

LEASE MANAGEMENT

Tiruppur Kumaran College For Women
Asbru3j

TEAM ID : NM2025TMID28356

TEAM MEMBERS:

TEAM LEADER : Azhagu Shalini.K

Email : azhagushalini67@gmail.com

TEAM MEMBER 1: Asai Priya.M

Email : asaipriya796@gmail.com

TEAM MEMBER 2: Aneesh Fathima.M

Email : aneeshfathima1605@gmail.com

TEAM MEMBER 3: Bathmini.A

Email : bathminianandhan@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>

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
- ✓ Integrate your data with Data Cloud
- ✓ Connect 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: Last name:

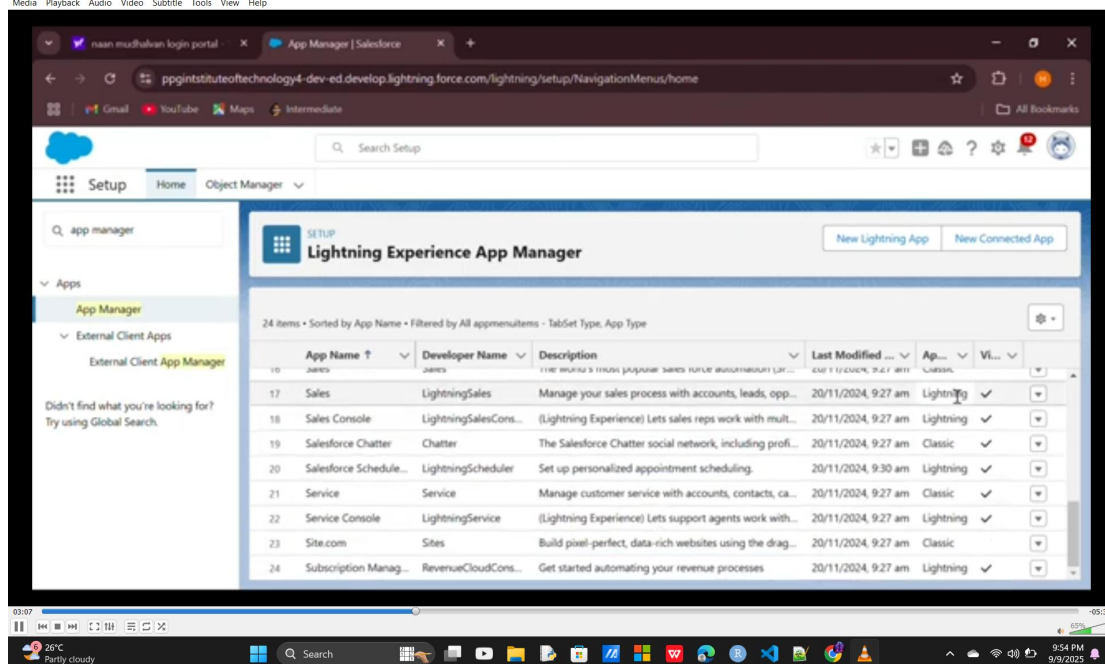
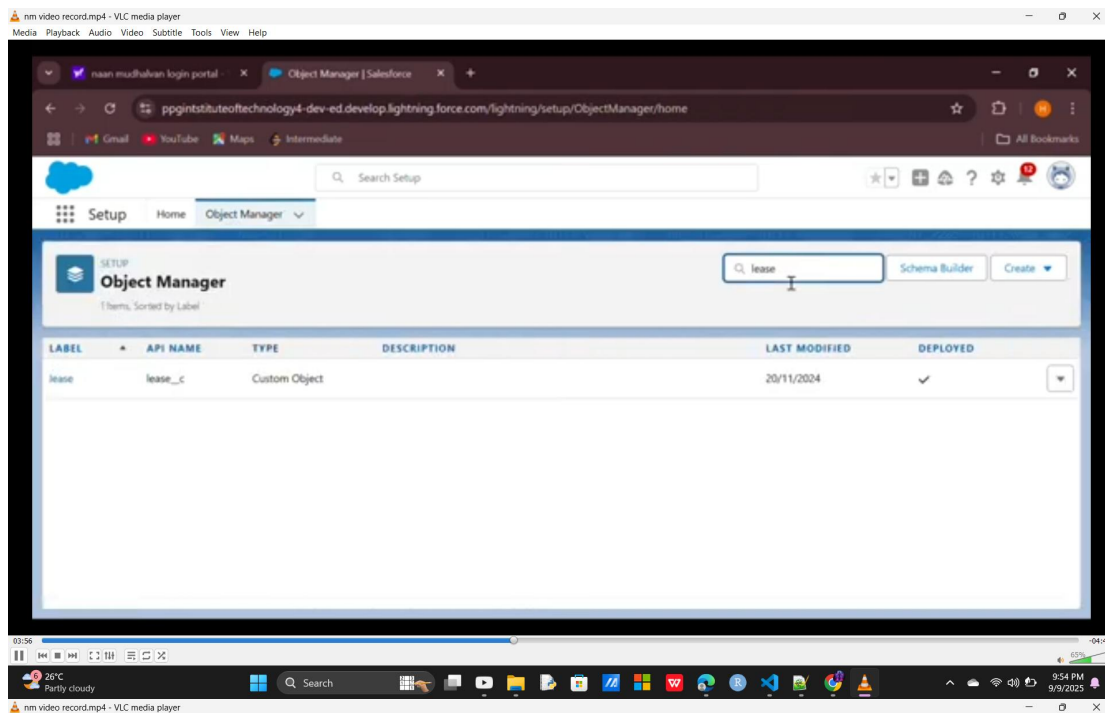
Job title: Work email:

Company: Country/Region:

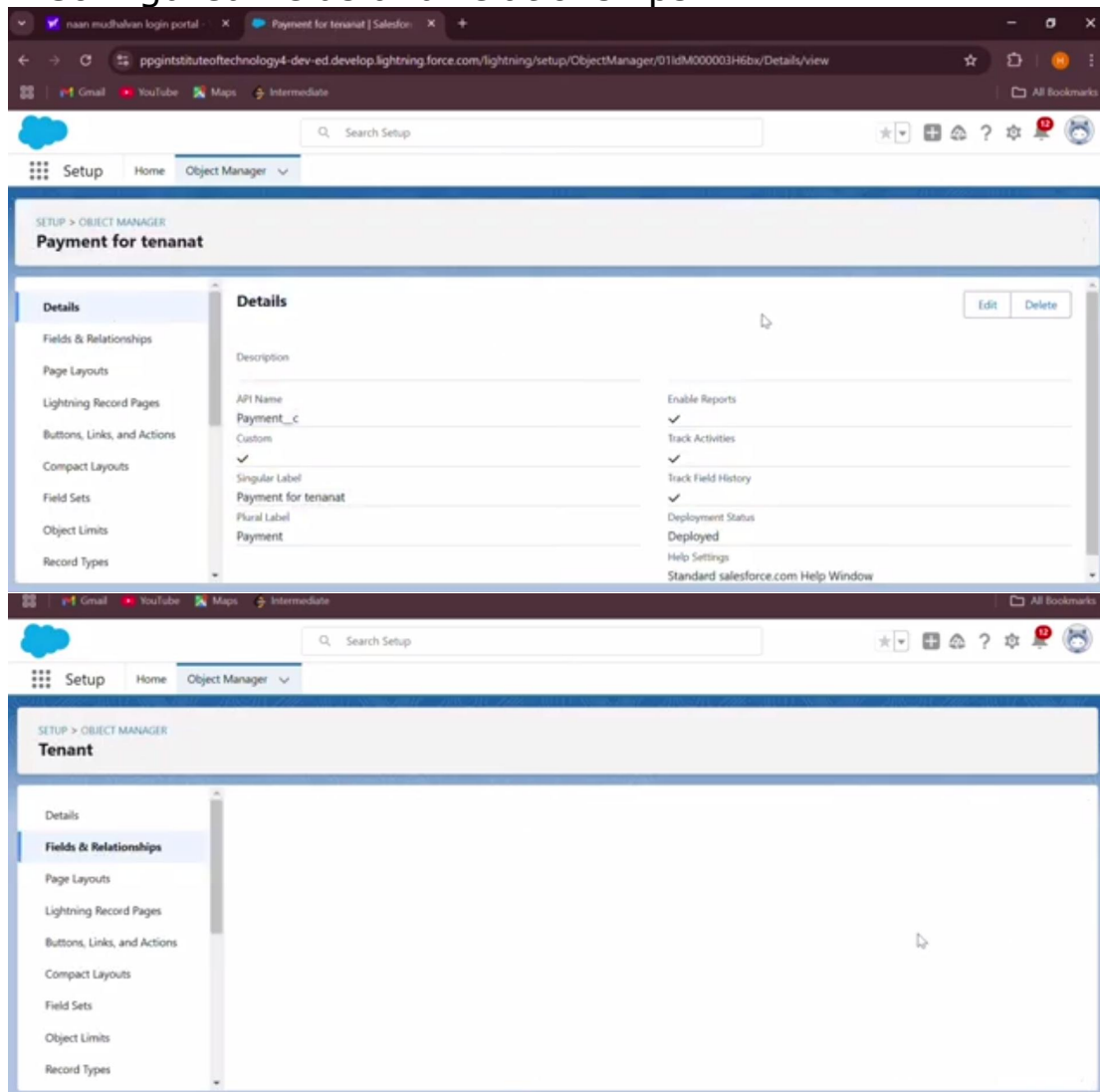
☐ I agree to the Salesforce Developer Edition Terms of Service, which includes the Salesforce Privacy Policy and the Salesforce Developer Edition Terms of Service. I understand that my use of these features may not be available for all days.

We value your privacy. To learn more, visit our Privacy Information.

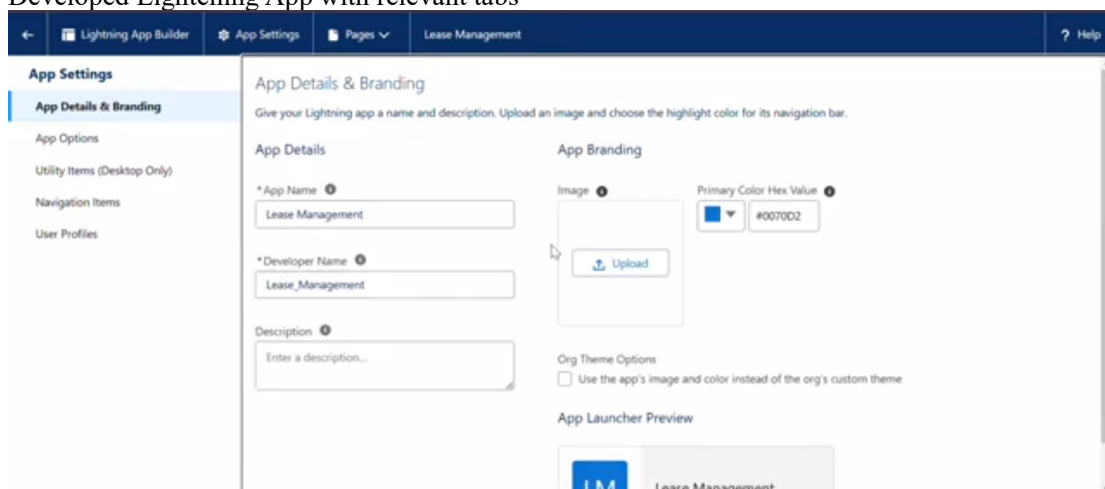
- Created objects: Property, Tenant, Lease, Payment

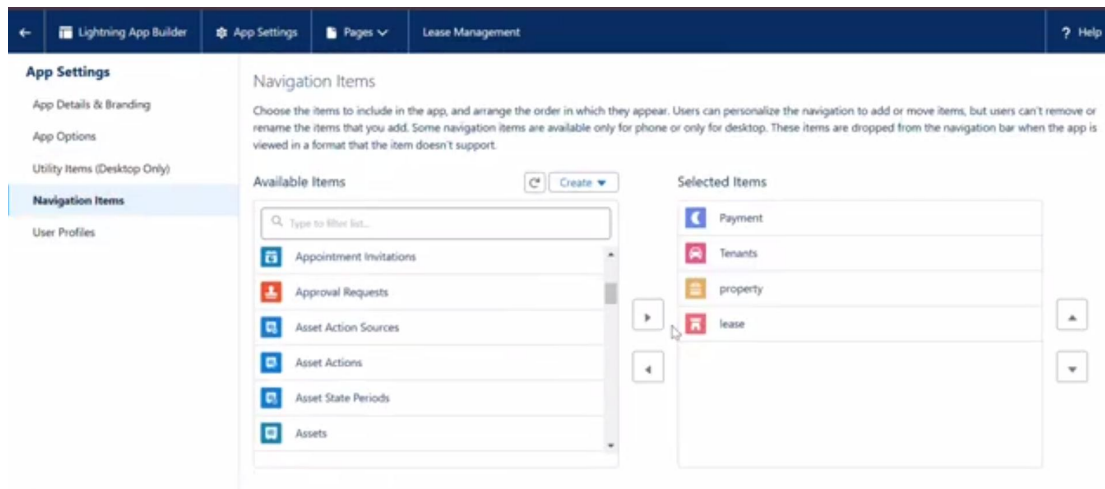


- Configured fields and relationships

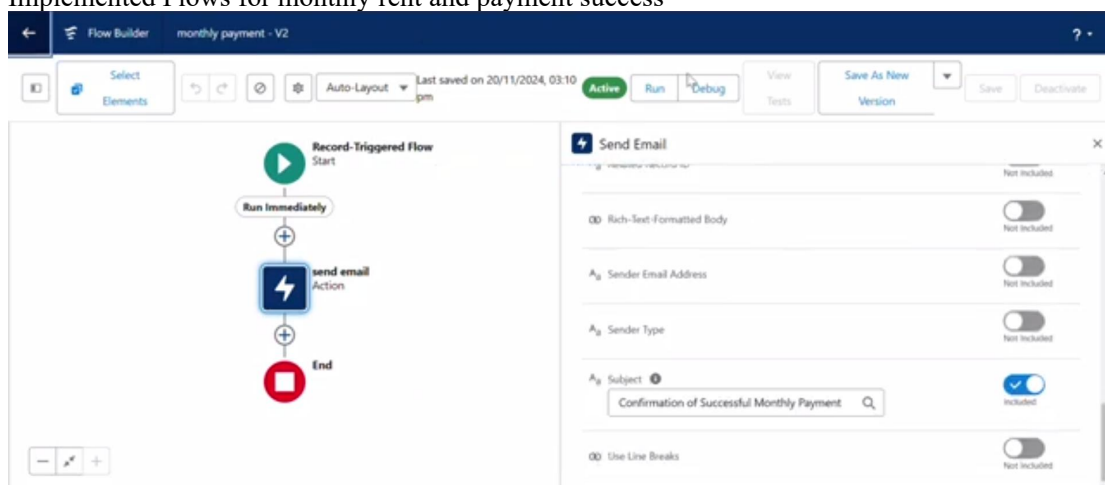


Developed Lightning App with relevant tabs

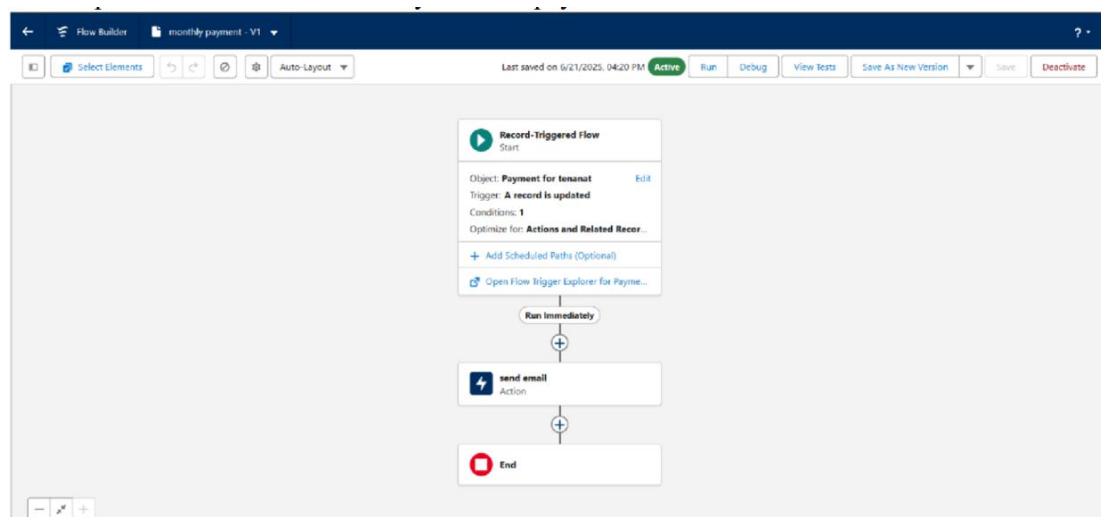


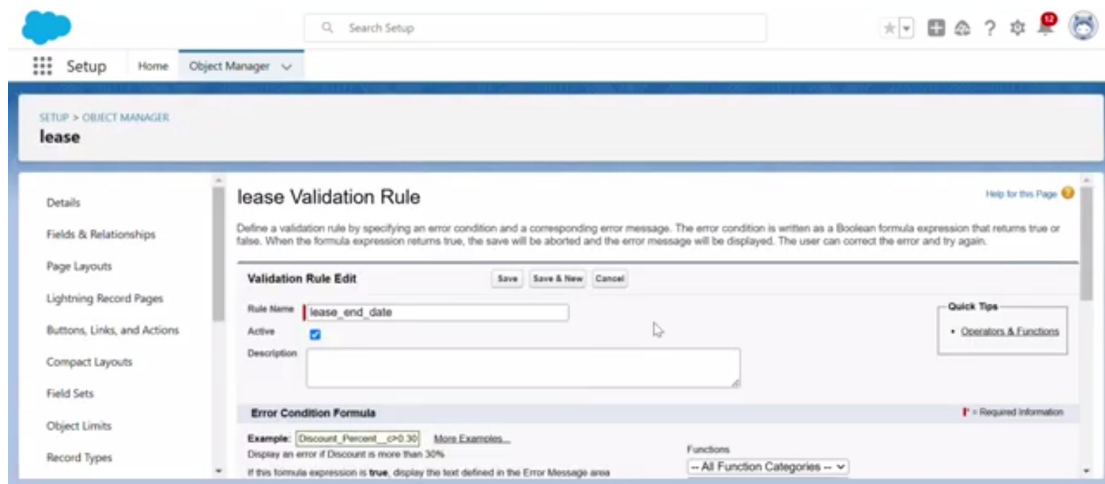


Implemented Flows for monthly rent and payment success

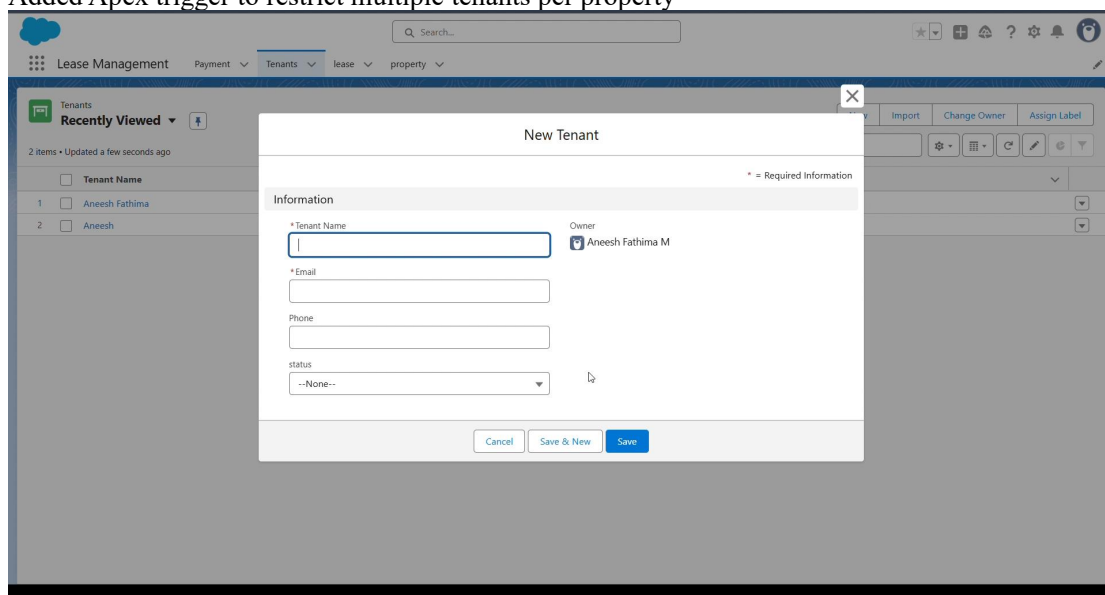


To create a validation rule to a Lease Object

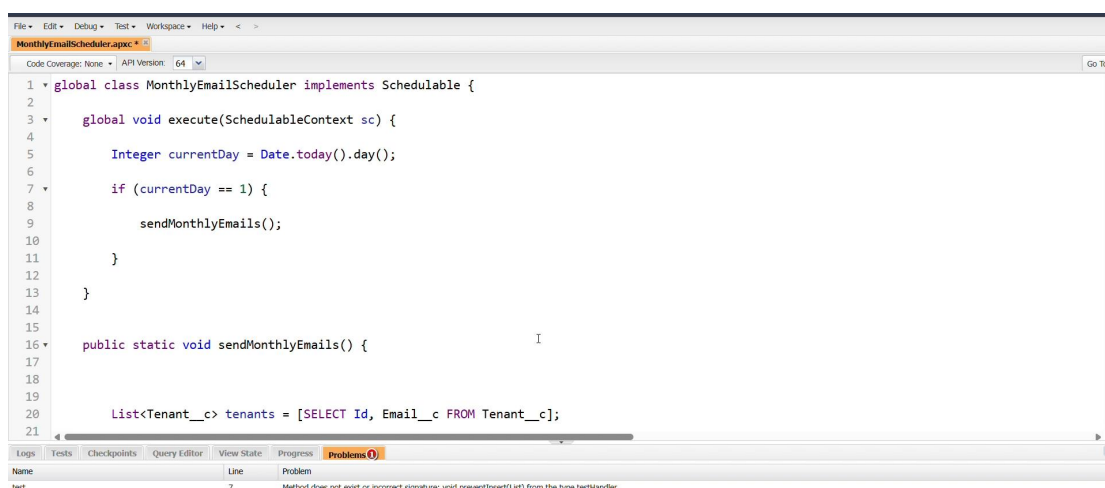




Added Apex trigger to restrict multiple tenants per property



Scheduled monthly reminder emails using Apex class



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

Setup

Home

Object Manager

Search Setup

class

Classic Email Templates

Classic Letterheads

Custom Code

Apex Classes

Data Classification

Data Classification Download

Data Classification Settings

Data Classification Upload

Didn't find what you're looking for?

Try using Global Search.

Classic Email Templates

Email Template

New Template

Step 1. Email Template: New Template

Choose the type of email template you would like to create.

☒ Text

☐ HTML (using Classic Letterhead)

☐ Custom (without using Classic Letterhead)

☐ Visualforce

Next

Cancel

Approval Process creation For Tenant Leaving:

Setup

Home

Object Manager

Search Setup

approval

Mass Transfer Approval Requests

Feature Settings

Approval Settings

Process Automation

Approval Processes

Didn't find what you're looking for?

Try using Global Search.

Approval Processes

New Approval Process

Tenants

Step 2. Specify Entry Criteria

If only certain types of records should enter this approval process, enter that criteria below. For example, only expense reports from employees at headquarters should use this approval process.

Specify Entry Criteria

Use this approval process if the following criteria are met

Field	Operator	Value	
Tenant: status	not equal to		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		

Add Filter Logic...

Previous

Next

Cancel

For Check for vacant:

Setup

Home

Object Manager

Search Setup

approval

Mass Transfer Approval Requests

Feature Settings

Approval Settings

Process Automation

Approval Processes

Didn't find what you're looking for?

Try using Global Search.

Approval Processes

Approval Process Management: Create Approval Process

Initial Submission By Tenant Owner

Created By Anesh Fathima M. 12/09/2025, 11:15 am

Modified By Anesh Fathima M. 12/09/2025, 11:16 am

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited

Approval Steps

You have not yet defined any approval steps

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited

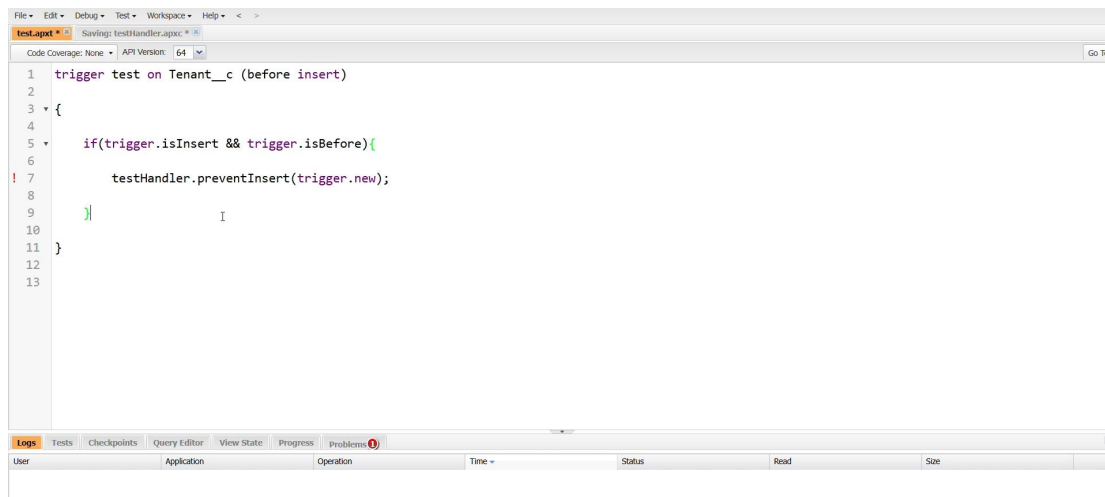
Final Rejection Actions

Action	Type	Description
Record Lock		Unlock the record for editing

Recall Actions

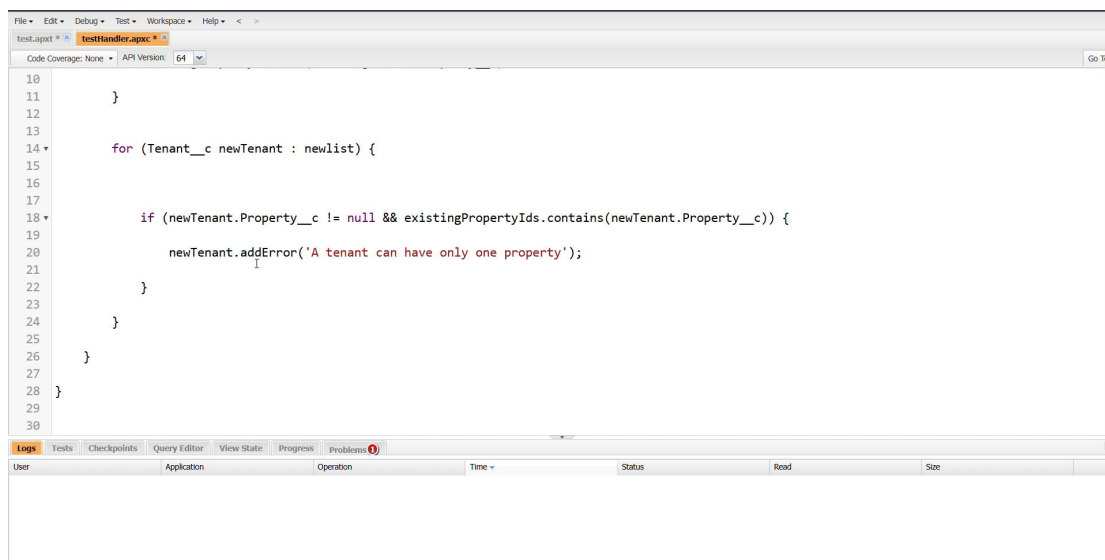
Action	Type	Description
--------	------	-------------

Apex Trigger Create an Apex Trigger



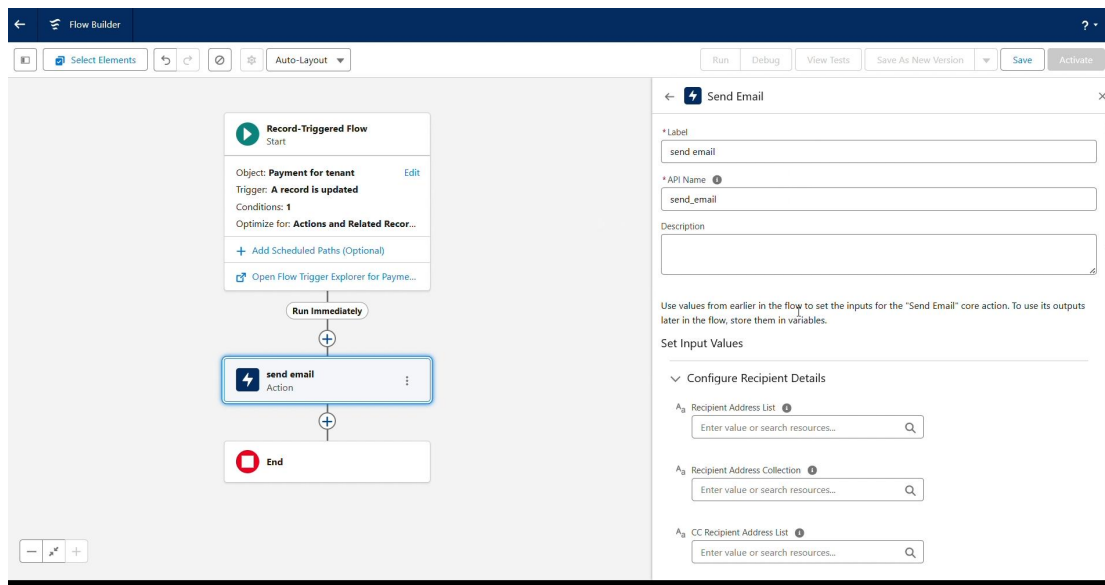
```
1 trigger test on Tenant__c (before insert)
2 {
3   {
4     if(trigger.isInsert && trigger.isBefore){
5       testHandler.preventInsert(trigger.new);
6     }
7   }
8 }
9
10
11
12
13
```

Create an Apex Handler class



```
10
11 }
12
13
14 for (Tenant__c newTenant : newList) {
15
16
17
18   if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
19     newTenant.addError('A tenant can have only one property');
20   }
21 }
22
23 }
24
25 }
26
27 }
28
29 }
30
```

FLOWS



The screenshot shows the Salesforce Flow Builder interface. On the left, a flow diagram is visible with the following steps:

- Record-Triggered Flow** (Start)
- Object: **Payment for tenant**
- Trigger: **A record is updated**
- Conditions: **1**
- Optimize for: **Actions and Related Records**
- Run Immediately** (Intermediate Step)
- send email** (Action)
- End** (Final Step)

On the right, the **Send Email** action configuration panel is open. It includes the following options:

- Configure Email Content**
 - Compose Email Content** (Selected): Set email details and compose message.
 - Use Email Template**: Select an existing email template.
- Subject**: Enter value or search resources...
- Body**: Insert a resource... or View as Rich Text
- Formatting**: Includes a font family dropdown (Salesforce Sans), font size (12), and a color picker. Below these are buttons for Bold (B), Italic (I), Underline (U), and text alignment (Left, Center, Right, Justify).

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     public static void sendMonthlyEmails() {
15
16         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
17
18     }
19
20 }
21

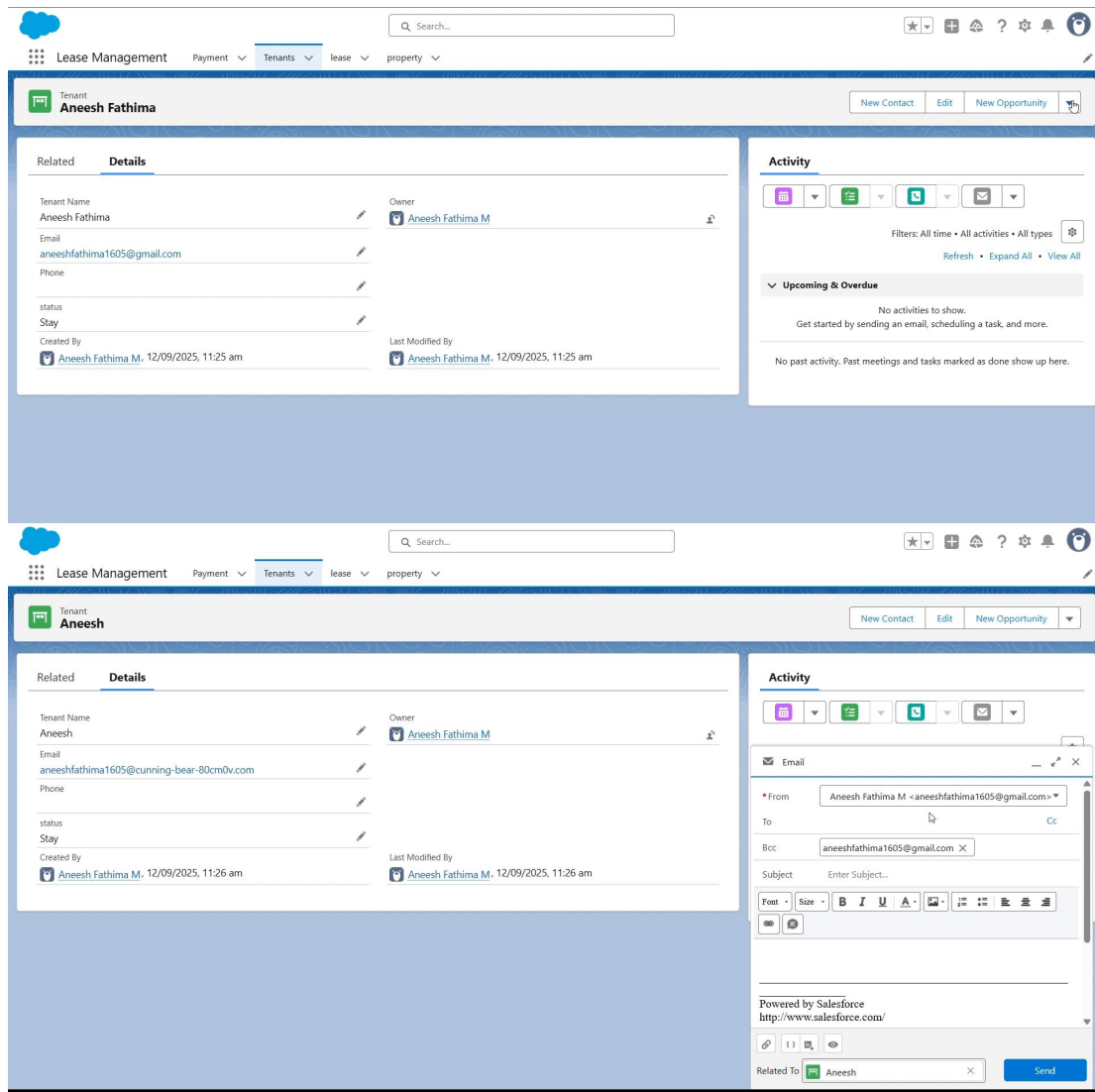
```

The screenshot shows the Salesforce Setup page for scheduling an Apex class. The page title is **Schedule Apex**. Below the title, there is a description: "Schedule an Apex class that implements the Schedulable interface to be automatically executed on a specified interval."

The configuration form includes the following fields and options:

- Job Name**: MonthlyEmailScheduler
- Apex Class**: MonthlyEmailScheduler
- Schedule Using**: ☒ Schedule Builder ☐ Cron Expression
- Frequency**: ☒ Weekly ☒ Monthly
 - On day**: 1 of every month
 - On the 1st**: Sunday of every month
- Start**: 12/09/2025 [12/09/2025]
- End**: 12/10/2025 [12/09/2025]
- Preferred Start Time**: --None--

At the bottom of the form, there is a note: "Exact start time will depend on job queue activity."



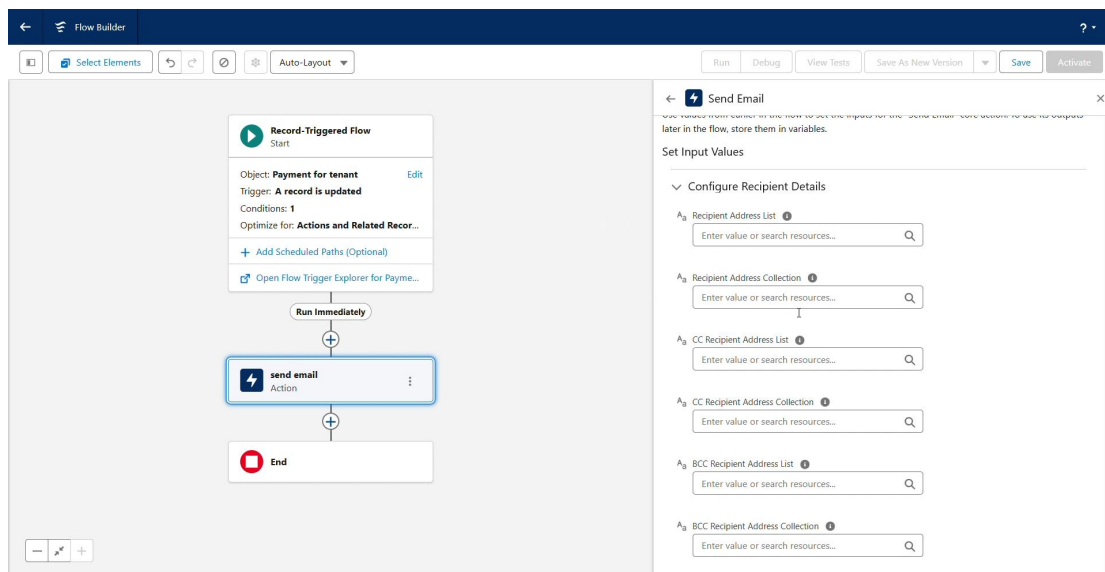
FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Trigger validation by entering duplicate tenant-property records

The screenshot shows a web application interface for 'Lease Management'. A 'New Tenant' modal form is open. The form has a title bar with a close button. Below the title, there's a section for 'Information' with a legend indicating that fields with an asterisk (*) are required. The form contains the following fields: 'Tenant Name' (required, with the value 'Aneesh'), 'Email' (required, empty), 'Phone' (empty), and 'status' (a dropdown menu with '--None--' selected). There's also an 'Owner' field showing a user profile for 'Aneesh Fathima M'. At the bottom of the form are three buttons: 'Cancel', 'Save & New', and 'Save'. In the background, a table titled 'Recently Viewed' is visible, showing one item: 'Aneesh Fathima'.

Test flows on payment update



Approval process validated through email alerts and status updates

Lease Management

Payment

Tenants

lease

property

Search...

Star

Grid

Home

Help

Settings

Notifications

User

Tenant

Aneesh

Email was sent.

New Contact

Edit

New Opportunity

Related

Details

Tenant Name

Aneesh

Email

aneeshfathima1605@cunning-bear-80cm0v.com

Phone

status

Stay

Created By

Aneesh Fathima M. 12/09/2025, 11:26 am

Owner

Aneesh Fathima M.

Last Modified By

Aneesh Fathima M. 12/09/2025, 11:26 am

Activity

Calendar

Tasks

Chats

Messages

Filters: All time • All activities • All types

Refresh • Expand All • View All

Upcoming & Overdue

No activities to show.

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

September • 2025

This Month

leave

11:52 AM | Today

You sent an email to aneeshfathima1605@cunning-bear-80cm0...

RESULTS Output Screenshots

Setup

Home

Object Manager

Search Setup

Star

Grid

Home

Help

Settings

Notifications

User

Q tabs

User Interface

Rename Tabs and Labels

Tabs

Didn't find what you're looking for?

Try using Global Search.

SETUP

Tabs

Custom Tabs

Help for this Page

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

New What Is This?

Action	Label	Tab Style	Description
Edit Del	lease	COMPOUND	
Edit Del	Payment	Camera	
Edit Del	property	Diamond	
Edit Del	Tenants	Desk	

Web Tabs

New What Is This?

No Web Tabs have been defined

Visualforce Tabs

New What Is This?

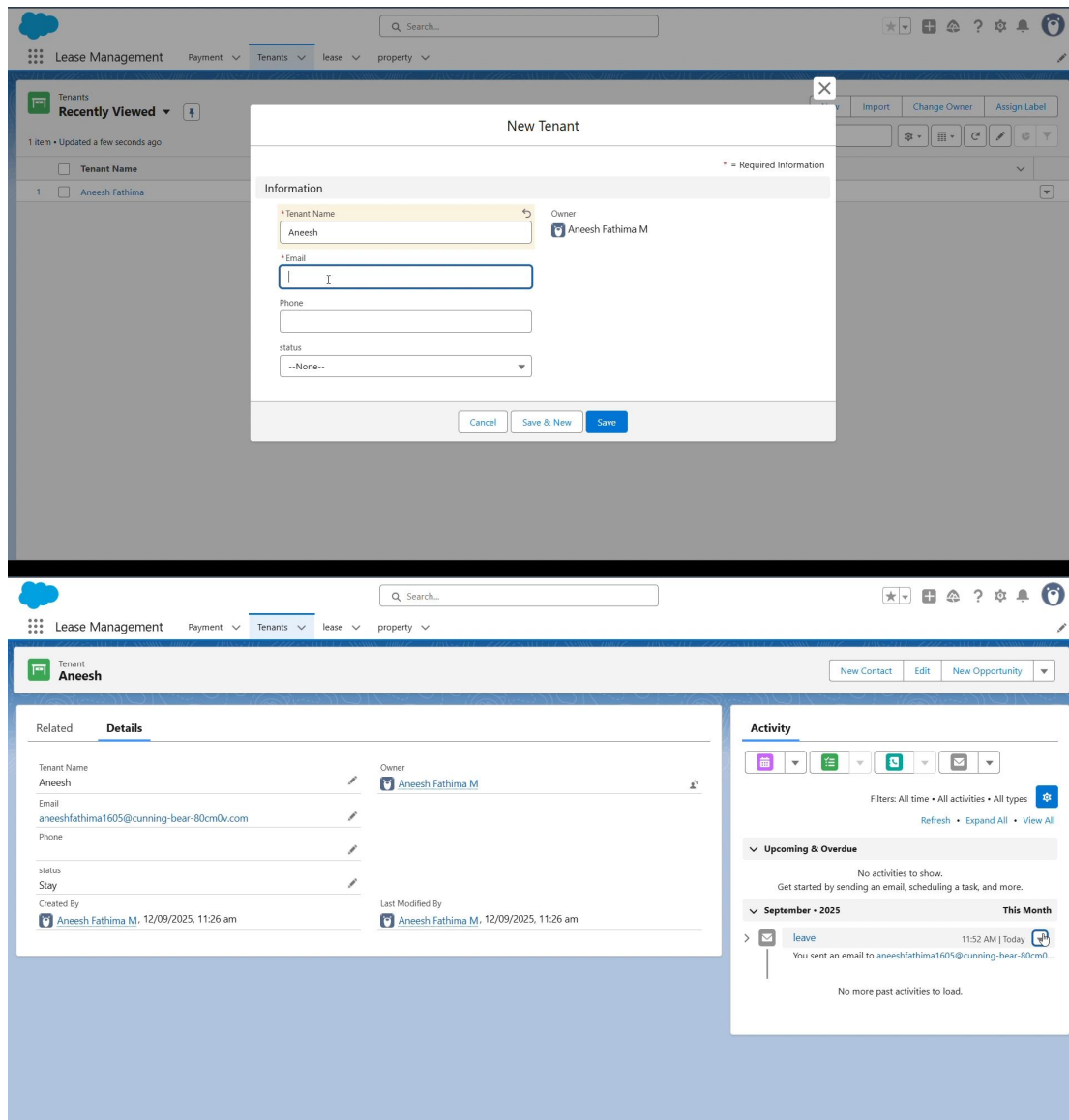
No Visualforce Tabs have been defined

Lightning Component Tabs

New What Is This?

Action	Label	Tab Style	Description
--------	-------	-----------	-------------

Trigger error messages



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 existingPropertyIds = new Set() 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');
            }
        }
    }
}

```

```

MonthlyEmailScheduler.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 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});
        }
    }
}

```