

LEASE MANAGEMENT

College Name: **vidyasagar college of arts and science**

College Code: **bru30**

TEAM ID: **NM2025TMID26850**

MEMBERS:

Team LeaderName: JANANI B.S

Email: Jananibs1030@gmail.com

Team Member1: KARRUPUSAMY T

Email: karuppusamy231033@gmail.com

Team Member2: PRIYA DHARSHINI J

Email: priyadharshinij231042@gamil.com

Team Member3: KATHIRESHAN S

Email: kathireshan231044@gmail.com

\.INTRODUCTION

\.\ProjectOverview

TheLease ManagementSystem is aSalesforce-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

orgfarm-5df1e80502-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01sgk000000z73h4/Details/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

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

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

orgfarm-5df1e80502-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01sgk000000z73h4/Details/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

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

Details

Description

API Name: lease__c

Custom

✓ Singular Label: lease

Plural Label: leases

Enable Reports

✓

Track Activities

✓

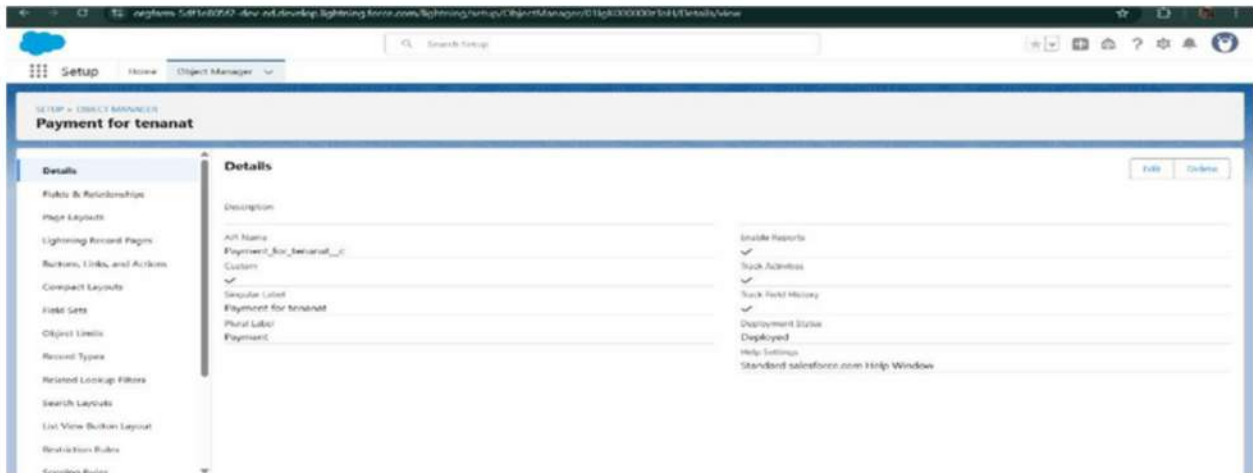
Track Field History

✓

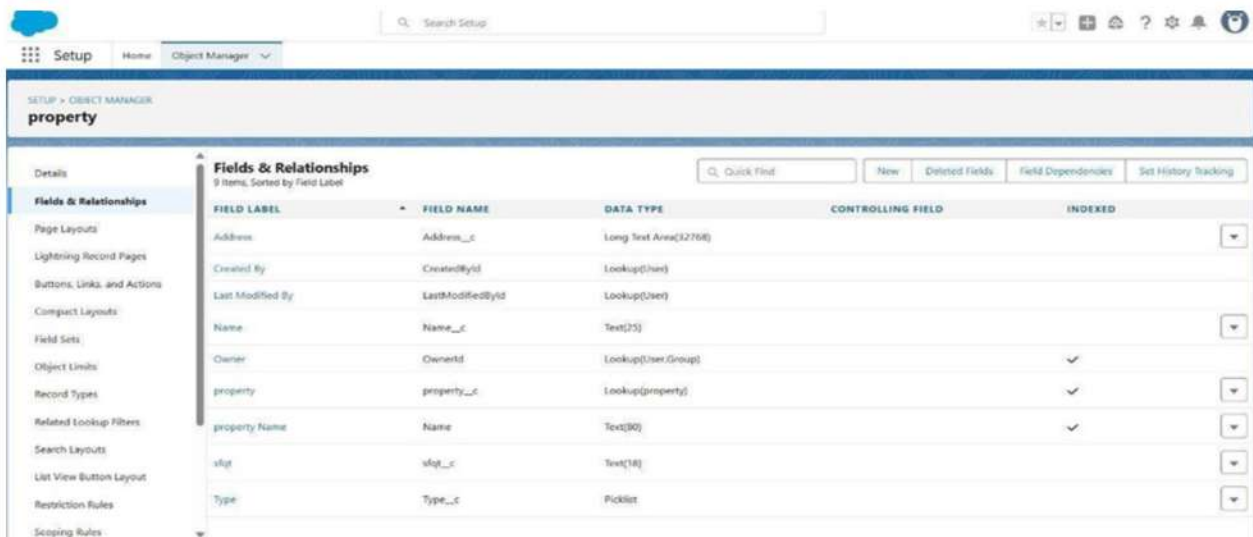
Deployment Status: Deployed


Help Settings: Standard salesforce.com Help Window

Edit Delete









- Configured fields and relationships





Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

Payment for tenanat


Details

Fields & Relationships







7 Items, Sorted by Field Label

Quick FindNewDeleted FieldsField DependenciesSet History Tracking

| 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) | | ✓ |



Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

lease

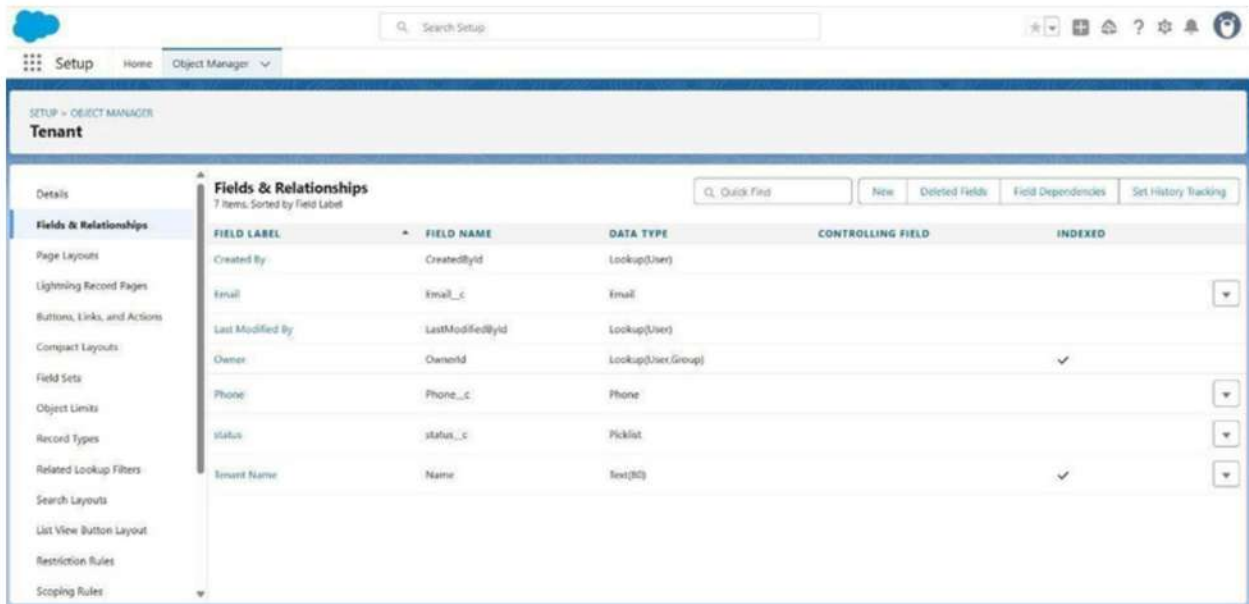
Details

Fields & Relationships

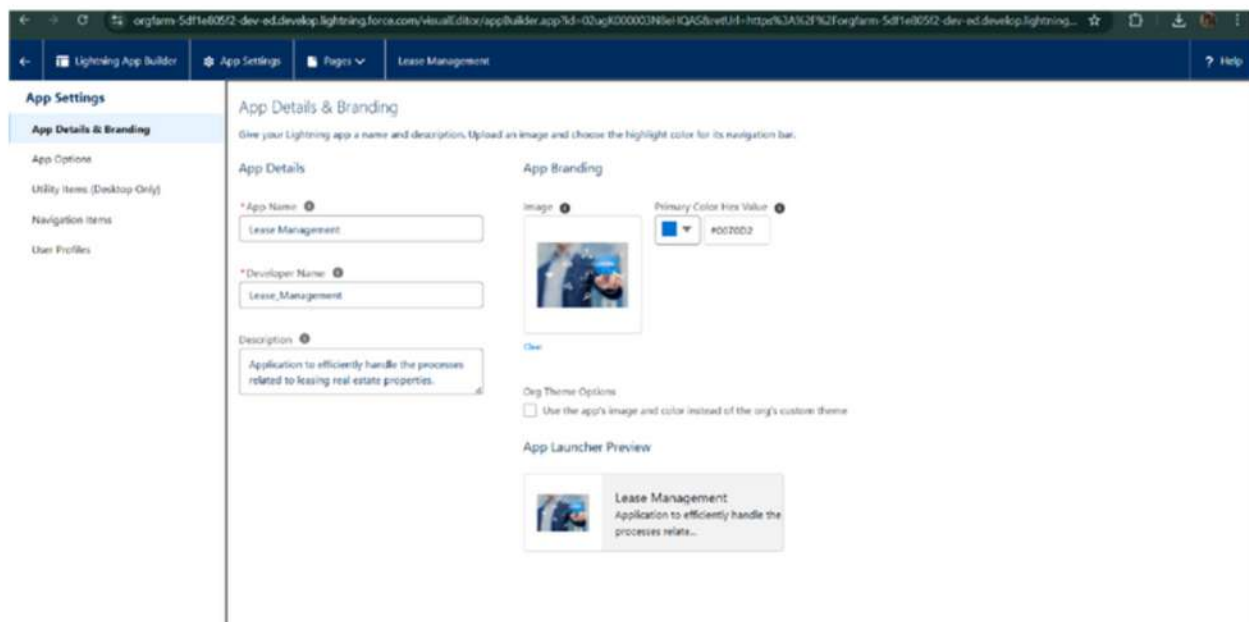
7 Items, Sorted by Field Label

Quick FindNewDeleted FieldsField DependenciesSet History Tracking

| 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 | | |



- Developed Lightning App with relevant tabs



org/iam-5d1e8052-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02ag000003NlsHQA5&vrfM=http%3A%2F%2Forg/iam-5d1e8052-dev-ed.develop.lightning...

Lightning App Builder App Settings Pages Lease Management Help

App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items**
- User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Type to filter list...

Accounts

Activation Targets

Activations

All Tabs

Alternative Payment Methods

Analytics

App Launcher

Appointment Categories

Appointment Invitations

Approval Requests

Selected Items

Payment

Sevents

property

lease

➔

⬅

⬆

⬆

org/iam-5d1e8052-dev-ed.develop.lightning.force.com/visualEditor/appBuilder.app?id=02ag000003NlsHQA5&vrfM=http%3A%2F%2Forg/iam-5d1e8052-dev-ed.develop.lightning...

Lightning App Builder App Settings Pages Lease Management Help

App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items
- User Profiles**

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

Analytics Cloud Integration User

Analytics Cloud Security User

Anypoint Integration

Authenticated Website

Authenticated Website

B2B Reordering Portal Buyer Profile

Contract Manager

Custom Marketing Profile

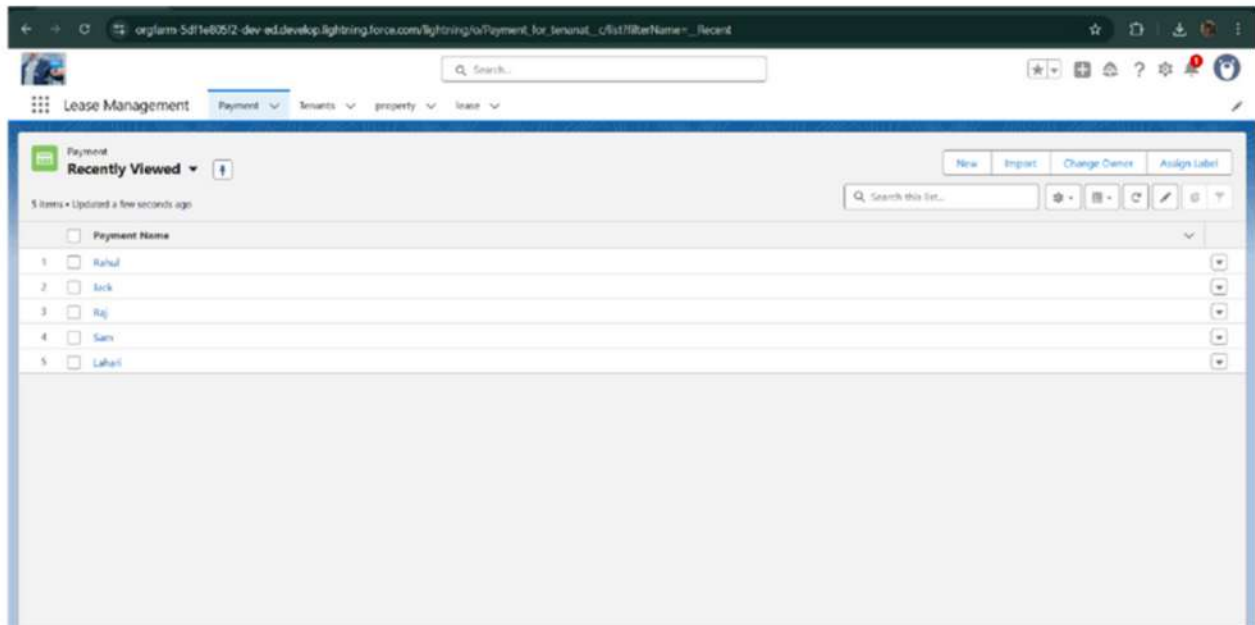
Custom Sales Profile

Custom Support Profile

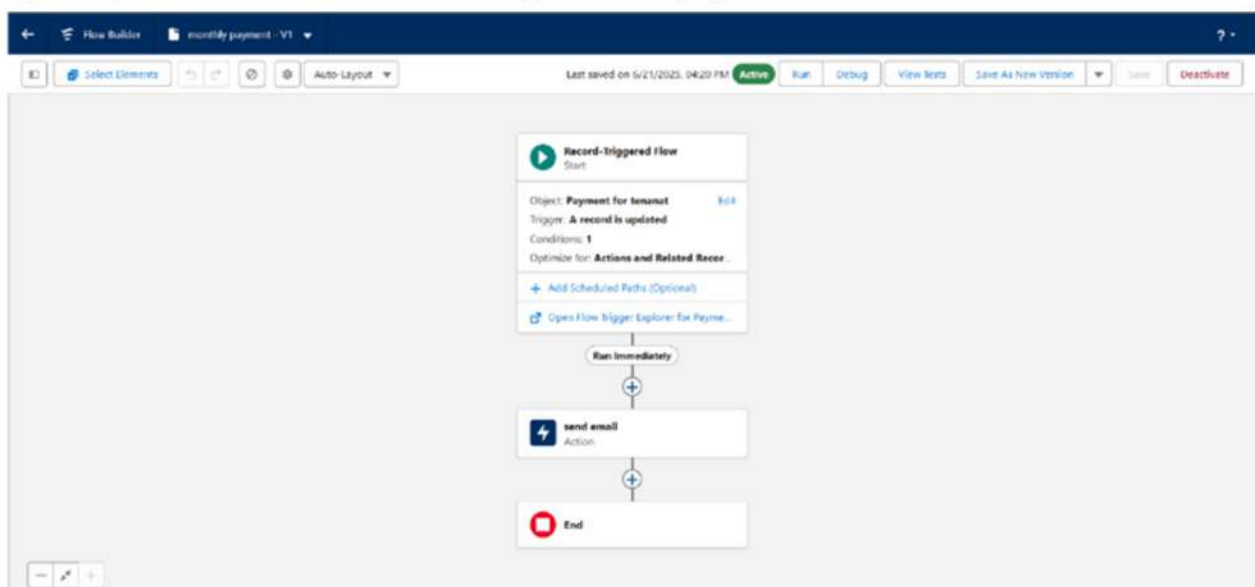
Customer Community Login User

Selected Profiles

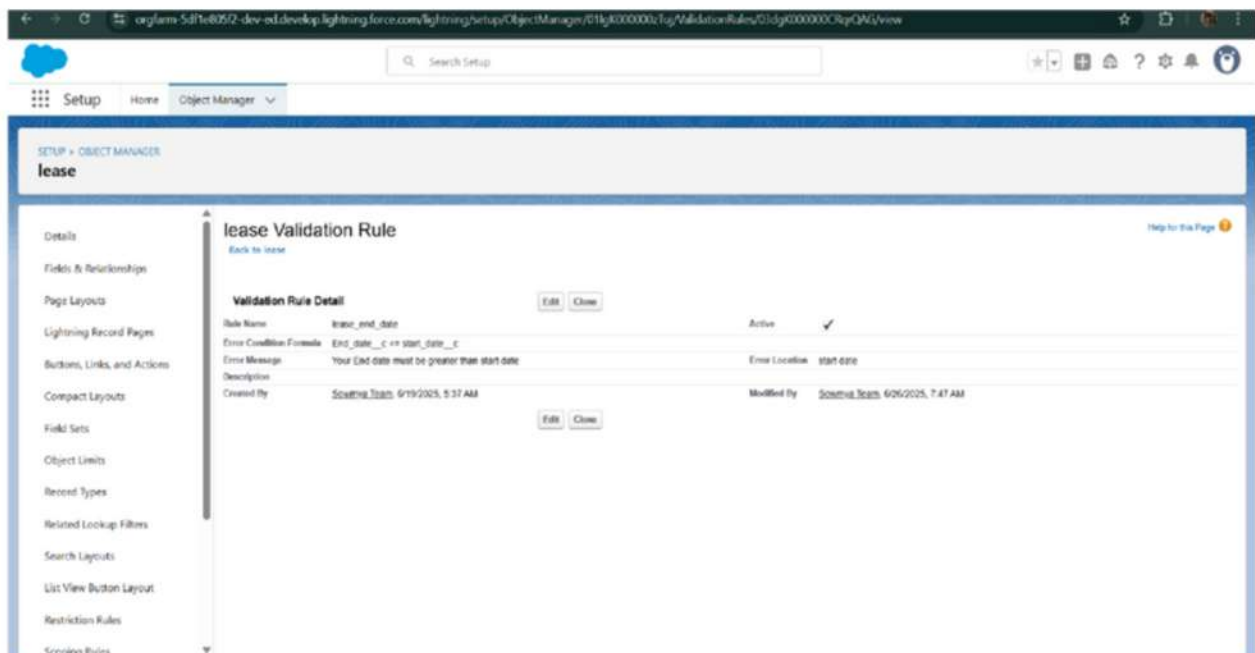
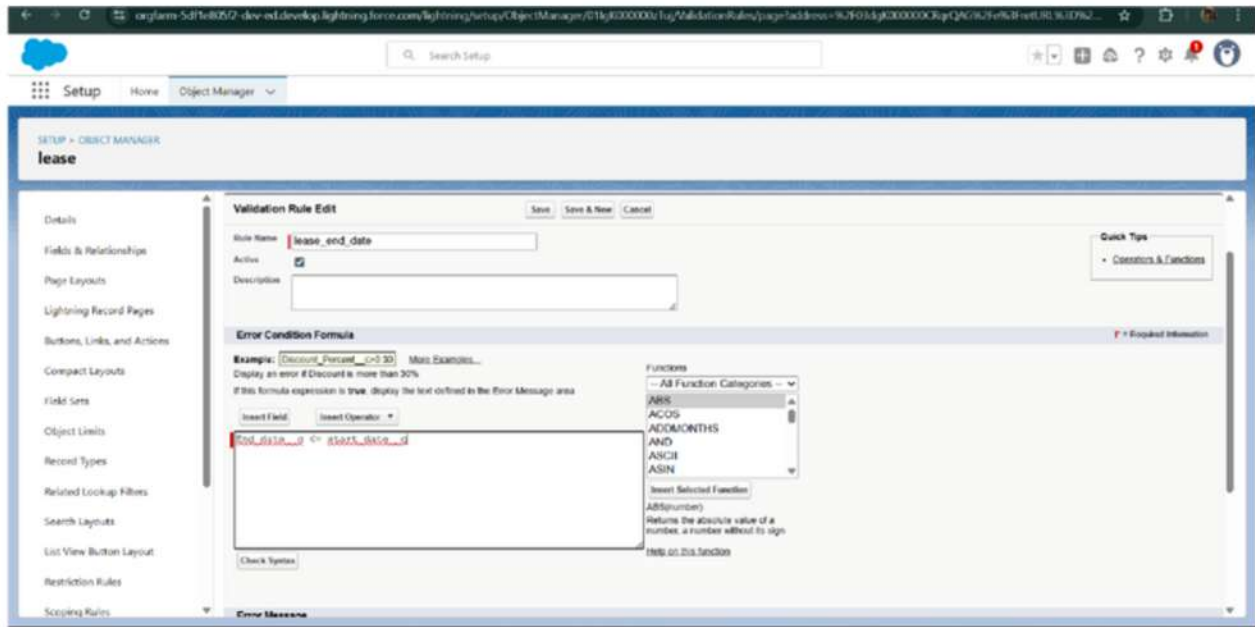
System Administrator



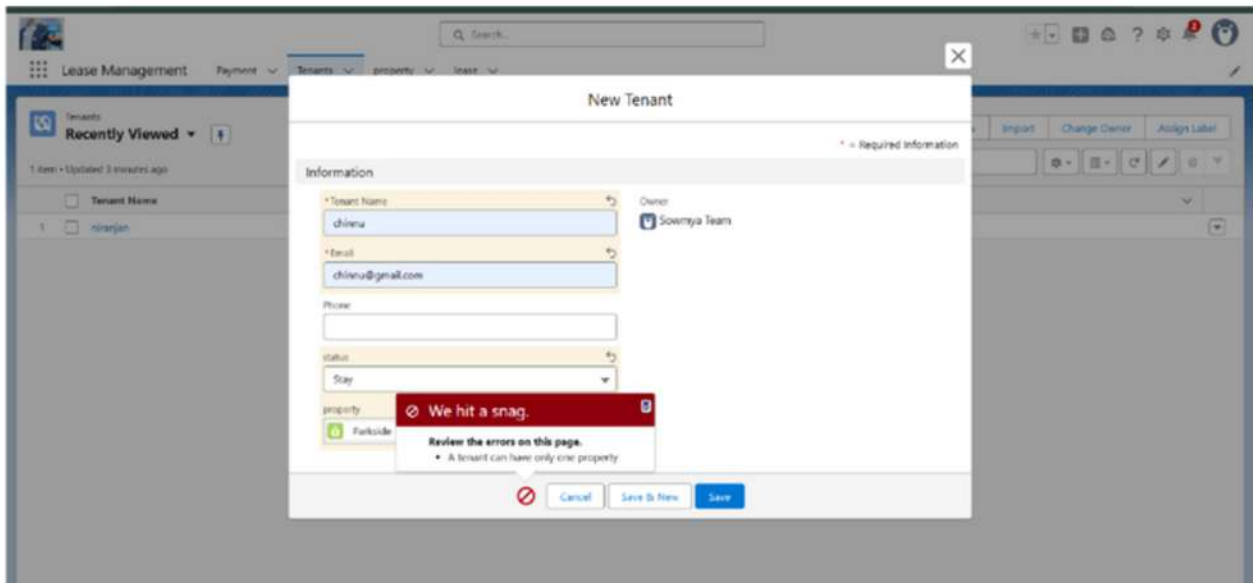
- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object



- Added Apex trigger to restrict multiple tenants per property



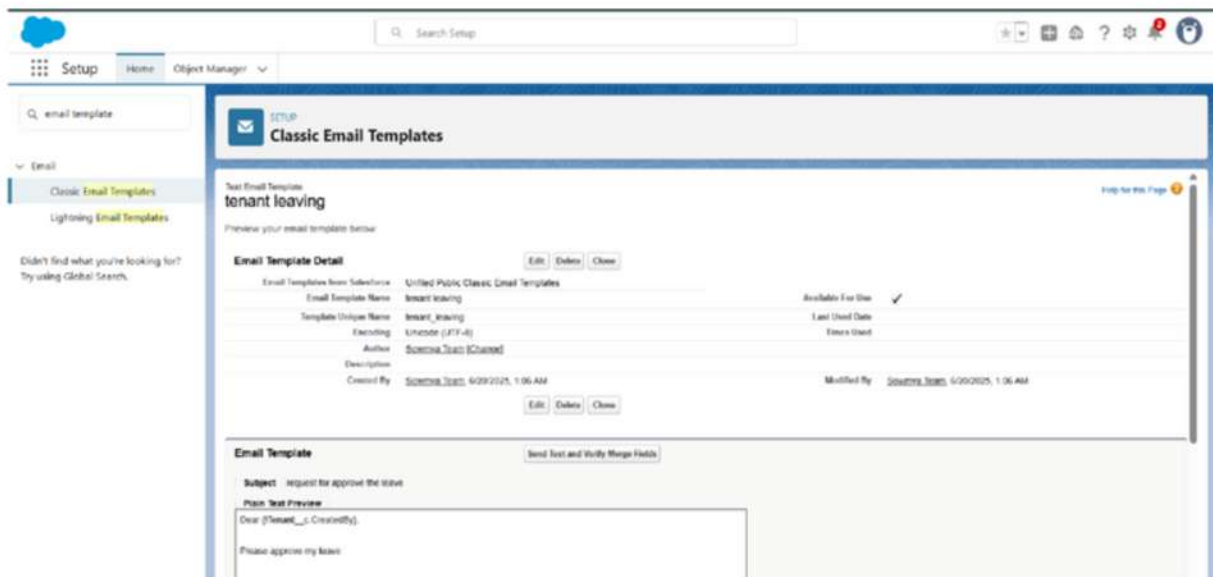
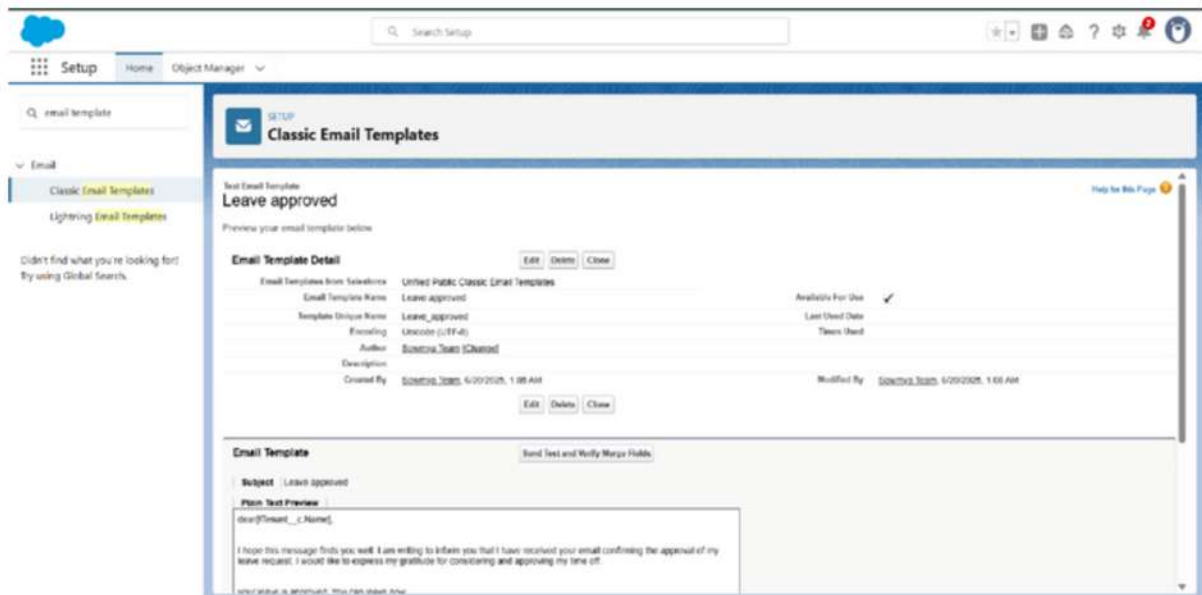
- Scheduled monthly reminder emails using Apex class

```

1 // Global class MonthlyRentScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmail();
10
11         }
12     }
13 }
14
15 public static void sendMonthlyEmail() {
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 }

```

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



Setup

Home

Object Manager

Search Setup

email template

Classic Email Templates

Lightning Email Templates

Didn't find what you're looking for?
Try using Global Search.

SETUP

Classic Email Templates

Test Email Template

Leave rejected

Preview your email template below.

Email Template Detail

Email Template Name

Leave rejected

Available For Use

✓

Template Unique Name

Leave_rejected

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

System Administrator

Modified By

System Administrator

Description

Created By

System Administrator

Created On

6/20/2025, 1:11 AM

Modified On

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

Setup

Home

Object Manager

Search Setup

email template

Classic Email Templates

Lightning Email Templates

Didn't find what you're looking for?
Try using Global Search.

SETUP

Classic Email Templates

Test Email Template

Tenant Email

Preview your email template below.

Email Template Detail

Email Template Name

Tenant Email

Available For Use

✓

Template Unique Name

Tenant_Email

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

System Administrator

Modified By

System Administrator

Description

Created By

System Administrator

Created On

6/20/2025, 1:12 AM

Modified On

6/20/2025, 1:12 AM

Email Template

Send Test and Verify Merge Fields

Subject

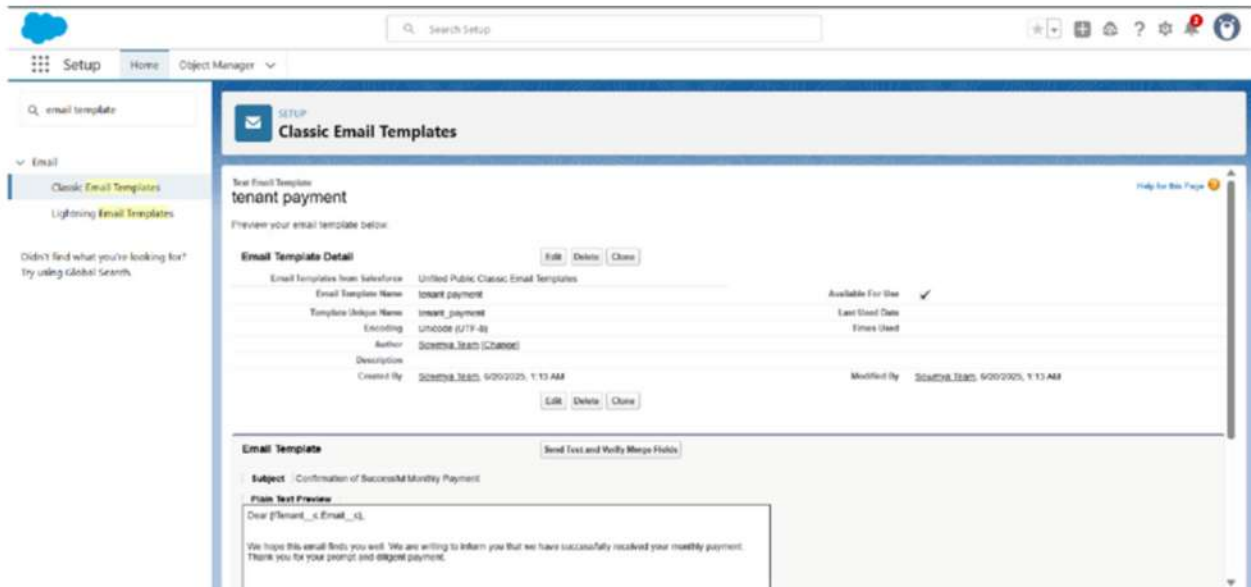
Urgent: Monthly Rent Payment Reminder

Plain Text Preview

Dear {Tenant__c.Name},

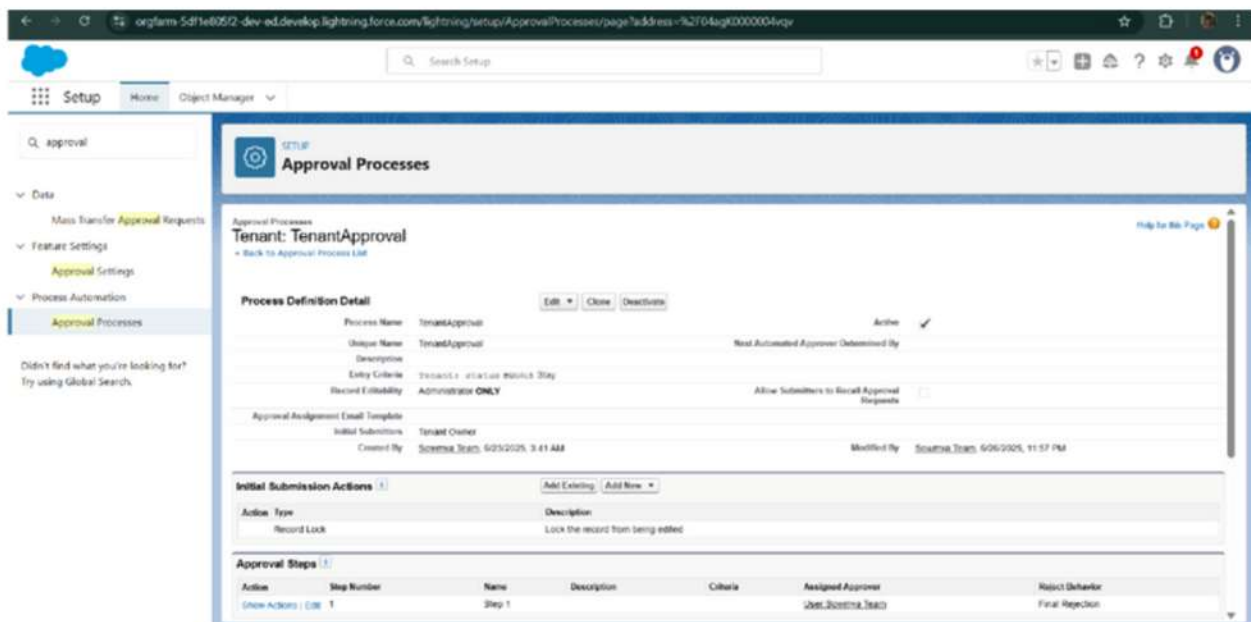
I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

Your cooperation is a highly valuable asset and we look forward to your continued partnership. It is not our

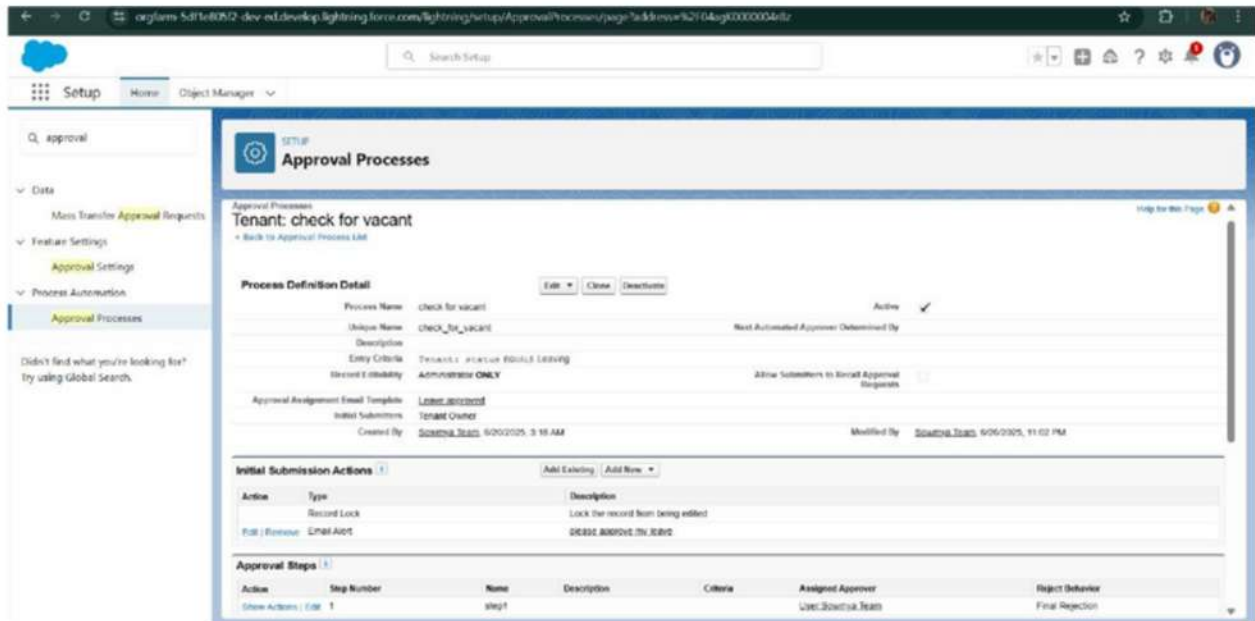


- Approval Process creation

For Tenant Leaving:

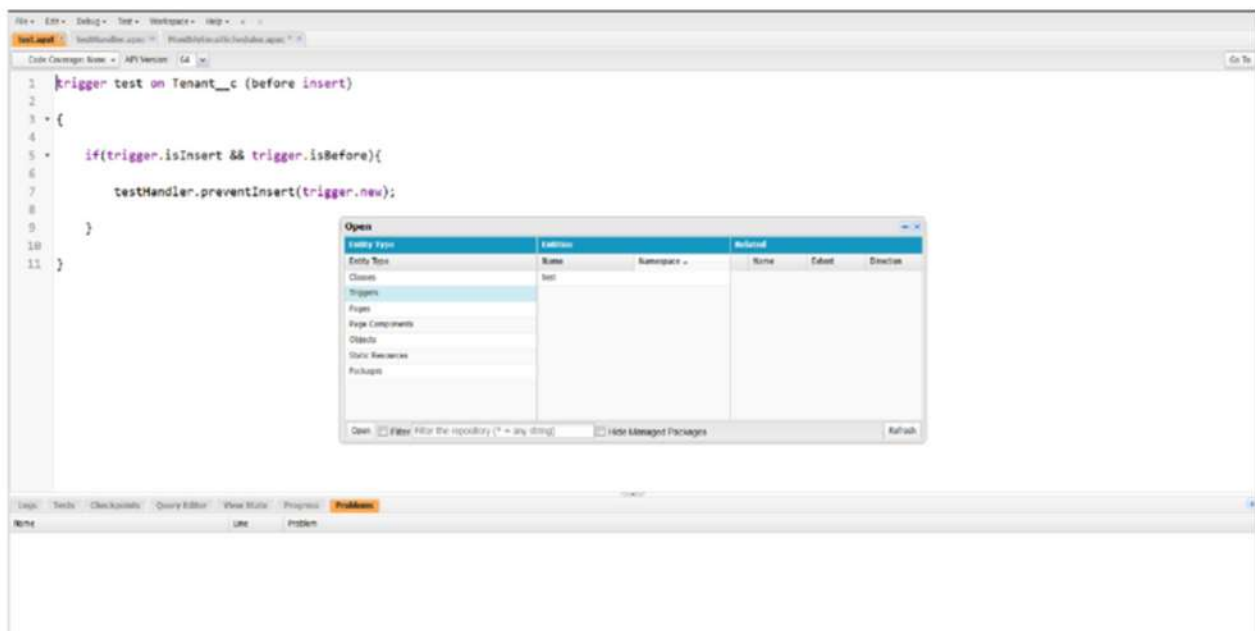


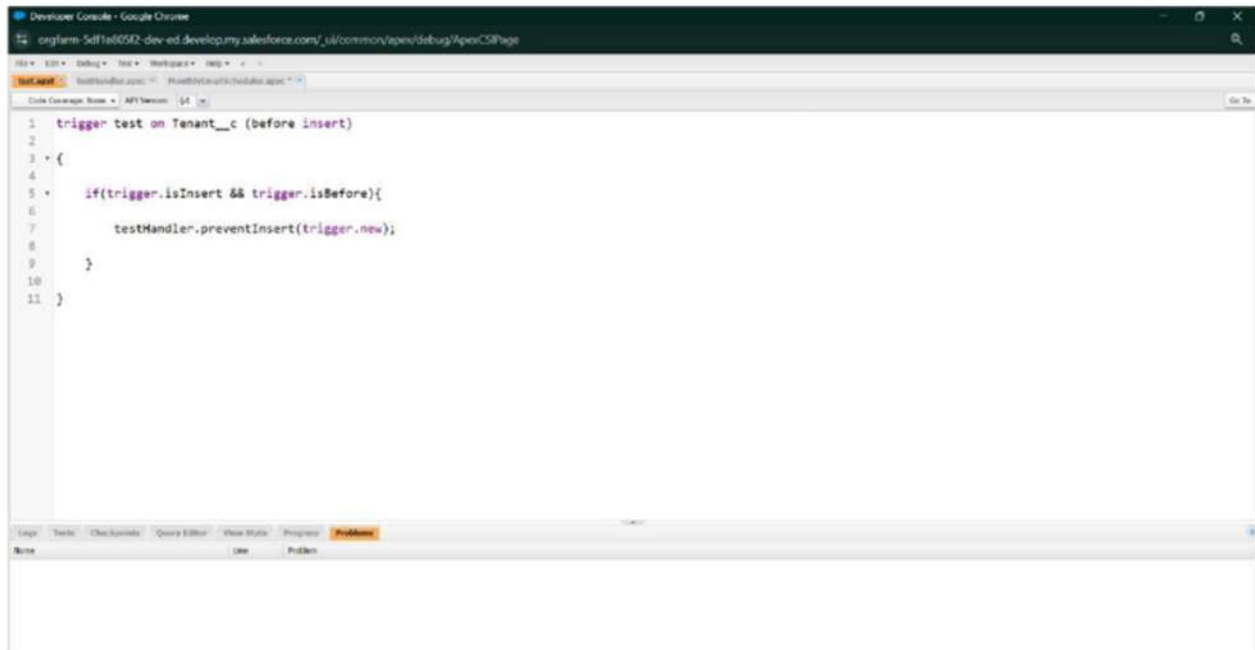
For Check for Vacant:



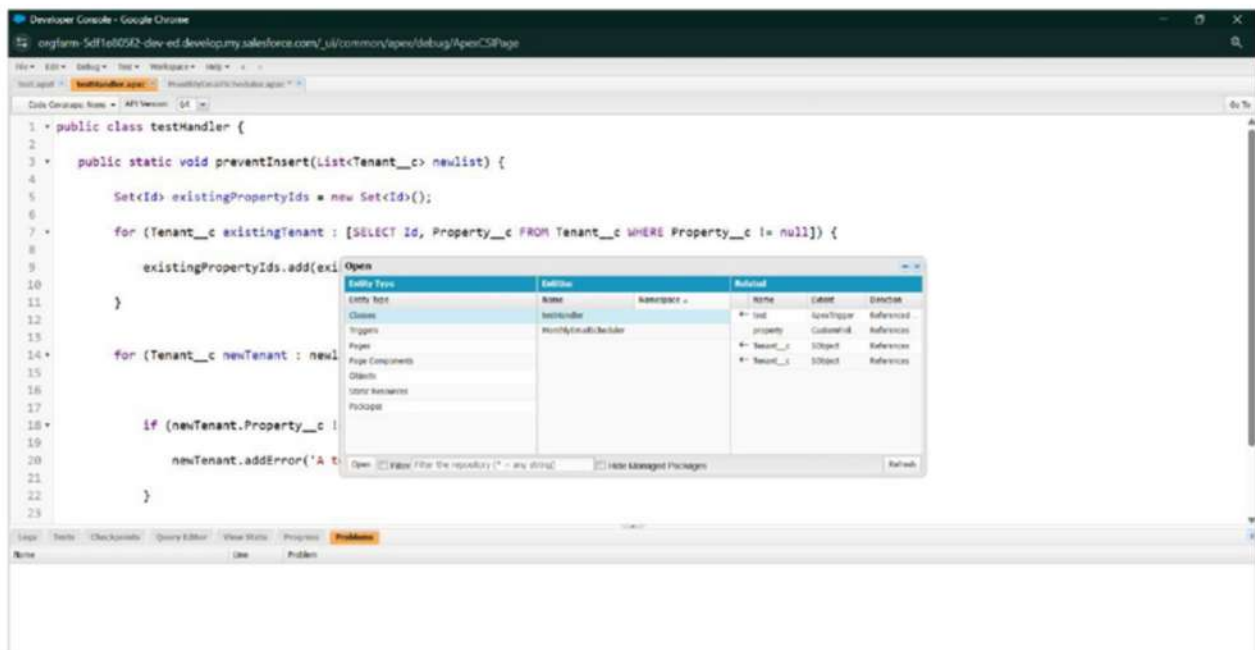
- Apex Trigger

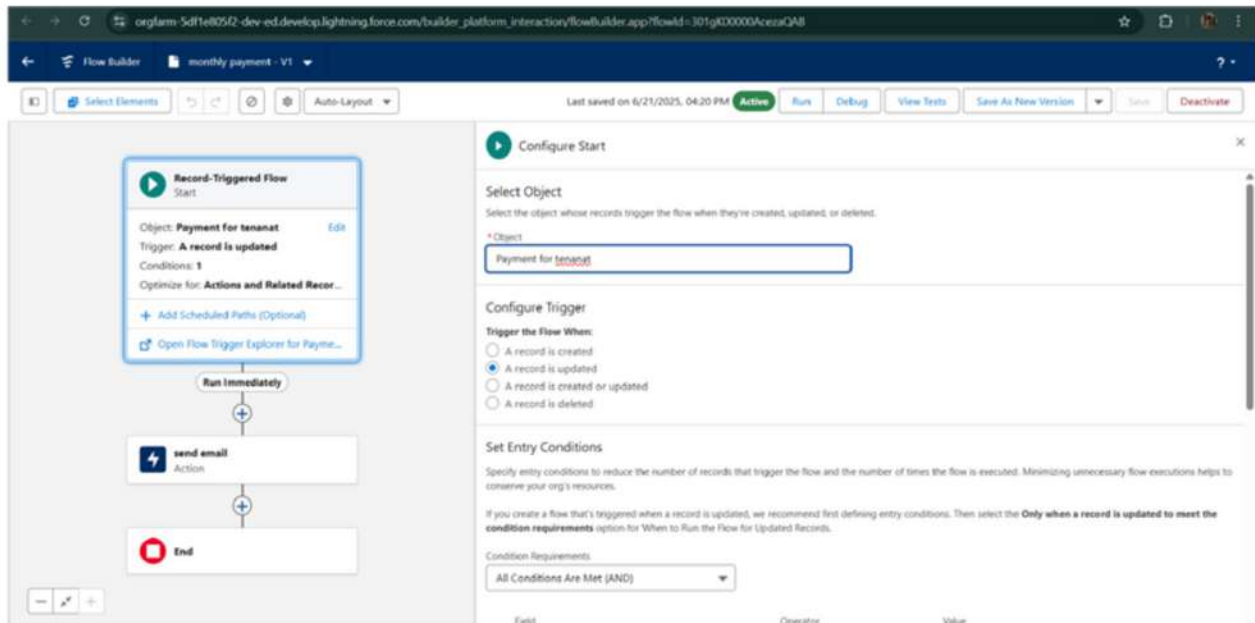
Create an Apex Trigger



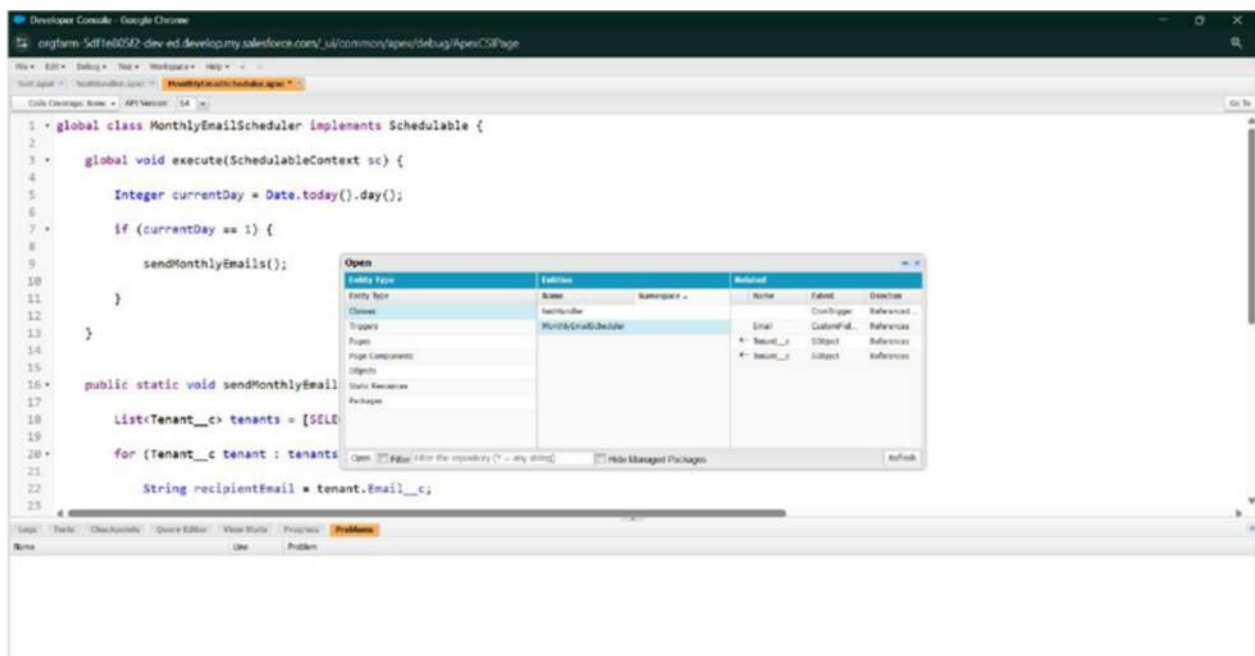


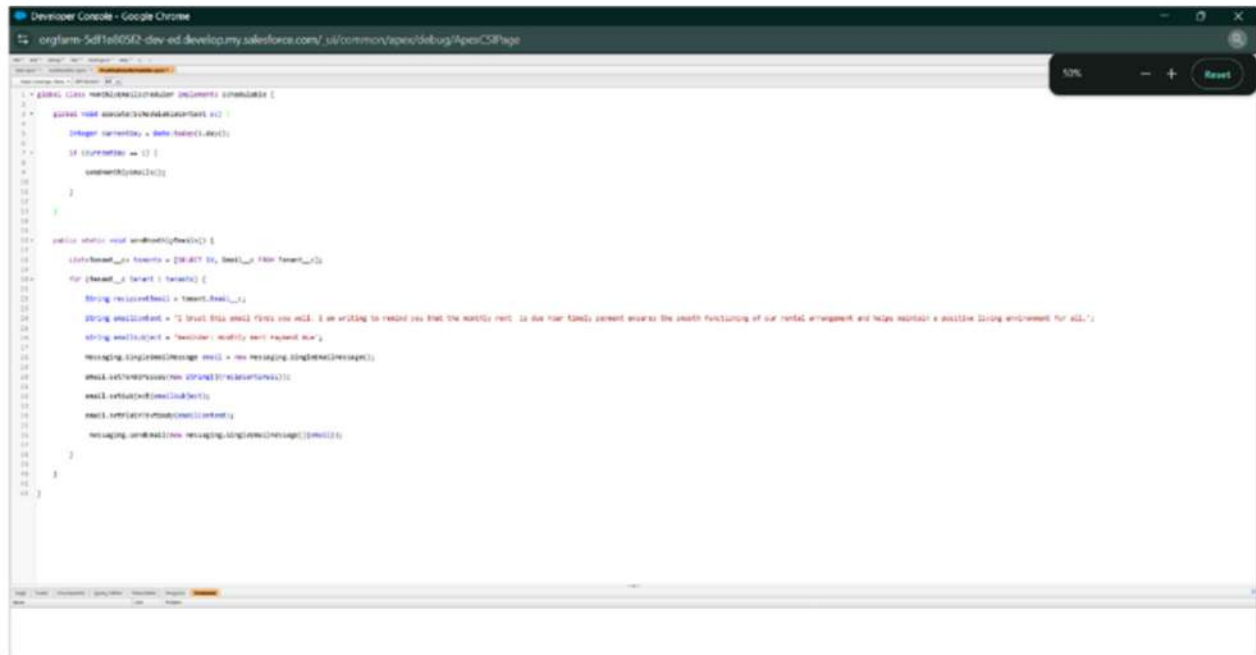
Create an Apex Handler class





- Schedule class:
Create an Apex Class

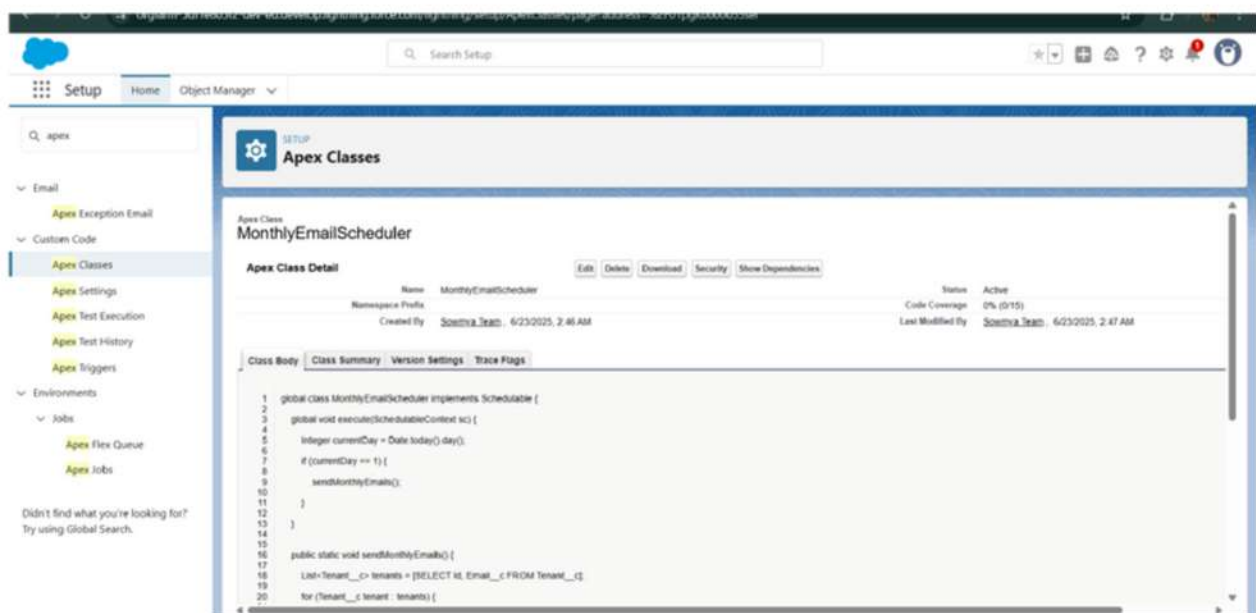




The screenshot shows a Chrome Developer Console window with the URL `orgname-5d1a0592-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage`. The console displays the source code for the `MonthlyEmailScheduler` class, which implements the `Schedulable` interface. The code includes a `execute` method that sends an email to tenants on the first day of the month. The email body contains a message about a monthly rent increase. The code is as follows:

```
1 global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4         Integer currentDay = Date.today().day();
5         if (currentDay == 1) {
6             sendMonthlyEmails();
7         }
8     }
9
10    public static void sendMonthlyEmails() {
11        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
12        for (Tenant__c tenant : tenants) {
13            String emailContent = 'Hi ' + tenant.Email__c + ', we are writing to remind you that the monthly rent for the first time is about to increase the month functioning of our rental arrangement and helps maintain a positive living environment for all.';
14            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
15            email.setToAddresses(new String[] {tenant.Email__c});
16            email.setSubject('Monthly Rent Increase');
17            email.setBody(emailContent);
18            Messaging.sendEmail(new Messaging.SingleEmailMessage(email));
19        }
20    }
21 }
```

Schedule Apex class



The screenshot shows the Salesforce Setup page with the 'Apex Classes' section selected. The class `MonthlyEmailScheduler` is listed with the following details:

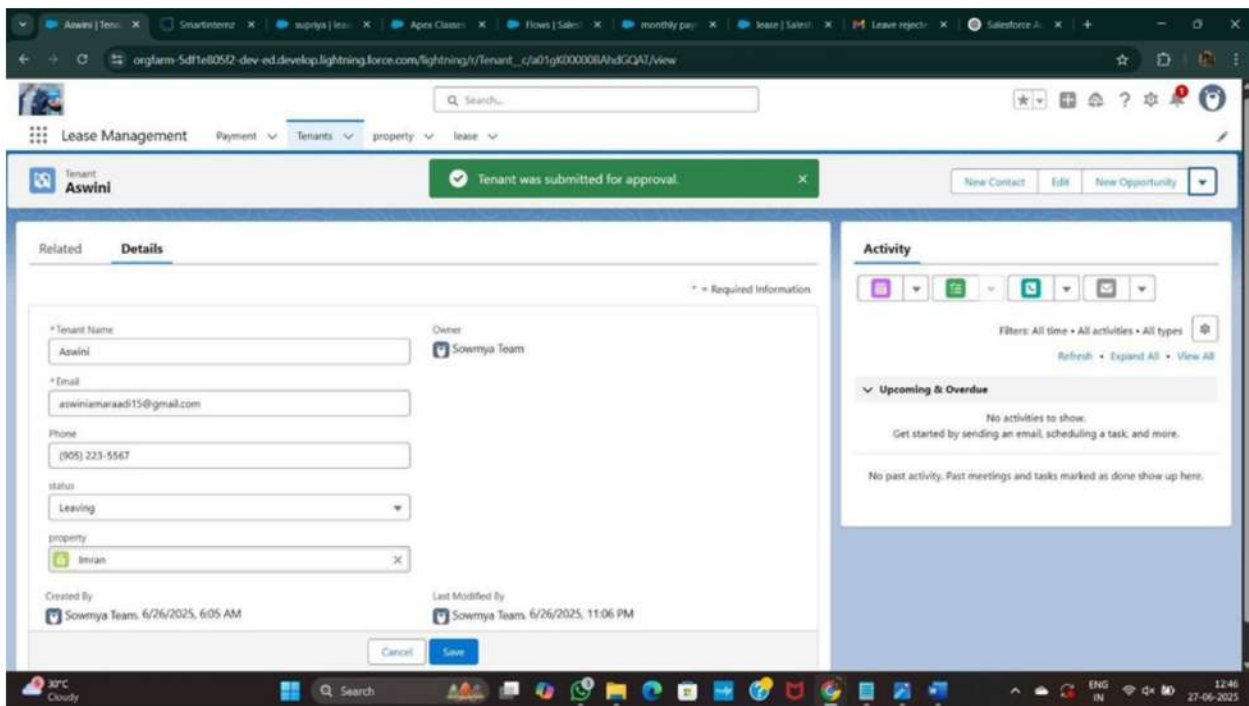
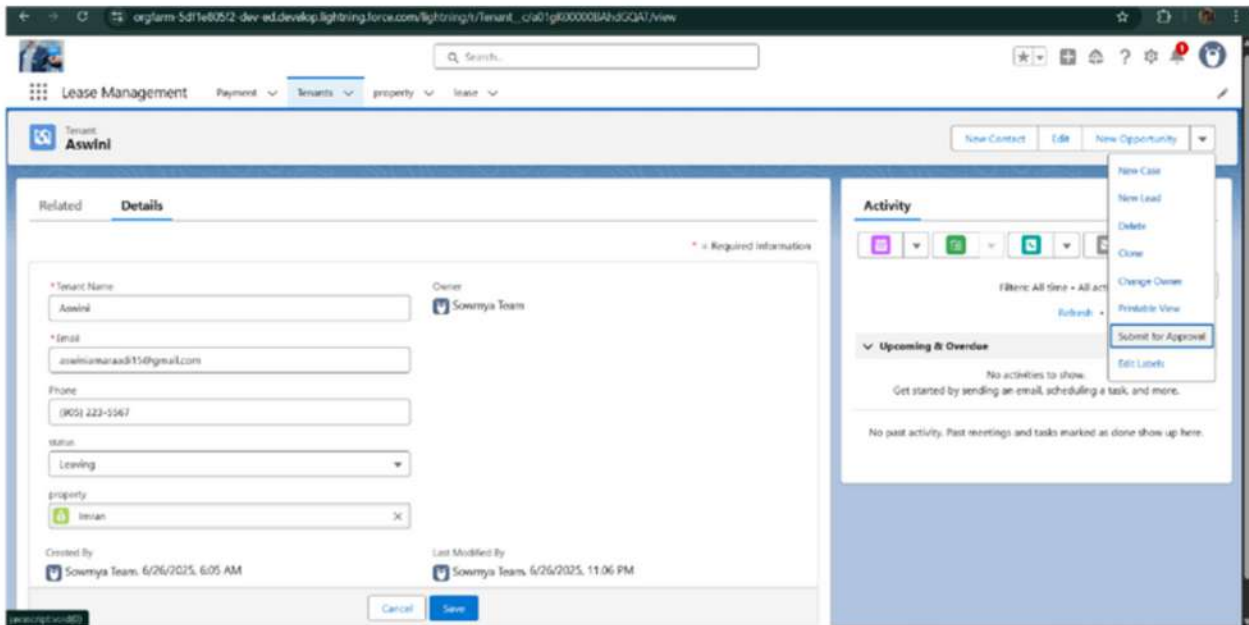
| Name | Namespace | Status |
|-----------------------|-----------|--------|
| MonthlyEmailScheduler | | Active |

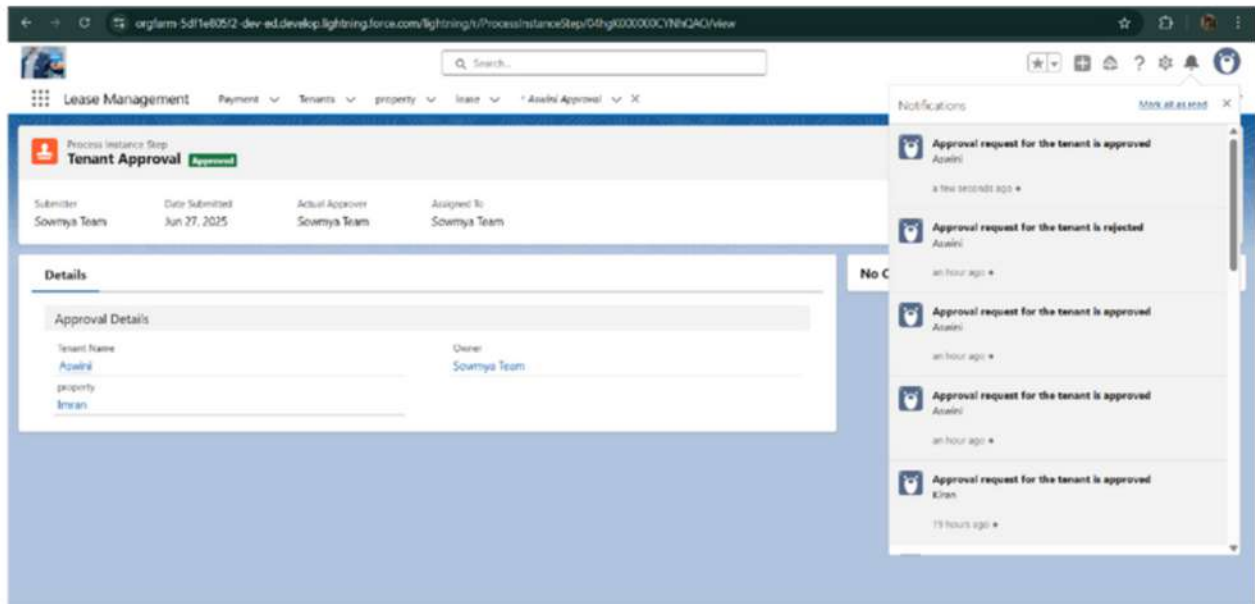
Additional details for the class:

- Created By: ScottyA_Thorn
- Last Modified By: ScottyA_Thorn
- Code Coverage: 0% (0/15)

The class body is displayed below the details, showing the same code as in the first screenshot:

```
1 global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4         Integer currentDay = Date.today().day();
5         if (currentDay == 1) {
6             sendMonthlyEmails();
7         }
8     }
9
10    public static void sendMonthlyEmails() {
11        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
12        for (Tenant__c tenant : tenants) {
13            String emailContent = 'Hi ' + tenant.Email__c + ', we are writing to remind you that the monthly rent for the first time is about to increase the month functioning of our rental arrangement and helps maintain a positive living environment for all.';
14            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
15            email.setToAddresses(new String[] {tenant.Email__c});
16            email.setSubject('Monthly Rent Increase');
17            email.setBody(emailContent);
18            Messaging.sendEmail(new Messaging.SingleEmailMessage(email));
19        }
20    }
21 }
```





FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Triggervalidationbyentering duplicate tenant-property records

New Tenant

* = Required Information

Information

Tenant Name: chirna

Email: chirna@gmail.com

Phone:

Status: Shay

Property: Parkside

We hit a snag.

Review the errors on this page.

- A tenant can have only one property.

Cancel Save & New Save

- Validation Rule checking

Lease

Related Details

* = Required Information

Lease Name: supriya

Start Date: 6/28/2025

End Date: 6/18/2025

Property: Search property...

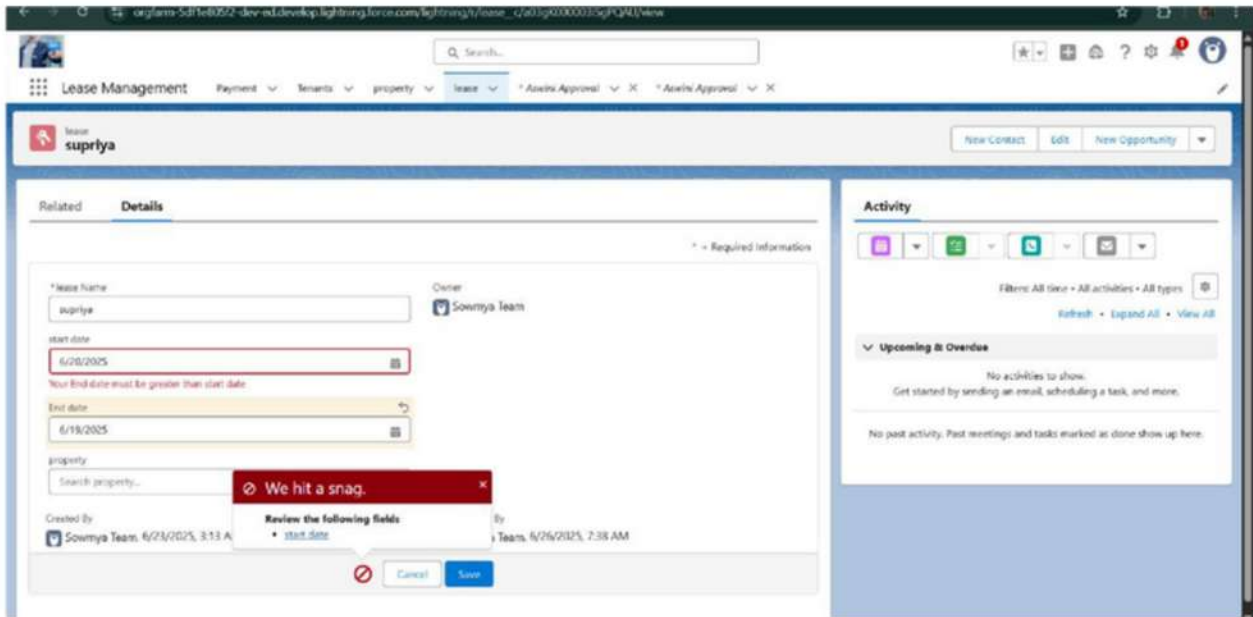
We hit a snag.

Review the following fields

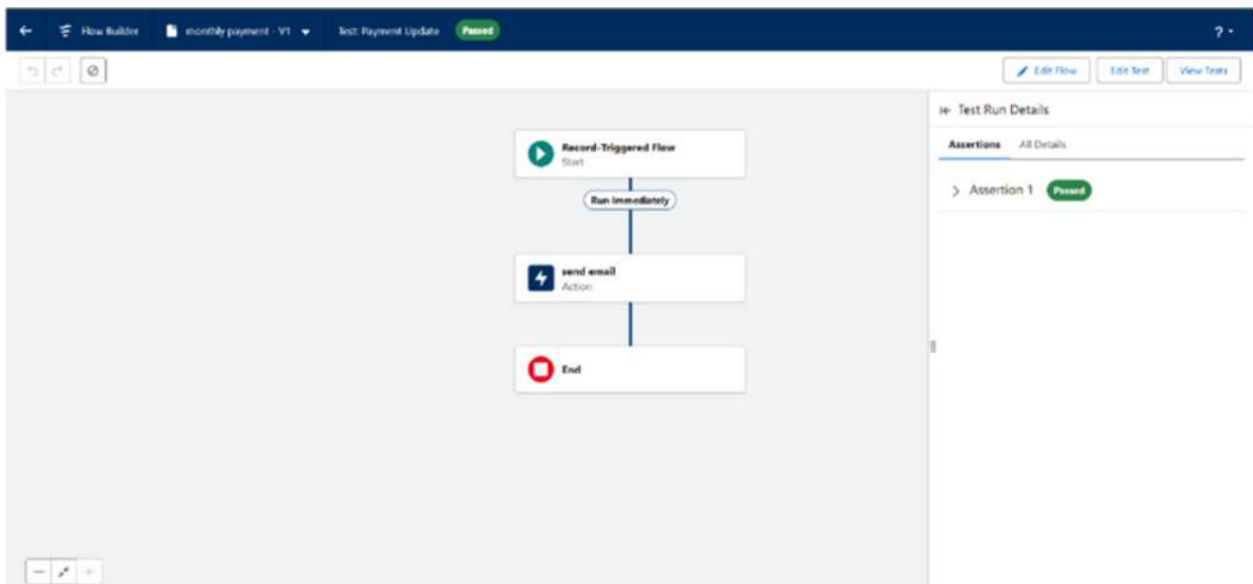
- start date

Your End date must be greater than start date.

Cancel Save



- Test flows on payment update



- Approval process validated through email alerts and status updates

Lease Management

Payment

Tenants

property

lease

niranjan Approval

X

Search...

Notifications

Mark all as read

Tenant

niranjan

Related

Details

* Required information

* Tenant Name

niranjan

Owner

Sowmya Team

* Email

niranjan1306@gmail.com

Phone

Status

Stay

Property

Parkside Lofts

Created By

Sowmya Team, 6/23/2025, 2:33 AM

Last Modified By

Sowmya Team, 6/23/2025, 8:58 AM

Cancel

Save

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

Learn how to guide reps through the sales process.

Jun 20, 2025, 1:28 PM

Lease Management

Payment

Tenants

property

lease

Search...

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

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

Tenant

niranjan

New Contact

Edit

New Opportunity

Approval History (6+)

| Step Name | Date | Status | Assigned To |
|----------------------------|--------------------|-----------|-------------|
| Step 1 | 6/25/2025, 5:39 AM | Approved | Sowmya Team |
| Approval Request Submitted | 6/25/2025, 5:29 AM | Submitted | Sowmya Team |
| Step 1 | 6/23/2025, 3:59 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, 3:50 AM | Submitted | Sowmya Team |

View All

Payment (2)

New

Payment Name

Jack

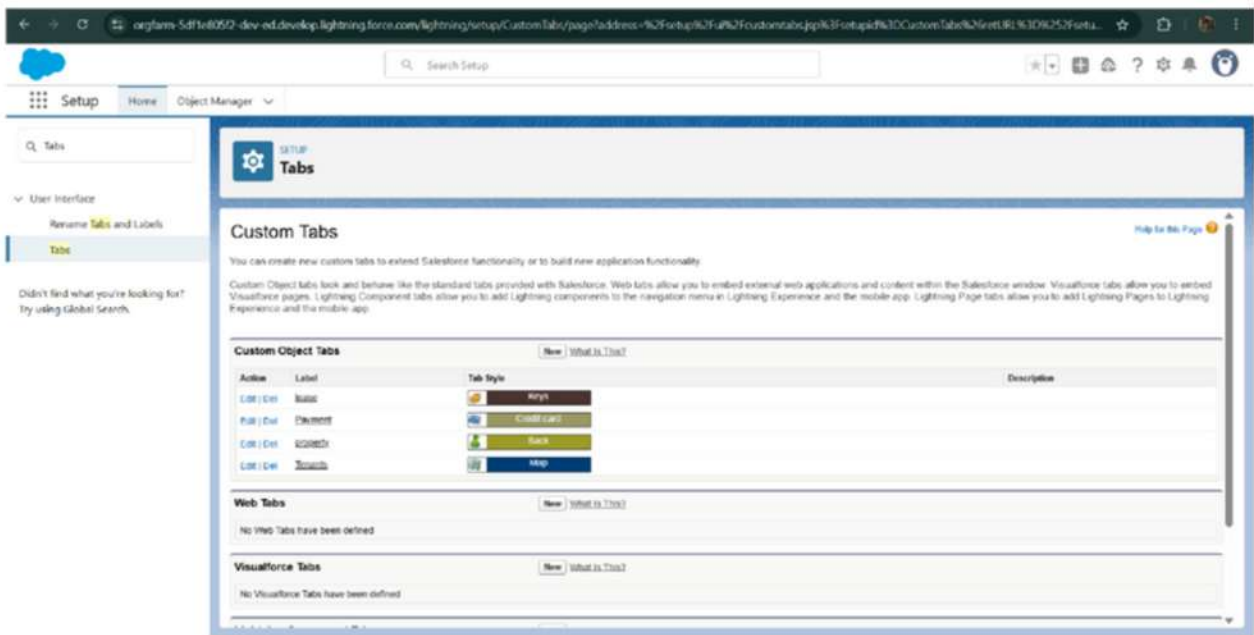
Rahul

View All

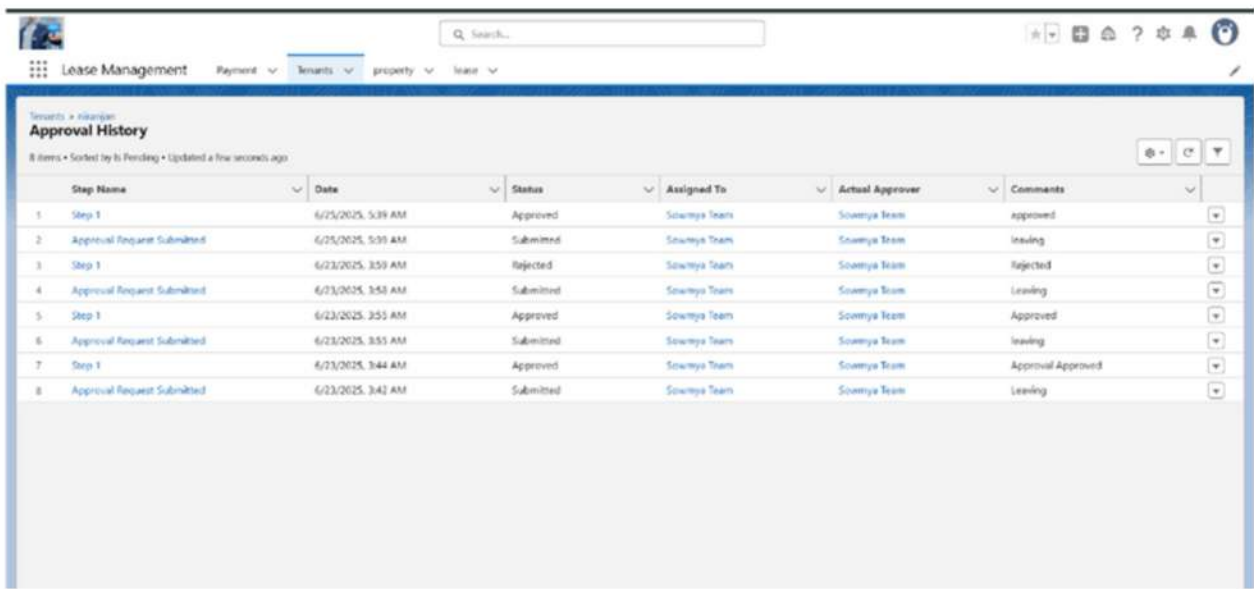
RESULTS

Output Screenshots

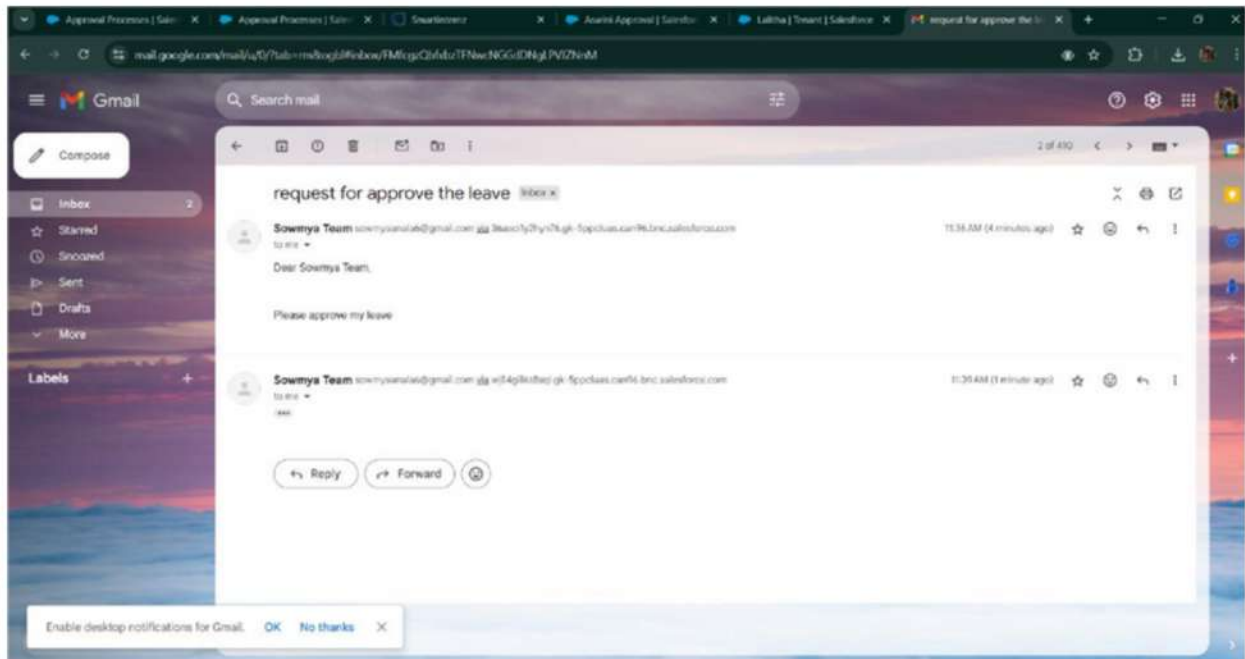
- Tabs for Property, Tenant, Lease, Payment



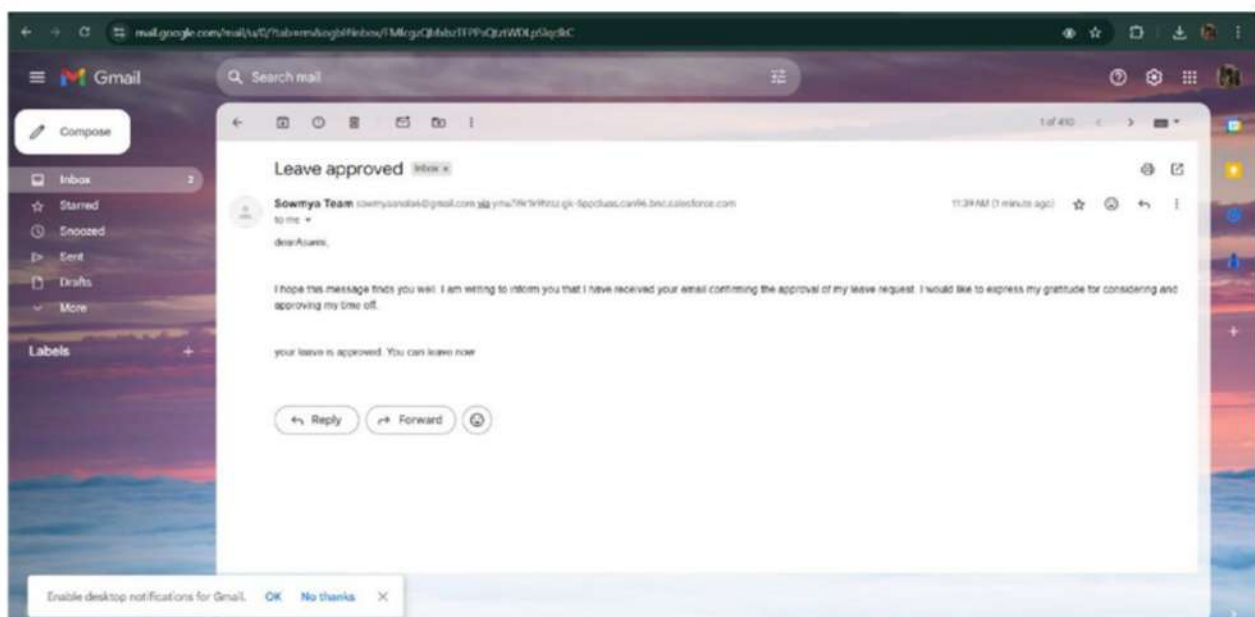
- Email alerts



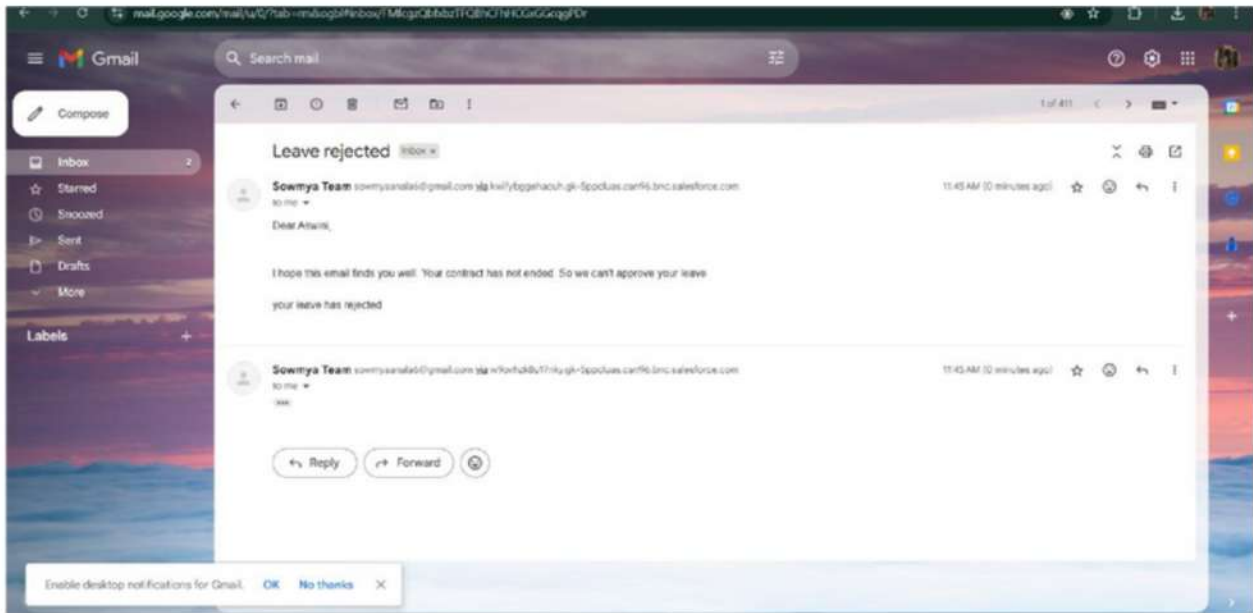
- Request for approve the leave



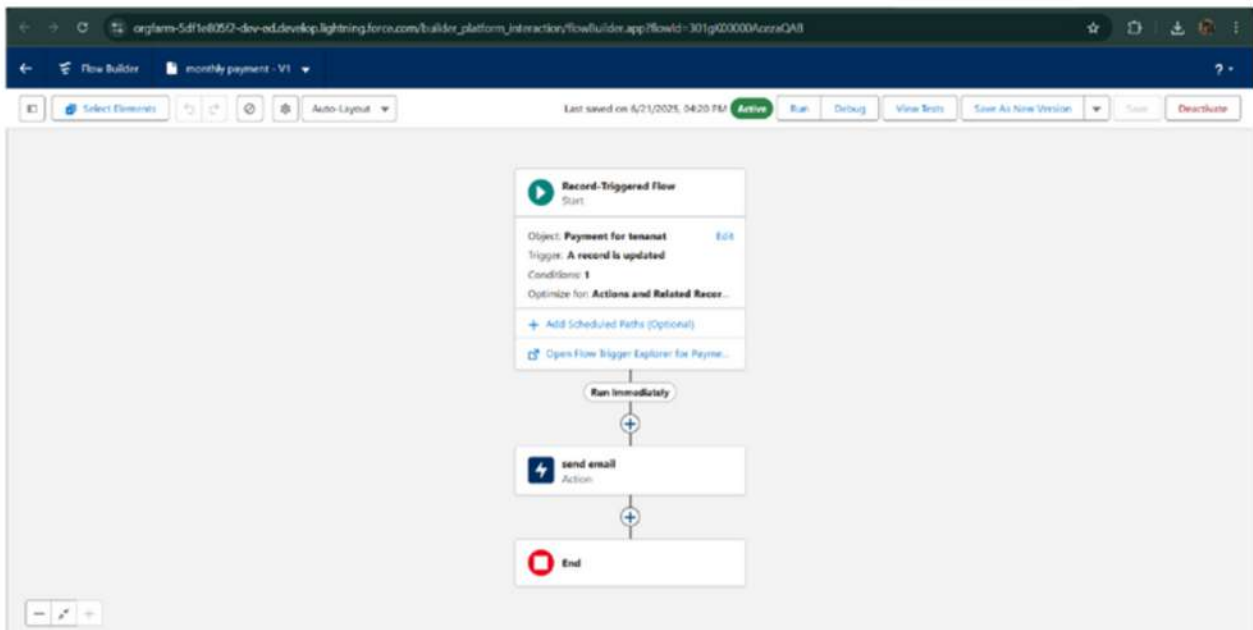
- Leave approved



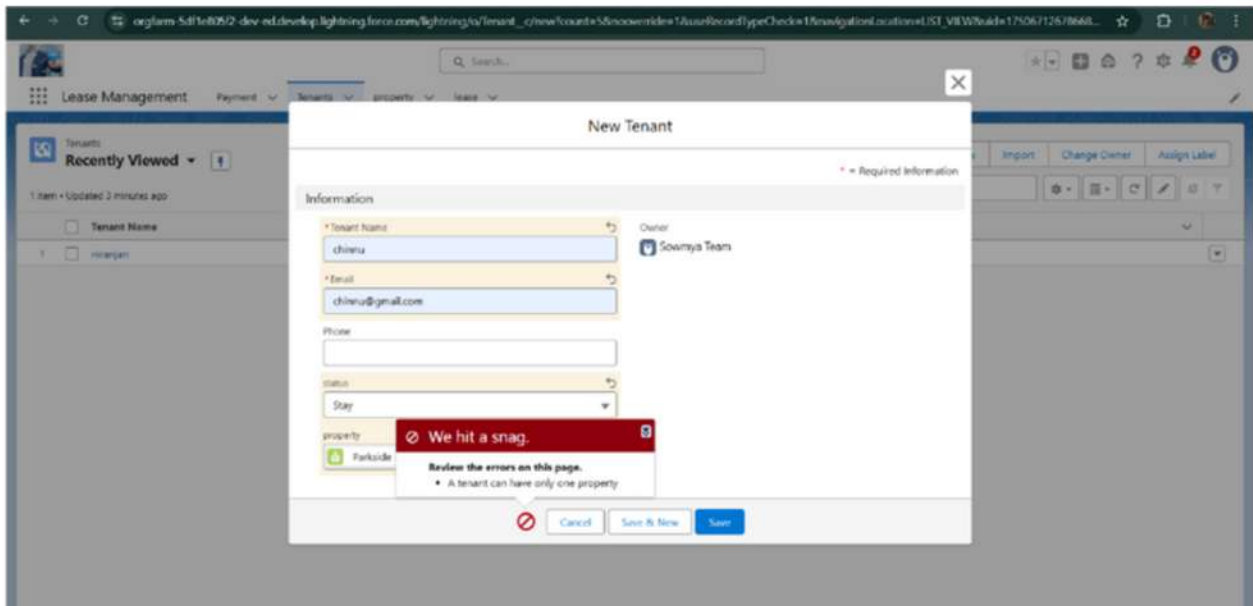
- Leave rejected



- 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 teston 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');
            }
        }
    }
}

```

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<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));
```

