrl-Z) | Redo (ctrl-Y) | Cut (ctrl-X) | Copy (ctrl-C) | Paste (ctrl-V) | Delete (ctrl-D) | Select All (ctrl-A) | Find (ctrl-F) | Replace (ctrl-H) | Go To (ctrl-G) | Run (ctrl-R) | Stop (ctrl-S) | Refresh (ctrl-F5) | New (ctrl-N) | Open (ctrl-O) | Save (ctrl-S) | Save As (ctrl-Shift-S) | Save All (ctrl-Shift-Shift-S) | Exit (ctrl-Q)
- Status Bar:** Code Coverage Rate: 64% | API Version: 44 | 57% | - + | Reset
- Code Area:** The code is for a class named `MonthlyEmailScheduler` which implements the `Schedulable` interface. It contains a static method `sendMonthlyEmails()` that sends emails to all tenants.

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

- Bottom Navigation:** Help | Tools | Checkstyle | Query Editor | View Setup | Progress | Problems

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



The screenshot shows the Microsoft Edge browser interface with the following details:

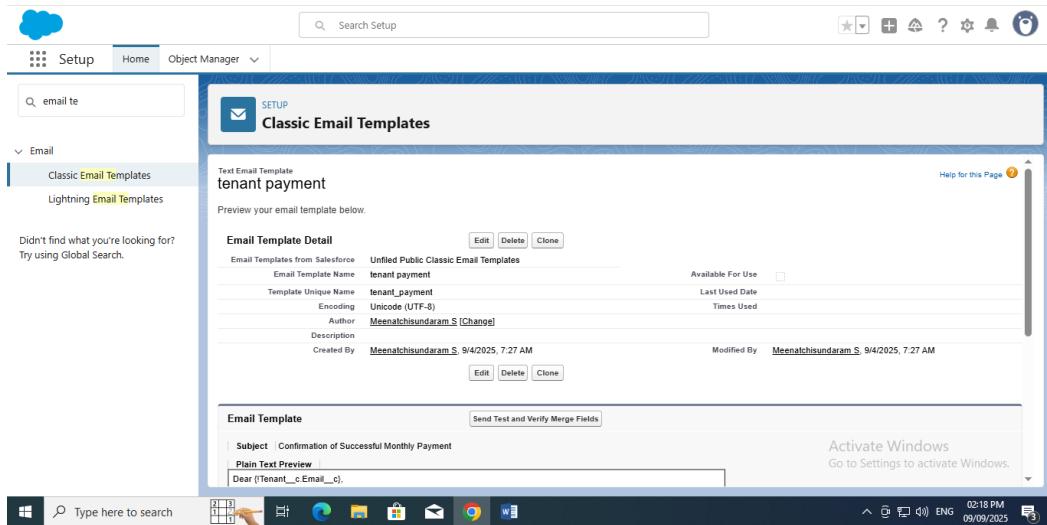
- Top Bar:** Includes the Edge logo, a search bar with "Search Setup", and a toolbar with icons for star, square, plus, cloud, question mark, user, and bell.
- Header:** Shows "Setup" as the active tab, "Home" and "Object Manager" other tabs, and a search bar with "email te".
- Left Sidebar:** Under "Email", it shows "Classic Email Templates" (selected) and "Lightning Email Templates". A message says " Didn't find what you're looking for? Try using Global Search."
- Main Content:** The "Classic Email Templates" page for "tenant leaving".
  - Title:** Classic Email Templates
  - Template Type:** Text Email Template
  - Template Name:** tenant leaving
  - Description:** Preview your email template below.
  - Email Template Detail:** Shows the template's details:
    - Email Templates from Salesforce: Unified Public Classic Email Templates
    - Email Template Name: tenant leaving
    - Template Unique Name: tenant\_leaving
    - Encoding: Unicode (UTF-8)
    - Author: Meenatchisundaram S [Change]
    - Description: Created By: Meenatchisundaram S 9/4/2025, 7:22 AM
    - Available For Use: ✓
    - Last Used Date:
    - Times Used:
  - Email Template:** Subject: request for approve the leave  
Plain Text Preview  
Dear {Tenant\_\_c createdBy},
- Bottom Bar:** Shows the Windows taskbar with various pinned icons (File Explorer, Edge, Mail, etc.) and system status: 02:16 PM, ENG, 09/09/2025.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar at the top right contains the placeholder 'Search Setup'. The left sidebar has a 'Email' category expanded, showing 'Classic Email Templates' and 'Lightning Email Templates' 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 'Classic Email Templates' with a 'Text Email Template' icon. It displays a template named 'Leave rejected' with a preview message: 'Leave rejected'. Below this is the 'Email Template Detail' section, which includes fields for 'Email Templates from Salesforce' (set to 'Unfiled Public Classic Email Templates'), 'Email Template Name' ('Leave rejected'), 'Template Unique Name' ('Leave\_rejected'), 'Encoding' ('Unicode (UTF-8)'), 'Author' ('Meenatchisundaram S [Change]'), 'Description' (empty), 'Created By' ('Meenatchisundaram S 9/4/2025, 7:26 AM'), and 'Modified By' ('Meenatchisundaram S 9/4/2025, 8:14 AM'). There are 'Edit', 'Delete', and 'Clone' buttons for each row. At the bottom, there's an 'Email Template' section with a 'Send Test and Verify Merge Fields' button, a subject line 'Subject - Leave rejected', a plain text preview 'Plain Text Preview', and a merge field 'Dear {!Tenant\_c\_Name}.' On the far right, there's a 'Help for this Page' link with a question mark icon. The bottom of the screen features the Windows taskbar with various pinned icons and a system tray showing network, battery, and clock status.

The screenshot shows the Salesforce interface with the following details:

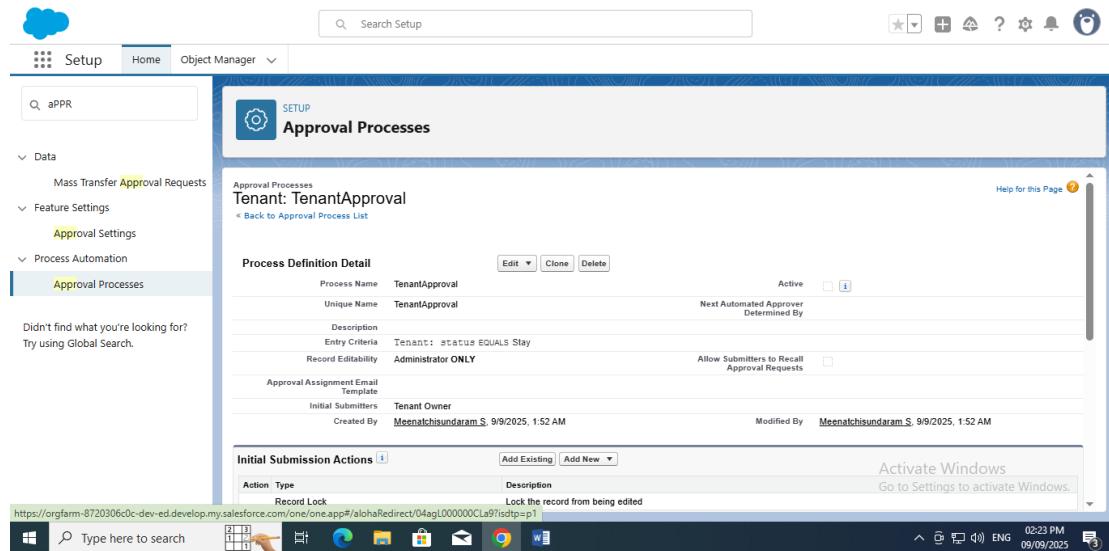
- Header:** Search bar with "Search Setup", a gear icon, and a bell icon.
- Left Navigation:** "Setup" button, "Home" link, "Object Manager" dropdown, and a search bar containing "email te". Below it, under "Email", are "Classic Email Templates" (selected) and "Lightning Email Templates". A note says " Didn't find what you're looking for? Try using Global Search."
- Main Content:** "SETUP" button and "Classic Email Templates" page.
  - Section:** "Text Email Template" - "Tenant Email".
  - Help:** "Help for this Page" with a yellow question mark icon.
  - Email Template Detail:** Preview of the email template below.
  - Fields:**

| Field                           | Value                                  |
|---------------------------------|--|
| Email Templates from Salesforce | Unfiled Public Classic Email Templates |
| Email Template Name             | Tenant Email                           |
| Template Unique Name            | Tenant_Email                           |
| Encoding                        | Unicode (UTF-8)                        |
| Author                          | Meenatchisundaram_S [Change]           |
| Description                     |  |
| Created By                      | Meenatchisundaram_S, 9/4/2025, 7:27 AM |
| Modified By                     | Meenatchisundaram_S, 9/4/2025, 7:54 AM |
  - Email Template:** Subject: Urgent Monthly Rent Payment Reminder, Plain Text Preview, and a "Send Test and Verify Merge Fields" button.
  - Bottom Bar:** "Activate Windows" link, "Go to Settings to activate Windows.", and a search bar with "javascriptrsrcUp(%27%2f000gL000005TBe%3fsetupid%3DCommunicationTemplatesEmail%26sdtp%3Dp1%27);".
  - Taskbar:** Icons for File, Home, Object Manager, and a search bar with "Type here to search".
  - System Bar:** Icons for Task View, Start, Taskbar View, and a clock showing "02:17 PM 09/09/2025".

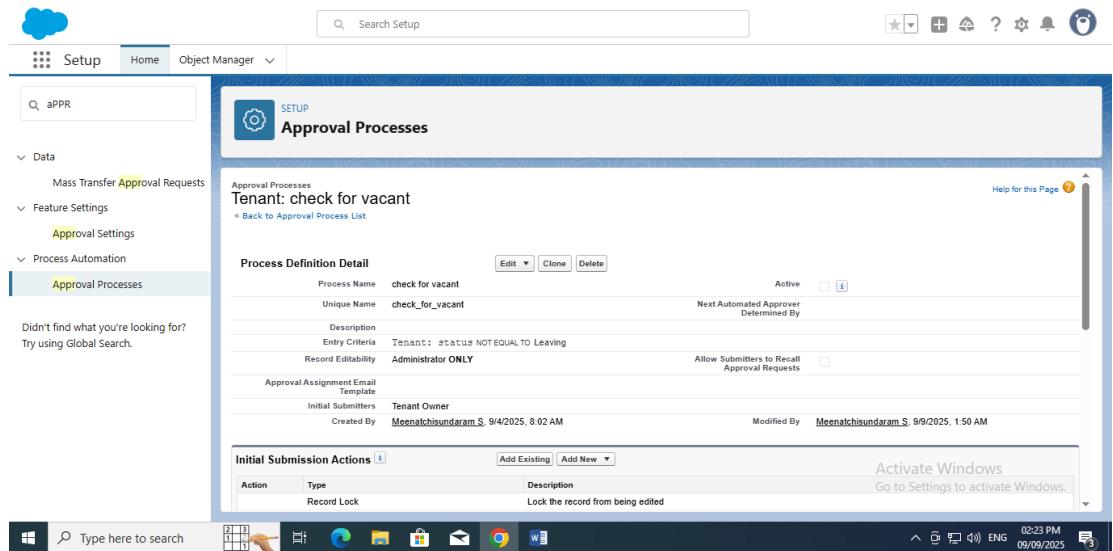


- Approval Process creation

For Tenant Leaving:



For Check for Vacant:



- Apex Trigger

### Create an Apex Trigger

```

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

```

Developer Console - Google Chrome

osgfarm-Sdf1e005f2-dev-ed develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help

test.apex [testHandler.apc] [MonthYearScheduler.apc] [Go To]

Code Coverage: None API Version: 64

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

Logs Tests Checkpoints Query Editor View State Progress Problems

## Create an Apex Handler class

Developer Console - Google Chrome

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

File Edit Debug Test Workspace Help < >

testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 64 Go To

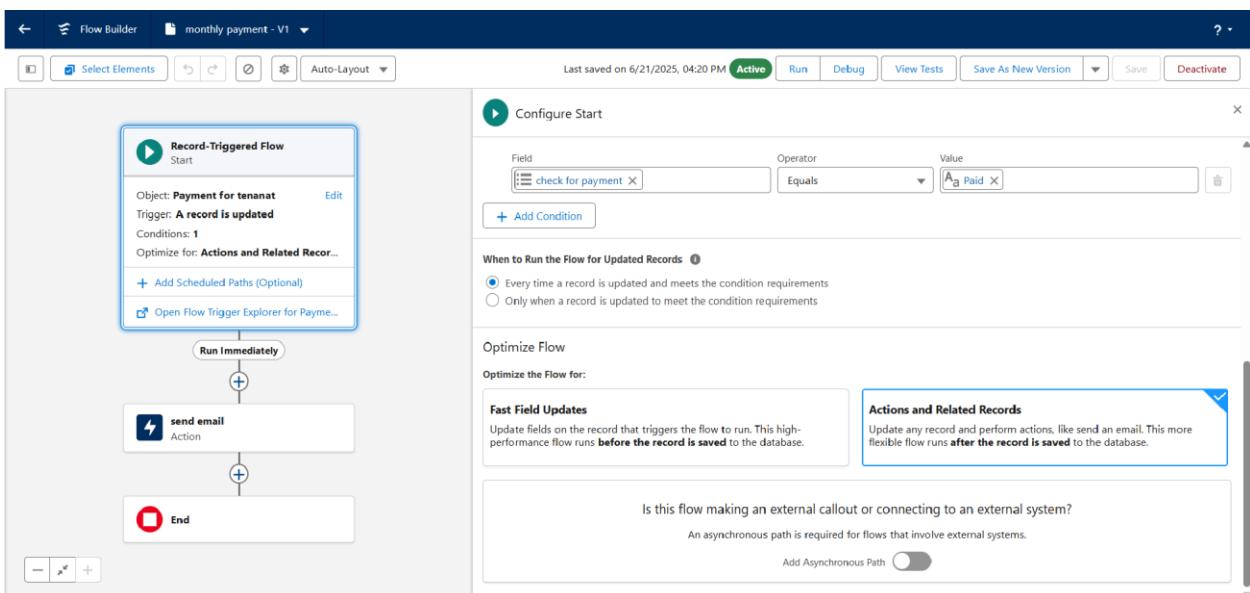
```
1 * public class testHandler {  
2       
3     public static void preventInsert(List<Tenant__c> newList) {  
4           
5             Set<Id> existingPropertyIds = new Set<Id>();  
6           
7             for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {  
8                   
9                     existingPropertyIds.add(exi Open Entity Type Name Namespace Related  
10                 } Classes testHandler test ApexTrigger References  
11             Triggers MonthlyEmailScheduler property CustomField References  
12             Pages   
13             Page Components   
14             Objects Tenant__c SObject References  
15             Static Resources   
16             Packages   
17               
18             if (newTenant.Property__c != null) {  
19                   
20                     newTenantaddError('A t Open Filter Filter the repository (* = any string) Hide Managed Packages Refresh  
21             }  
22         }  
23     }
```

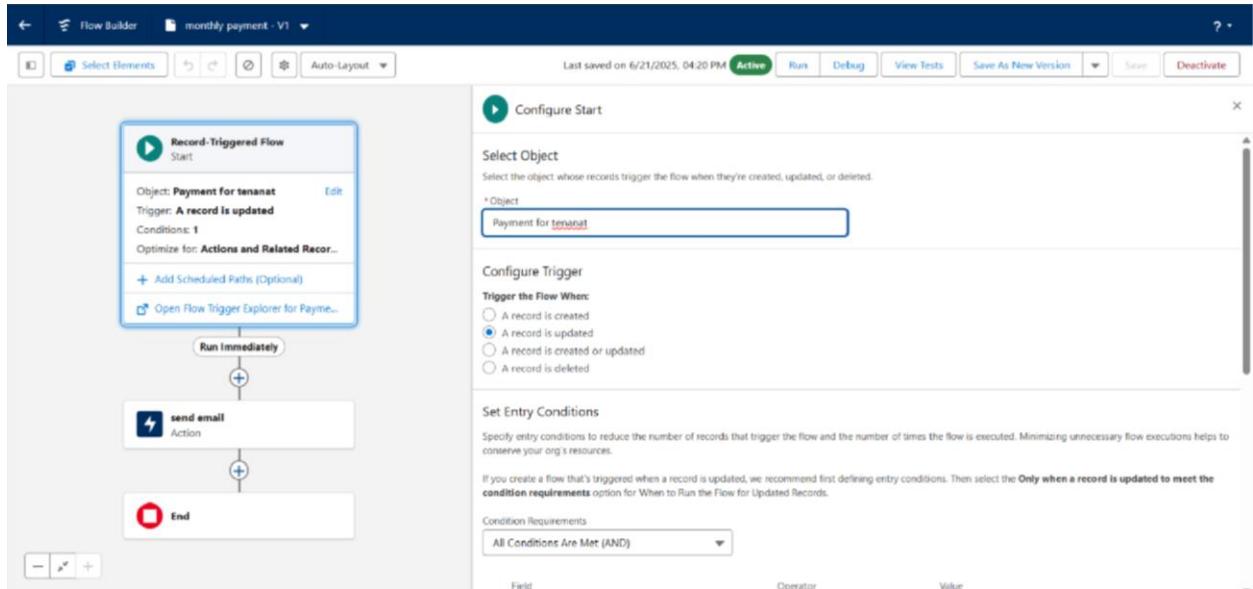
The screenshot shows the Developer Console interface in Google Chrome. The title bar reads "Developer Console - Google Chrome" and the URL is "orgfarm-Sdf1o00512-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSIPage". The menu bar includes File, Edit, Debug, Test, Workspace, Help, and a search icon. Below the menu is a toolbar with tabs: testHandler.apex (selected), MonthlyEmailScheduler.apex, Code Coverage, New, API Version (set to 54), and Go To. 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(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                newTenant.addError('A tenant can have only one property');
20
21            }
22
23        }
24    }
25}
```

At the bottom, there are tabs for Logs, Tests, Checkpoints, Query Editor, View Status, Progress, and Problems. The Problems tab is selected, showing a single error message: "Line 19 Problem A tenant can have only one property".

### • FLOWS





- 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 sendMonthlyEmails() {
17
18         List<Tenant__c> tenants = [SELECT
19
20             FOR (Tenant__c tenant : tenants
21
22                 String recipientEmail = tenant.Email__c;
23

```

```

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

```

## Schedule Apex class

The screenshot shows the Salesforce Setup interface with the search bar set to "apex". The left sidebar has sections for Email (Apex Exception Email, Apex Classes selected), Custom Code (Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers), and Environments (Jobs, Apex Flex Queue, Apex Jobs). The main area displays the "Apex Classes" page for "MonthlyEmailScheduler". The "Apex Class Detail" table shows the class name, namespace prefix (Meenatchisundaram\_S), created by (Meenatchisundaram\_S), status (Active), and last modified by (Meenatchisundaram\_S). The "Class Body" tab shows the Apex code:

```

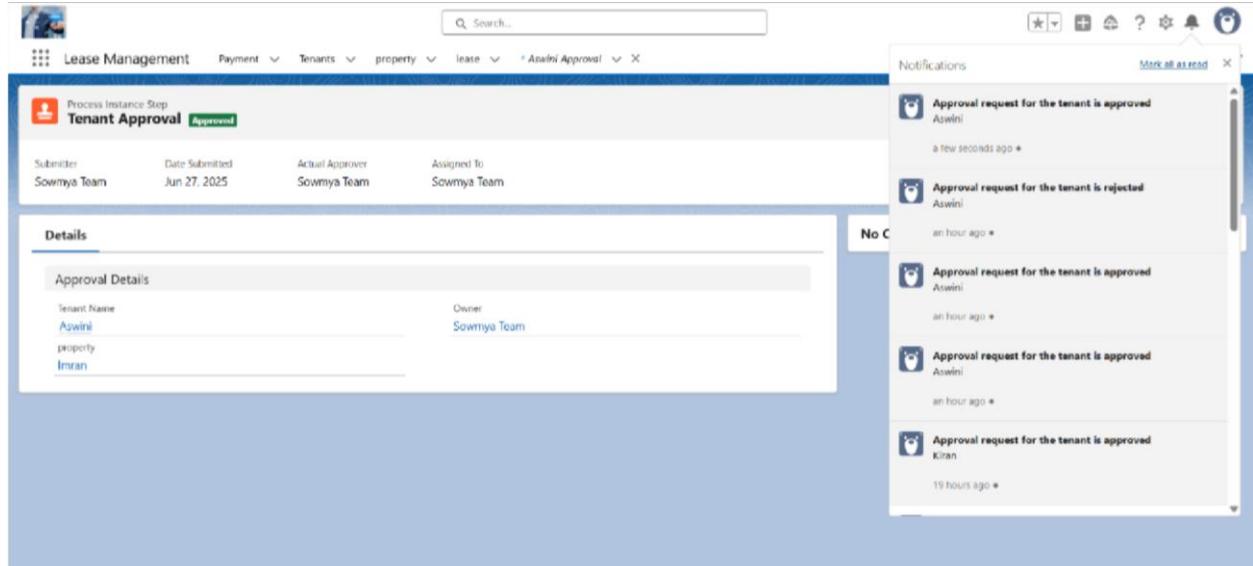
1. global class MonthlyEmailScheduler implements Schedulable {
2.
3.     global void execute(SchedulableContext sc) {
4.
5.         Integer currentDay = Date.today().day();
6.
7.         if (currentDay == 1) {
8.             sendMonthlyEmails();
9.         }
10.    }
11.
12. }
13.
14.
15. public static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

```

The "Schedule" tab indicates the class runs daily at 12:01 AM. A message at the bottom right says "Activate Windows Go to Settings to activate Windows." The taskbar at the bottom shows various application icons.

The screenshot shows a Microsoft Dynamics 365 Lease Management application window. At the top, there's a navigation bar with icons for Home, Properties, Tenants, Lease, and Payment. A search bar is located at the top right. Below the navigation bar, the main area displays a tenant record for "meenu". The "Details" tab is selected, showing fields for Tenant Name (meenu), Phone ((902) 554-0321), status (Stay), Email (meenatchis40@gmail.com), and Property (Hyd). The "Owner" field shows "Meenatchisundaram S". The "Created By" field shows "Meenatchisundaram S" with a timestamp of 9/4/2025, 8:48 AM. The "Last Modified By" field shows "Meenatchisundaram S" with a timestamp of 9/8/2025, 4:45 AM. To the right of the details, there's an "Activity" section with a button to "Activate Windows". The system status bar at the bottom indicates it's 02:28 PM on 09/09/2025.

This screenshot shows the same Microsoft Dynamics 365 Lease Management application after a new tenant record has been created. A green success message "Tenant \"meenu\" was created." is displayed prominently at the top. The tenant record for "meenu" now includes a "Phone" field with the value "(123) 654-7890" and a "status" field with the value "Leaving". The rest of the fields remain the same as in the previous screenshot. The "Activity" section and system status bar are also present.

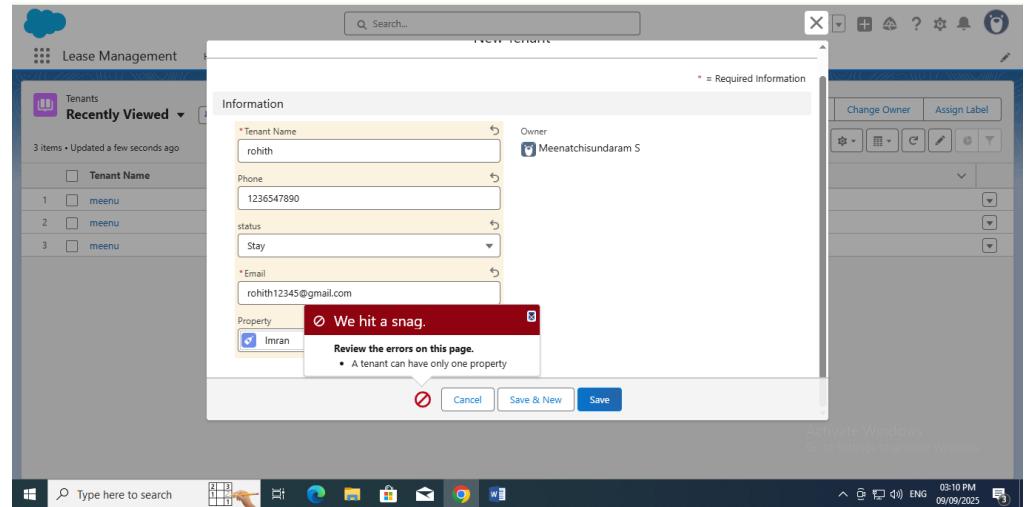


---

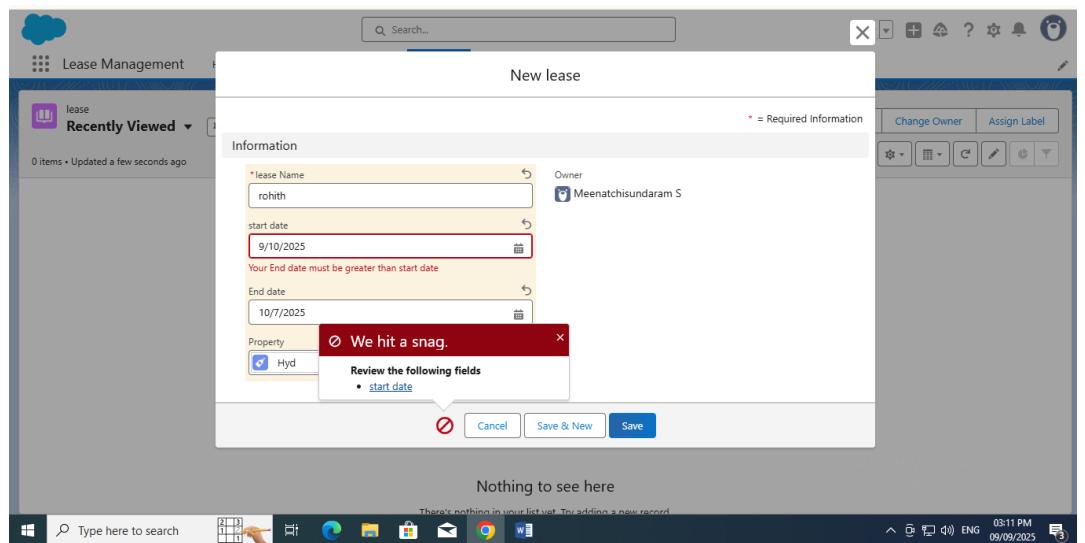
# FUNCTIONAL AND PERFORMANCE TESTING

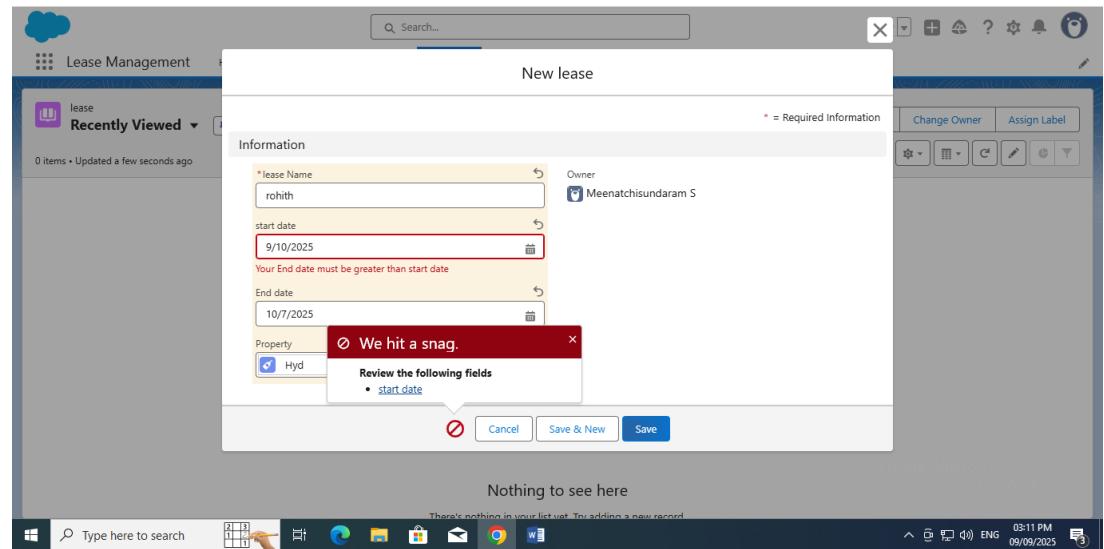
## Performance Testing

- Trigger validation by entering duplicate tenant-property records

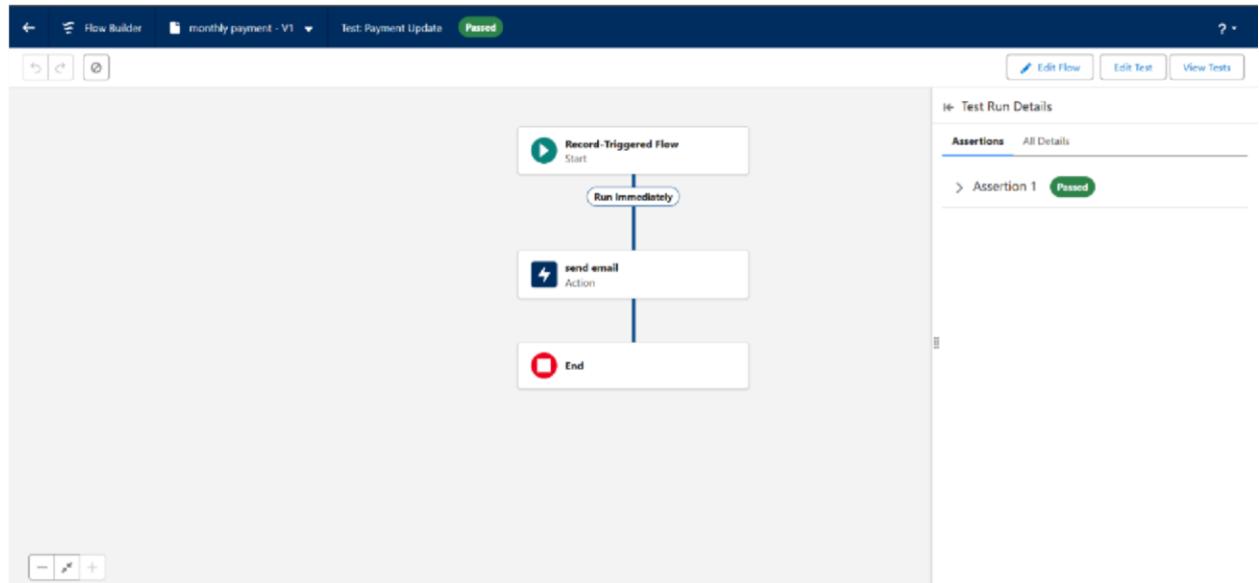


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the 'Lease Management' application interface. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', 'lease', and 'niranjan Approval'. The main content area displays the 'Details' tab for tenant 'niranjan'. The form fields include:

- Tenant Name: niranjan
- Email: niranjan1506@gmail.com
- Phone: (empty)
- Status: Stay
- Property: Parkside Lofts
- Created By: Sowmya Team
- Last Modified By: Sowmya Team

Below the form are 'Cancel' and 'Save' buttons. To the right, a sidebar titled 'Notifications' lists recent activity:

- Approval request for the tenant is approved niranjan (a few seconds ago)
- Approval request for the tenant is rejected niranjan (Jun 23, 2025, 4:29 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:25 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:14 PM)
- New Guidance Center learning resource available: Define Your Sales Process (Jun 20, 2025, 1:28 PM)

At the bottom right of the sidebar is a 'Mark all as read' button.

This screenshot shows the same 'Lease Management' application interface for tenant 'niranjan'. The 'Approval History' section is expanded, displaying a table of approval steps:

| Step Name                  | Date               | Status    | Assigned To |
|----------------------------|--------------------|-----------|-------------|
| Step 1                     | 6/25/2025, 5:39 AM | Approved  | Sowmya Team |
| Approval Request Submitted | 6/25/2025, 5:39 AM | Submitted | Sowmya Team |
| Step 1                     | 6/23/2025, 2:59 AM | Rejected  | Sowmya Team |
| Approval Request Submitted | 6/23/2025, 3:58 AM | Submitted | Sowmya Team |
| Step 1                     | 6/23/2025, 3:55 AM | Approved  | Sowmya Team |
| Approval Request Submitted | 6/23/2025, 3:55 AM | Submitted | Sowmya Team |

Below the approval history is the 'Payment' section, which lists two payments:

| Payment Name |
|--------------|
| Jack         |
| Rahul        |

At the top right of the main content area are 'New Contact', 'Edit', and 'New Opportunity' buttons. A sidebar on the right provides a summary of past activity:

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

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

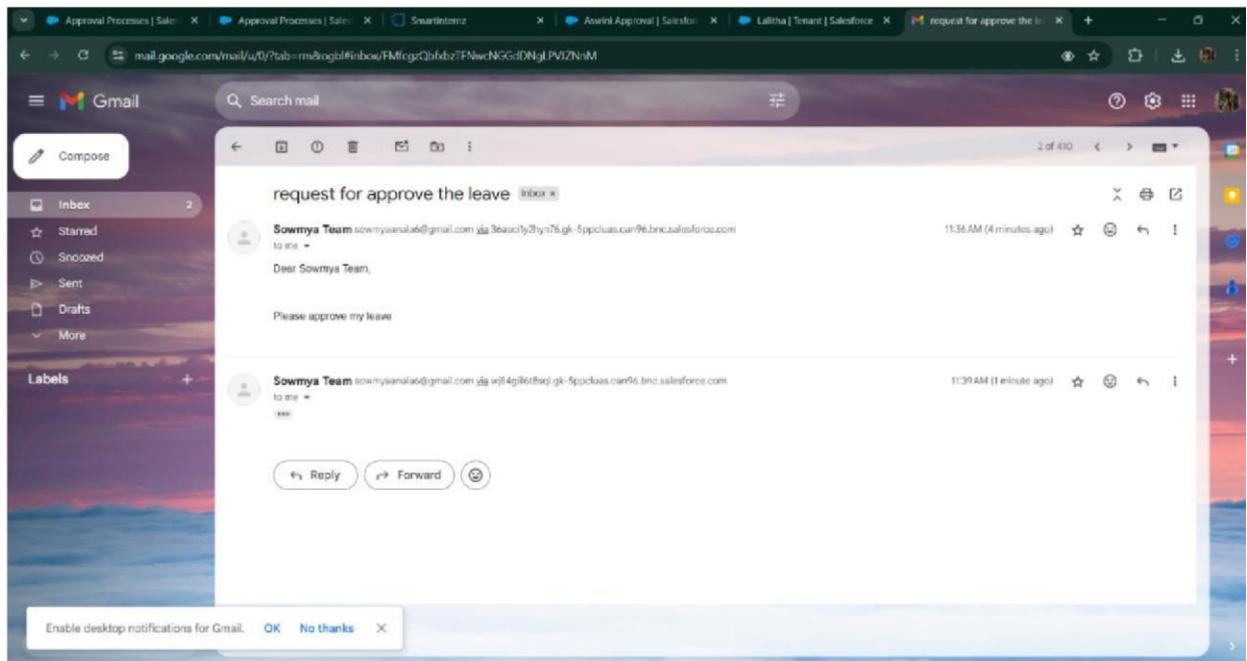
The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The 'Custom Object Tabs' section lists four tabs: 'Lease' (Keys icon), 'Payment' (Credit card icon), 'property' (Sack icon), and 'Tenants' (Map icon). The 'Web Tabs' and 'Visualforce Tabs' sections both indicate that no tabs have been defined.

- Email alerts

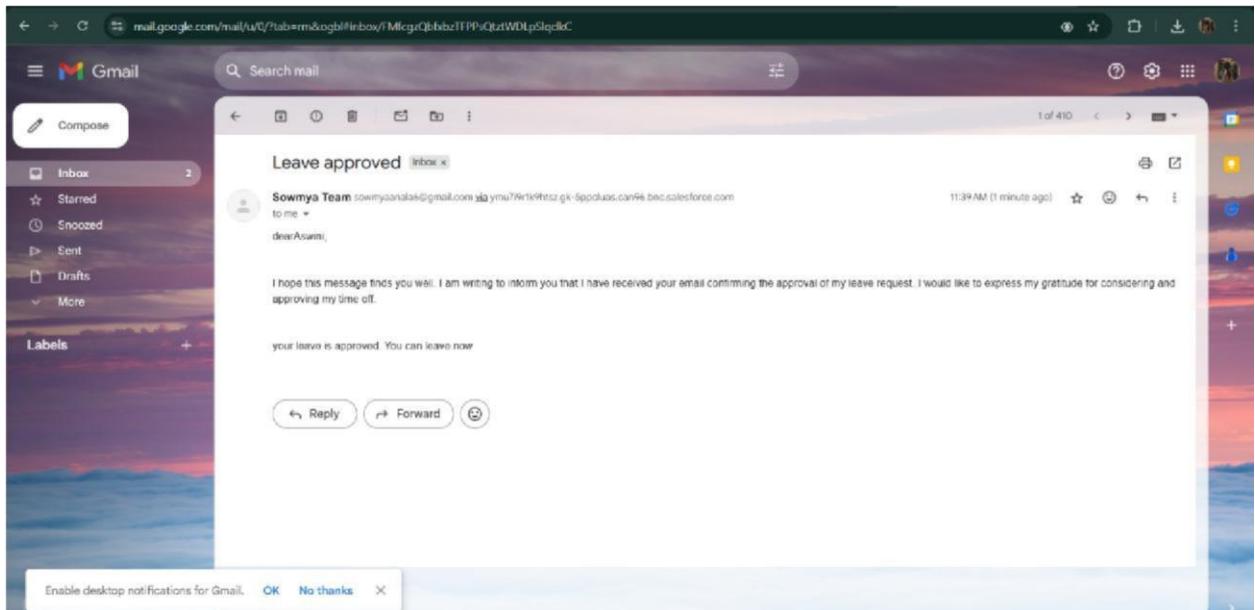
The screenshot shows the 'Approval History' page for a tenant named 'niranjan'. The table lists 8 items, all sorted by 'Is Pending' and updated a few seconds ago. The columns include Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data shows a sequence of approval requests and responses:

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

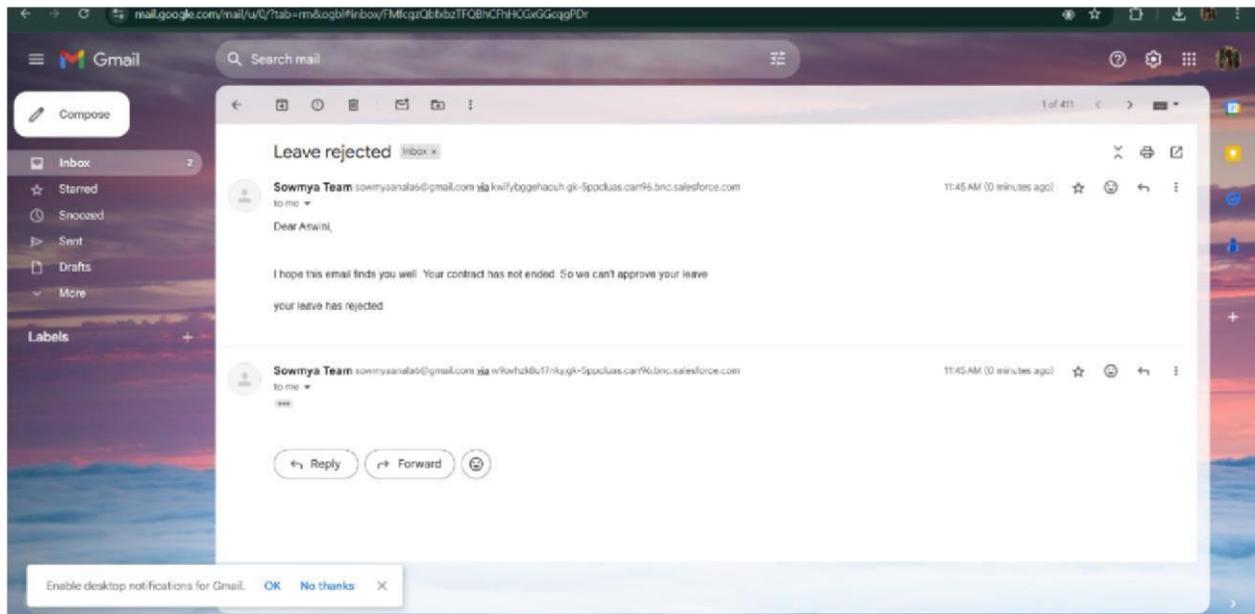
- Request for approve the leave



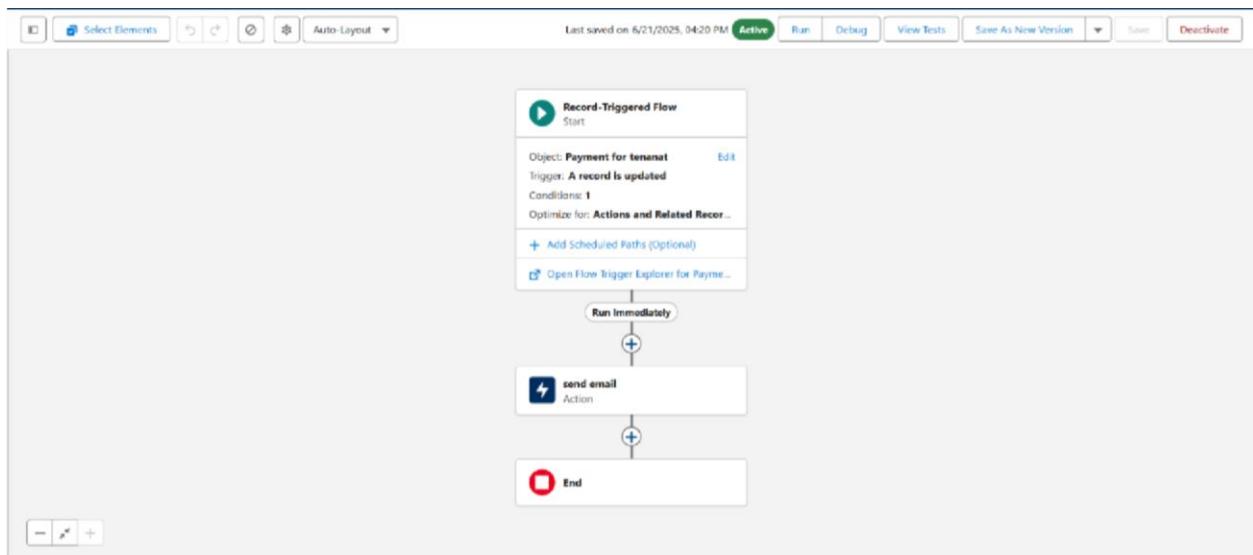
- Leave approved



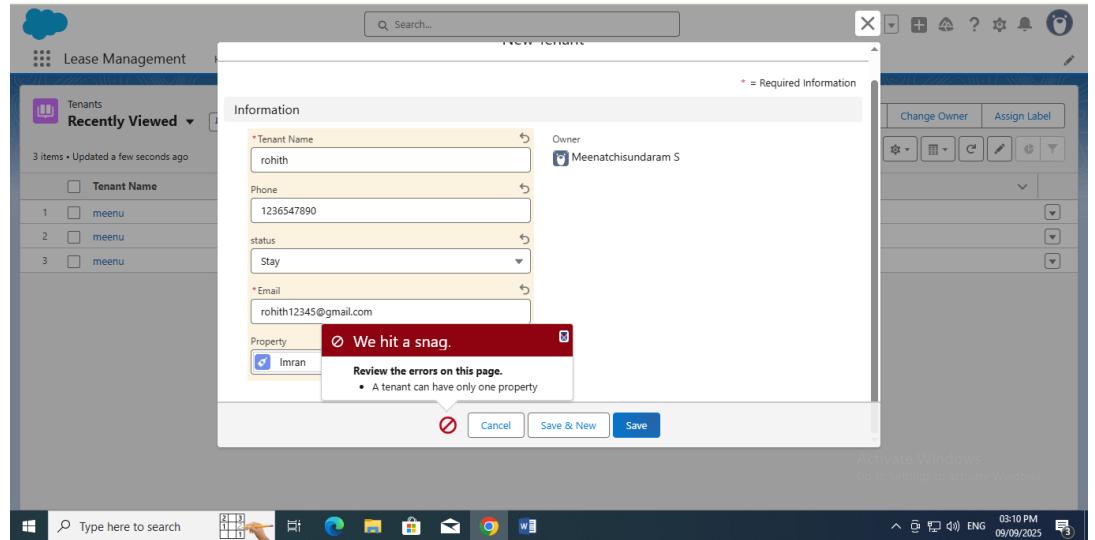
- Leave rejected



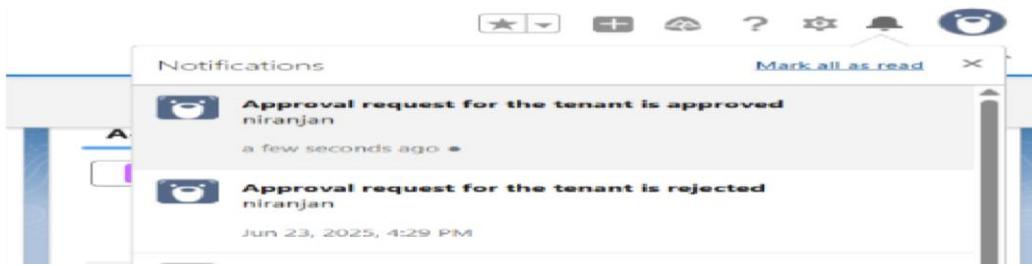
- Flow runs



- Trigger error messages



- Approval process notifications



## ADVANTAGES & DISADVANTAGES

# CONCLUSION

The Lease Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

---

## APPENDIX

- **Source Code:** Provided in Apex Classes and Triggers

**Test.apxt:** trigger test on Tenant\_\_c

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

**testHandler.apxc:**

```
public class
testHandler { public
static void
preventInsert(List<
Tenant__c> newlist)
{
    Set<Id>
existingPropertyIds
= new Set<Id>()
```

```

        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c
WHERE Property__c != null]) {

            existingPropertyIds.add(existingTenant.Property__c;

        } for (Tenant__c newTenant :

newlist) {

            if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) { newTenantaddError('A

tenant can have only one property');

        }

    }

}

```

### **MothlyEmailScheduler.apxc:**

```

global class MonthlyEmailScheduler implements Schedulable {
    global

        void execute(SchedulableContext sc) { Integer currentDay =

Date.today().day(); if (currentDay == 1) {

            sendMonthlyEmails();

        }

    }

} public static void

sendMonthlyEmails() { List<Tenant__c>

tenants = [SELECT Id, Email__c FROM

```

```
Tenant__c]; for (Tenant__c tenant :  
tenants) {  
  
    String recipientEmail = tenant.Email__c;  
    String emailContent = 'I trust this email finds you well. I am writing to remind you  
that the monthly rent is due Your timely payment ensures the smooth functioning of our  
rental arrangement and helps maintain a positive living environment for all.';  
  
    String emailSubject = 'Reminder: Monthly Rent Payment Due';  
    Messaging.SingleEmailMessage email = new  
  
        Messaging.SingleEmailMessage(); email.setToAddresses(new  
        String[]{recipientEmail}); email.setSubject(emailSubject);  
        email.setPlainTextBody(emailContent);  
  
    Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});  
}  
}  
}
```