# TITLE:

**LEASE MANAGEMENT**

**College Name: Navarasam Arts & Science College for Women in**

**Arachalur**

**College Code:** **BRU3A**

**TEAM ID: NM2025TMID27196**

**TEAM MEMBERS:**

Team LeaderName: RAGAVI S

**Email**: [ragavisaran123@gmail.com](mailto:ragavisaran123@gmail.com)

Team Member1**:** HARSHAVARSHINI S

**Email:**  harshavarshinisubramani2006@gmail.com

Team Member2: LUKSHYA S P

**Email**: lukshyaluks@gmail.com

**Team** M**ember3:** ABIRAMI G

**Email:** abirami4506@gmail.com

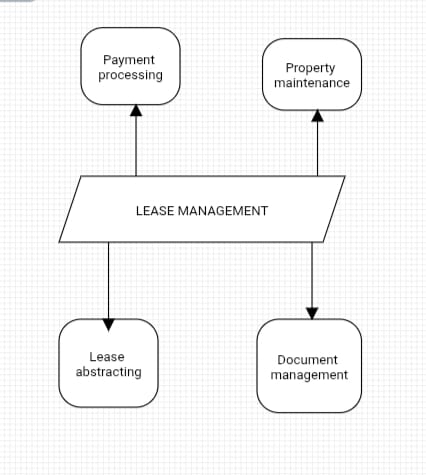
## 

# INTRODUCTION

## 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,reports,and dashboards to support decision-making.it ensures a structured approach to managing the complete lease lifecycle.



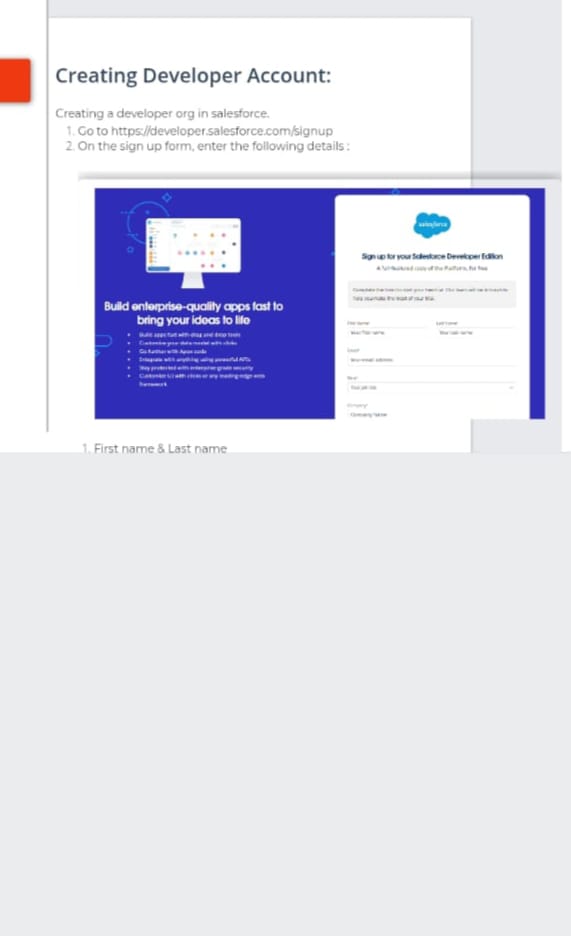
## Purpose :

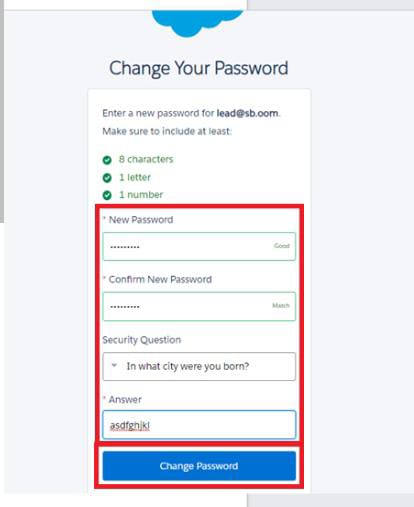
The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

# DEVELOPMENT PHASE

**Creating Developer Account:**

By using this URL[**- https://www.salesforce.com/form/developer-signup/?d=pb**](https://www.salesforce.com/form/developer-signup/?d=pb)

****



**.** Created objects: Property, Tenant, Lease, Payment

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

2. Enter the label name>> property,Tenant,Payment,lease

3. Plural label name>> property,Tenant,Payment,lease

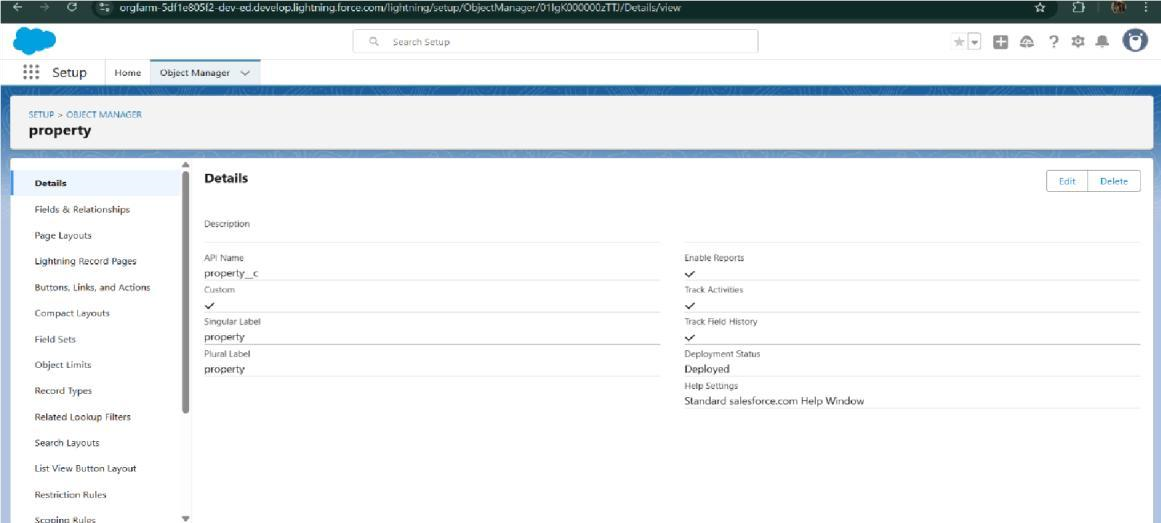
4. Enter Record Name Label and Format

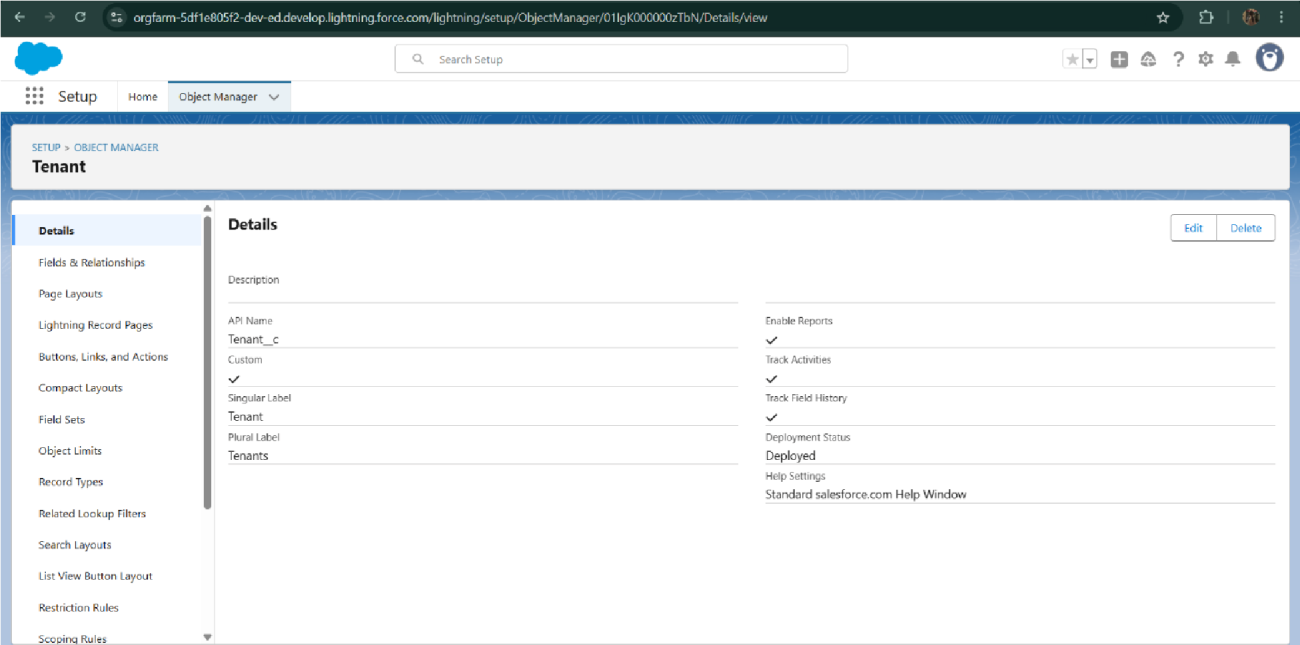
\* Record Name >>property Name,Tenant Name,Payment Name,lease Name

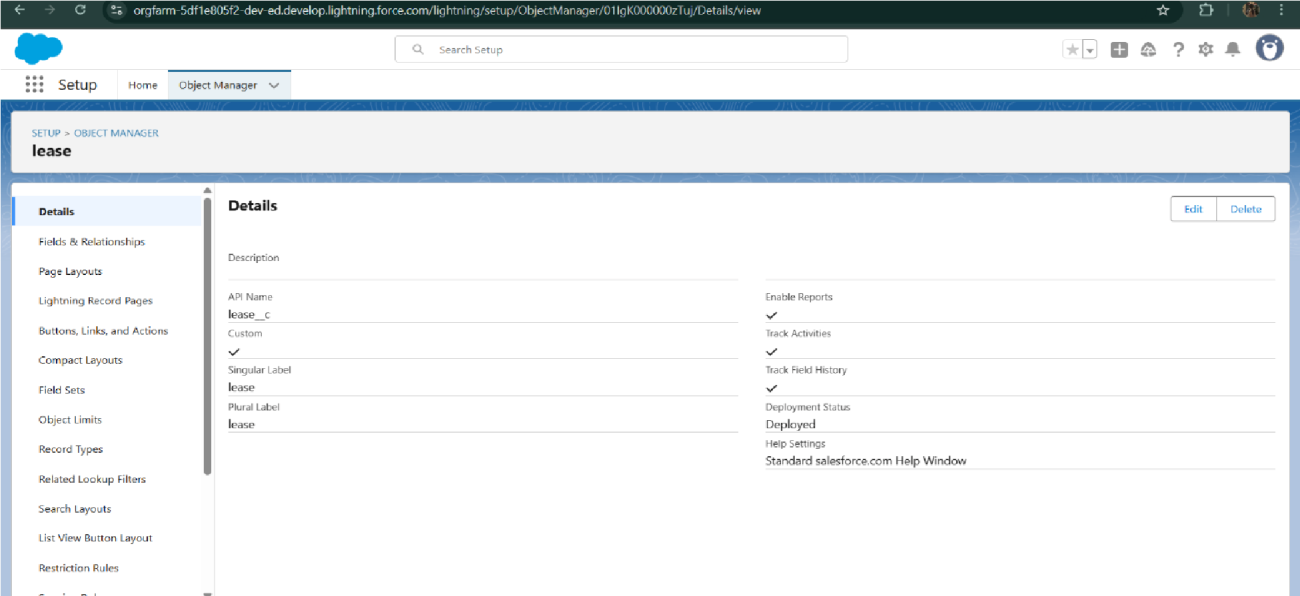
\* Data Type >> Text

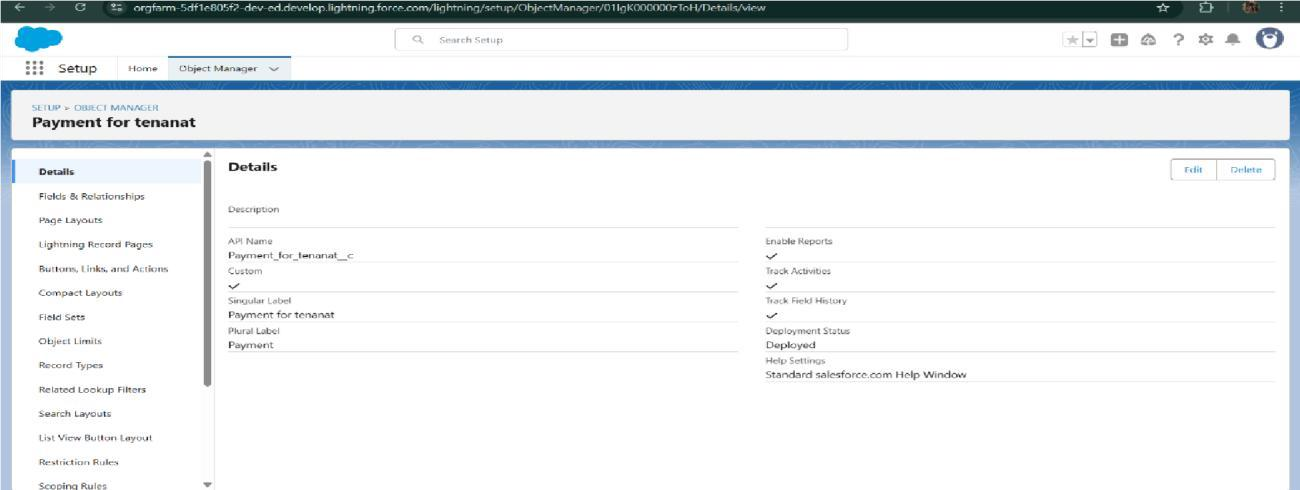
Click on Allow reports and Track Field History,Allow Activities

Allow search >> Save.





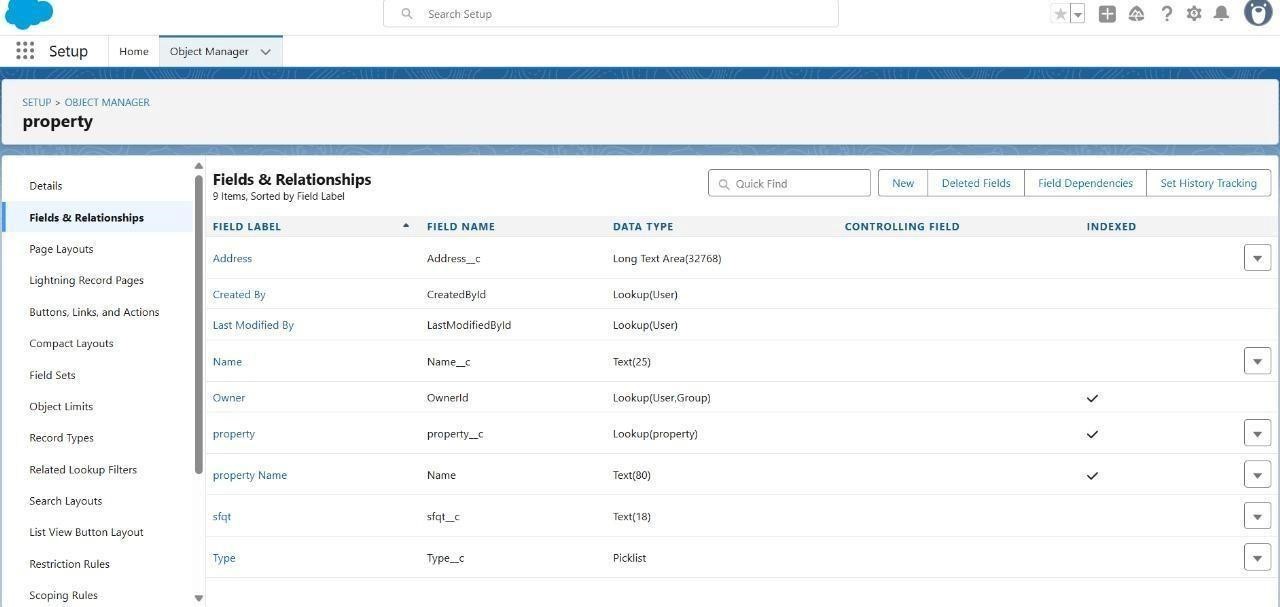


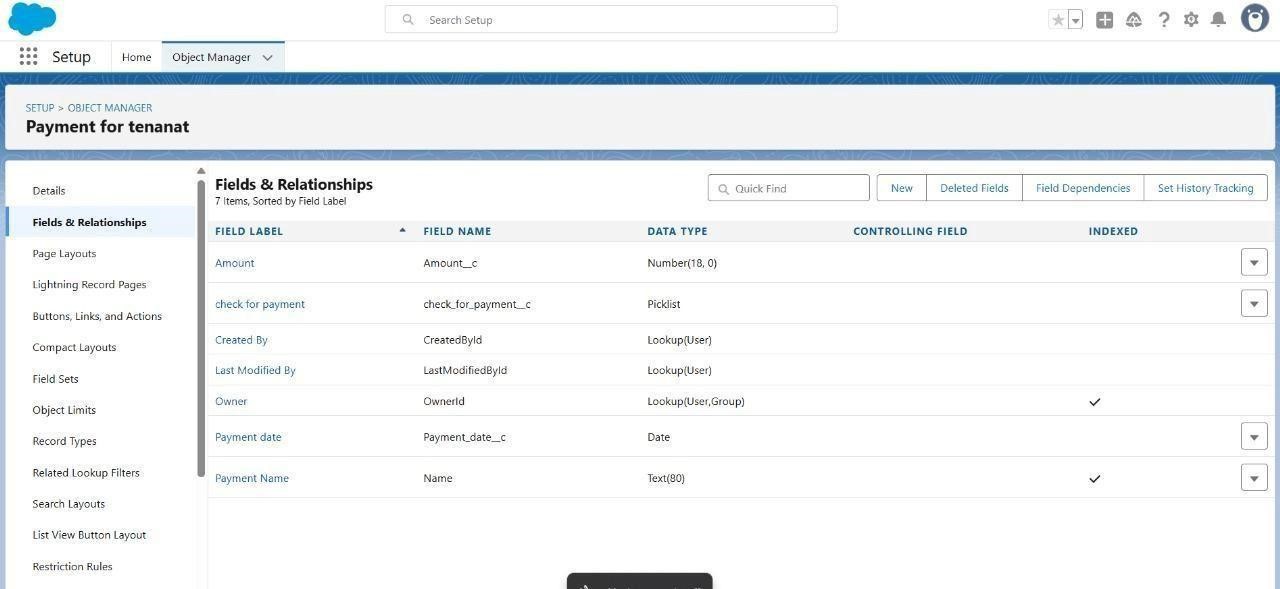


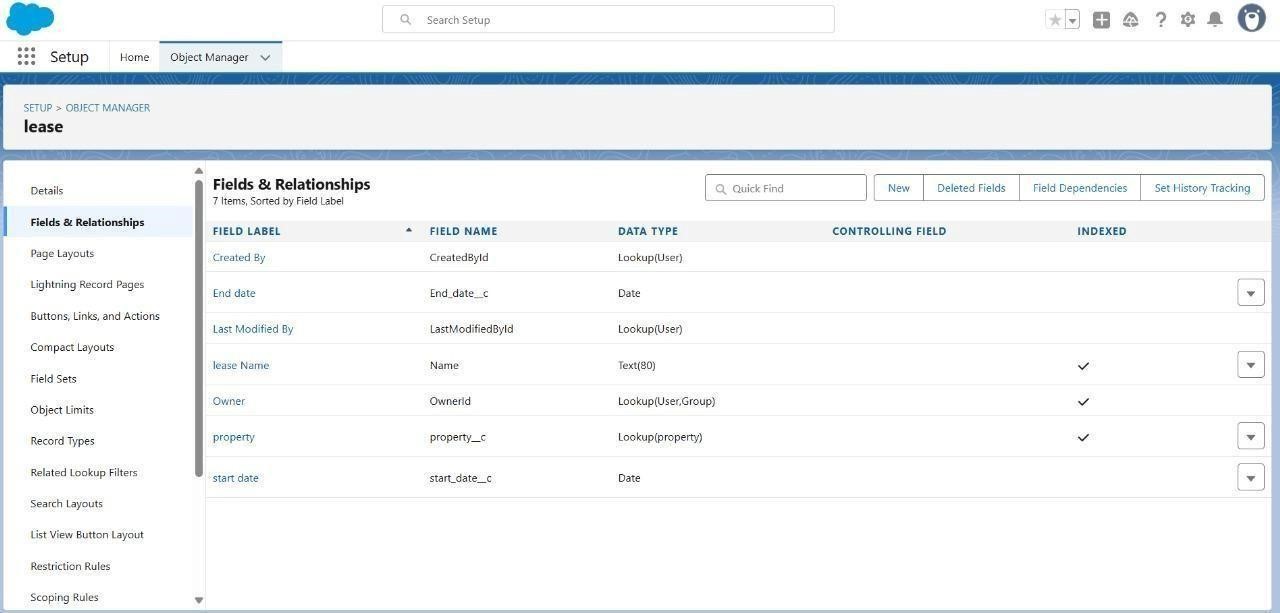
* Configured fields and relationships

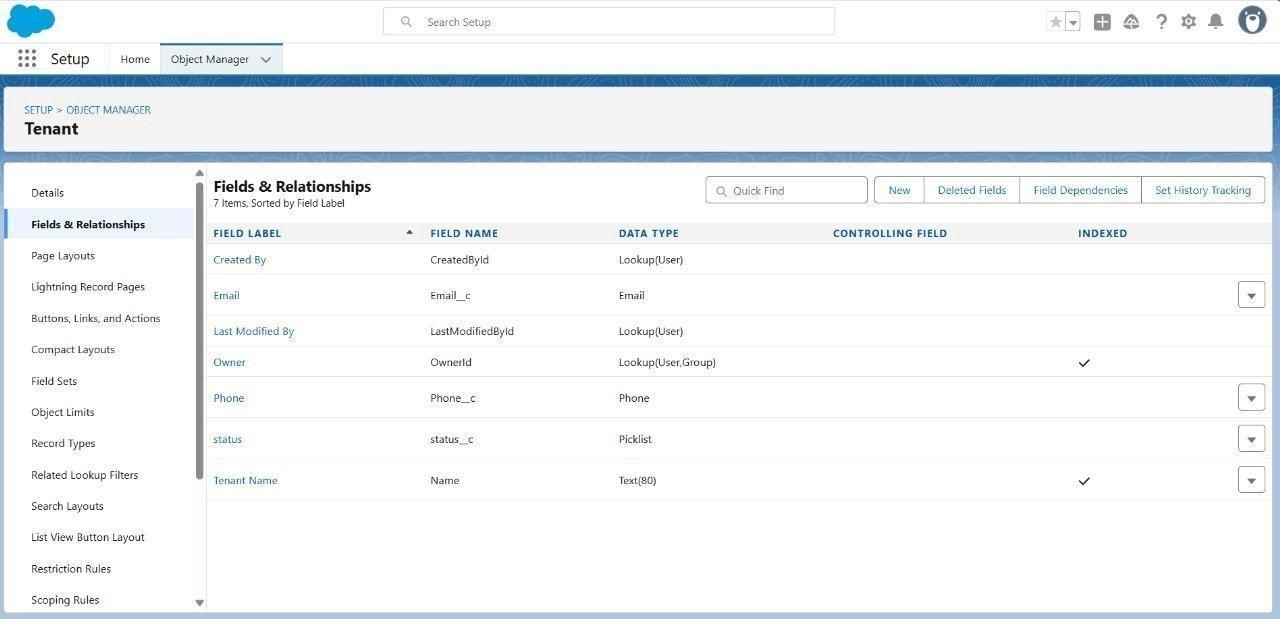
To create fields in an object:

1. Property object
2. Tenant object
3. Lease management
4. Payment for tenant object
5. Lookup fields





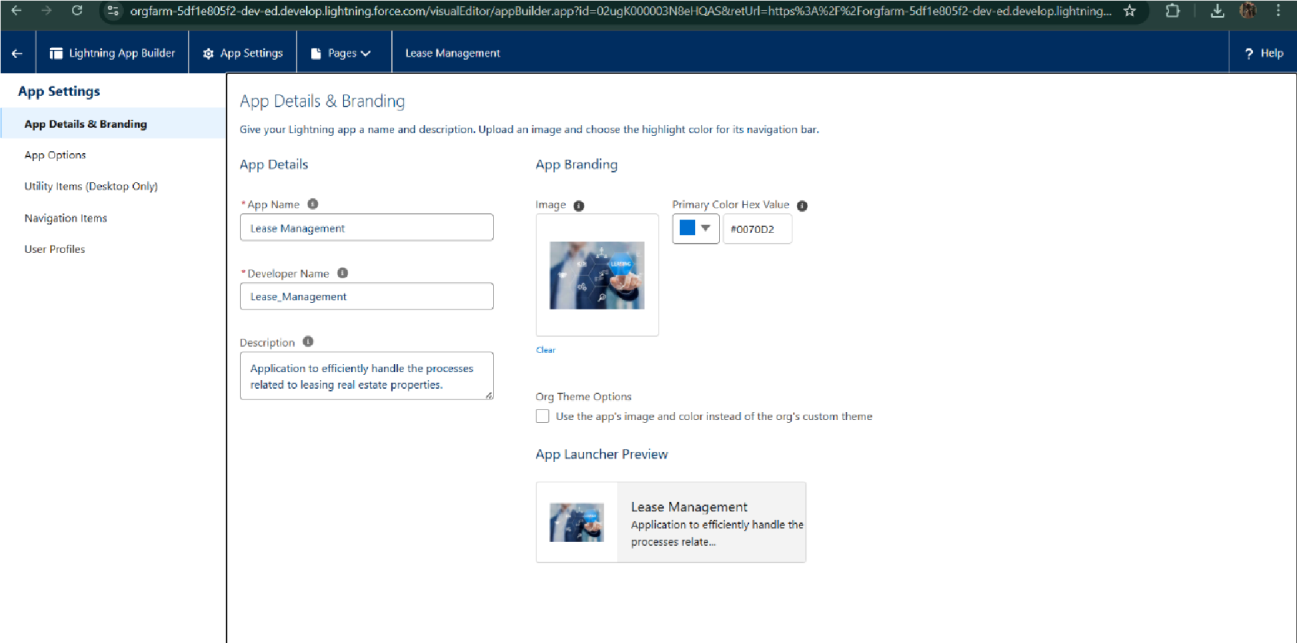




* Developed Lightning App with relevant tabs

To create a lightning app page

1. Go to setup page >> search “app manager” in quick find >> select “app manager”>> click on New lightning App.



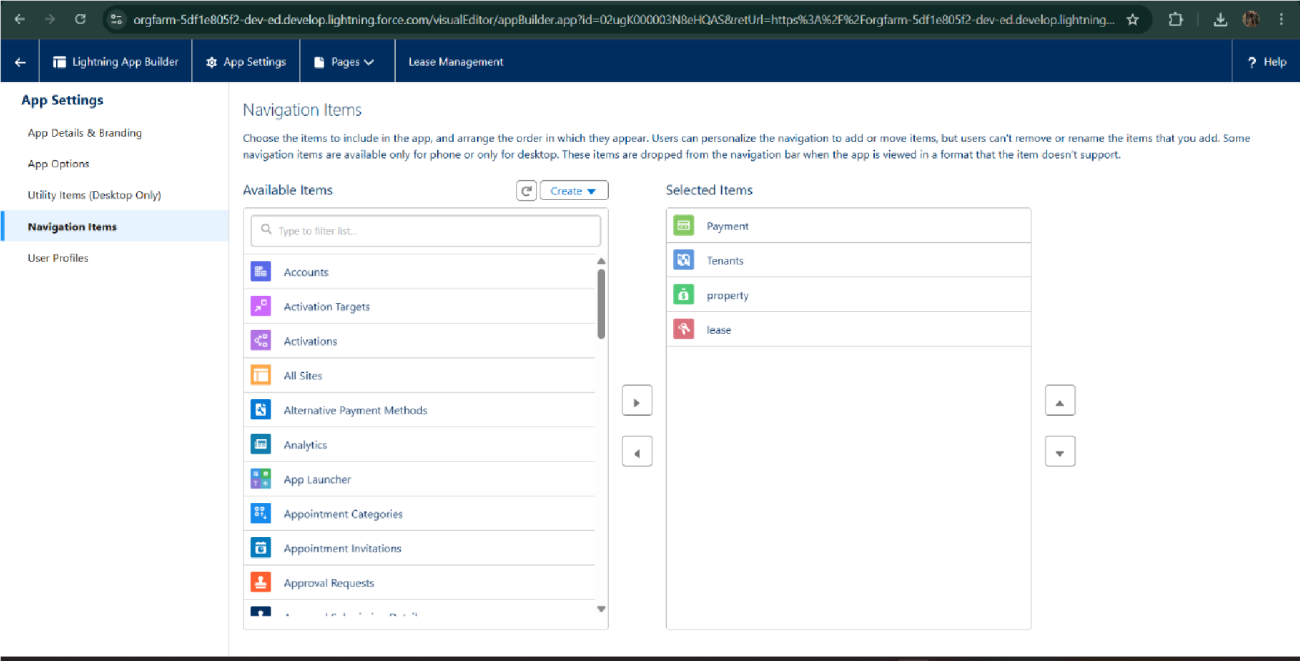
1. Fill the app name in app details and branding as follow

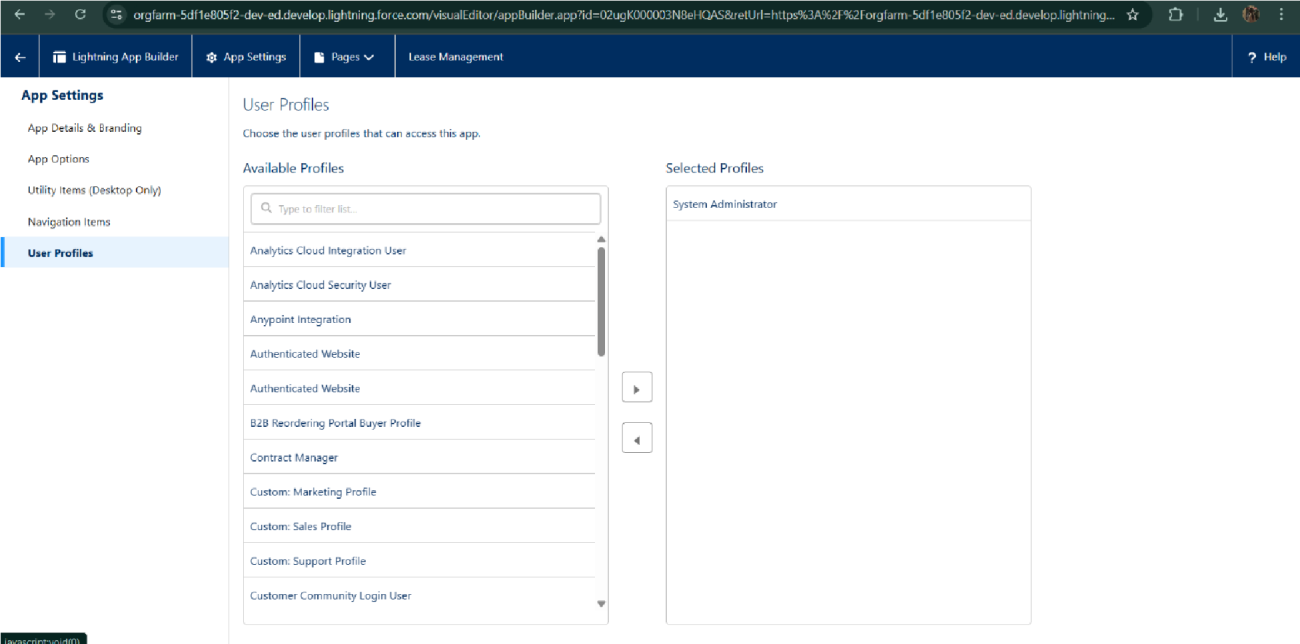
App Name : Lease ManagementDeveloper Name : This will auto populatedImage : optional (if you want to give any image you can otherwise not mandatory) Primary colour hex value : keep this default.

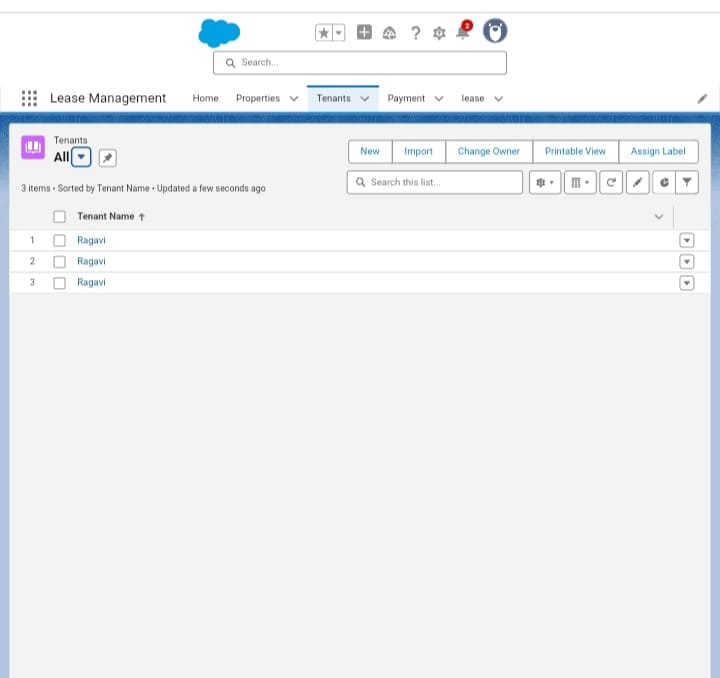
3.Then click Next >> (App option page)Set Navigation Style as Standard Navigation >> Next

1. To Add Navigation Items: Search for the item in the (Payment for tenant, Tenants,property,lease) from the search bar and move it using the arrow button ? Next? Next.

5. To Add User Profiles: Search profiles (System administrator) in the search bar >>click on the arrow button >> save & finish.







. Implemented Flows for monthly rent and payment success

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the record Triggered flow.Click on create
3. Under Object select ”Payment for tenant”. Click on A record is updated.

4. Set Entry Conditions

Under Condition Requirements

All Conditions are met

Field: check\_for\_payment\_\_c

Operator: Equals

Value : paid

5. Click on : Every time a record is updated and meets the condition requirements

6. Click on : Actions and related records,done

7. Under record trigger flow click on “+” icon and select action

In action search for send email then click on send email (check below image)

8. Label : send email

API Name : send\_email

9. Label : send email

10. API Name : send\_email

11. Enable Body

12. Click on new resource

Under resource type select “Text Template”

API Name : emailbody

Under body: (paste the below text)

Dear {!$Record.Tenant\_\_r.Name},

14. Click Done.

15. Enable recipient Address List

Paste this ?{!$Record.Tenant\_r.Email\_c}

16. Click Done

17. Enable subject

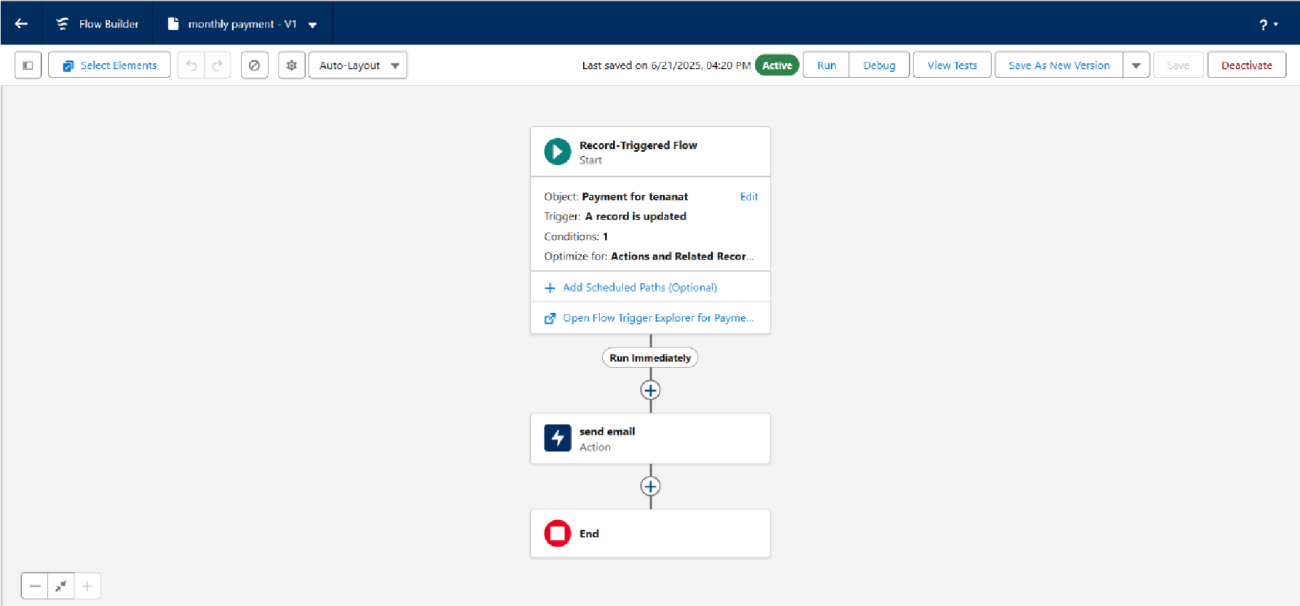
Pate this >> Confirmation of Successful Monthly Payment

18. Click on save

Flow label : monthly payment

Flow API Name : monthly\_payment

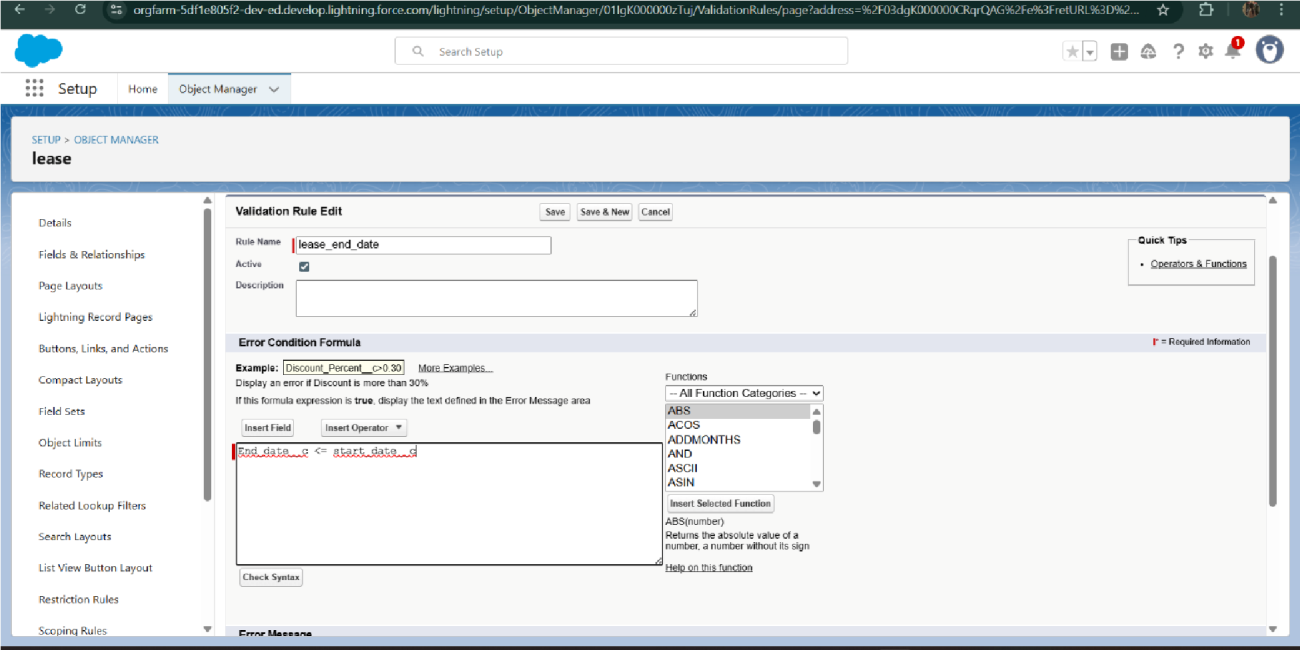
Click on activate



* To create a validation rule to a Lease Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.

2. Click on the validation rule >> click New.

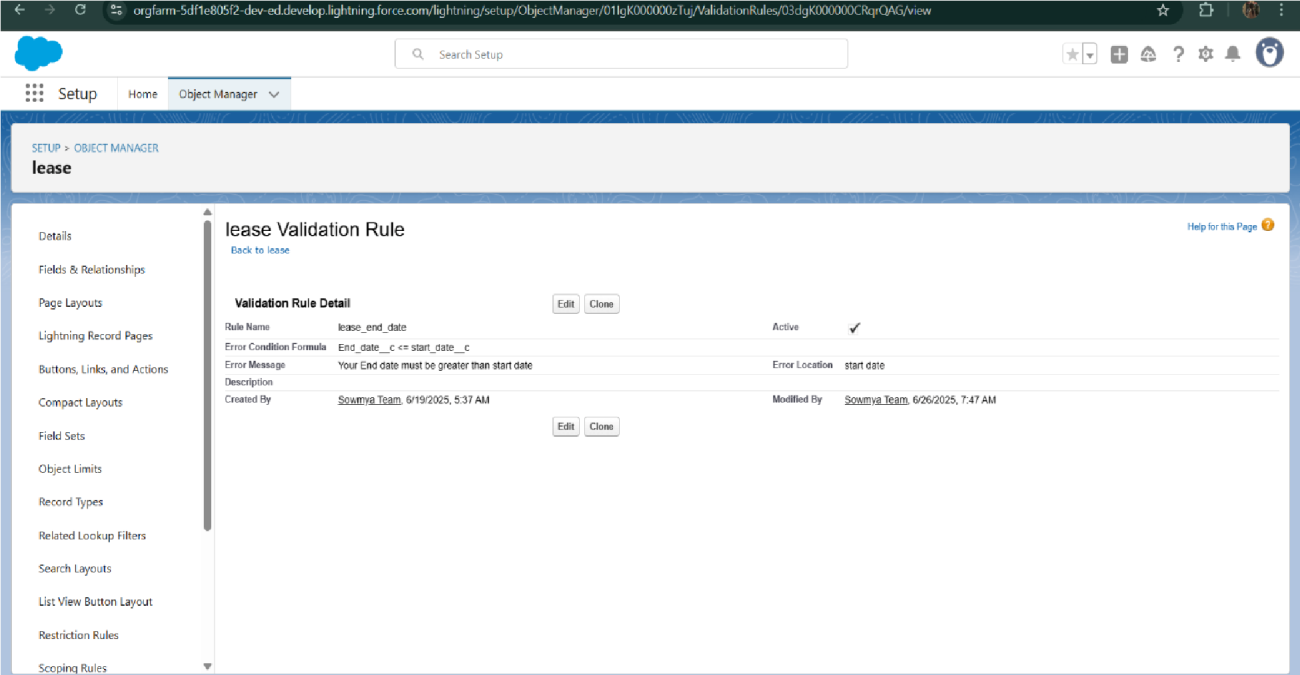


3. Enter the Rule name as “ lease\_end\_date”.

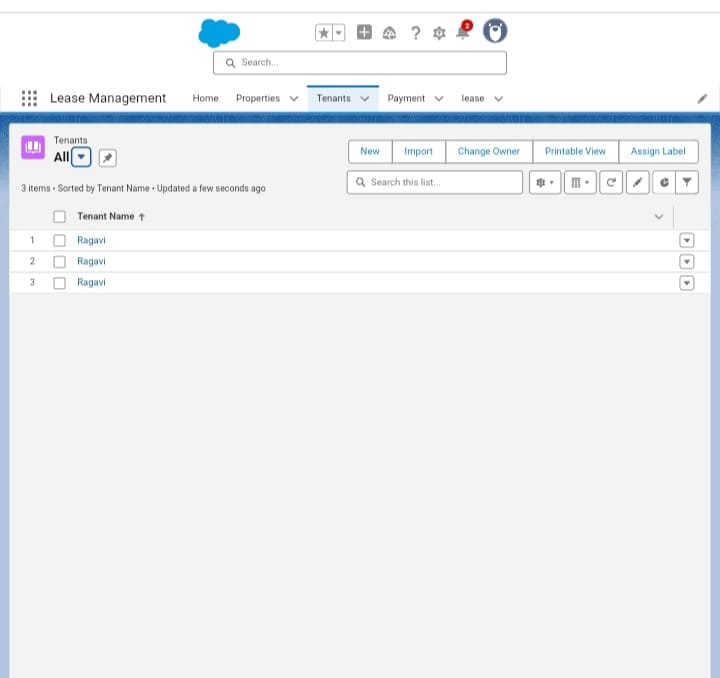
4. Insert the Error Condition Formula as :

End\_date\_c > start\_date\_c

5. Enter the Error Message as “Your End date must be greater than start date”, select the Error location as Field and select the field as “start date”, and click Save.



* Added Apex trigger to restrict multiple tenants per property



* Scheduled monthly reminder emails using Apex class

1. To create a new Apex Class follow the below steps:

Click on the file >> New >>Apex Class.

1. Enter class name as testHandler.

Apex logic:

public class testHandler {

public static void preventInsert(List<Tenant\_\_c> newlist) {

Set<Id> existingPropertyIds = new Set<Id>()

for (Tenant\_c existingTenant : [SELECT Id, Propertyc FROM Tenantc WHERE Property\_c != null]) {

existingPropertyIds.add(existingTenant.Property\_\_c);

}

for (Tenant\_\_c newTenant : newlist) {

if (newTenant.Property\_c != null && existingPropertyIds.contains(newTenant.Property\_c)) {

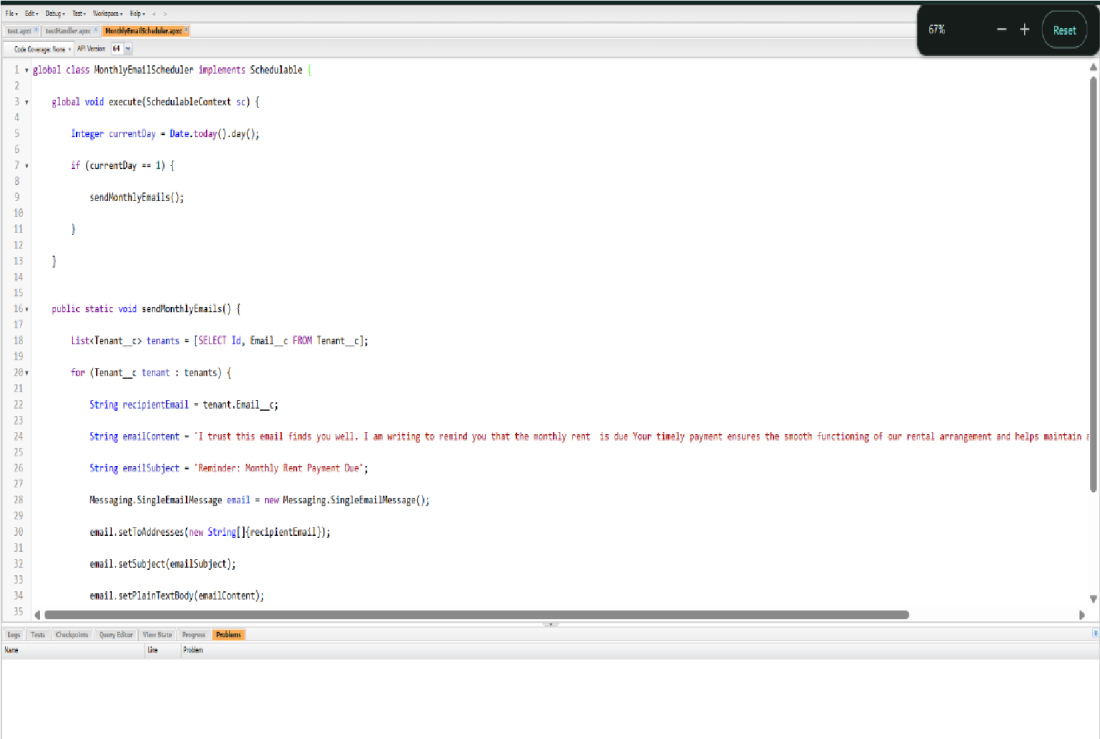
newTenant.addError('A tenant can have only one property');

}

}

}

}



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

To create Email Template:

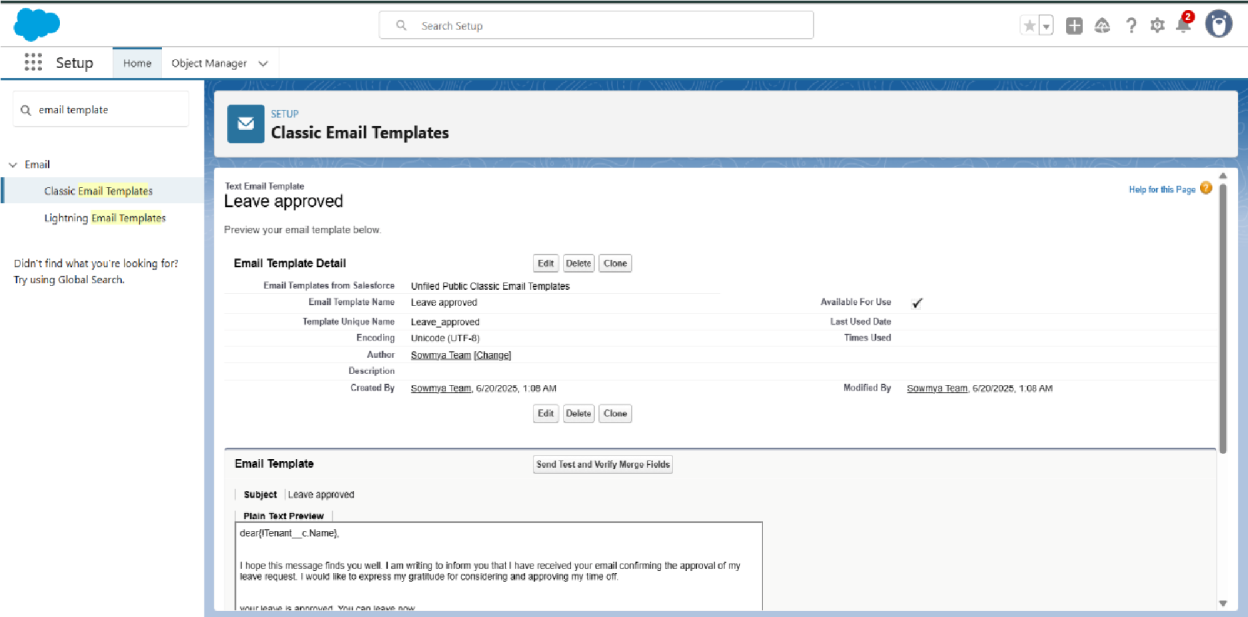
1.Tenant leaving

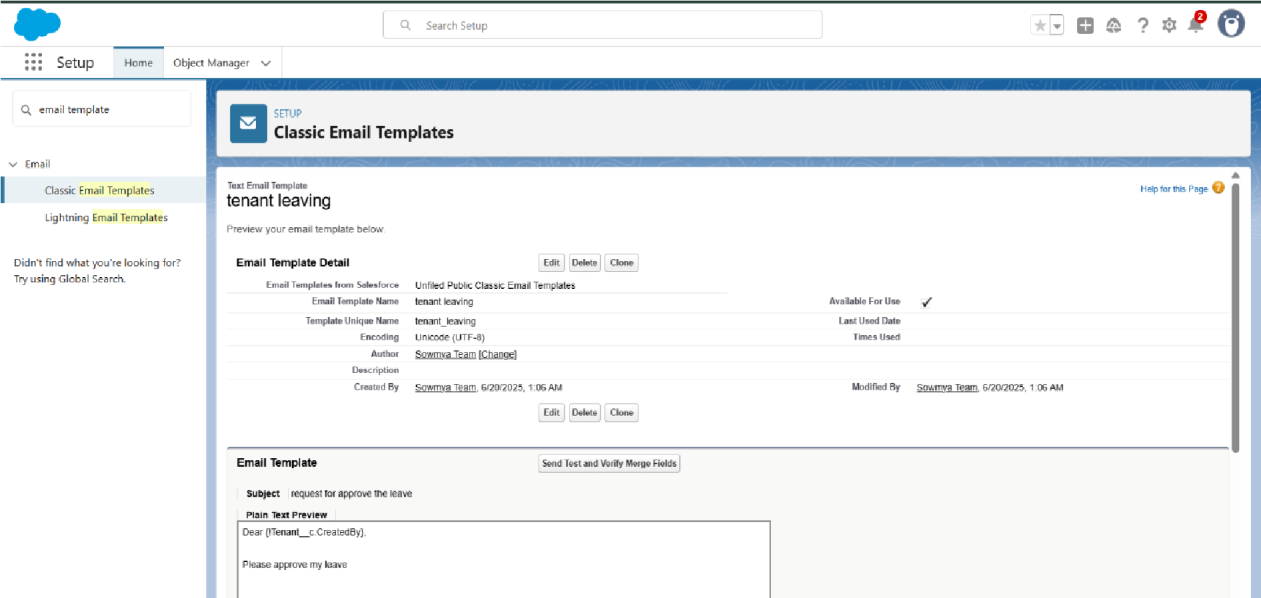
2. leave approved

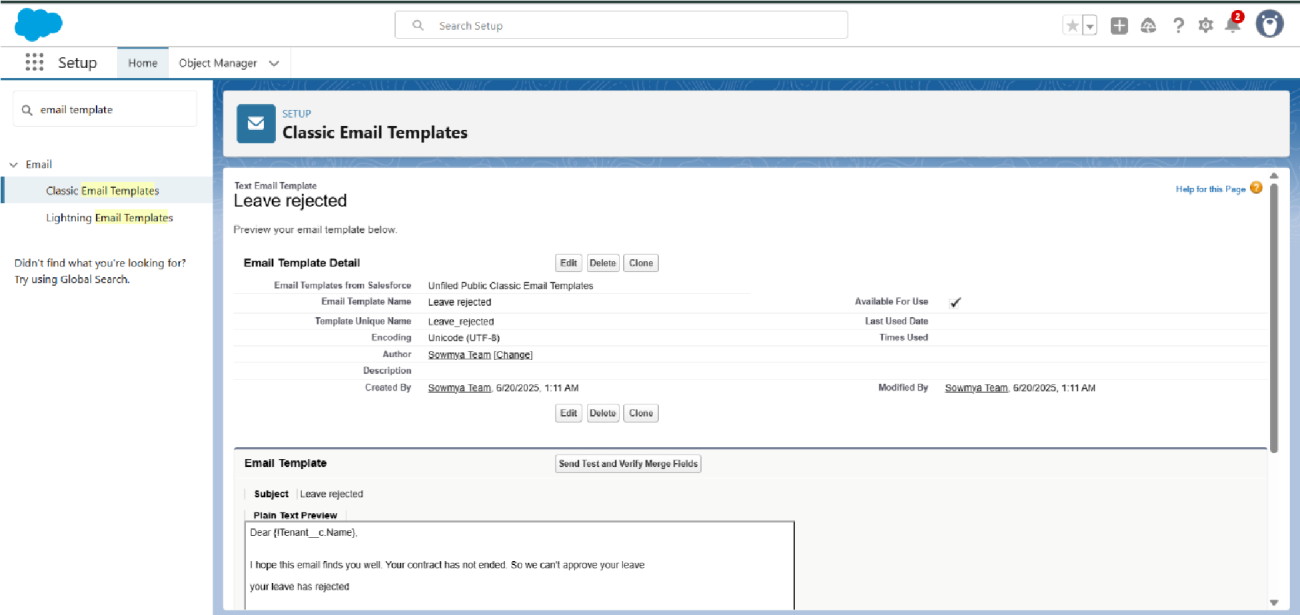
3. rejection for leave

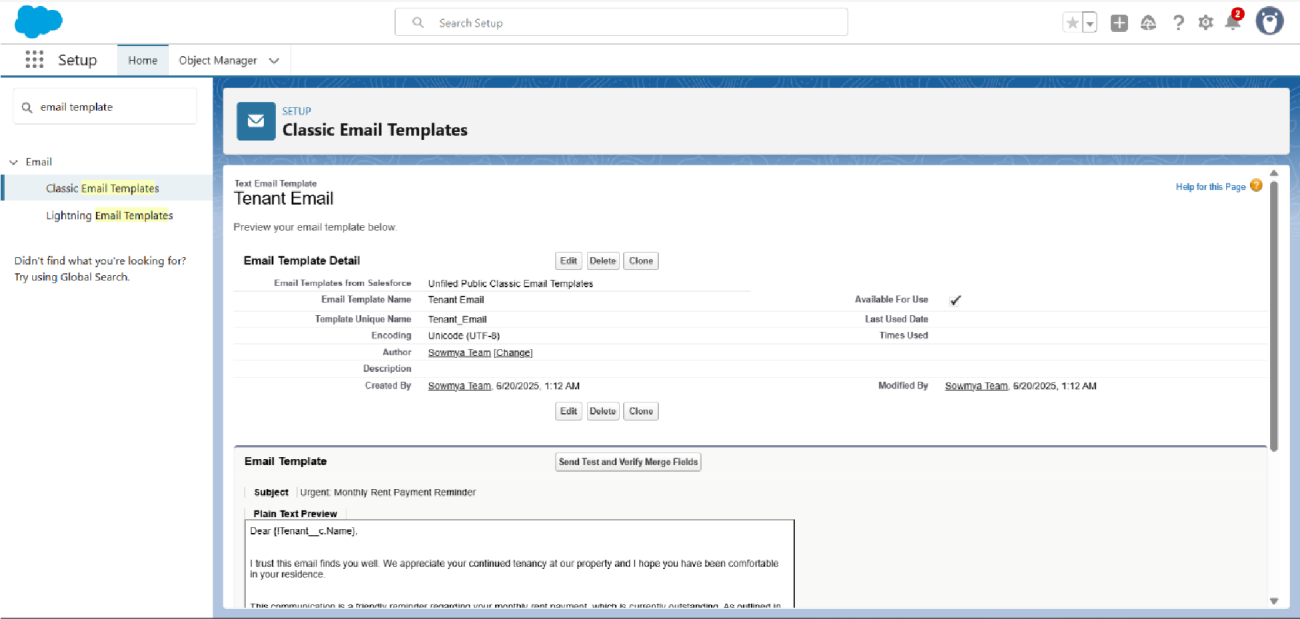
4. monthly payment

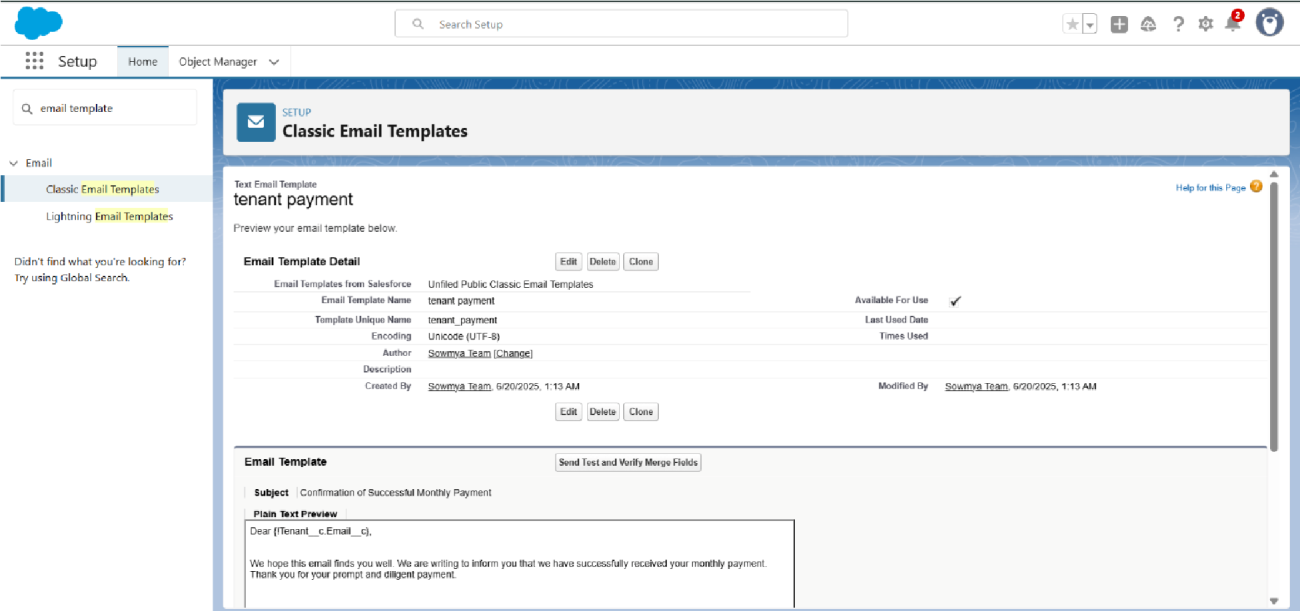
5. successful payment





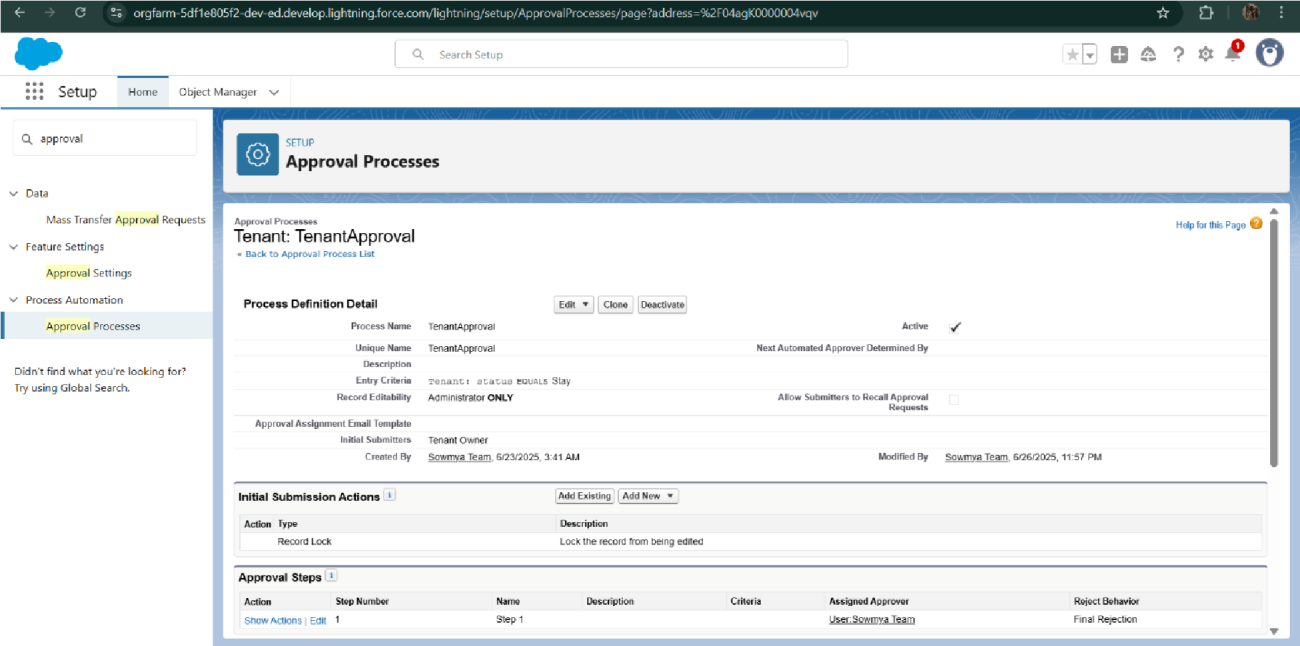




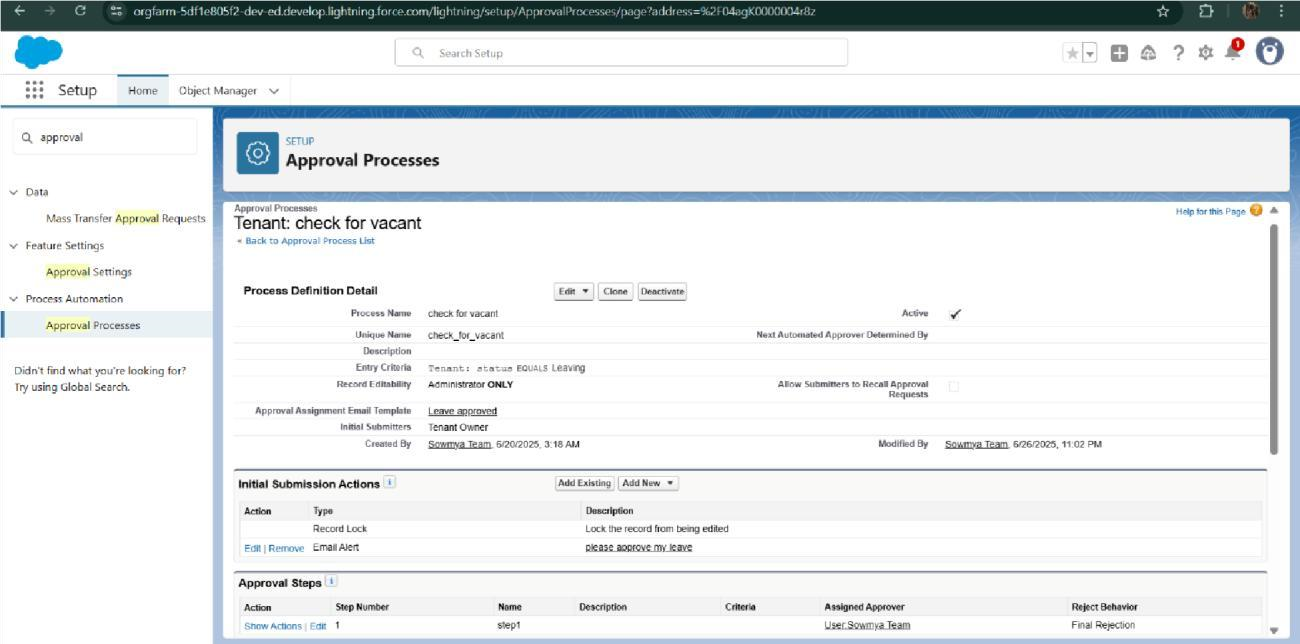


* Approval Process creation

For Tenant Leaving:



For Check for Vacant:



* Apex Trigger

Create an Apex Trigger

1. To create a new Apex Class follow the below steps:

Click on the file >> New ? Apex Class.

1. Give the Apex Trigger name as “test”, and select “Tenant\_\_c” from the dropdown for sObject.
2. Click Submit.
3. Now write the code logic here

Trigger Code:

trigger test on Tenant\_\_c (before insert)

{

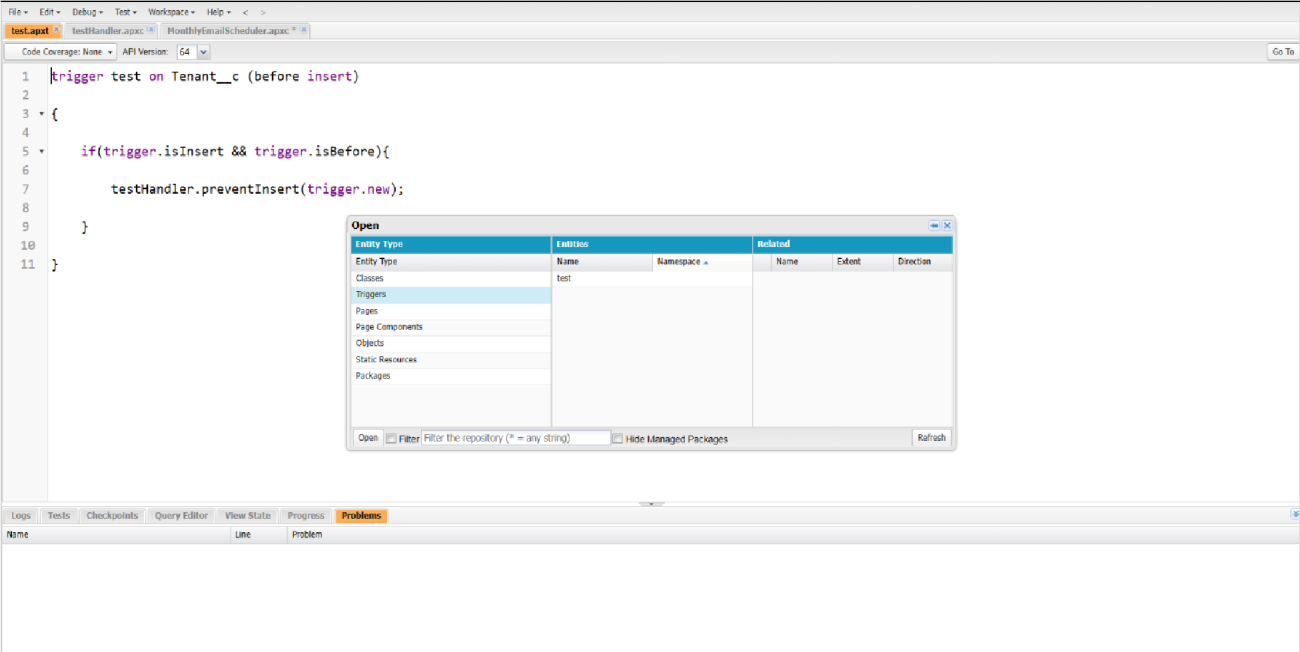
if(trigger.isInsert && trigger.isBefore)

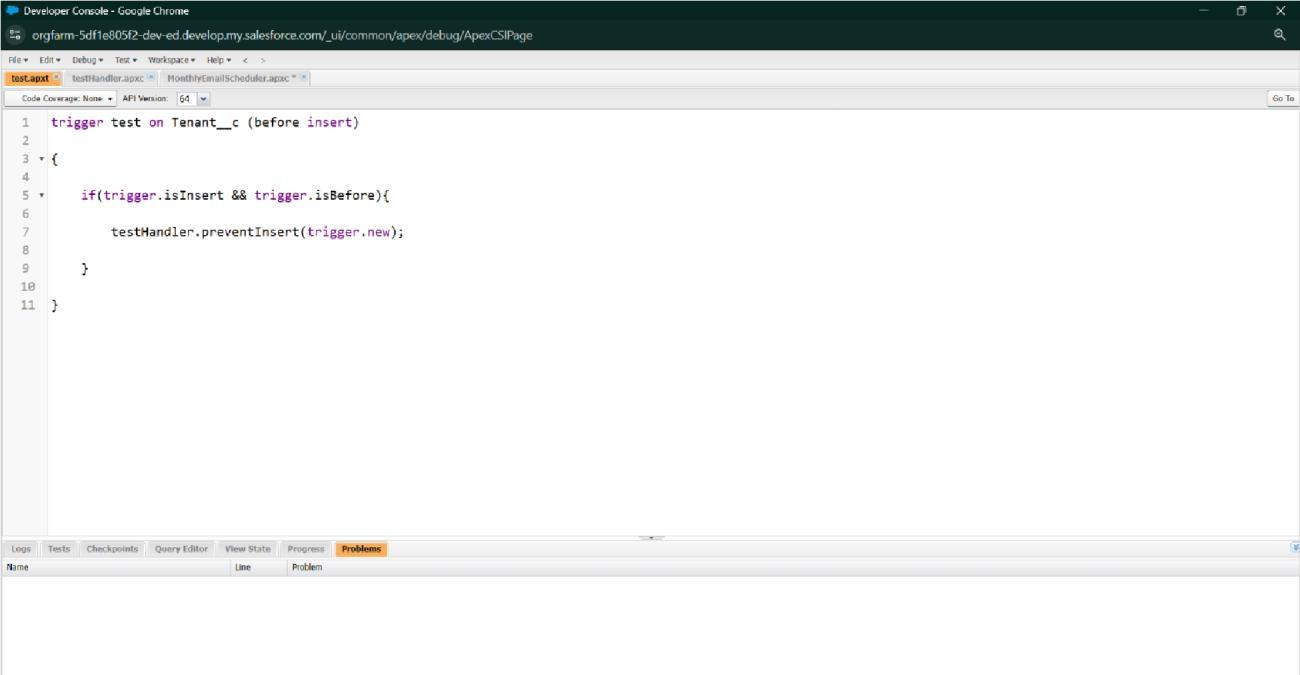
{

testHandler.preventInsert(trigger.new);

}

}





.Create an Apex Handler class

1. To create a new Apex Class follow the below steps:

Click on the file >> New >>Apex Class.

1. Enter class name as testHandler.

Apex logic:

public class testHandler {

public static void preventInsert(List<Tenant\_\_c> newlist) {

Set<Id> existingPropertyIds = new Set<Id>();

for (Tenant\_c existingTenant : [SELECT Id, Propertyc FROM Tenantc WHERE Property\_c != null]) {

existingPropertyIds.add(existingTenant.Property\_\_c);

}

for (Tenant\_\_c newTenant : newlist) {

if (newTenant.Property\_c != null && existingPropertyIds.contains(newTenant.Property\_c)) {

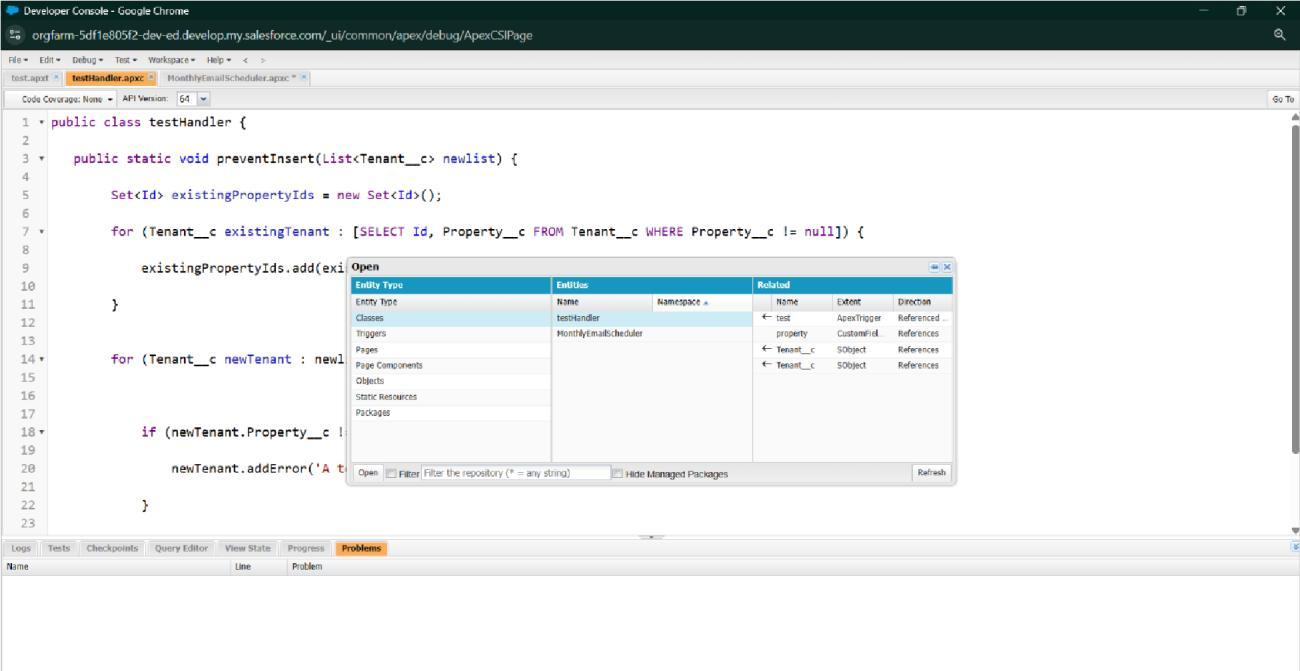
newTenant.addError('A tenant can have only one property');

}

}

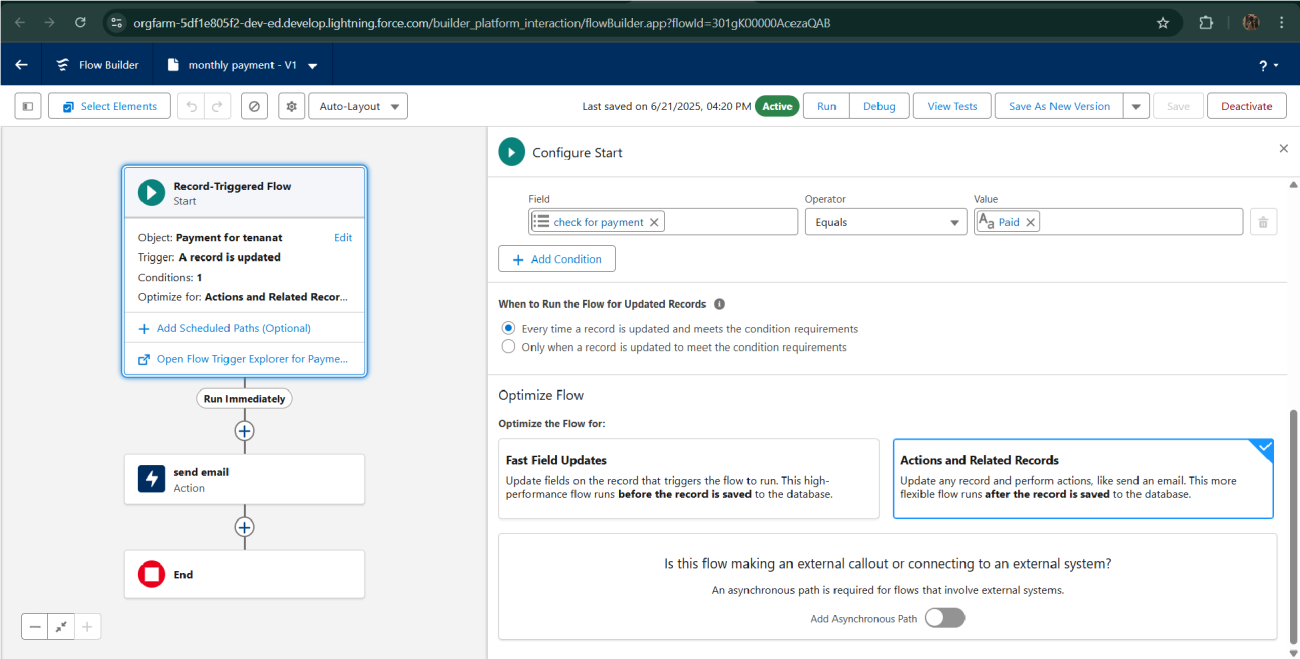
}

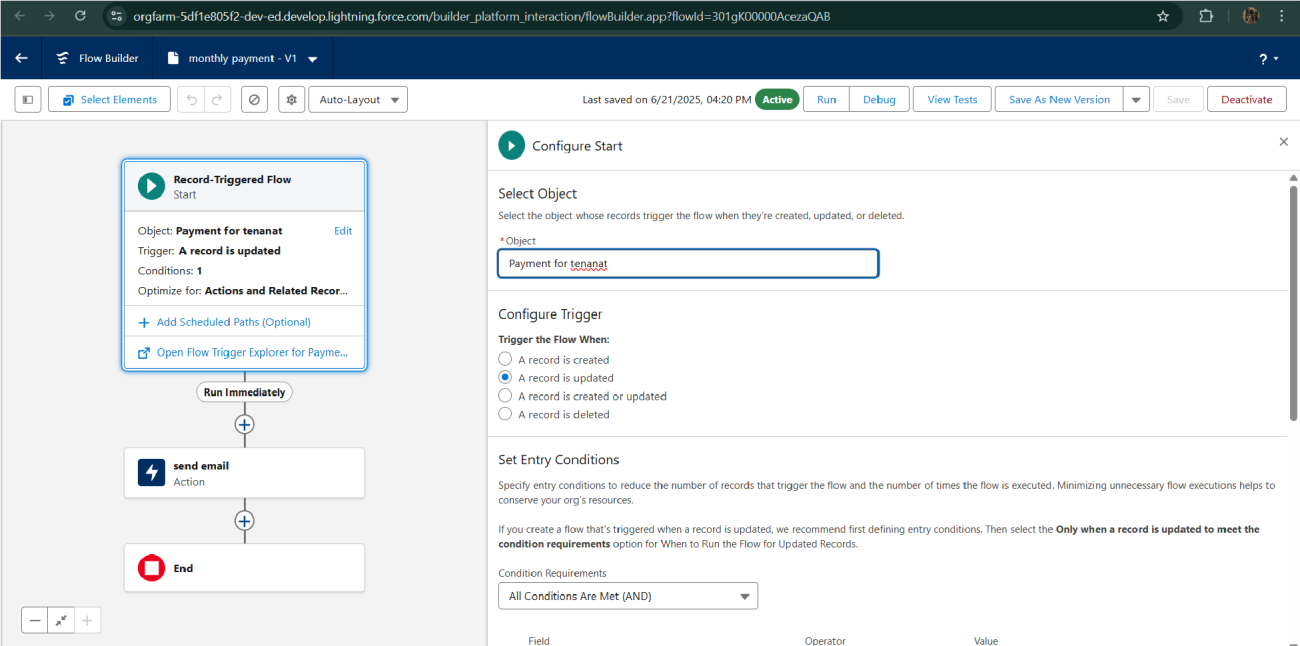
}





* FLOWS





* Schedule class:

Create an Apex Class

Apex logic:

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, Emailc 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);

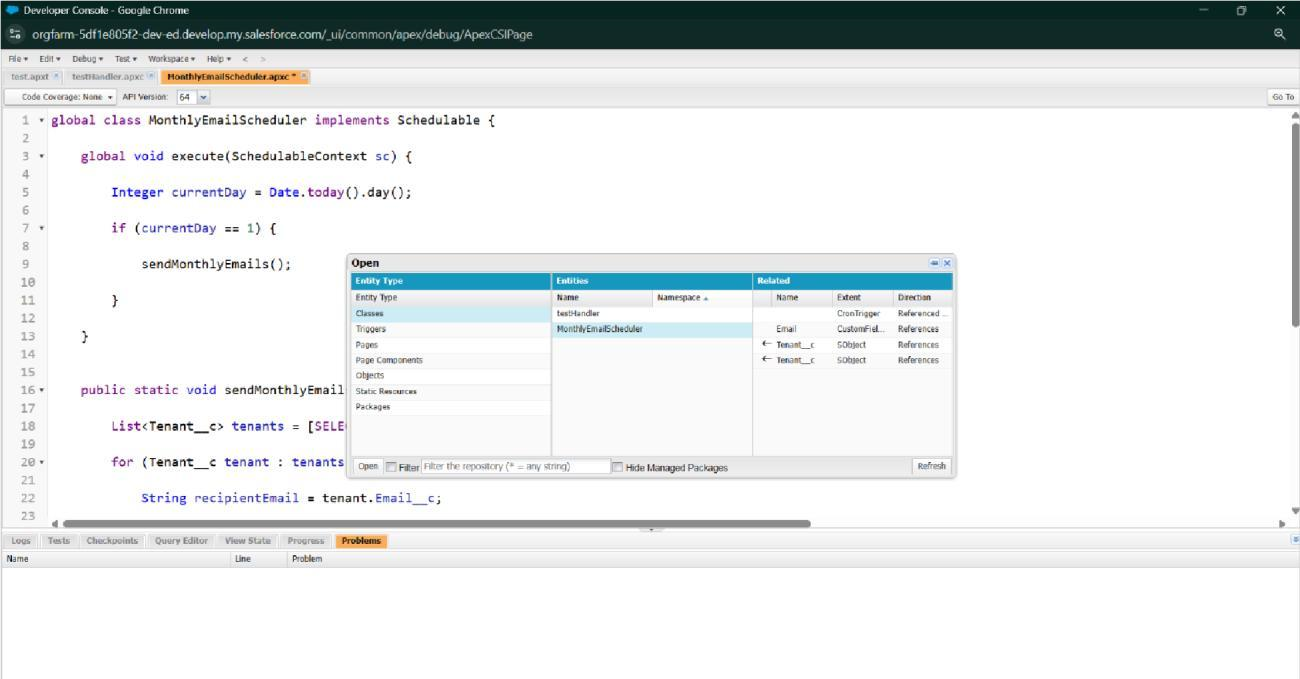
email.setPlainTextBody(emailContent);

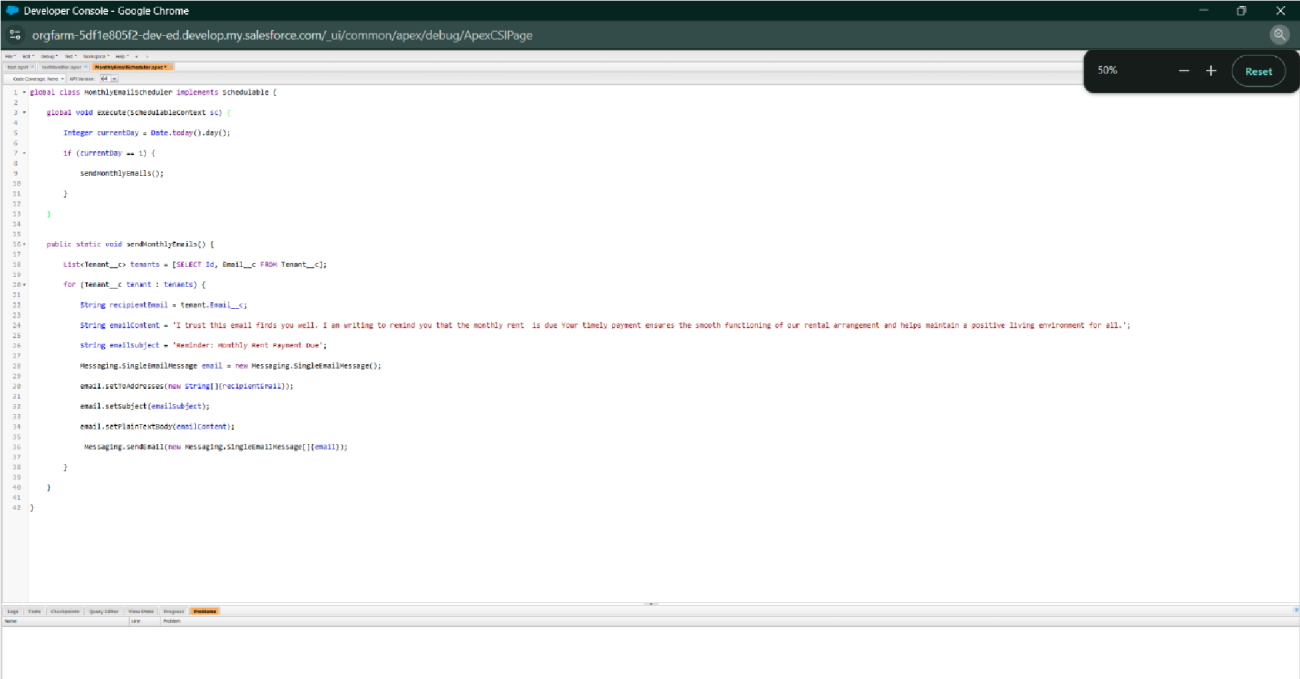
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});

}

}

}





Schedule Apex class

1. Enter Apex class in quick find box
2. Select schedule Apex
3. Enter job Name : MonthlyEmailScheduler

4 Apex class : MonthlyEmailScheduler

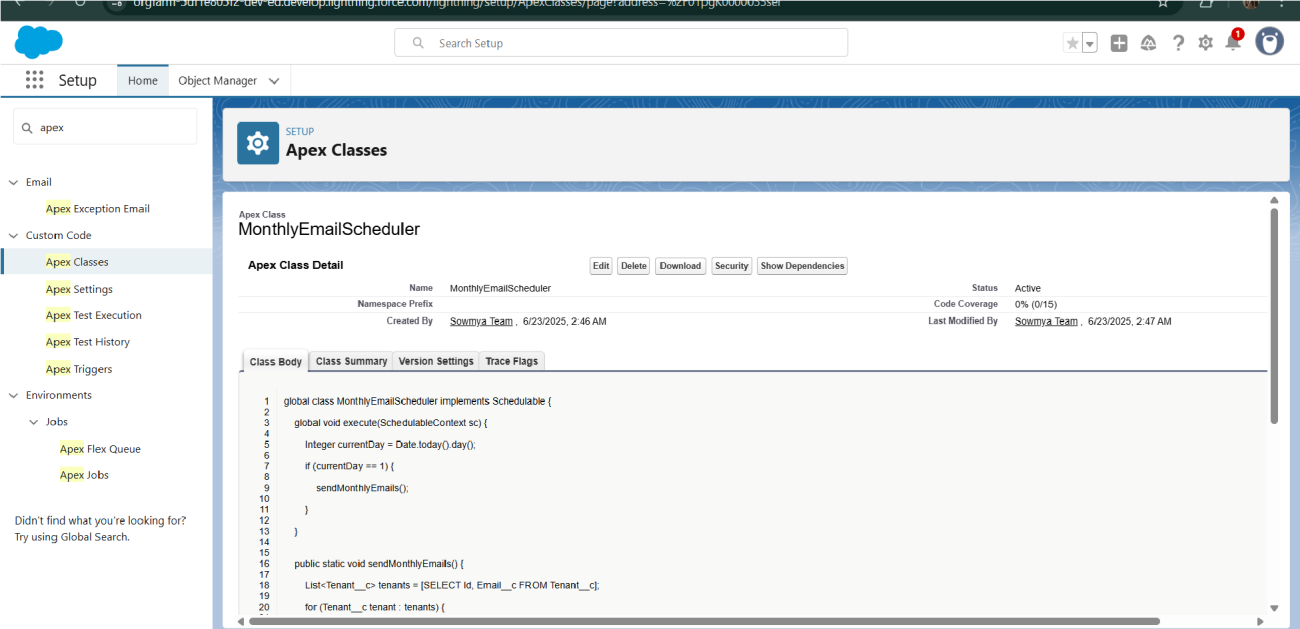
Frequency : Monthly===>select on day 1

Start date : 04/12/2023

End date : 04/01/2024

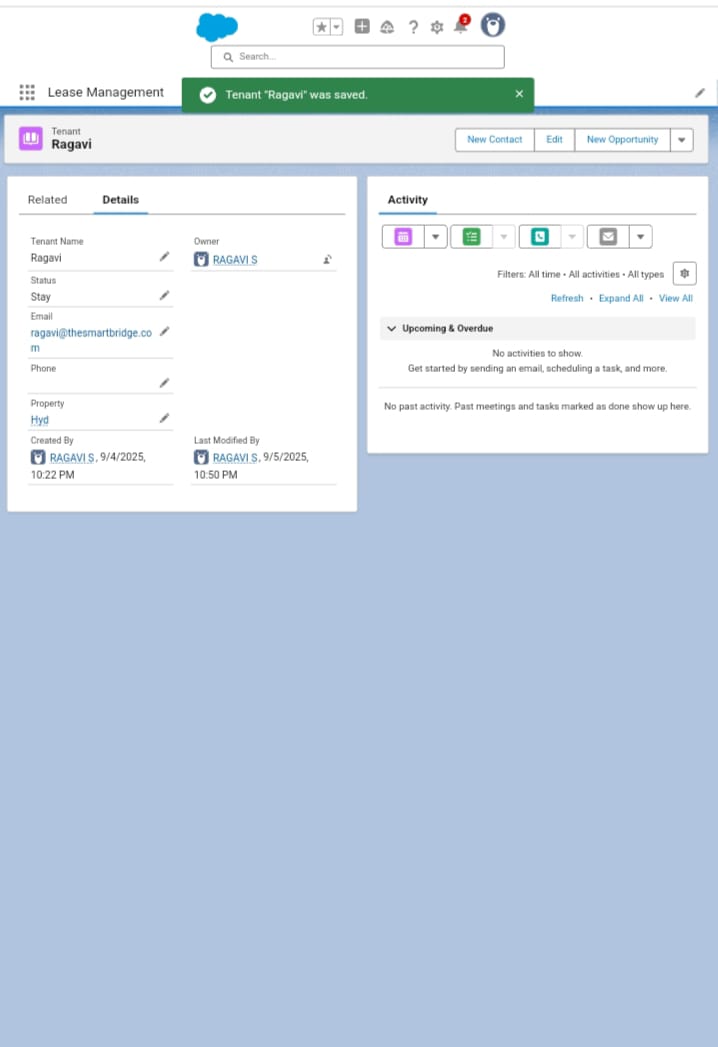
Preferred start time : 09:00 am

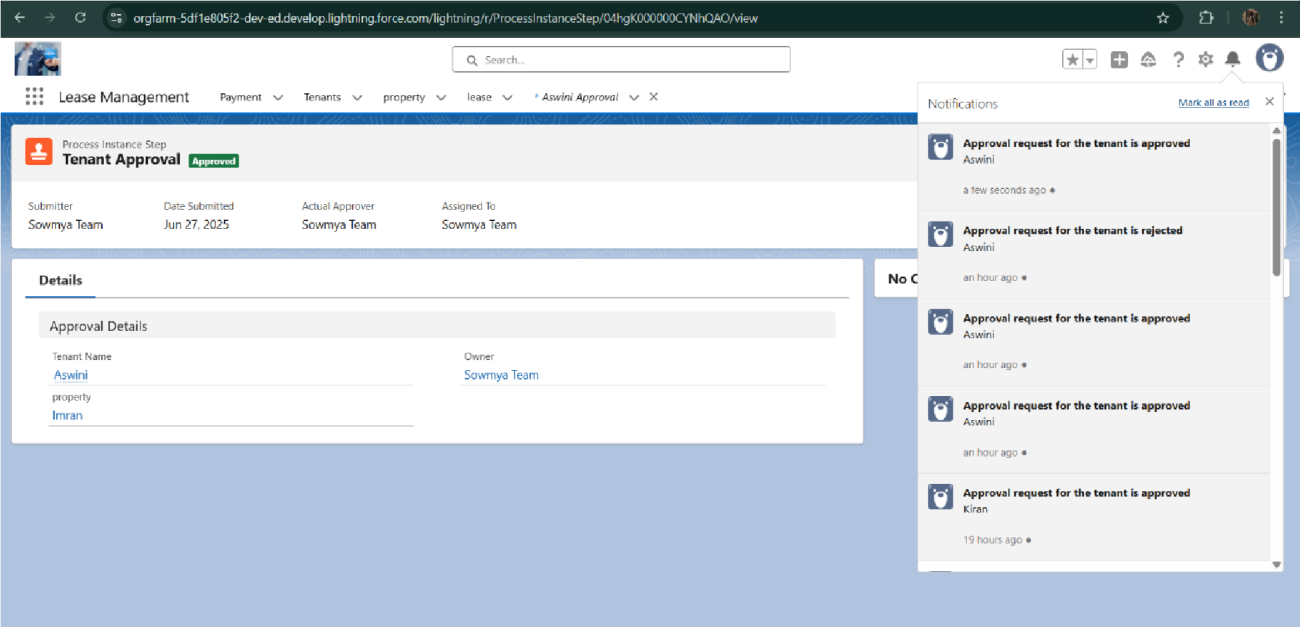
save



Approval process







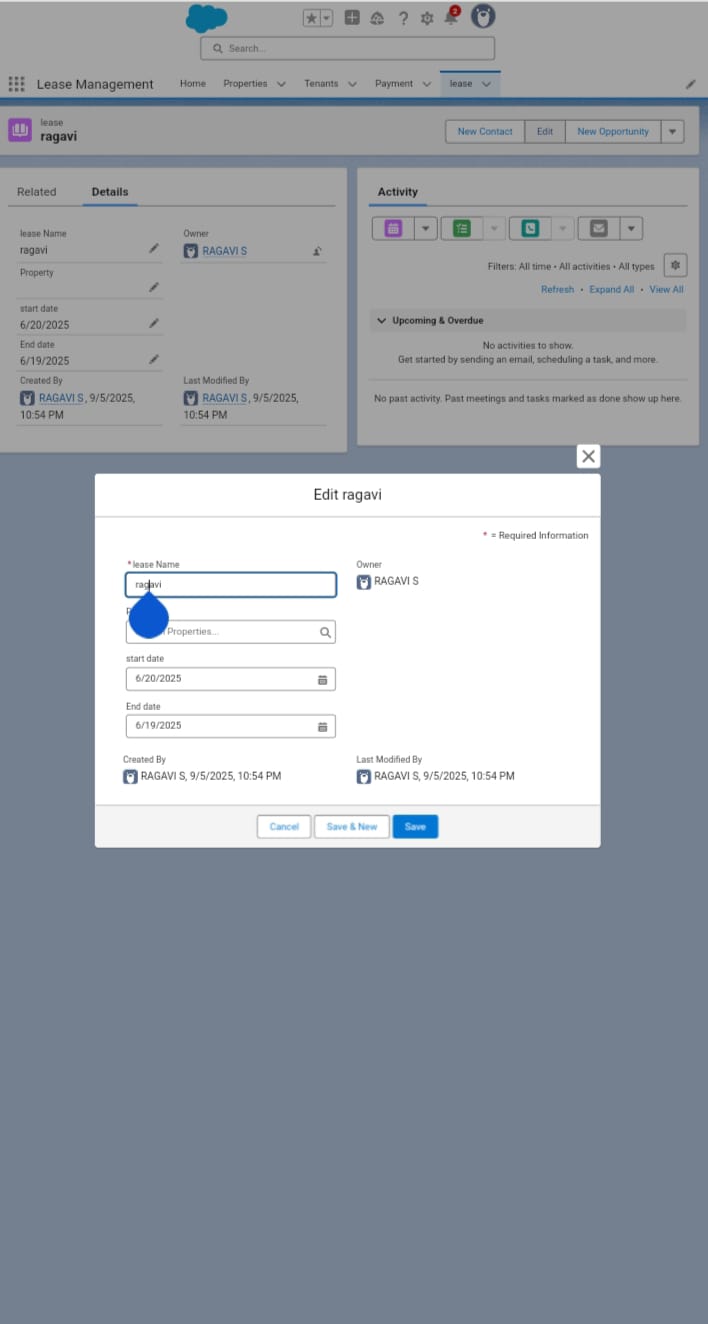
# FUNCTIONAL AND PERFORMANCE TESTING

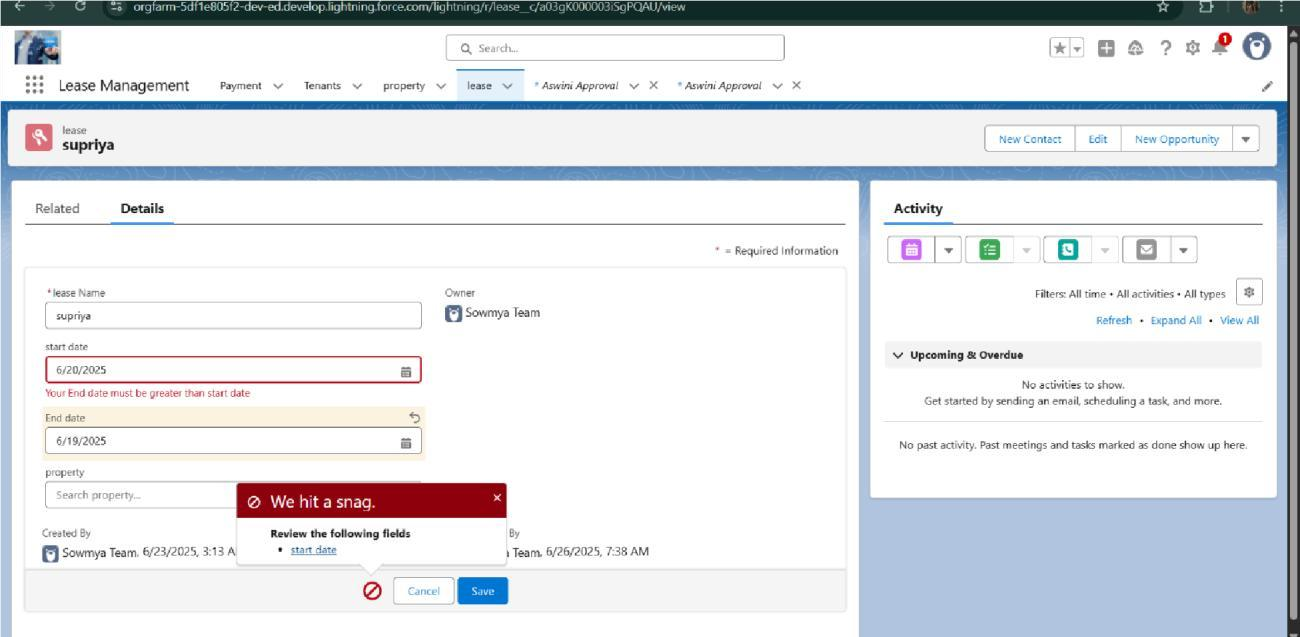
## Performance Testing

## 

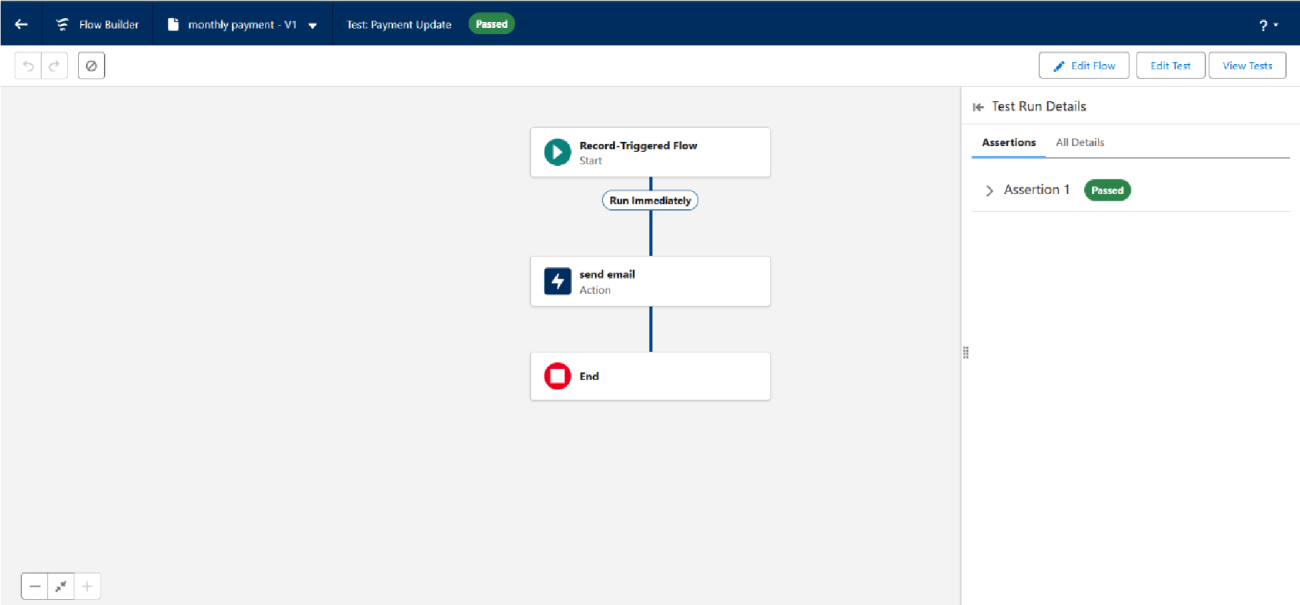
* Trigger validation by entering duplicate tenant-property records



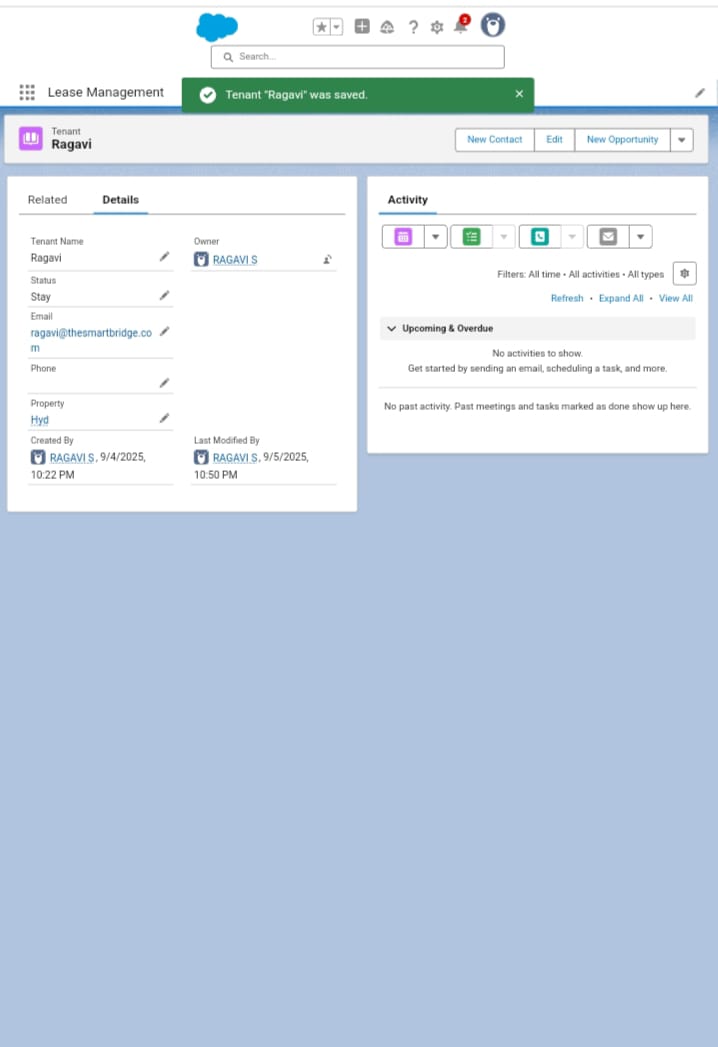


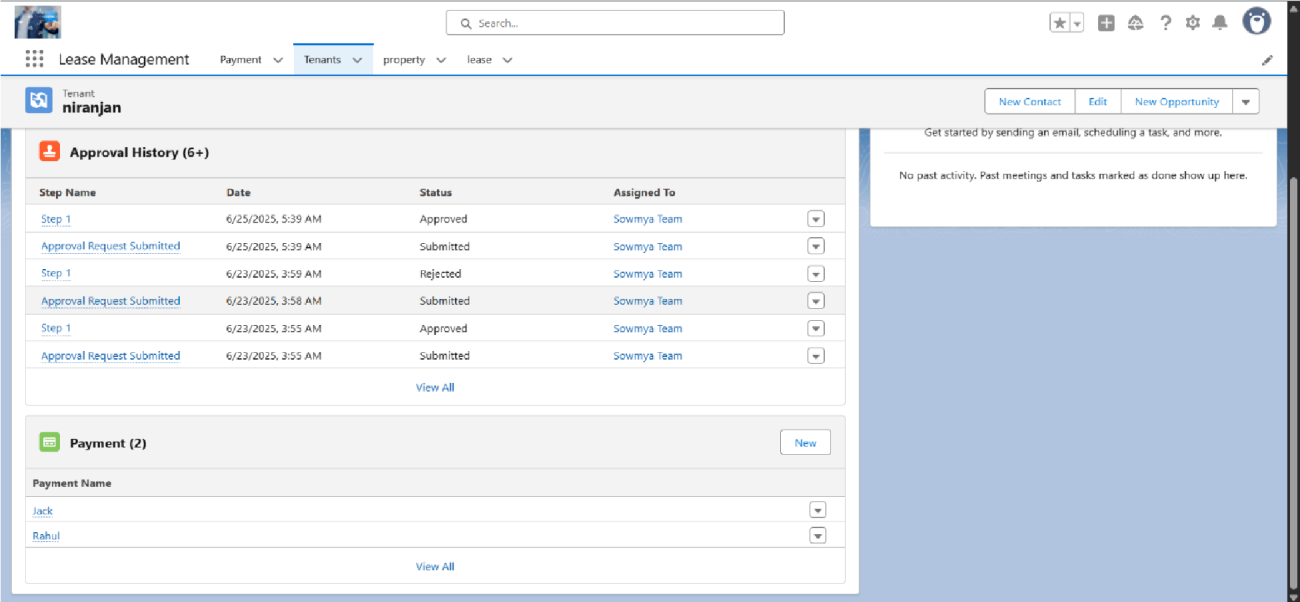


* Test flows on payment update



* Approval process validated through email alerts and status updates

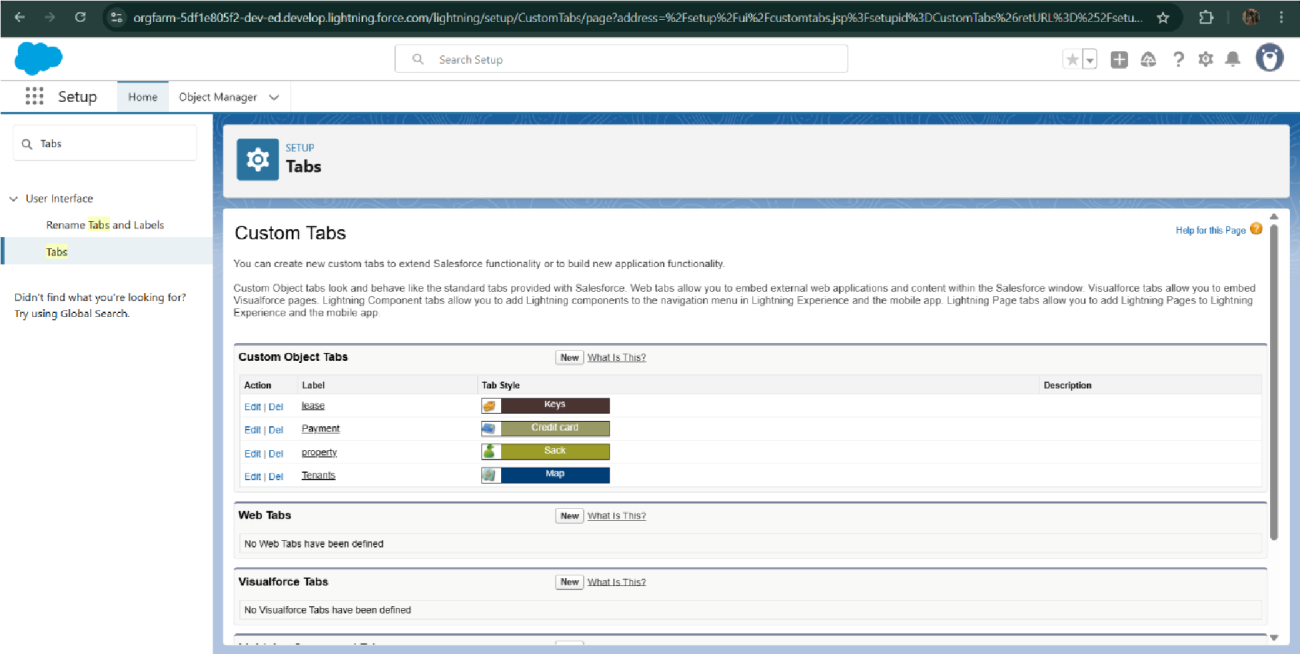




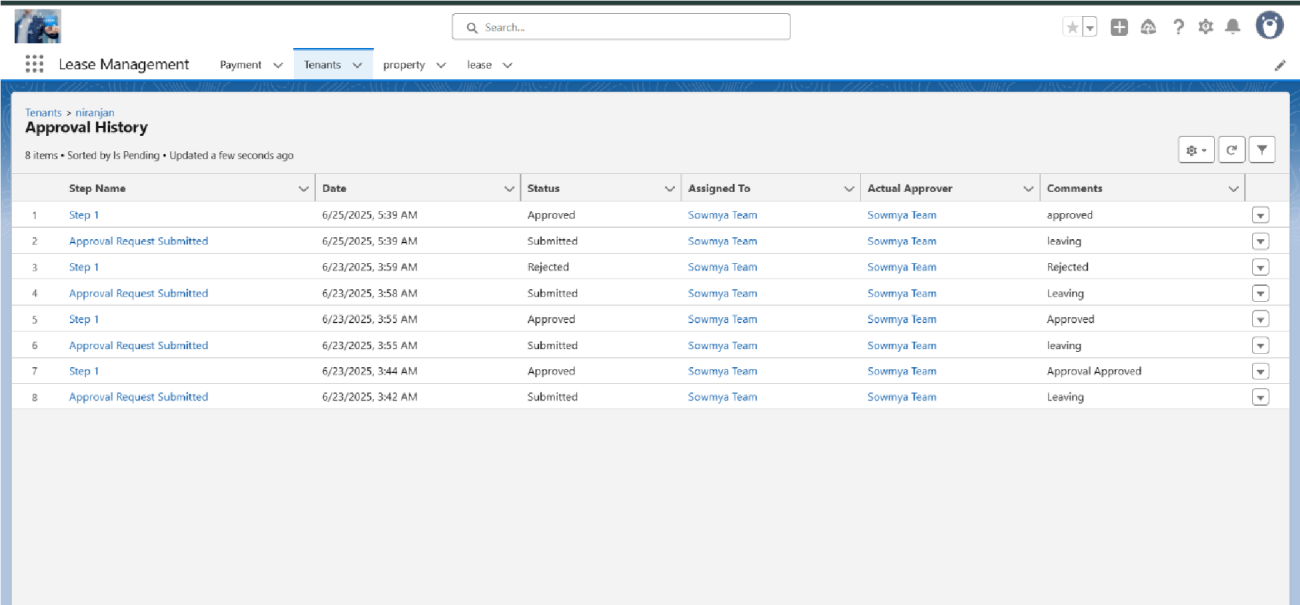
# RESULTS

## Output Screenshots

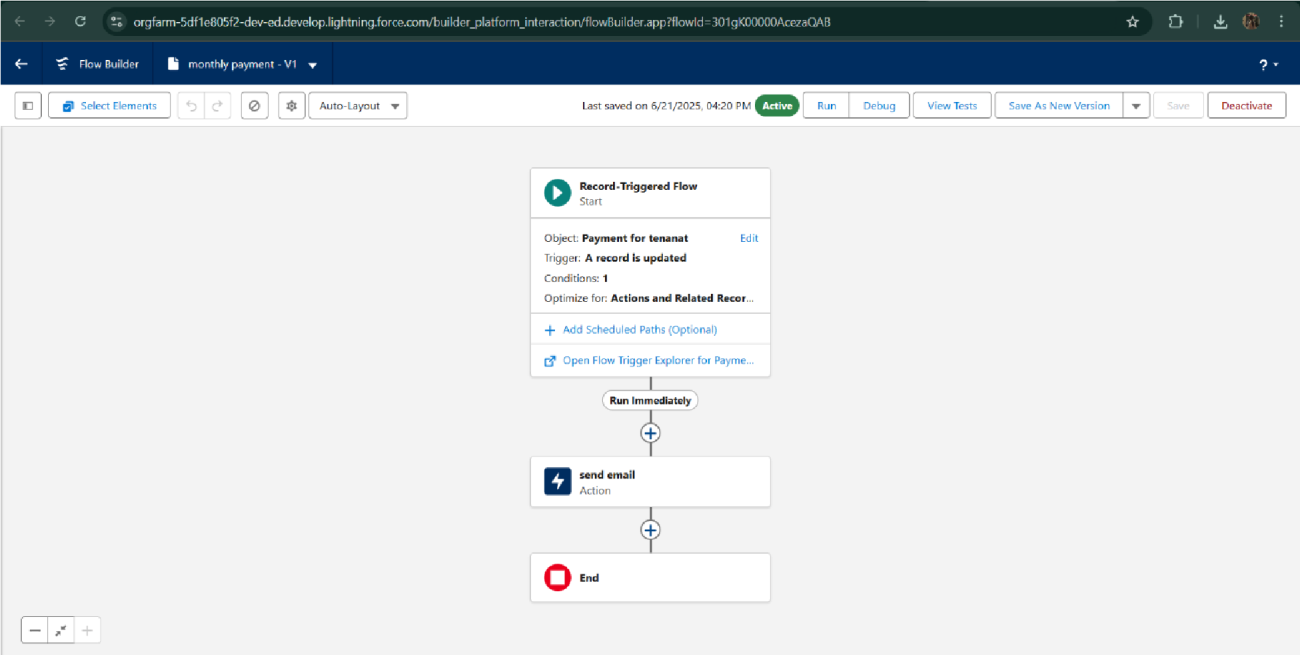
.Tabs for Property, Tenant, Lease, Payment



. Email alerts



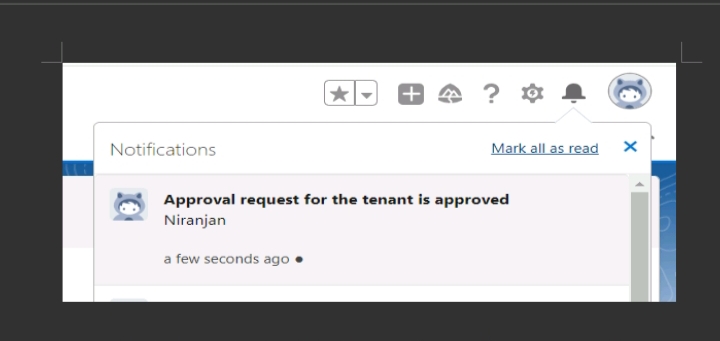
* Flow runs



Trigger message:



Approval process notifications:



.

# 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.