

LEASE MANAGEMENT

College Name: Sree Narayana Guru College, Coimbatore, Chavadi

College Code: BRU36

TEAM ID: NM2025TMID26568

TEAM MEMBERS:

Team LeaderName: NANDHITHA A C

Email: ambunandhitha@gmail.com

Team Member1: MUHAMMAD ALFAS P A

Email: alfasmuhammad312@gmail.com

Team Member: MUHAMMED SAALIM K T

Email: muhammedsaalimkt@gmail.com

Team Member: MUHAMMED ADHIL M

Email: mohammedadhil1012@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.



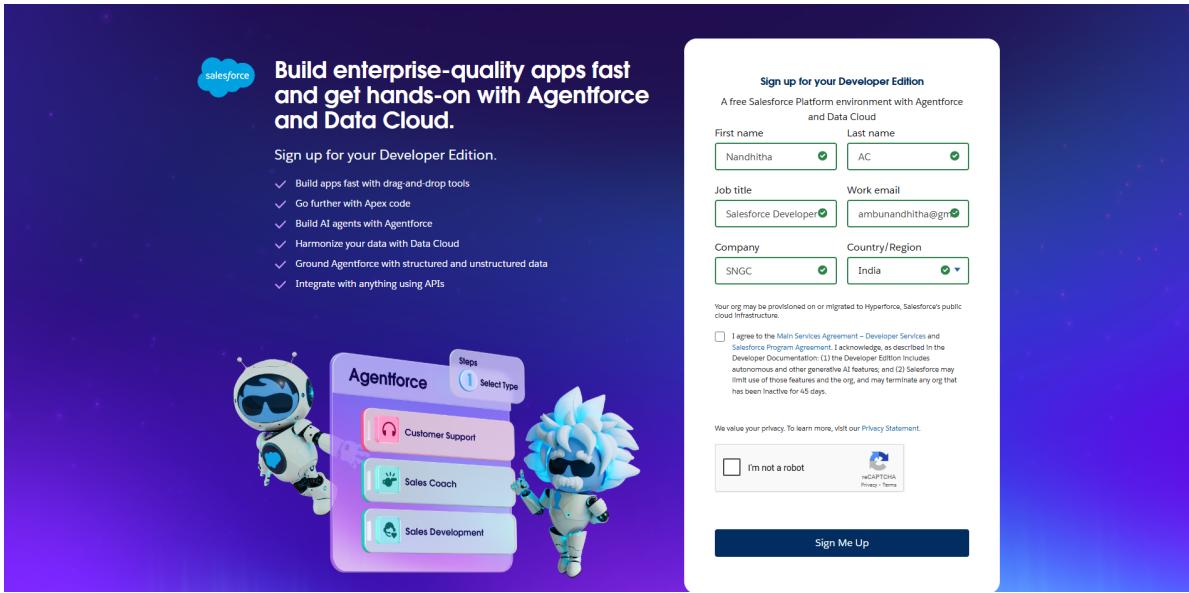
1.2 Purpose

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

Recently Viewed | Payment for | Tenant | Salesforce

orgfarm-46a3ec6bd-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gK00000273xh/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER Tenant

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

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

Object Access

Triggers

Flow Triggers

Validation Rules

Conditional Field Formatting

2:30 AM 9/16/2023

Recently Viewed | Payment for | lease | Salesforce

orgfarm-46a3ec6bd-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gK00000274hv/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER lease

Details

Description

API Name: lease__c

Custom

Singular Label: lease

Plural Label: lease

Enable Reports

✓ Track Activities

✓ Track Field History

✓ Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Flow Triggers

Validation Rules

Conditional Field Formatting

2:30 AM 9/16/2023

- Configured fields and relationships

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)		
Type	Type__c	Picklist		

Recently Viewed | Payment for tenant | Salesforce

orgfarm-46a3ec6bad-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01gK00000274FR/FieldsAndRelationships/view

Setup Home Object Manager

Search Setup

Payment for tenant

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

Object Access

Triggers

Flow Triggers

Validation Rules

Conditional Field Formatting

Fields & Relationships

8 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)		
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)		
property	property_c	Master-Detail(Property)		
Tenant	Tenant_c	Lookup(Tenant)		

Quick Find New Deleted Fields Field Dependencies Set History Tracking

2:32 AM 9/18/2023

Recently Viewed | lease | Salesforce

orgfarm-46a3ec6bad-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01gK00000274FR/FieldsAndRelationships/view

Setup Home Object Manager

Search Setup

lease

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Fields & Relationships

7 Items, Sorted by Field Label

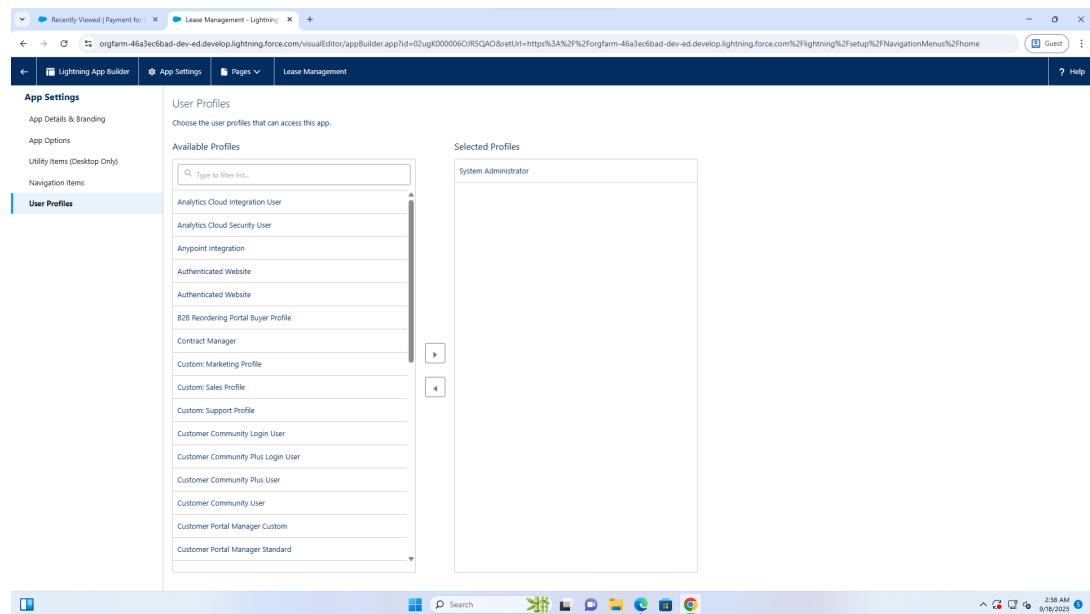
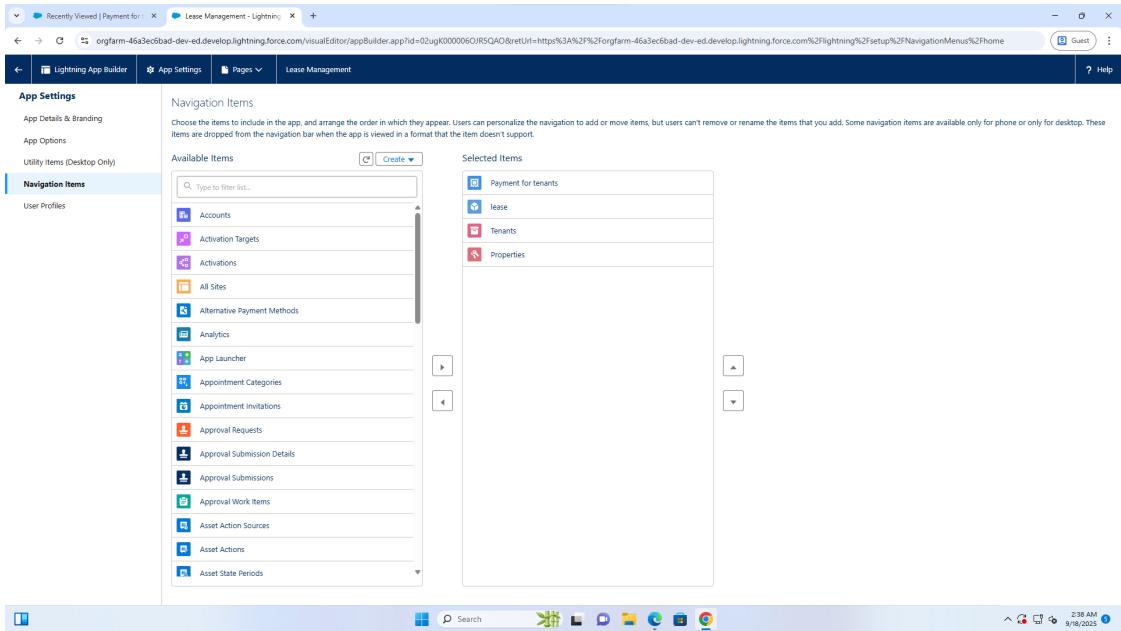
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		
Owner	OwnerId	Lookup(User/Group)		
property	property_c	Lookup(property)		
start date	start_date_c	Date		

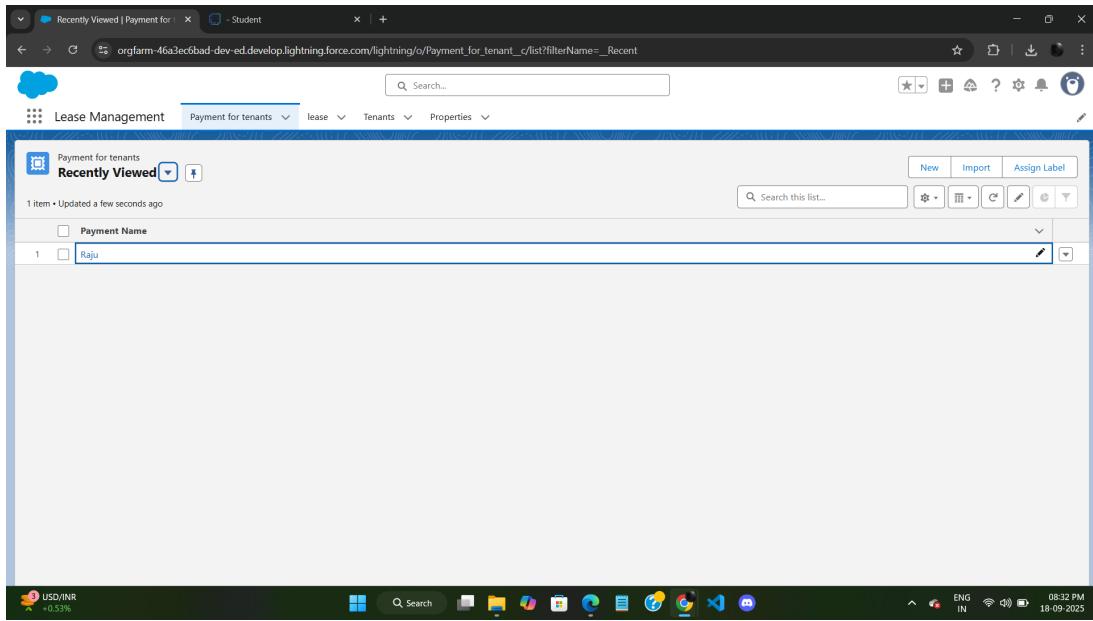
Quick Find New Deleted Fields Field Dependencies Set History Tracking

The screenshot shows the Salesforce Setup interface under the Object Manager tab for the Tenant object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled 'Fields & Relationships' and displays a table with 7 items, sorted by Field Label. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The rows show fields such as Created By, Email, Last Modified By, Phone, Property, status, and Tenant Name.

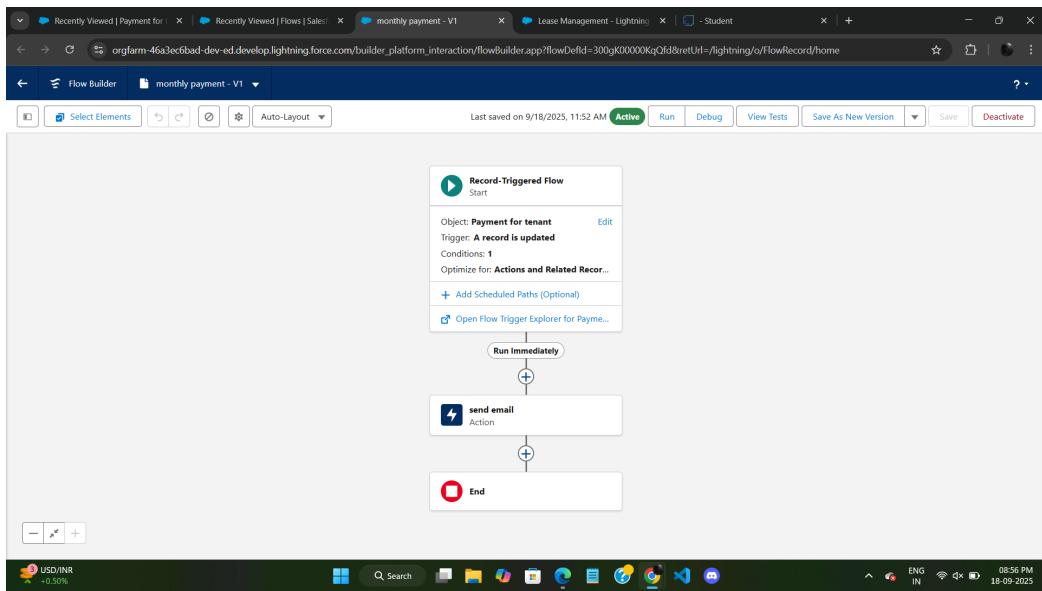
- Developed Lightning App with relevant tabs

The screenshot shows the Lightning App Builder interface with the 'Lease Management - Lightning' tab selected. The left sidebar shows 'App Settings' and 'App Details & Branding'. The main content area is titled 'App Details & Branding' and includes sections for 'App Details' and 'App Branding'. In the 'App Details' section, fields are filled with 'Lease Management' for App Name and Developer Name. In the 'App Branding' section, there is an 'Image' field with an upload button, a color picker set to #0070D2, and a checkbox for 'Org Theme Options'. Below these, an 'App Launcher Preview' shows a blue button labeled 'LM' and a label 'Lease Management'.

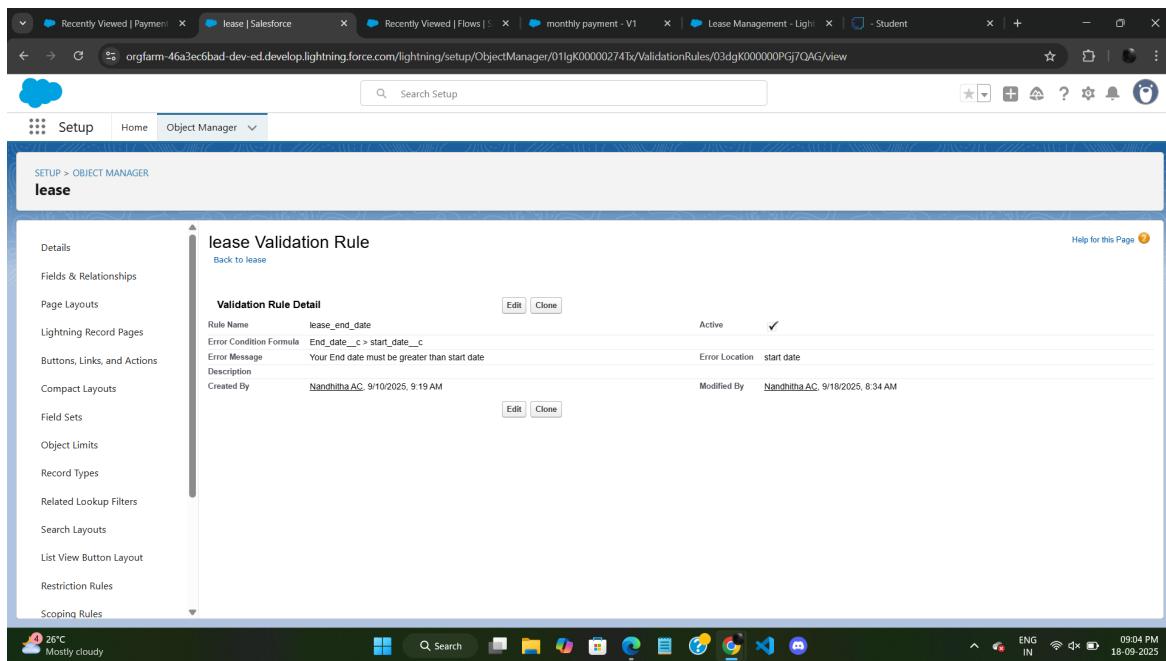
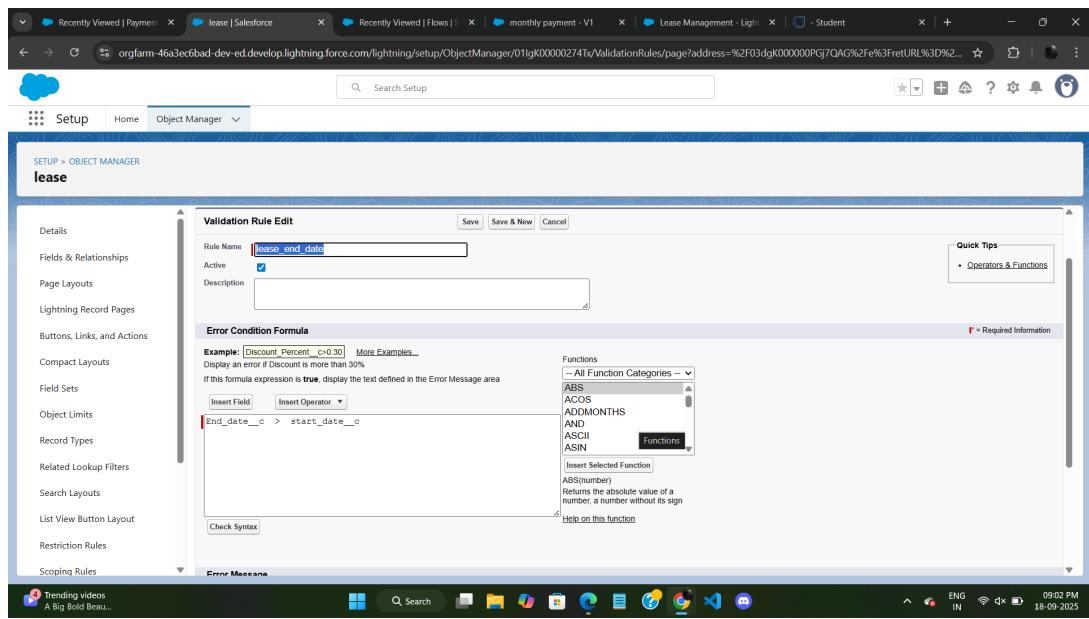




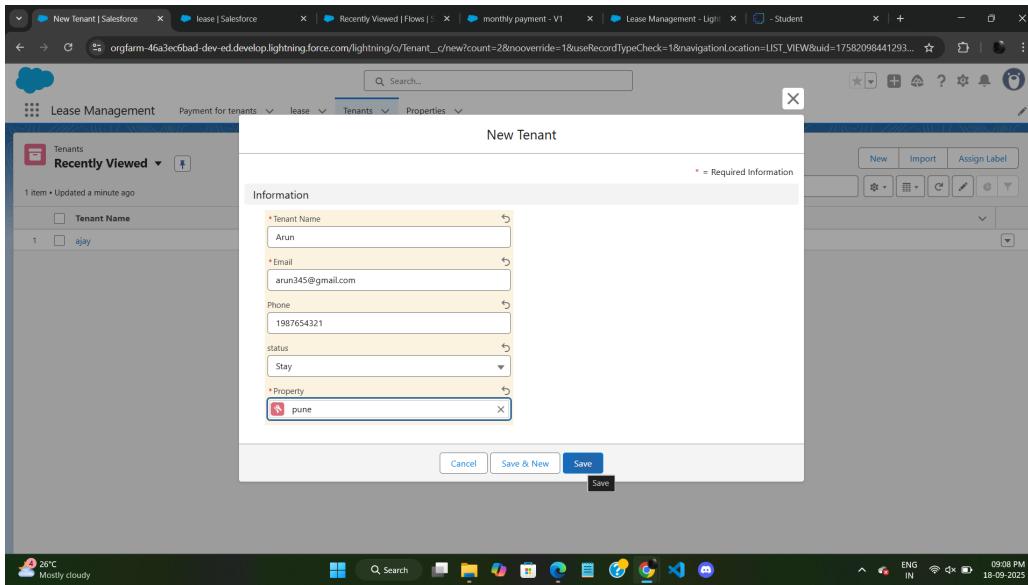
- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object



- Added Apex trigger to restrict multiple tenants per property

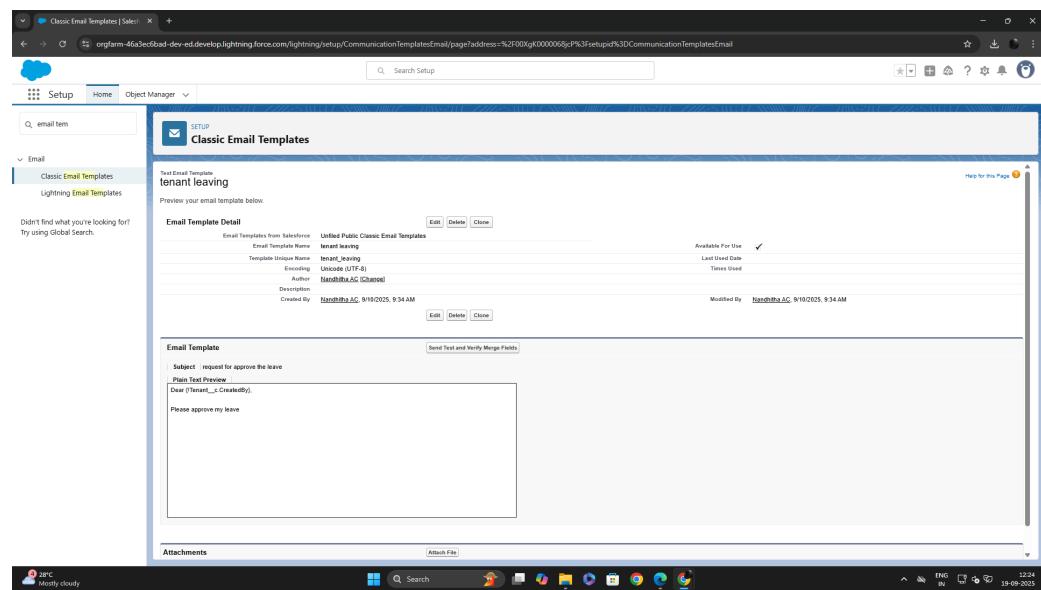
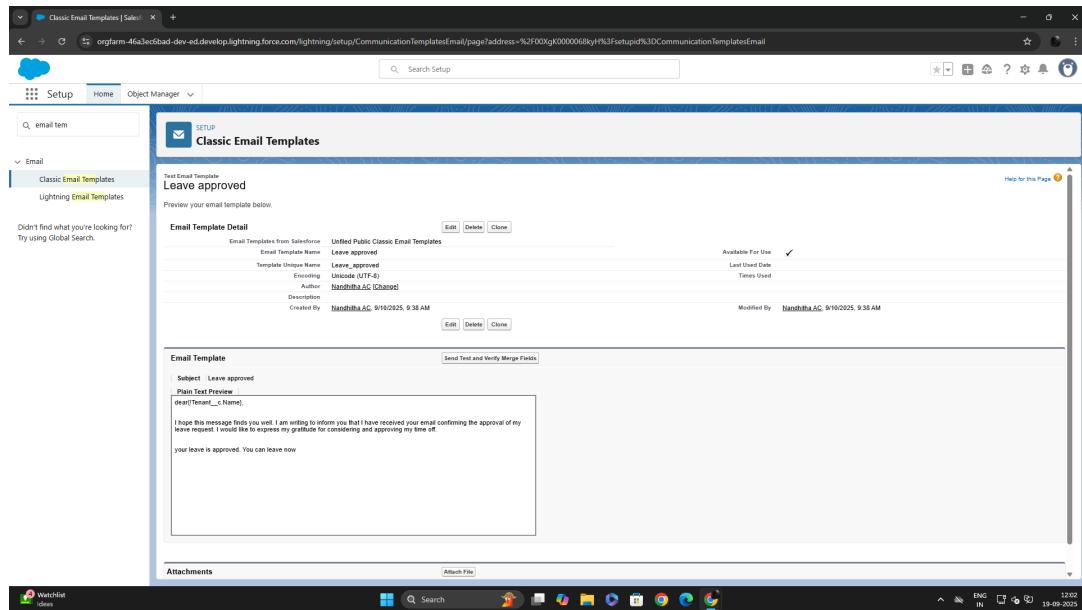


- Scheduled monthly reminder emails using Apex class

```
15
16  public static void sendMonthlyEmails() {
17
18
19
20      List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
21
22
23
24      for (Tenant__c tenant : tenants) {
25
26          String recipientEmail = tenant.Email__c;
27
28          String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment is appreciated.';
29
30          String emailSubject = 'Reminder: Monthly Rent Payment Due';
31
32
33          Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
34
35          email.setTo(recipientEmail);
36          email.setSubject(emailSubject);
37          email.setPlainTextEmailContent(emailContent);
38
39          Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{email});
```

A screenshot of the Salesforce Developer Console. It shows an Apex class named 'MonthlyEmailScheduler.apex'. The code within the class is annotated with comments explaining its purpose. The comments describe the selection of tenant records, the iteration through them to build an email message, and the final step of sending the email using the 'Messaging.SingleEmailMessage' class. The developer console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Problems tab currently selected.

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



Classic Email Templates | Sales

orgfarm-46a3ec6bad-dev-edt.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2FOXgkC0000068Nl%3FsetupId%3DCommunicationTemplateEmail

Setup Home Object Manager

Search Setup

Classic Email Templates

Leave rejected

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Nanditha.AC [Change]
Description	
Created By	Nanditha.AC 9/10/2025, 9:40 AM
Modified By	Nanditha.AC 9/10/2025, 9:40 AM

Available For Use Last Used Date Times Used

Email Template

Send Text and Verify Merge Fields

Subject: Leave rejected

Plain Text Preview

Dear {Tenant__Name},

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

Attachments

Attach File

Attachments

Attach File

12:28 19-09-2025

Classic Email Templates | Sales

orgfarm-46a3ec6bad-dev-edt.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2FOXgkC0000068Nl%3FsetupId%3DCommunicationTemplateEmail

Setup Home Object Manager

Search Setup

Classic Email Templates

Tenant Email

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Nanditha.AC [Change]
Description	
Created By	Nanditha.AC 9/10/2025, 9:42 AM
Modified By	Nanditha.AC 9/10/2025, 9:42 AM

Available For Use Last Used Date Times Used

Email Template

Send Text and Verify Merge Fields

Subject: Urgent: Monthly Rent Payment Reminder

Plain Text Preview

Dear {Tenant__Name},

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

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due . To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

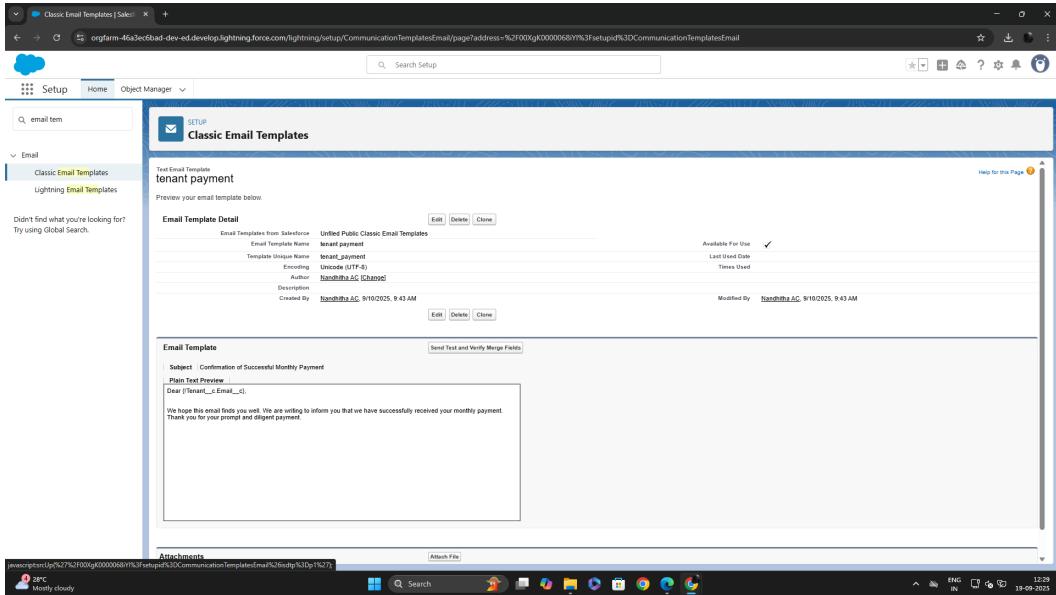
Attachments

Attach File

Attachments

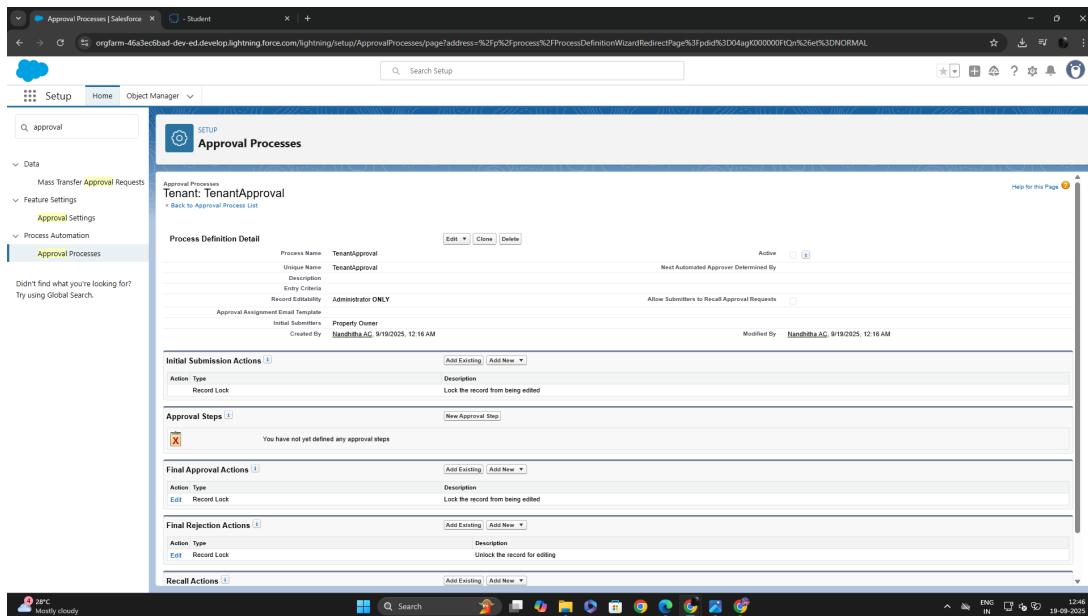
Attach File

12:28 19-09-2025

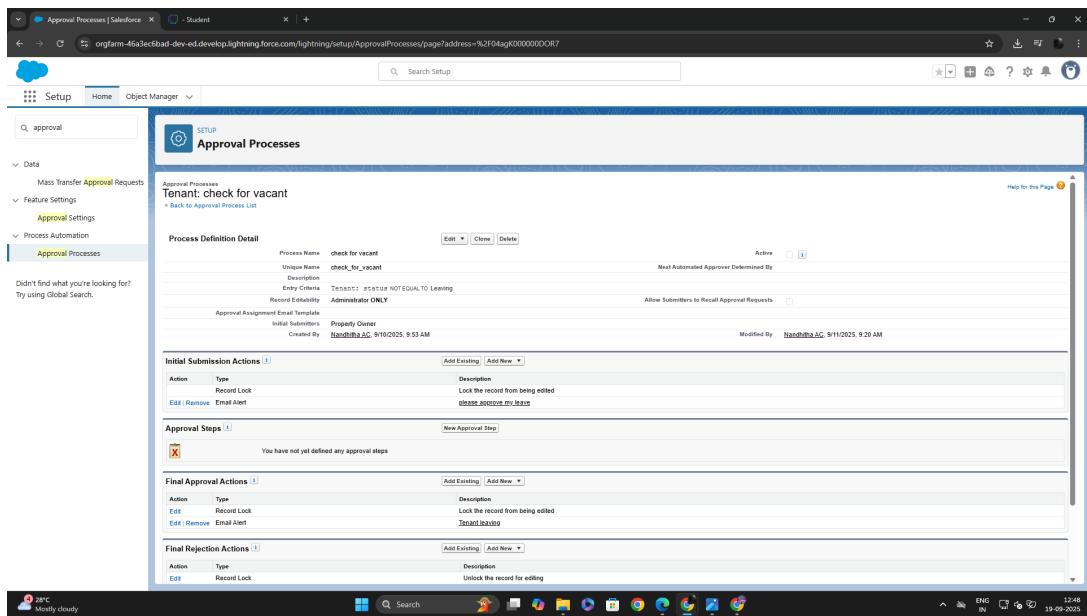


● Approval Process creation

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);
    }
}

```

The screenshot shows the Salesforce Developer Console interface. The main area displays the Apex code for a trigger test:

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

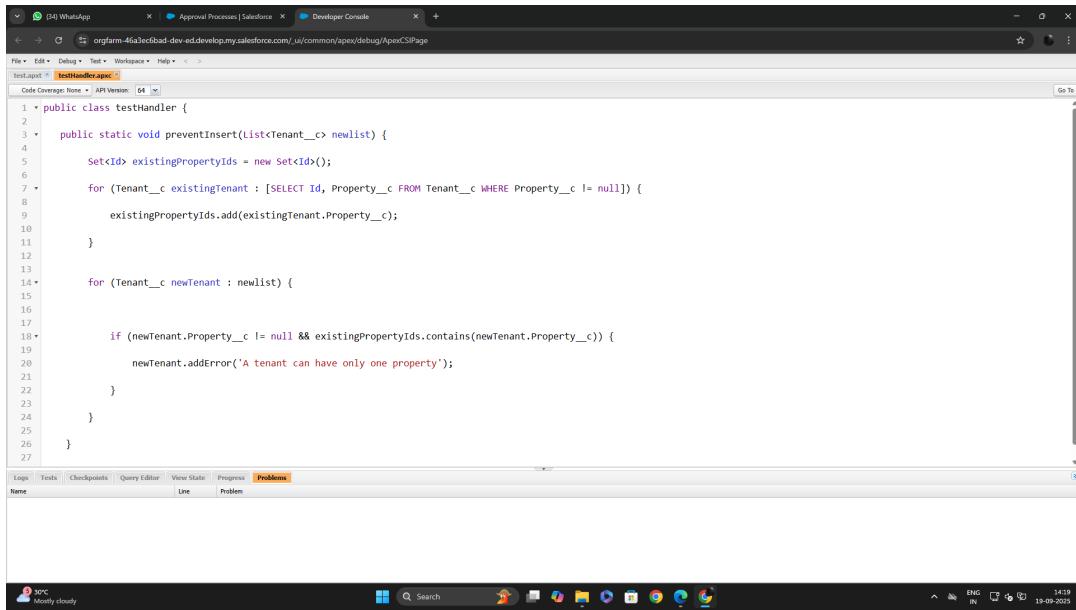
The tabs at the bottom include Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no errors. The status bar at the bottom right indicates it's 13:03 on 19-09-2025.

Create an Apex Handler class

The screenshot shows the Salesforce Developer Console interface. A new Apex class named "TestHandler" is being created:

```
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');
            }
        }
    }
}
```

A modal dialog titled "Open" is displayed, listing various entities and triggers. The "Entities" section shows "testHandler" and "MonthlyEmailScheduler". The "Triggers" section is empty. The status bar at the bottom right indicates it's 14:19 on 19-09-2025.



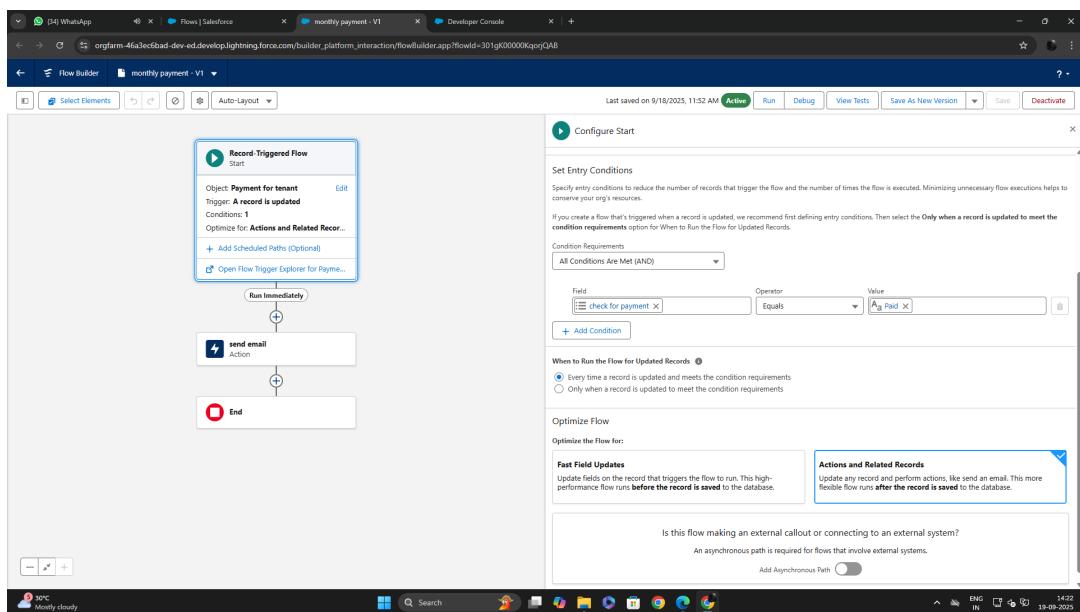
```

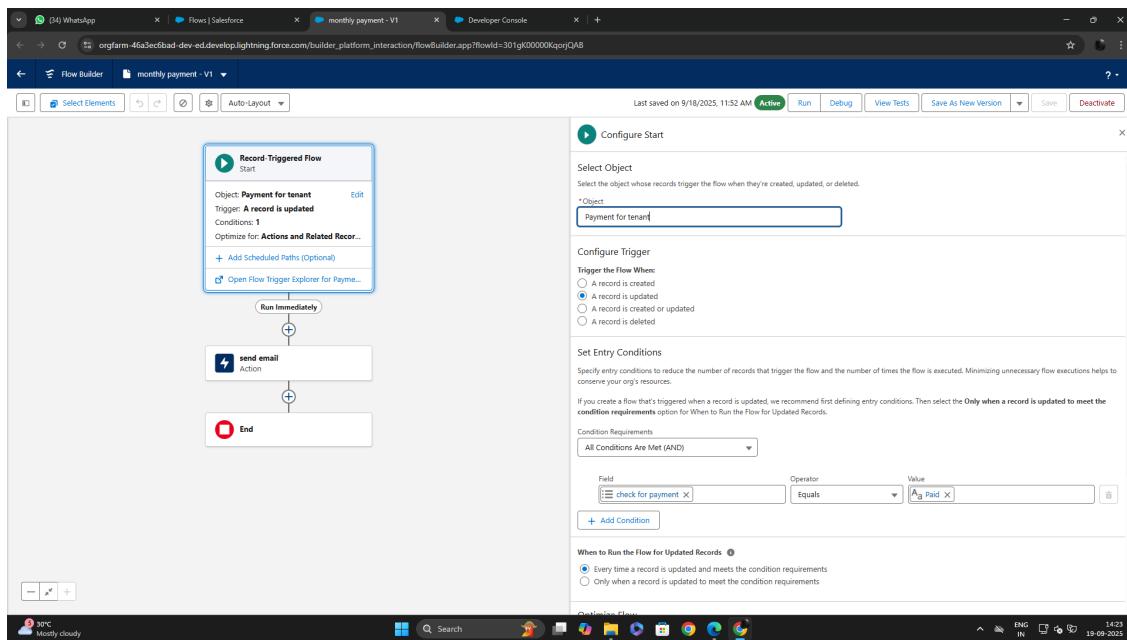
1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
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     }
26
27 }

```

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for File, Edit, Debug, Test, Workspace, and Help. Below the tabs, there's a code editor window with the file name 'testHandler.apex' and line numbers 1 through 27. The code implements a static method 'preventInsert' that takes a list of new tenants. It first retrieves all existing tenants with assigned properties from the database. Then, it iterates through the new tenants and adds an error to any new tenant that already has a property assigned. The bottom of the screen shows a toolbar with Log, Tests, Checkpoints, Query Editor, View State, Progress, and Problems tabs, with the Problems tab currently selected. The status bar at the bottom right shows the date as 19-09-2023.

● FLOWS



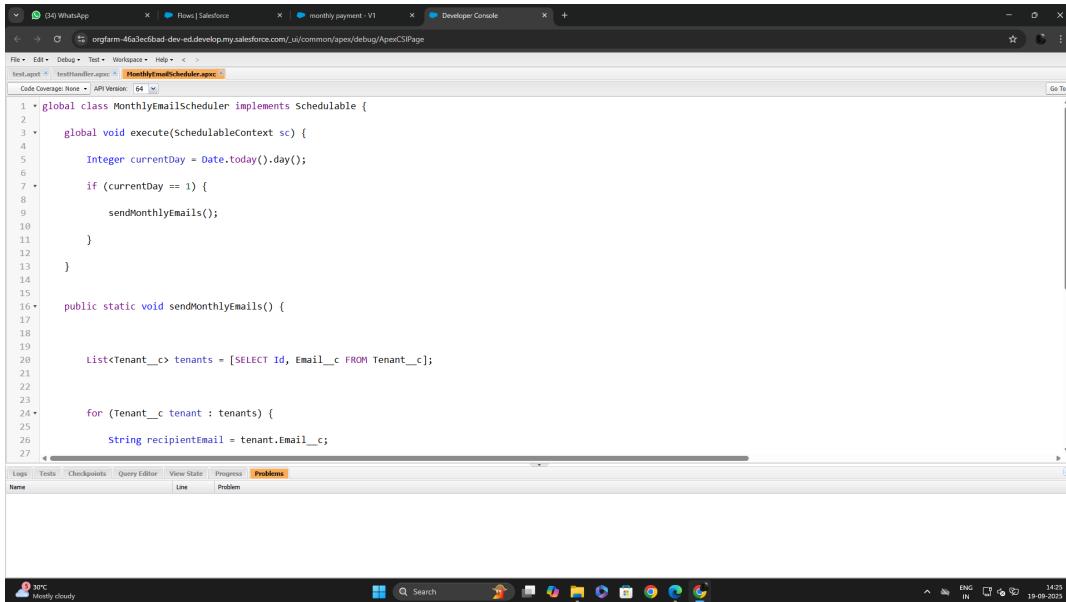


- Schedule class:
Create an Apex Class

```

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
19     List<Tenant__c> tenants = [SELECT Id, Email__c
20
21
22
23
24     for (Tenant__c tenant : tenants) {
25
26         String recipientEmail = tenant.Email__c;
27
}

```

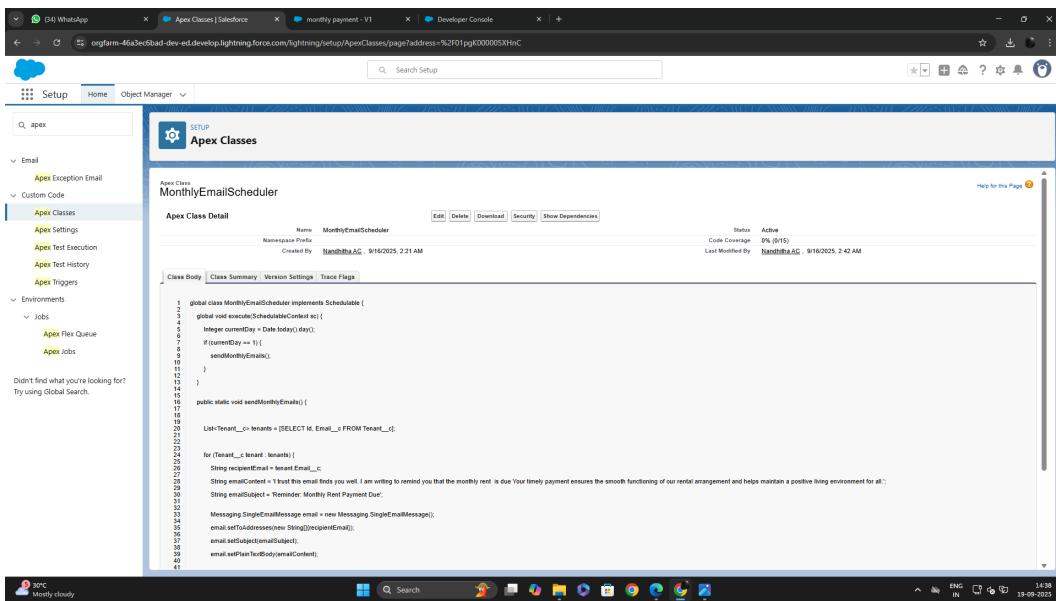


The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Flows, Apex Classes, monthly payment - V1, and Developer Console. The main area displays the code for the `MonthlyEmailScheduler` class:

```
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;
        }
    }
}
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing one error: "Name" is highlighted in red.

Schedule Apex class



The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar lists categories like Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, and Apex Triggers. The main content area shows the `MonthlyEmailScheduler` class details:

Apex Class Detail

Name	Namespace Prefix	Status	Active
MonthlyEmailScheduler		Code Coverage	0% (0/15)
		Last Modified By	Nanditha AG, 9/16/2025, 2:42 AM

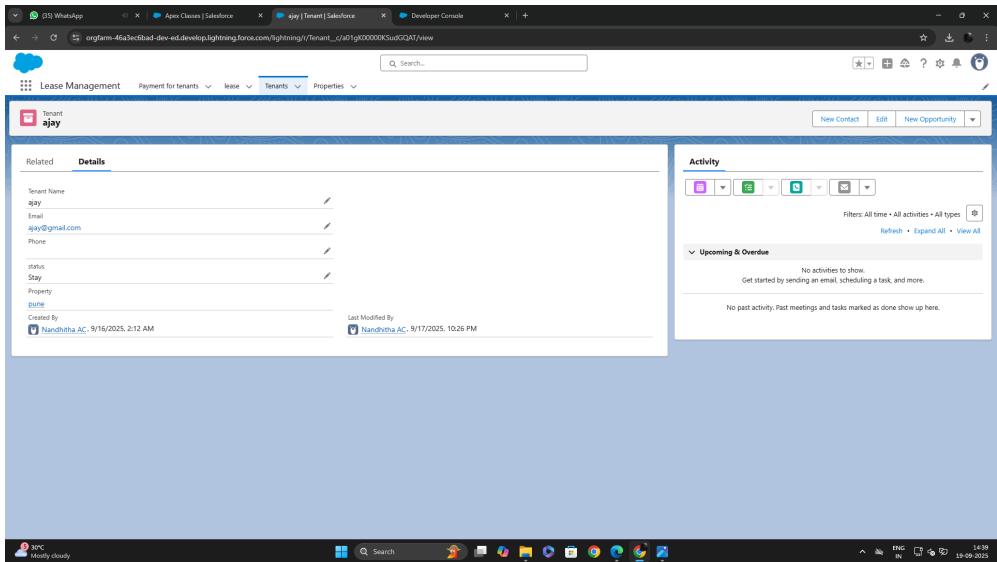
Class Body

```
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;
        }
    }
}
```

At the bottom of the page, there is a note: "Didn't find what you're looking for? Try using Global Search."

Screenshot of a Salesforce Lightning interface showing a Tenant record for 'ajay'. The Details tab is selected, displaying fields like Tenant Name, Email, Phone, status, Stay, and Property. The Activity sidebar shows no upcoming or overdue activities.

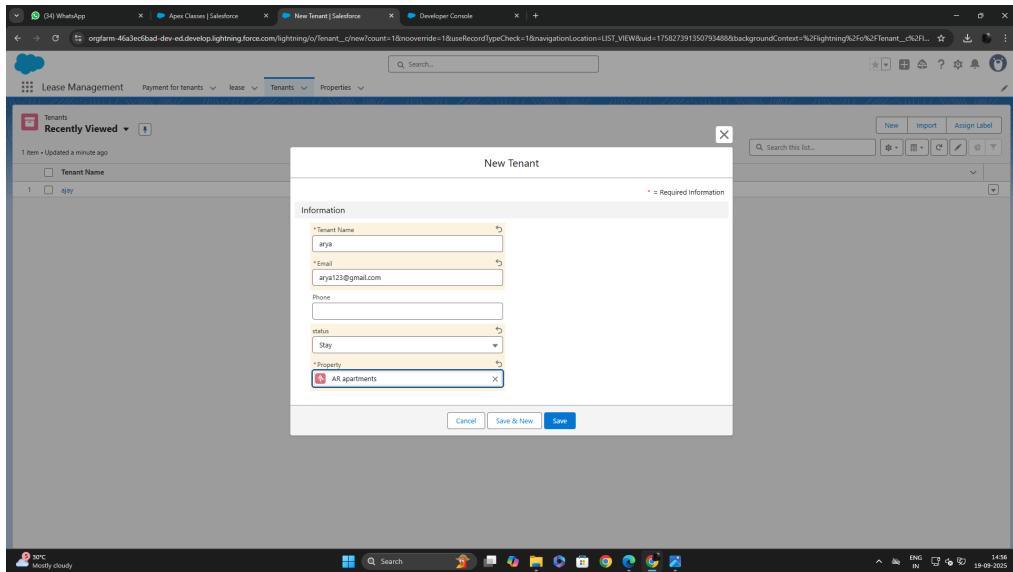
Screenshot of a Salesforce Lightning interface showing a Tenant record for 'ajay'. A success message 'Tenant was submitted for approval.' is displayed above the Details tab. The Details tab shows the same information as the previous screenshot. The Activity sidebar shows no upcoming or overdue activities.



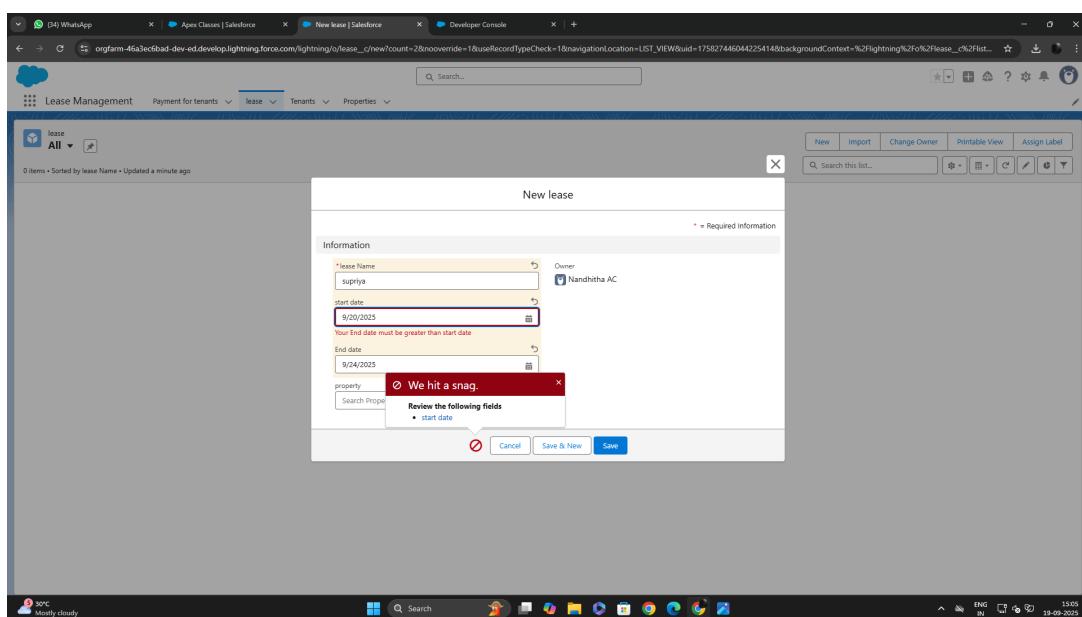
FUNCTIONAL AND PERFORMANCE TESTING

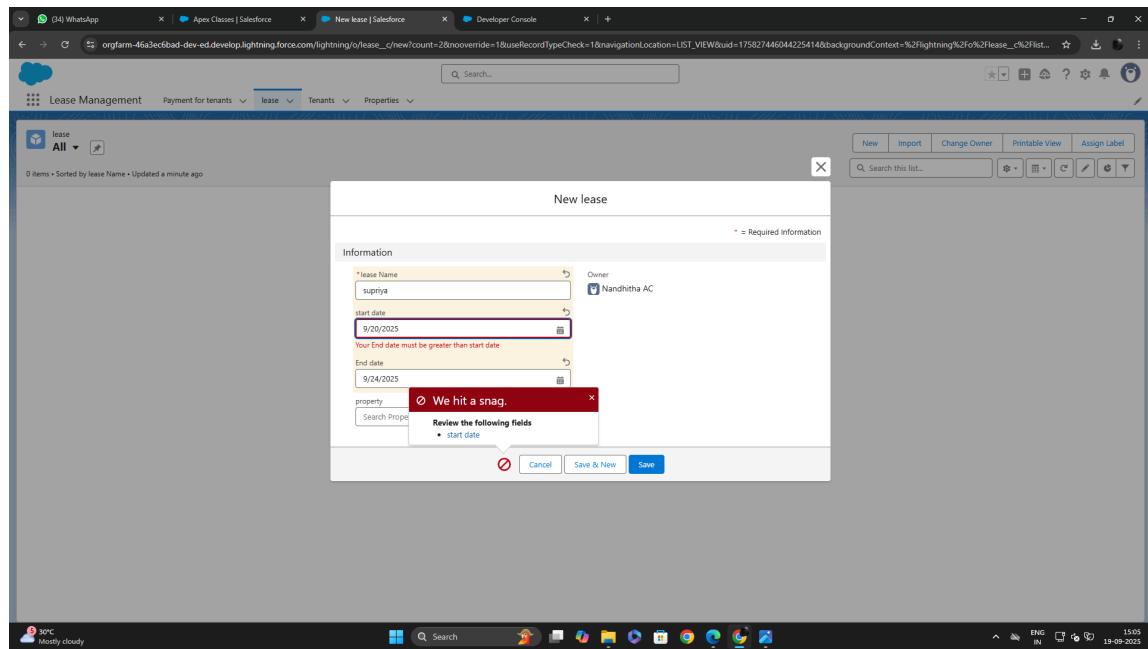
Performance Testing

- Trigger validation by entering duplicate tenant-property records

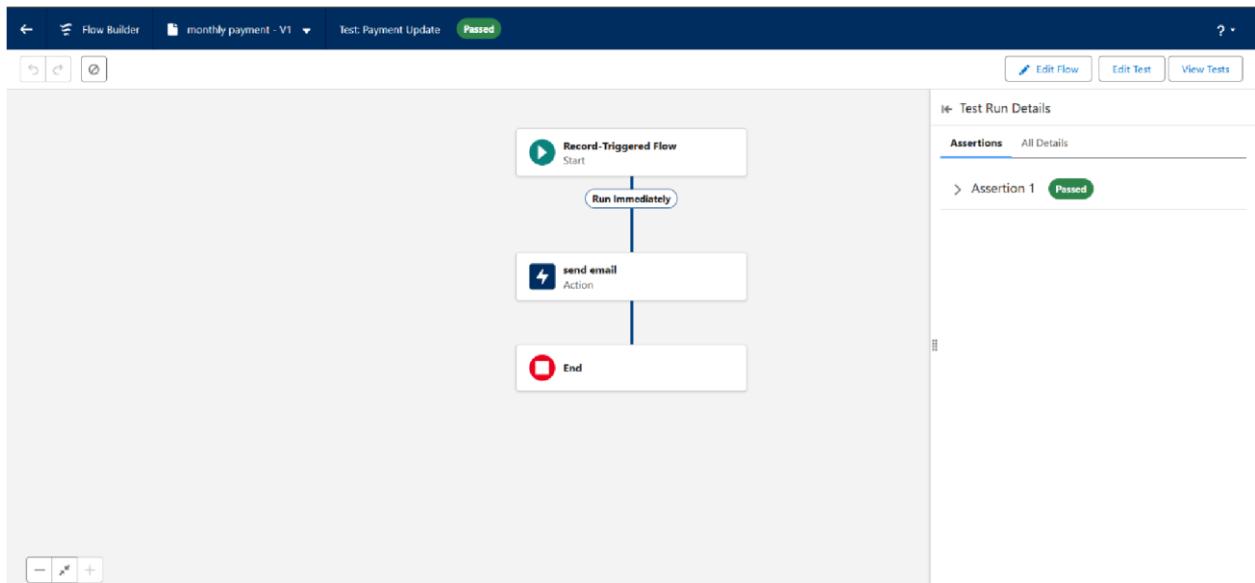


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

RESULTS

Output Screenshots

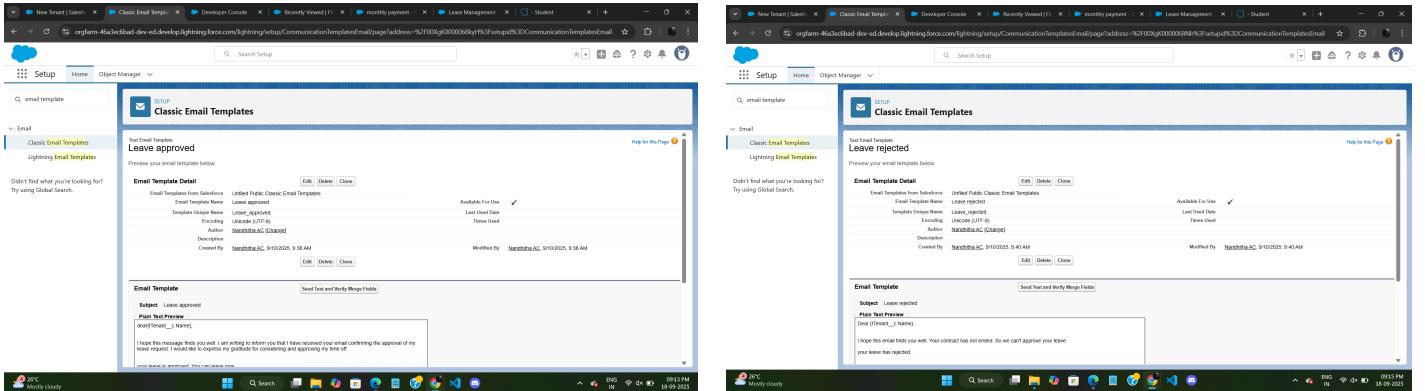
- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The left sidebar includes sections like Feature Settings, Analytics, Tableau, User Interface, and Console Settings. The main content area displays the 'Custom Tabs' section, which allows creating new custom tabs to extend Salesforce functionality. It shows a table for 'Custom Object Tabs' with four entries: 'Lease' (Building Block), 'Payment for tenants' (Postage), 'Properties' (Keys), and 'Tenants' (Box). Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both currently empty. The URL in the browser bar is <https://orgfarm-46a3ec6bad-dev-ed.lightning.force.com/lightning/setup/CustomTabs/home>.

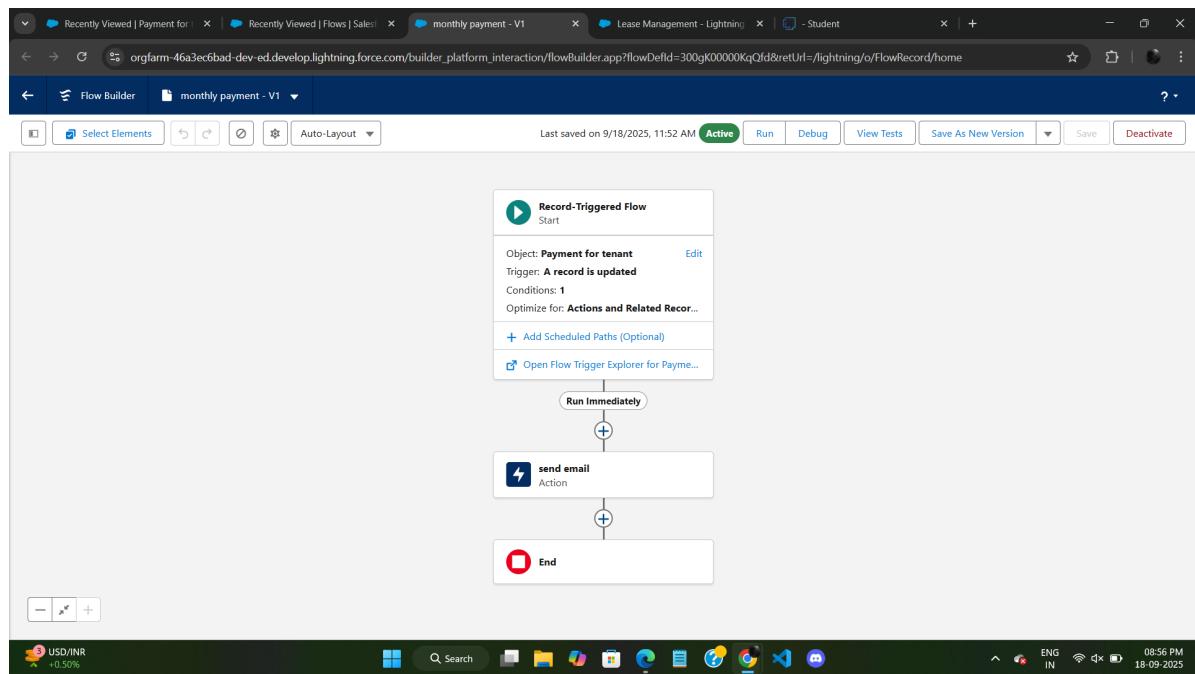
- Email Alerts

The screenshot shows the Salesforce Setup interface with the 'Email Alerts' page selected. The left sidebar includes sections like Process Automation and Workflow Actions. The main content area displays the 'All Email Alerts' section, which lists email alerts used to send emails from a flow or other automation. It shows three entries: 'please_approve_my_leave' (tenant leaving), 'Terant_leaving' (Leave approved), and 'your request for leave is rejected' (Leave rejected). The URL in the browser bar is <https://orgfarm-46a3ec6bad-dev-ed.lightning.force.com/lightning/setup/WorkflowEmails/page?address=%2F01W>.

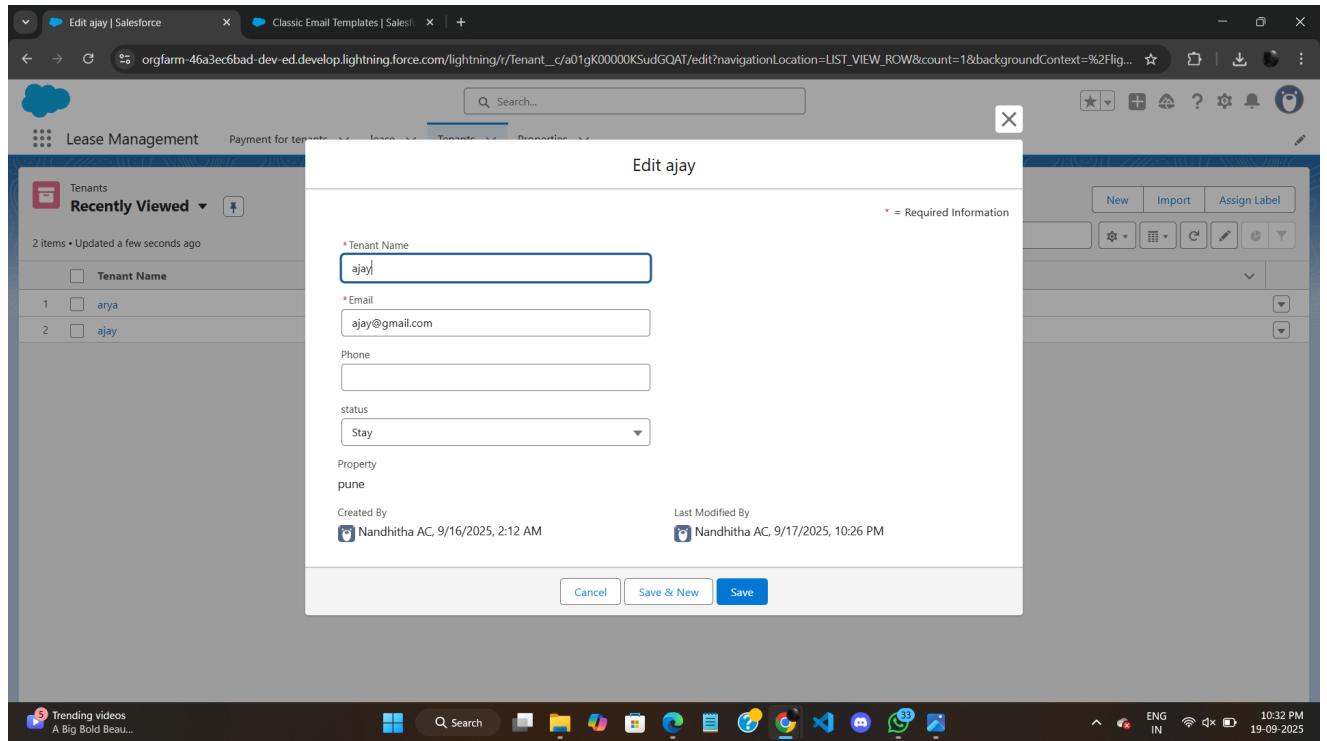
- Leave Approved, Leave Rejected



- Flow runs



- 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<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});  
    }  
}  
}
```