LEASE MANAGEMENT

College name: Sri Ramakrishna College Of Arts and Science for Women

College code: Bru28

Team id: NM2025TMID24283

TEAM MEMBERS:

Team Leader : HARINISRI R

Email id: srcw2326j118@srcw.ac.in

Team member 1 : BHAVANA DHARSHINI S

Email id: srcw2326j110@srcw.ac.in

Team member 2 : DHATCHAYINI P

Email id: srcw2326j159@srcw.ac.in

Team member 3 : ILAKKIYA R

Email id: srcw2326j121@srcw.ac.in

INTRODUCTION:

1.1 **PROJECT OVERVIEW:**

A lease management project involves creating a system or application to efficiently handle the processes related to leasing real estate properties, equipment, or other assets. The goal is to streamline and automate various tasks associate

ed with lease agreements, ensuring accurate record-keeping, compliance with regulations, and effective communication between parties involved.

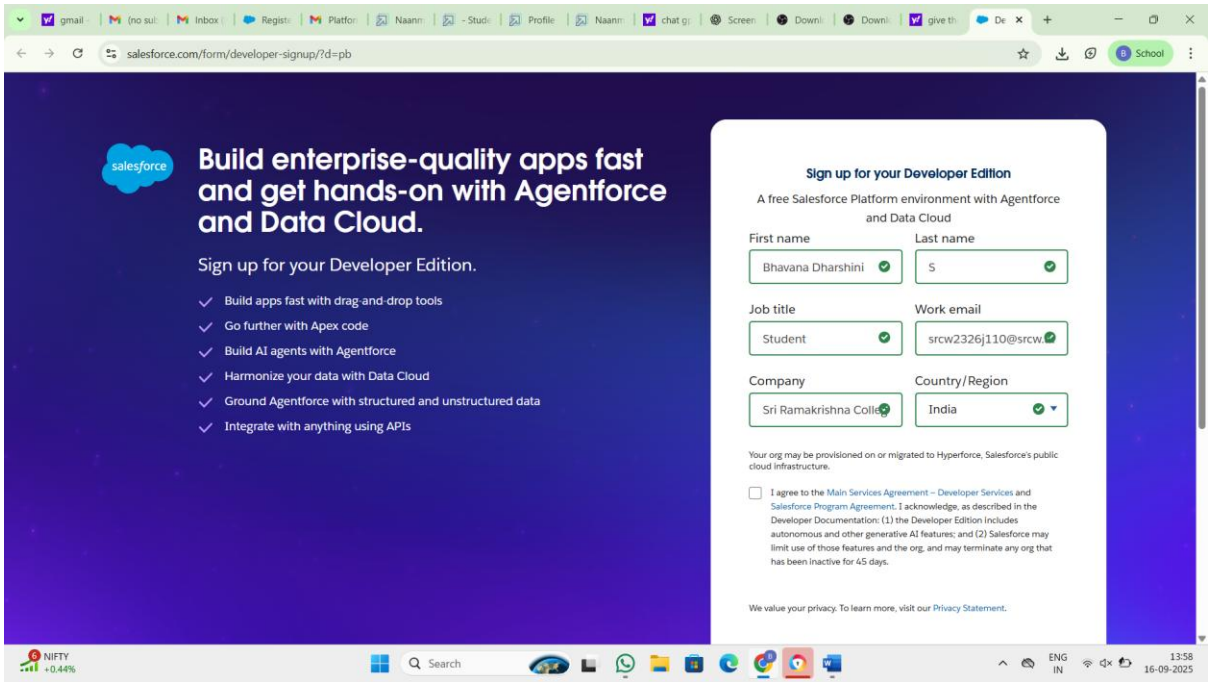
1.2 PURPOSE:

The purpose of the Lease Management project is to streamline and automate the end-to-end process of handling property leases. It helps in systematically managing property listings, creating accurate lease agreements, performing proper tenant screening, executing leases efficiently, and ensuring timely lease renewals. By following this structured workflow, the project aims to reduce manual errors, save time, and provide a clear and organized approach to managing properties, tenants, and payments within a single platform.

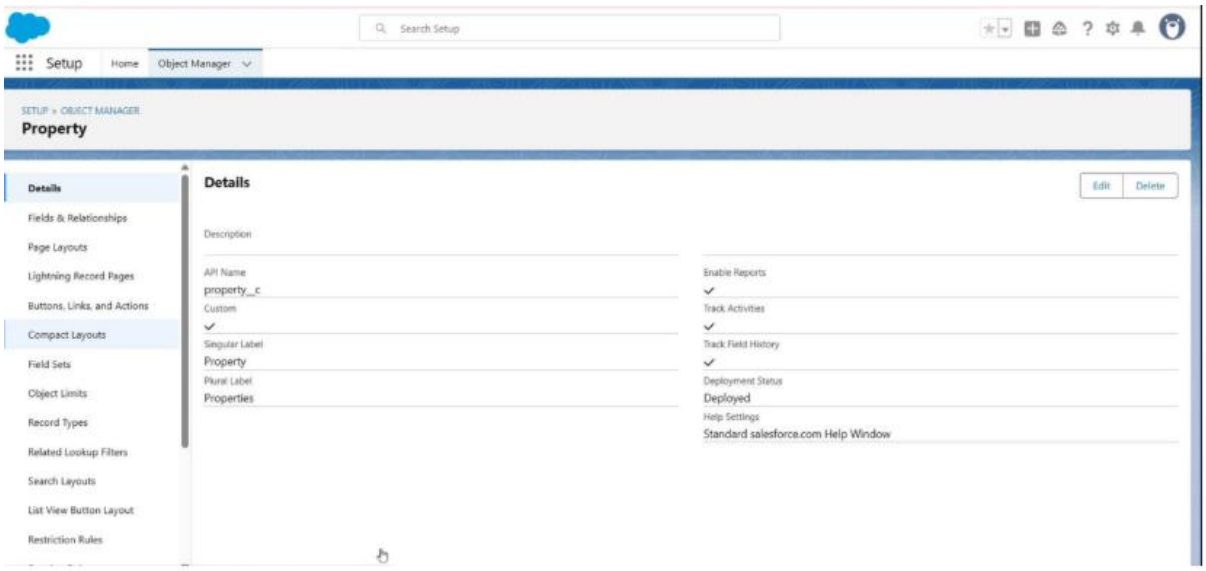
DEVELOPMENT PHASE:

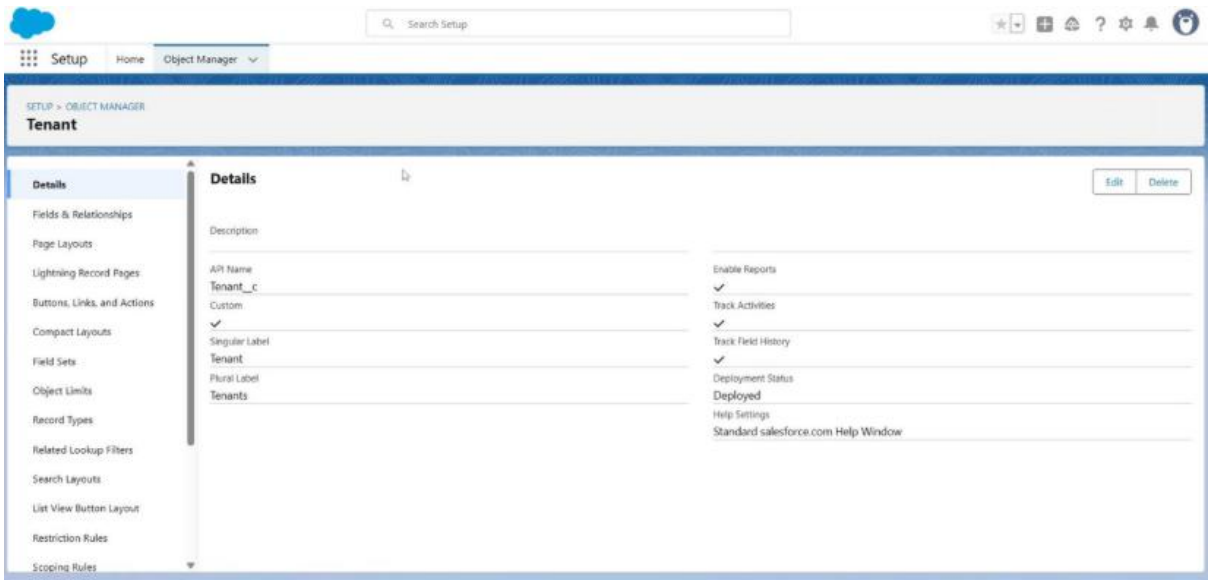
CREATING A DEVELOPER ACCOUNT:

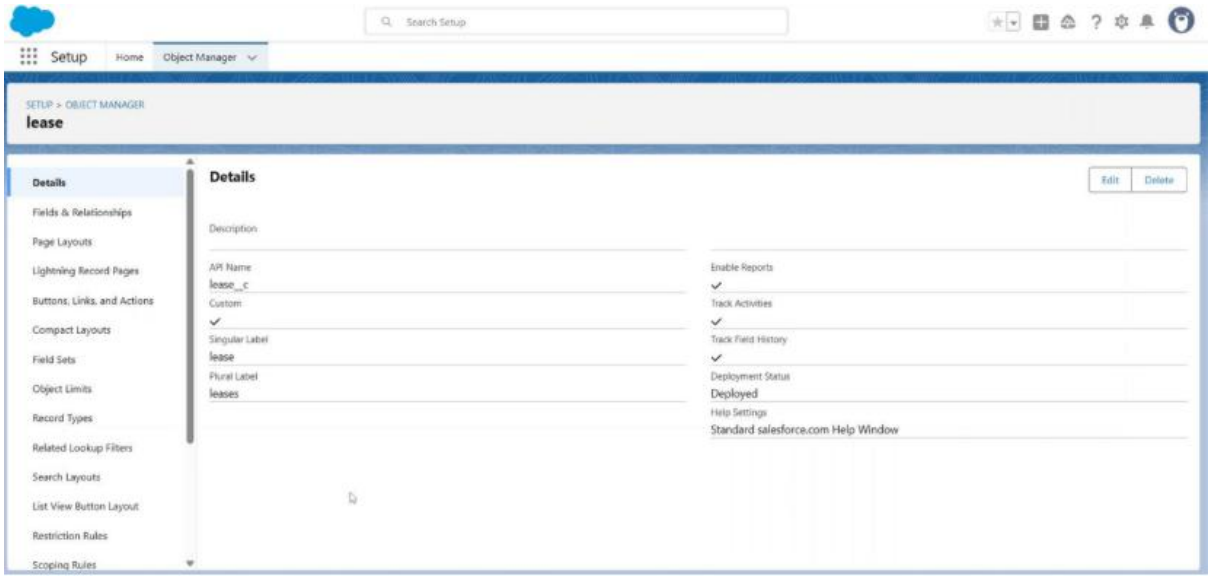
BY USING THE URL: https://www.salesforce.com/form/developer-signup/?d=pb

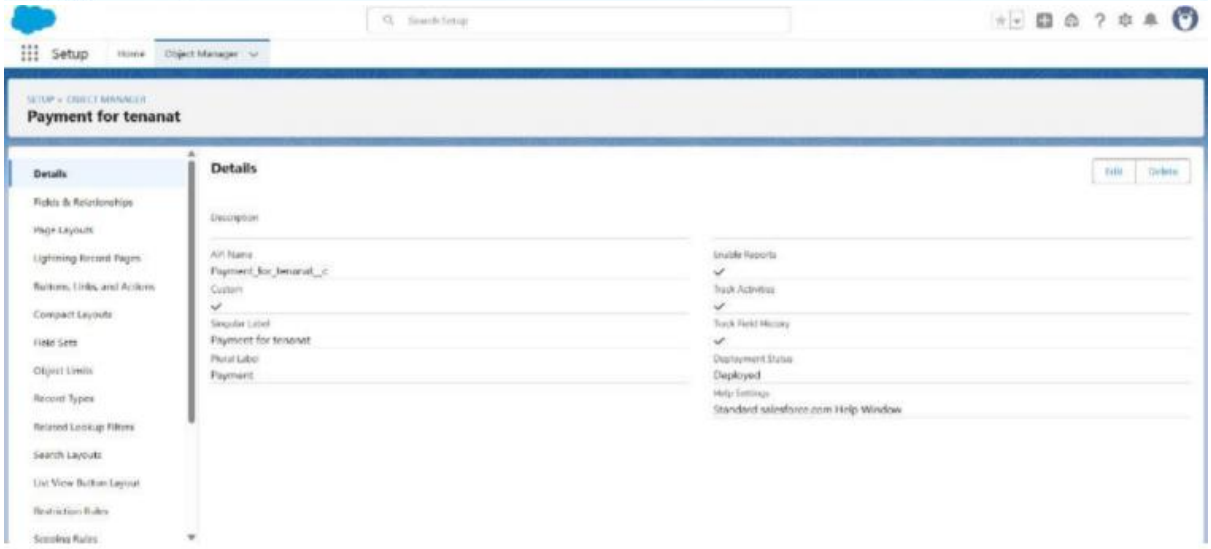


- Created objects : Property, Tenant,Lease,Payment

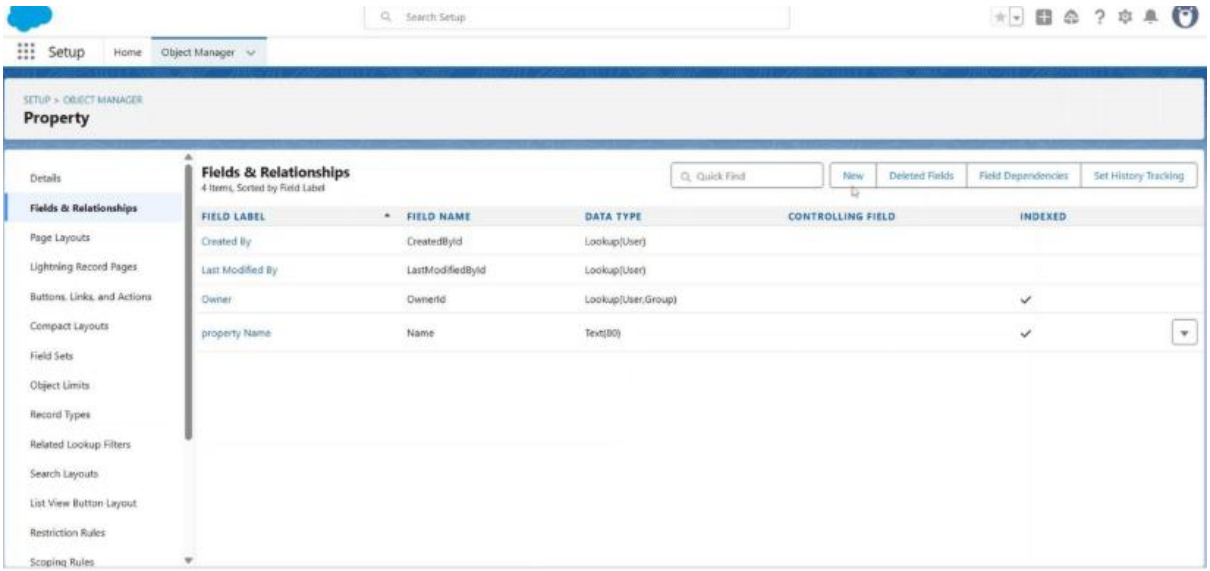


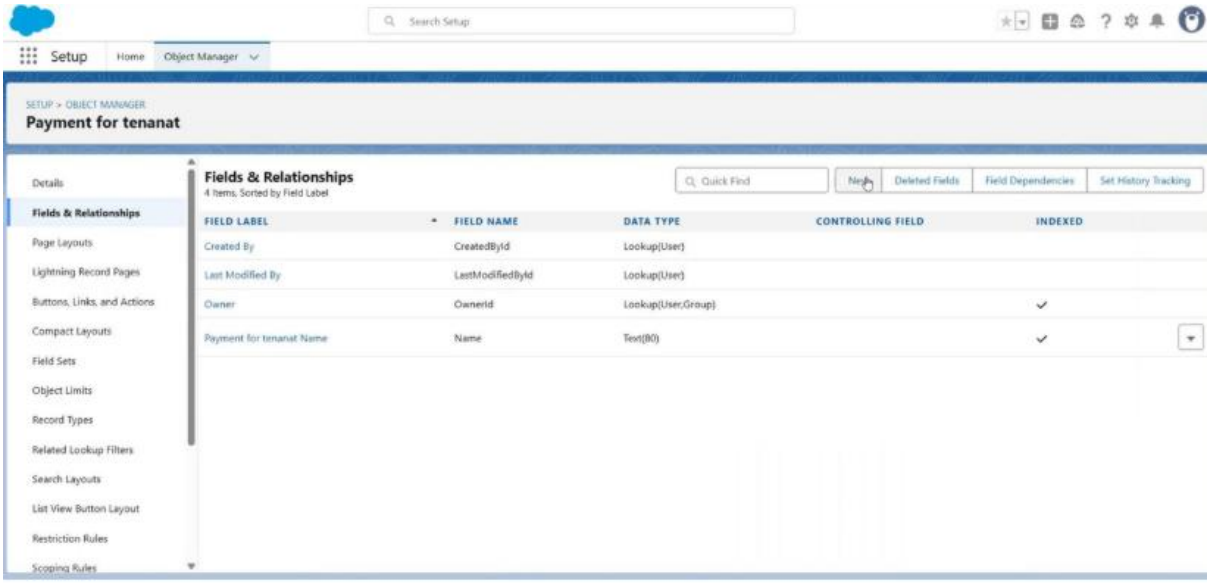

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Tenant
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Details
Edit
Delete
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

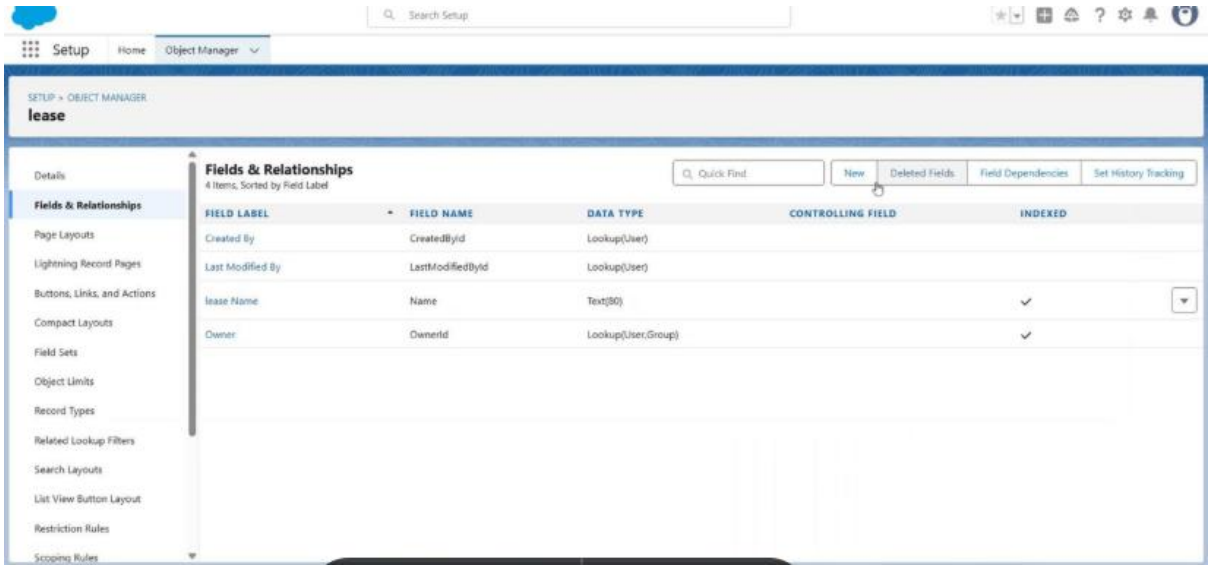

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
lease
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Details
Edit
Delete
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

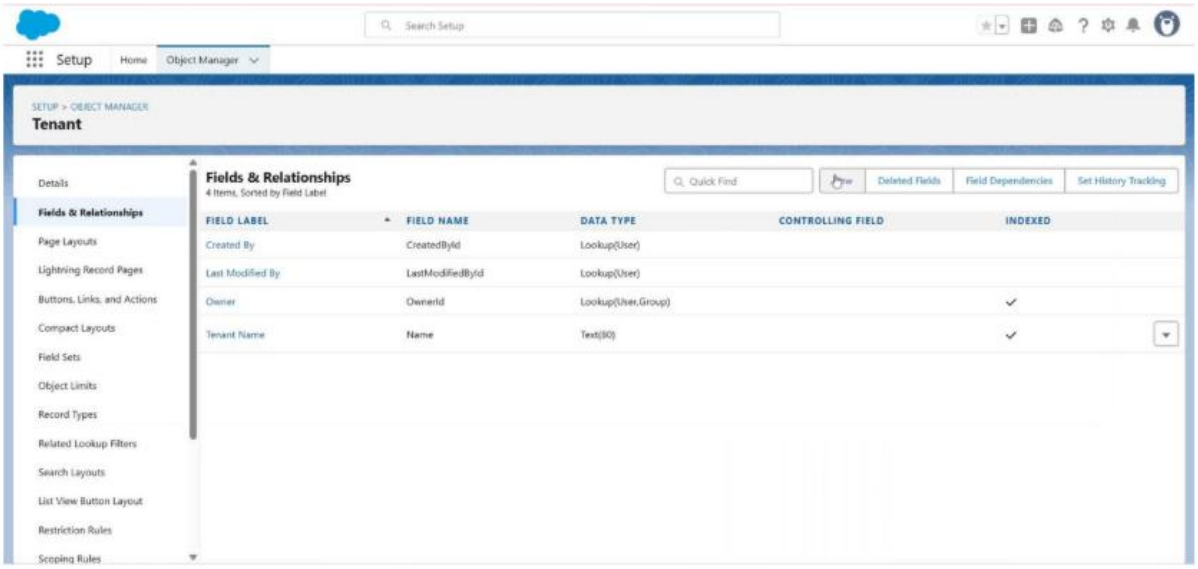

Setup
Payment for tenanat
Details
Description
Custom
Singular Label
Payment
Deployed
Standard salesforce.com Help Window

- Configured fields and relationships.

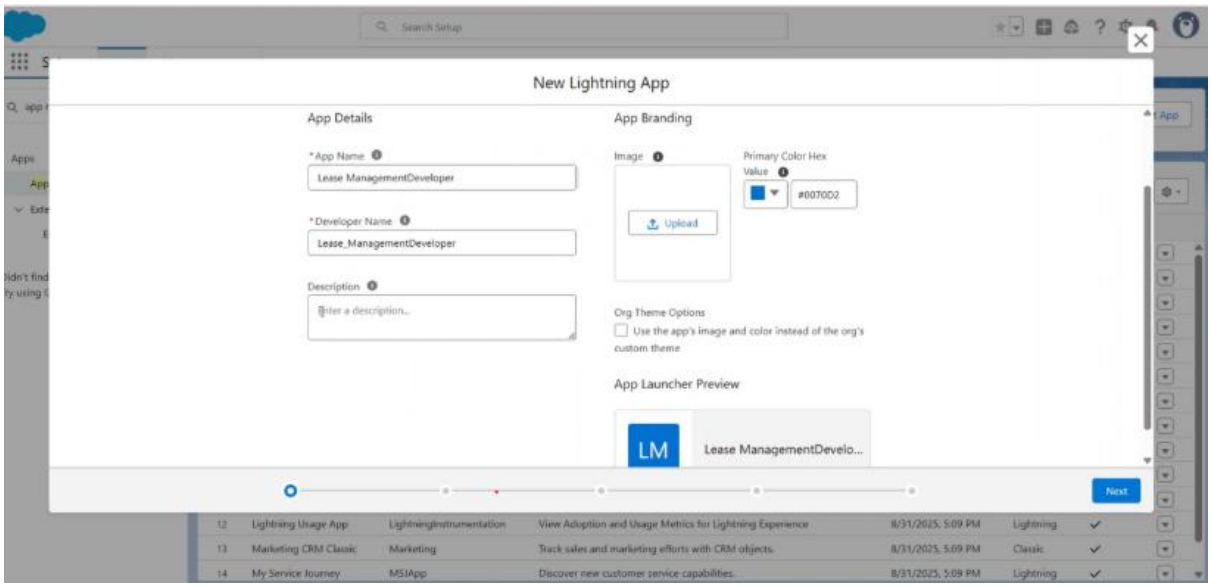


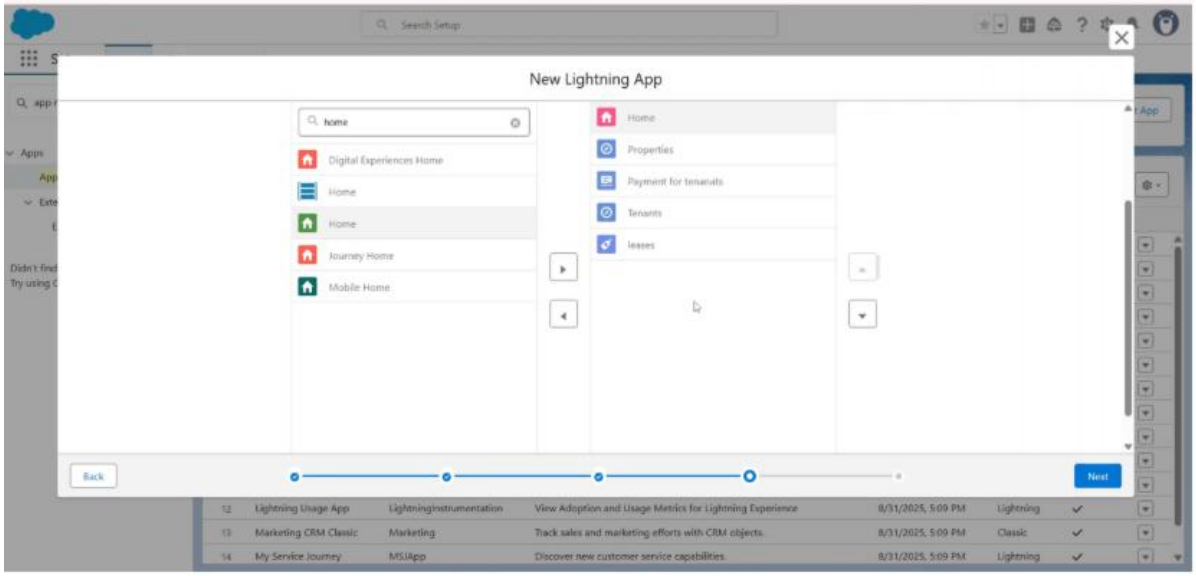


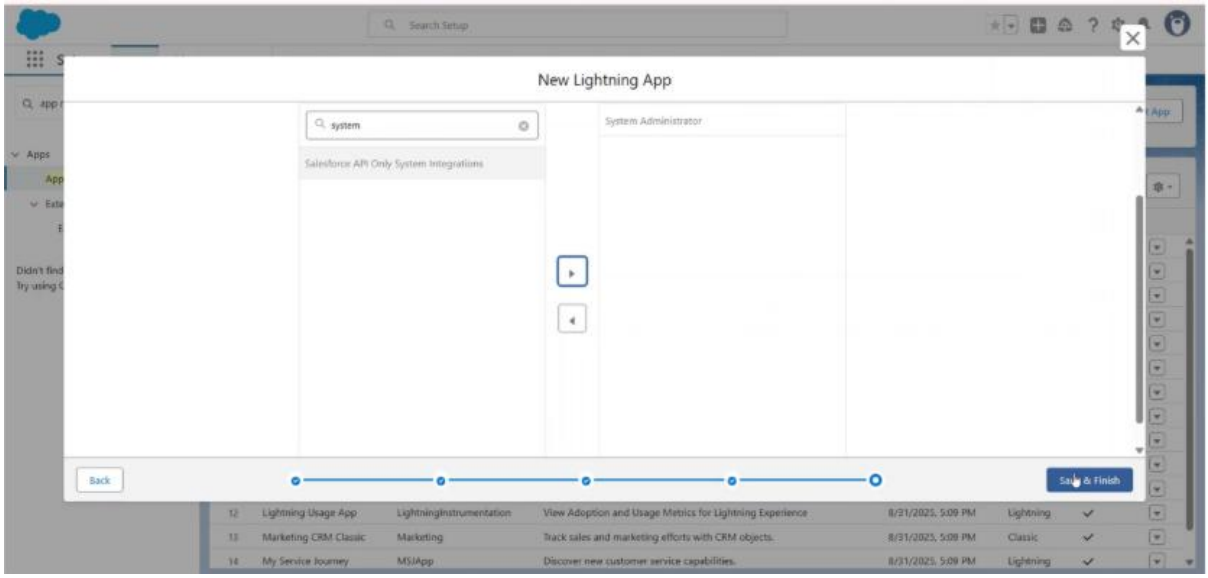


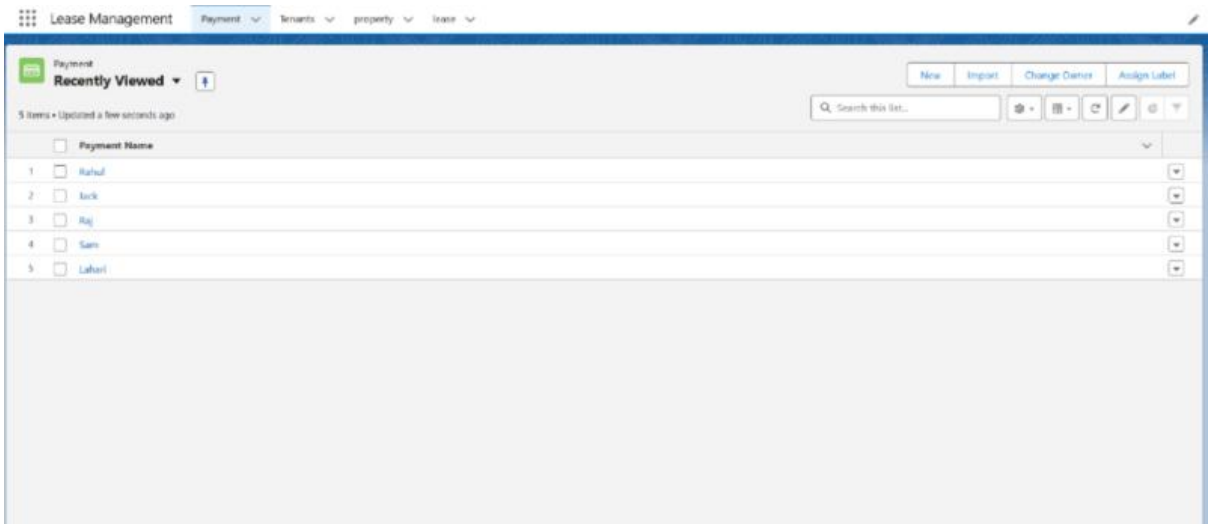


- Developed Lightning App with relevant tabs

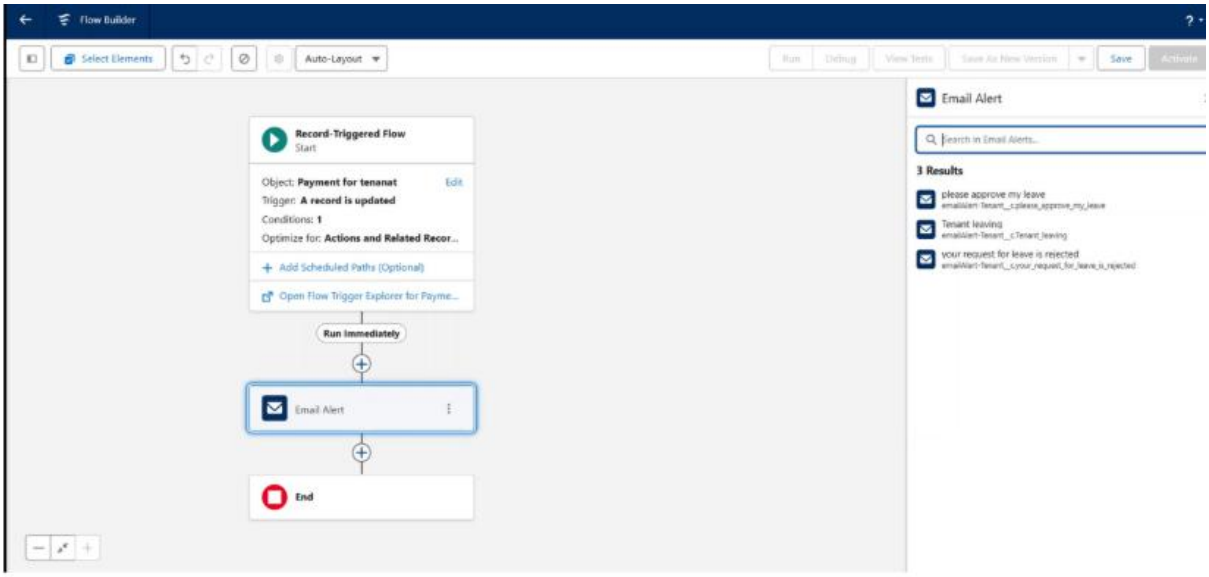



New Lightning App
home
Digital Experiences Home
Home
Home
Journey Home
Mobile Home
Home
Properties
Payment for tenanats
Tenants
leases
Back
Next

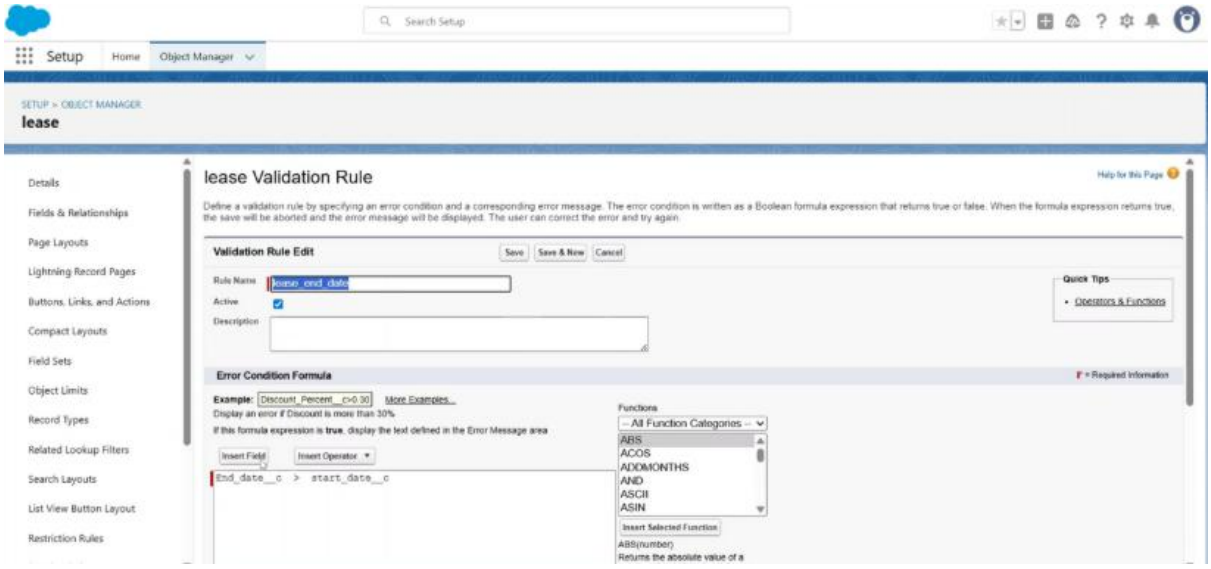

New Lightning App
system
Salesforce API Only System Integrations
System Administrator
Back
Save & Finish

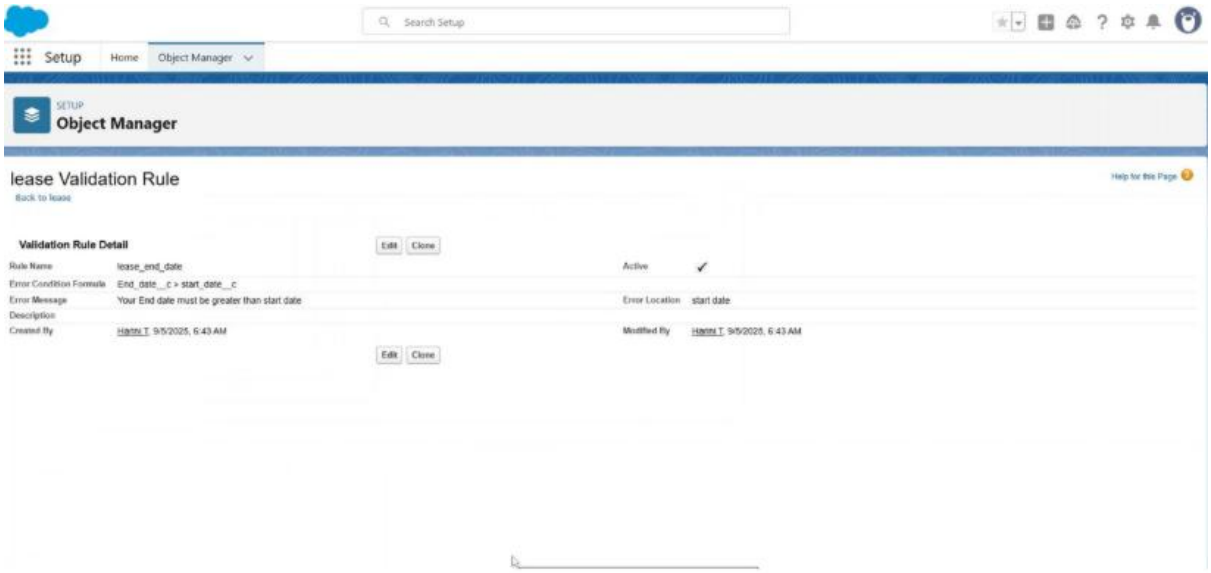

Lease Management
Payment
Tenants
property
lease
Recently Viewed
New
Import
Change Owner
Assign Label
Payment Name
Rahul
Jack
Raj
Sam
Lahari

- Implemented Flows for monthly rent and payment success

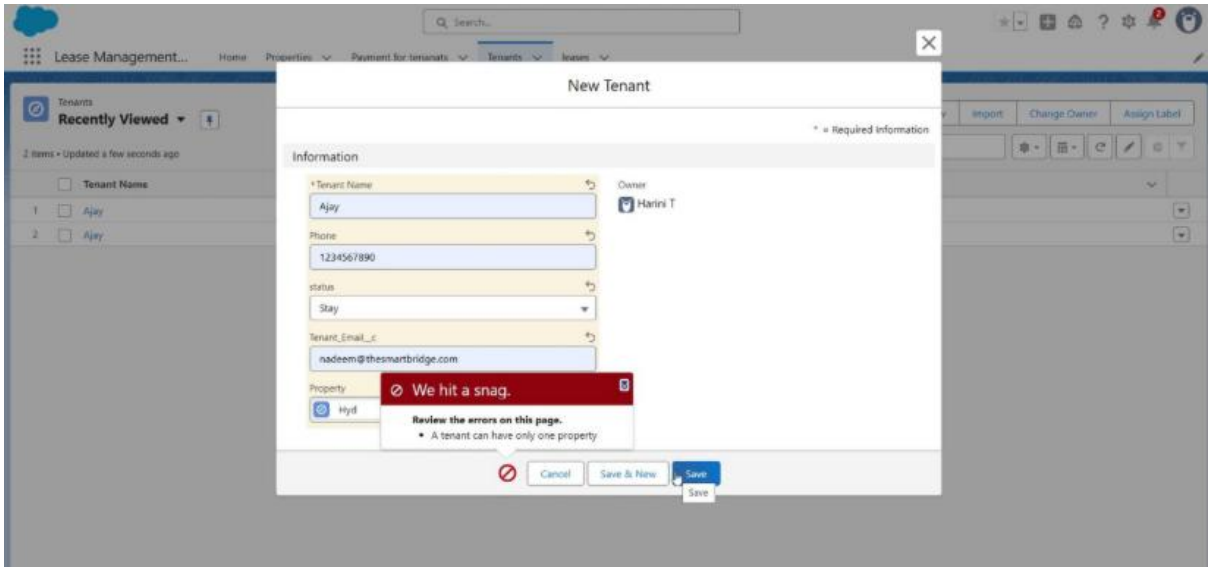


- To create a validation rule to a Lease Object





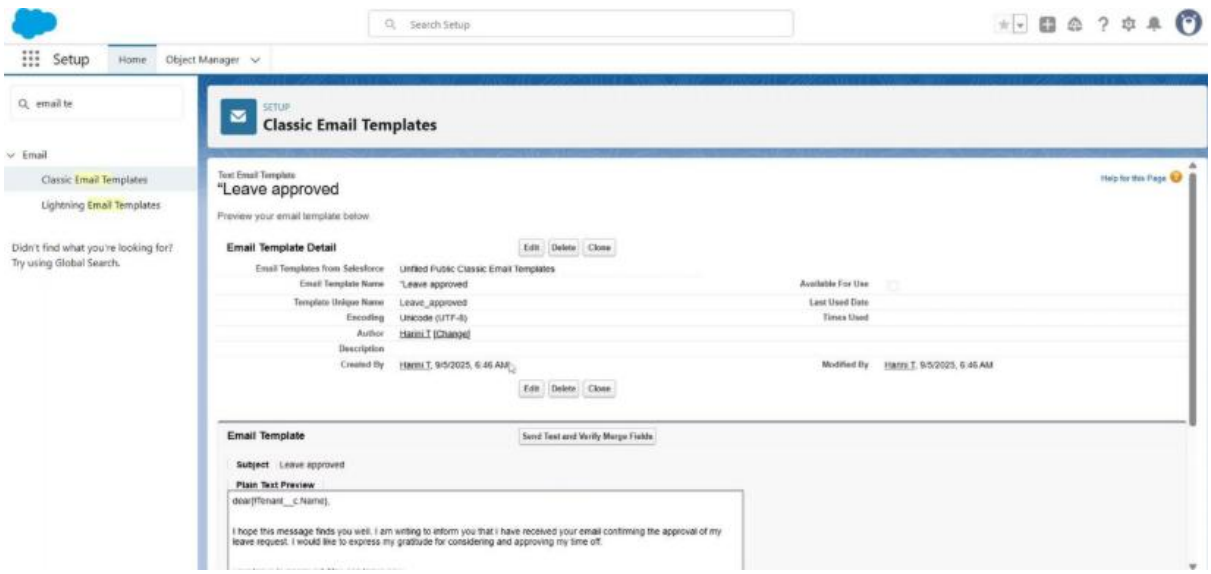
● Added Apex trigger to restrict multiple tenants per property

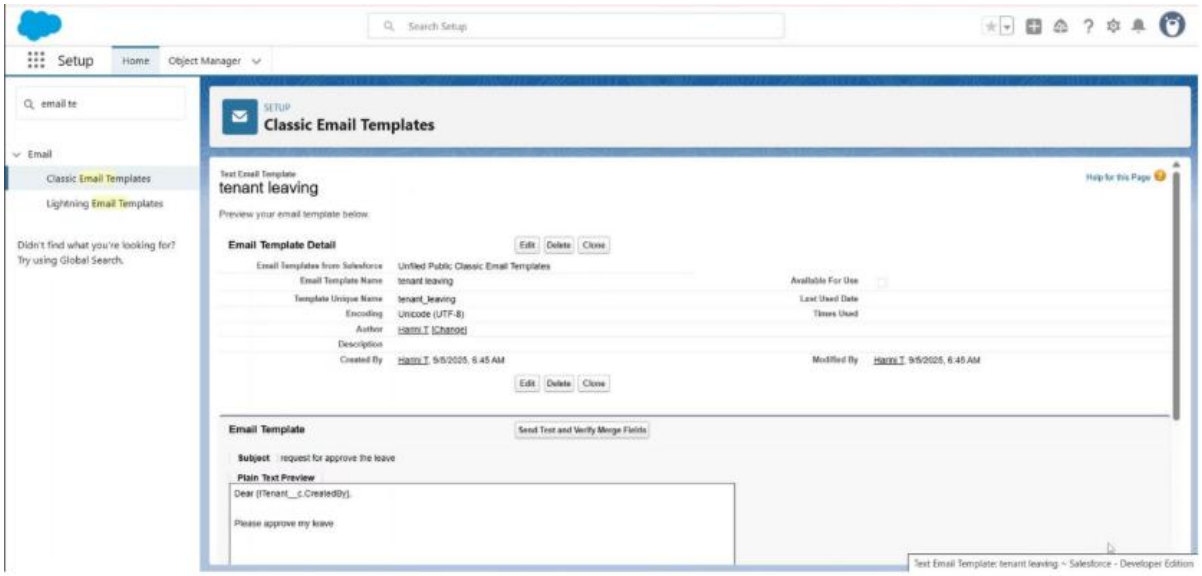


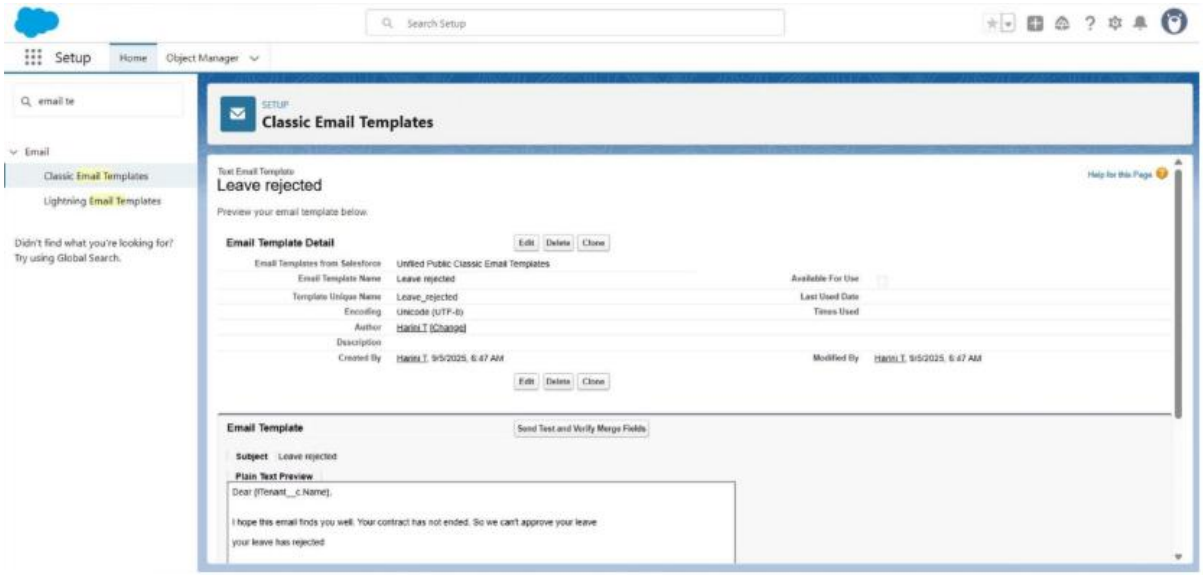
● Scheduled monthly reminder emails using Apex class

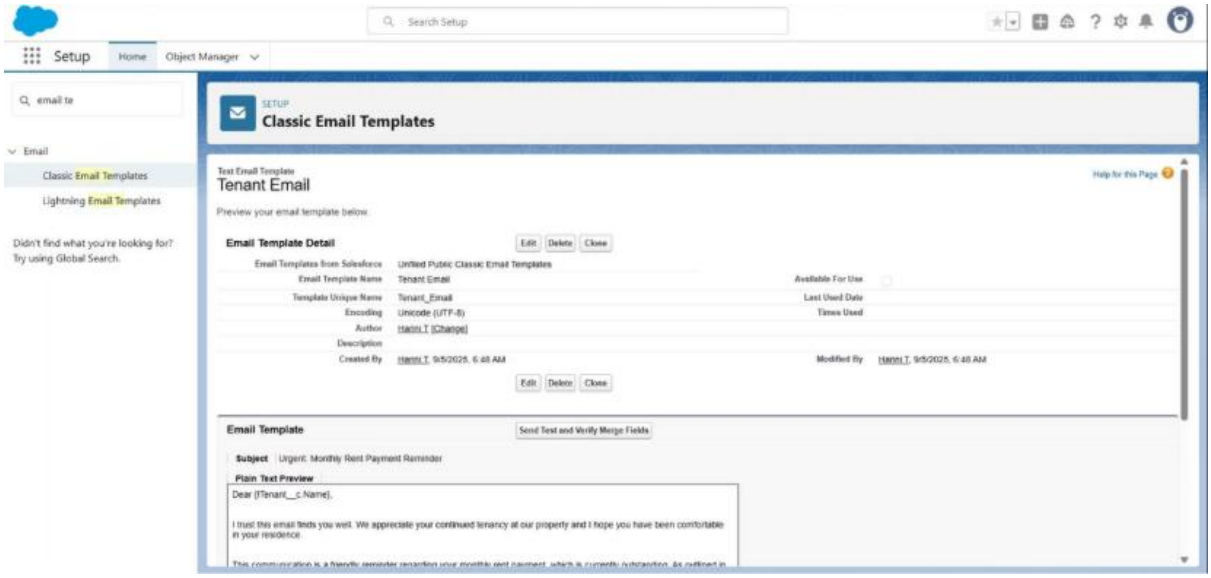
```
global class MonthlyEmailScheduler implements Schedulable {

    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

            String emailSubject = 'Reminder: Monthly Rent Payment Due';

            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

            email.setToAddresses(new String[]{recipientEmail});

            email.setSubject(emailSubject);

            email.setPlainTextBody(emailContent);
```

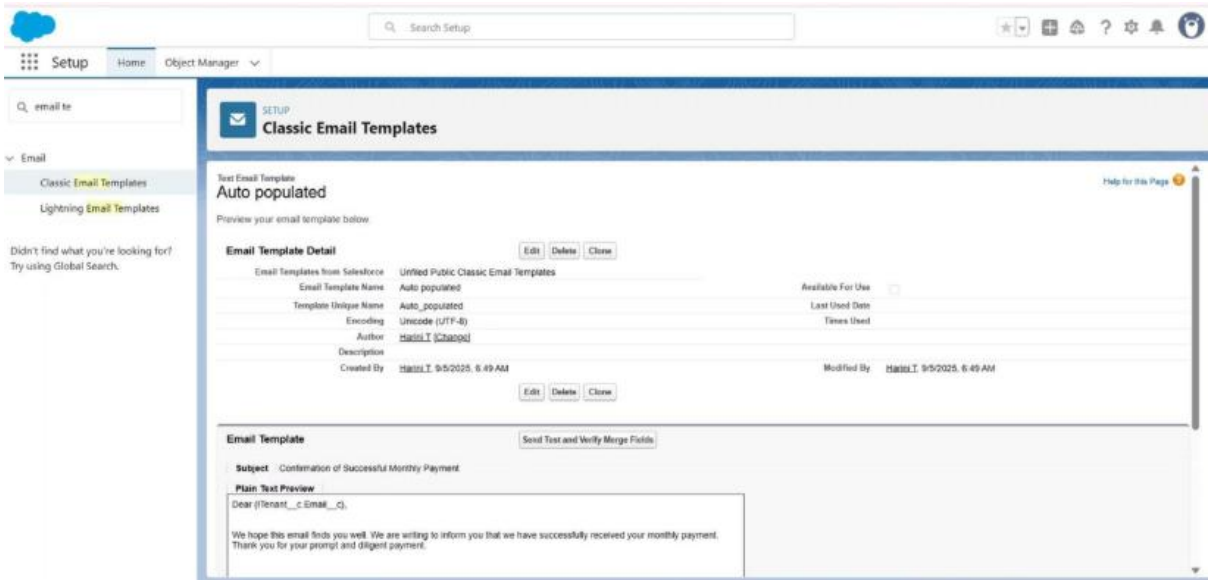
● Built and tested email templates for leave request, approval, rejection, payment, and reminders




Search Setup
Setup
Home
Object Manager
email te
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
tenant leaving
Help for this Page
Preview your email template below.
Email Template Detail
Edit
Delete
Clone
Email Templates from Salesforce
Unfiled Public Classic Email Templates
Email Template Name
tenant leaving
Available For Use
Template Unique Name
tenant_leaving
Last Used Date
Encoding
Unicode (UTF-8)
Times Used
Author
Hanni T [Change]
Description
Created By
Hanni T, 9/5/2025, 6:45 AM
Modified By
Hanni T, 9/5/2025, 6:45 AM
Email Template
Send Test and Verify Merge Fields
Subject
request for approve the leave
Plain Text Preview
Dear {!Tenant__c.CreatedBy},
Please approve my leave
Test Email Template: tenant leaving ~ Salesforce - Developer Edition

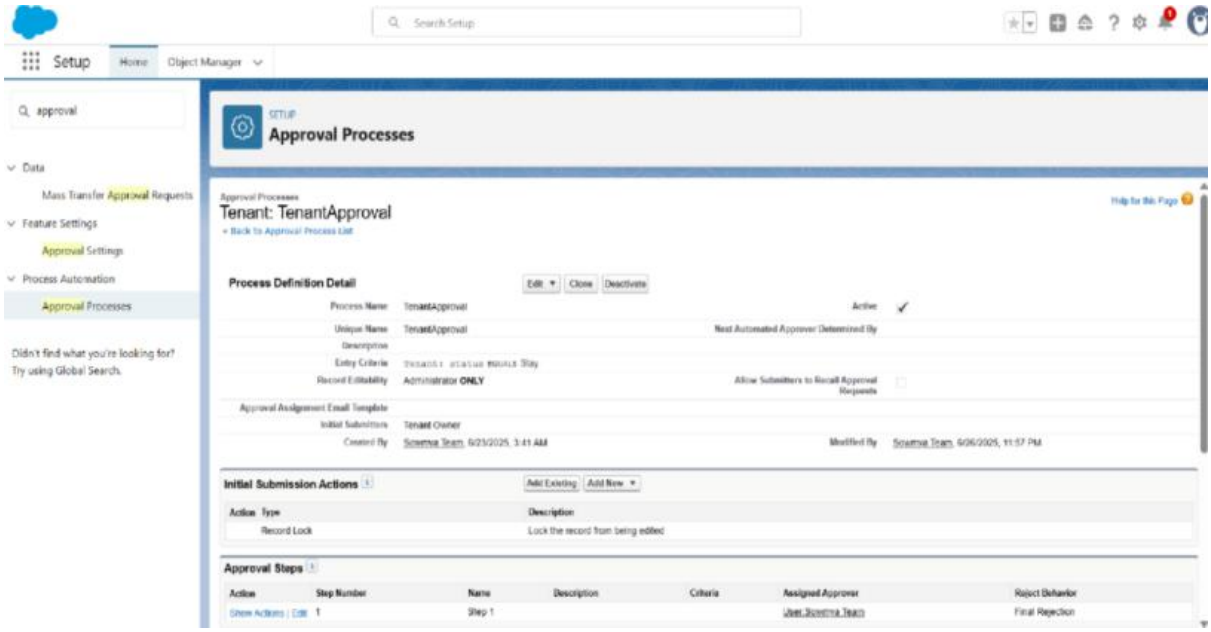

Search Setup
Setup
Home
Object Manager
email te
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
Leave rejected
Help for this Page
Preview your email template below.
Email Template Detail
Edit
Delete
Clone
Email Templates from Salesforce
Unfiled Public Classic Email Templates
Email Template Name
Leave rejected
Available For Use
Template Unique Name
Leave_rejected
Last Used Date
Encoding
Unicode (UTF-8)
Times Used
Author
Hanni T [Change]
Description
Created By
Hanni T, 9/5/2025, 6:47 AM
Modified By
Hanni T, 9/5/2025, 6:47 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


Search Setup
Setup
Home
Object Manager
email te
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
Tenant Email
Help for this Page
Preview your email template below.
Email Template Detail
Edit
Delete
Clone
Email Templates from Salesforce
Unfiled Public Classic Email Templates
Email Template Name
Tenant Email
Available For Use
Template Unique Name
Tenant_Email
Last Used Date
Encoding
Unicode (UTF-8)
Times Used
Author
Hanni T [Change]
Description
Created By
Hanni T, 9/5/2025, 6:48 AM
Modified By
Hanni T, 9/5/2025, 6:48 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.

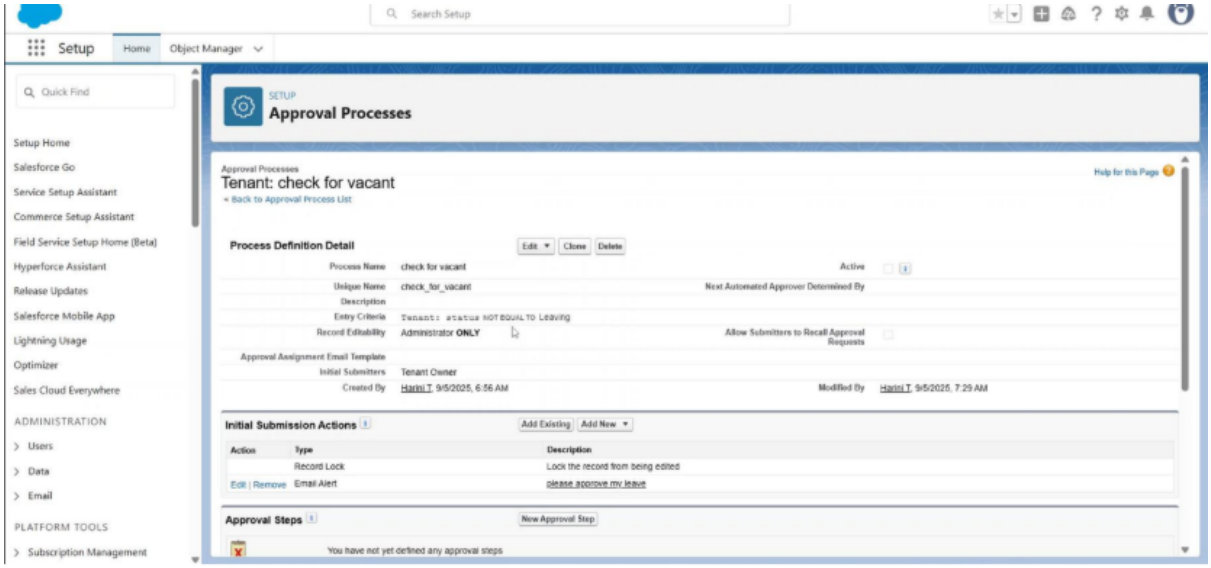


- Approval Process creation

For Tenant Leaving:

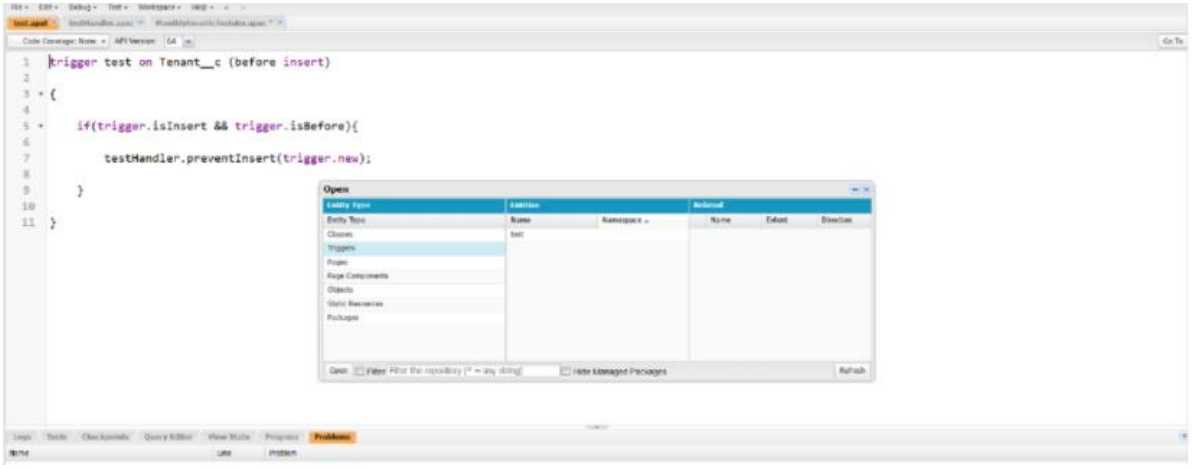


For Check for Vacant:

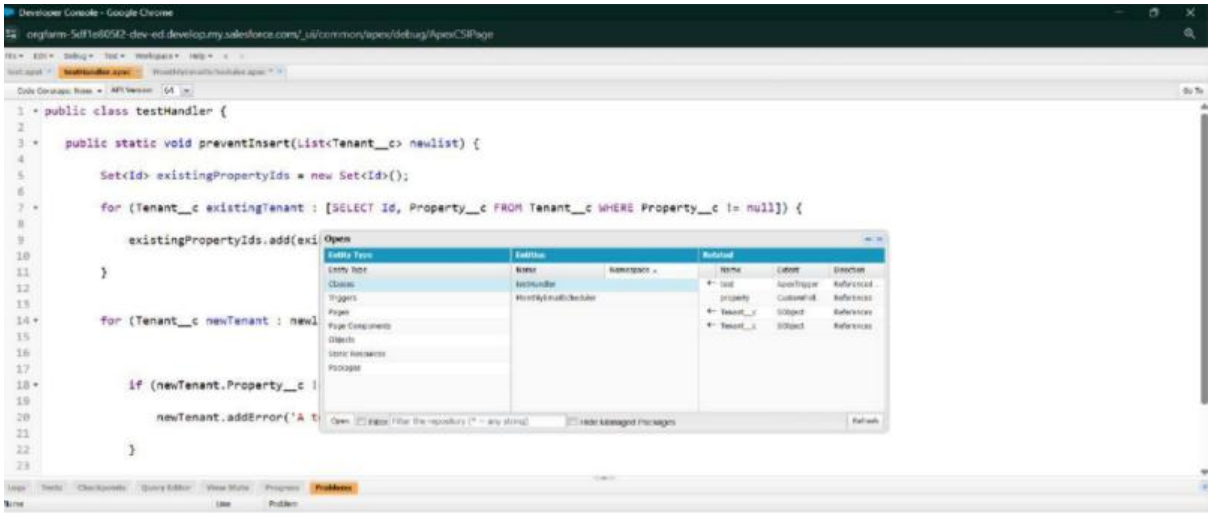


● Apex Trigger

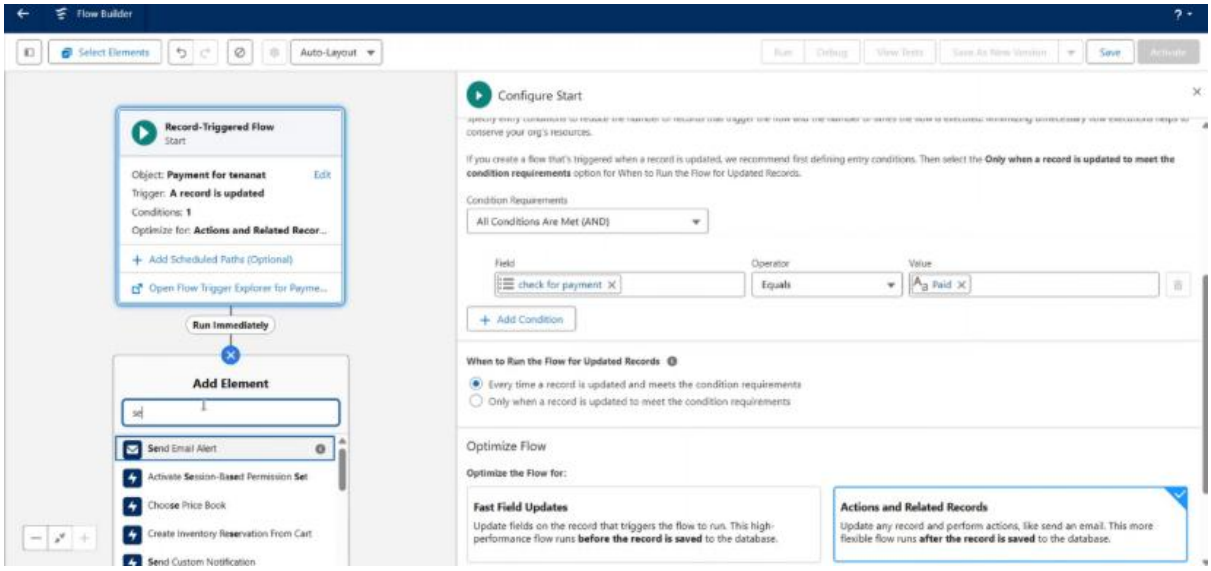
Create an Apex Trigger

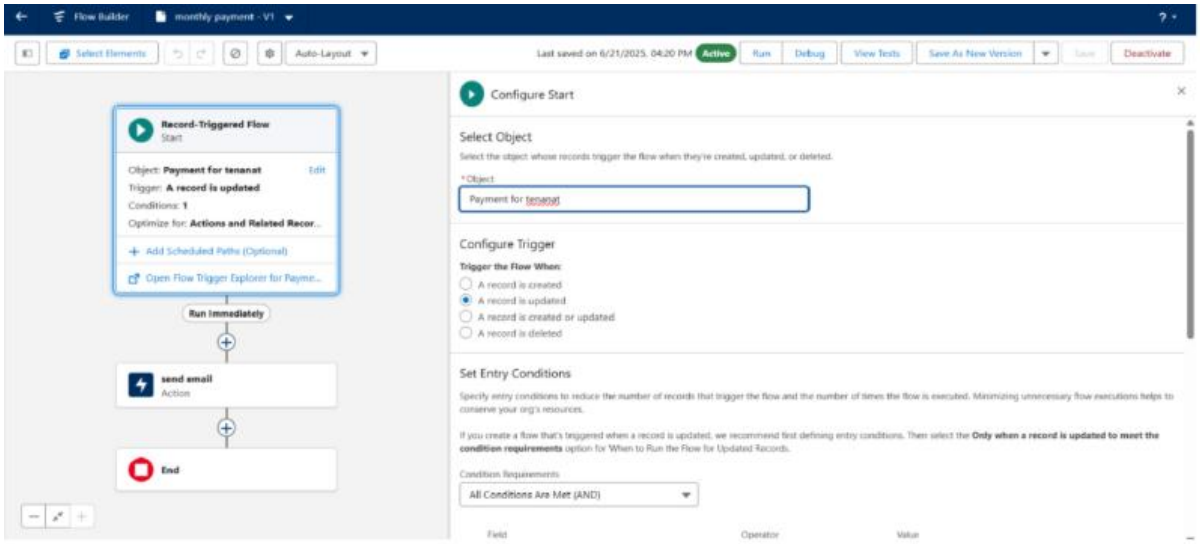


Create an Apex Handler class:



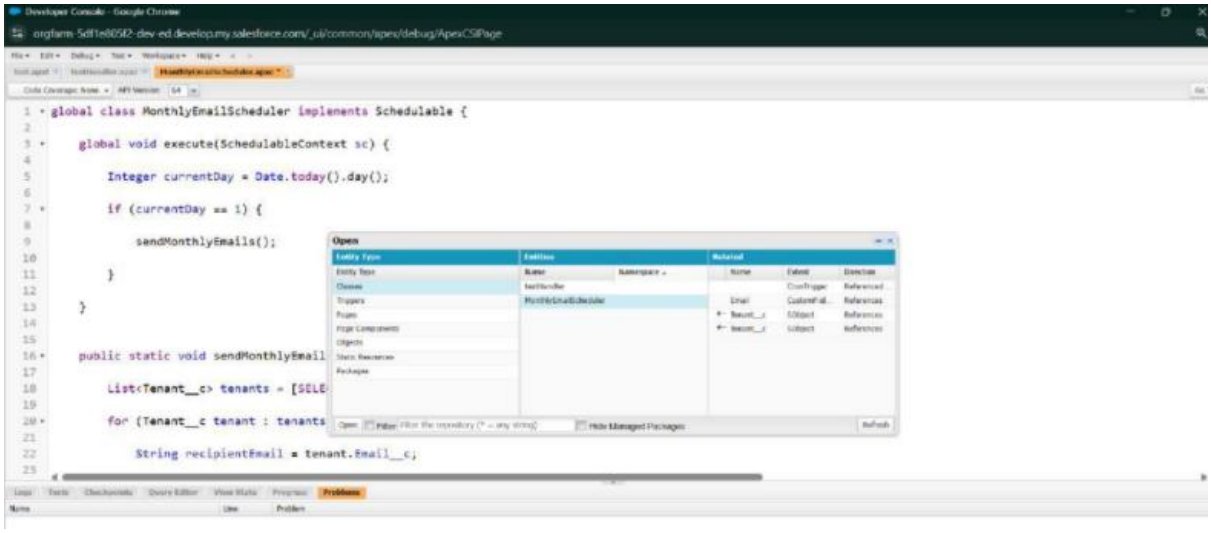
● FLOWS

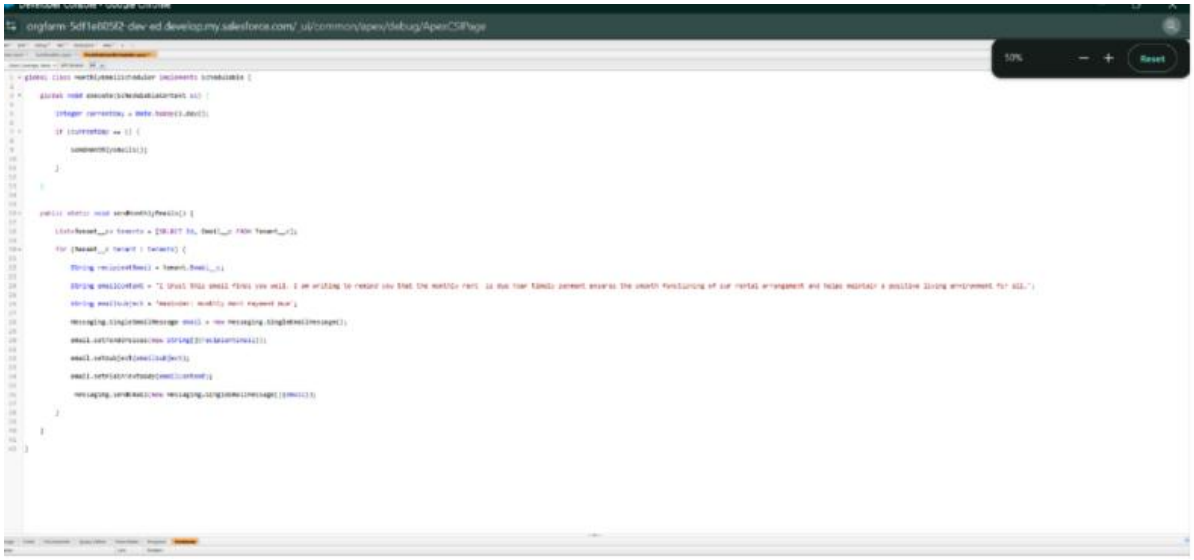




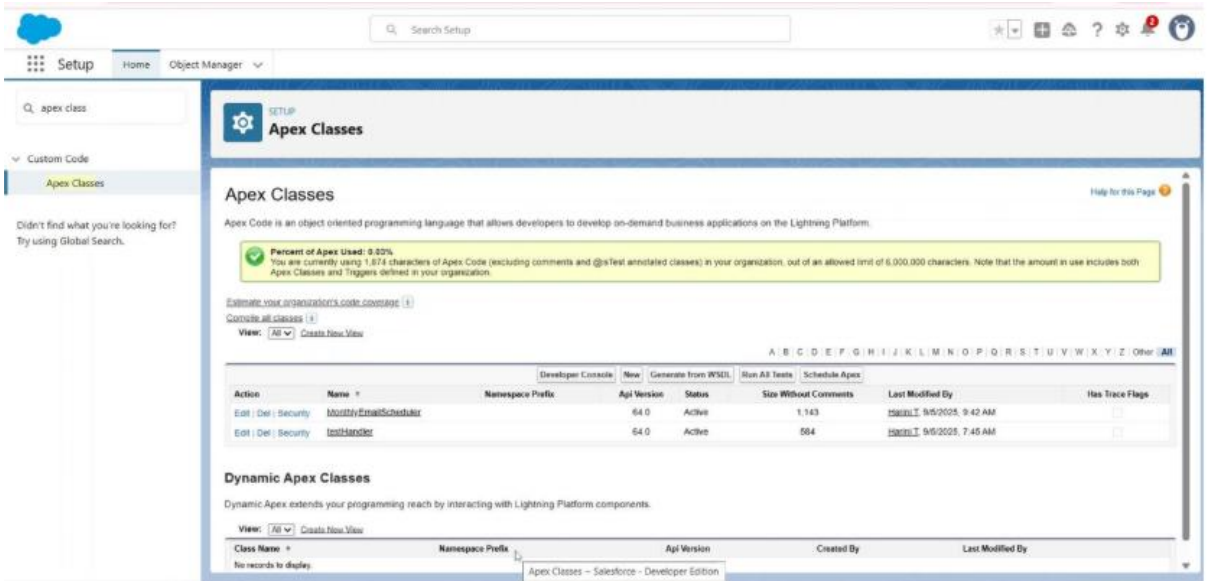
● Schedule class:

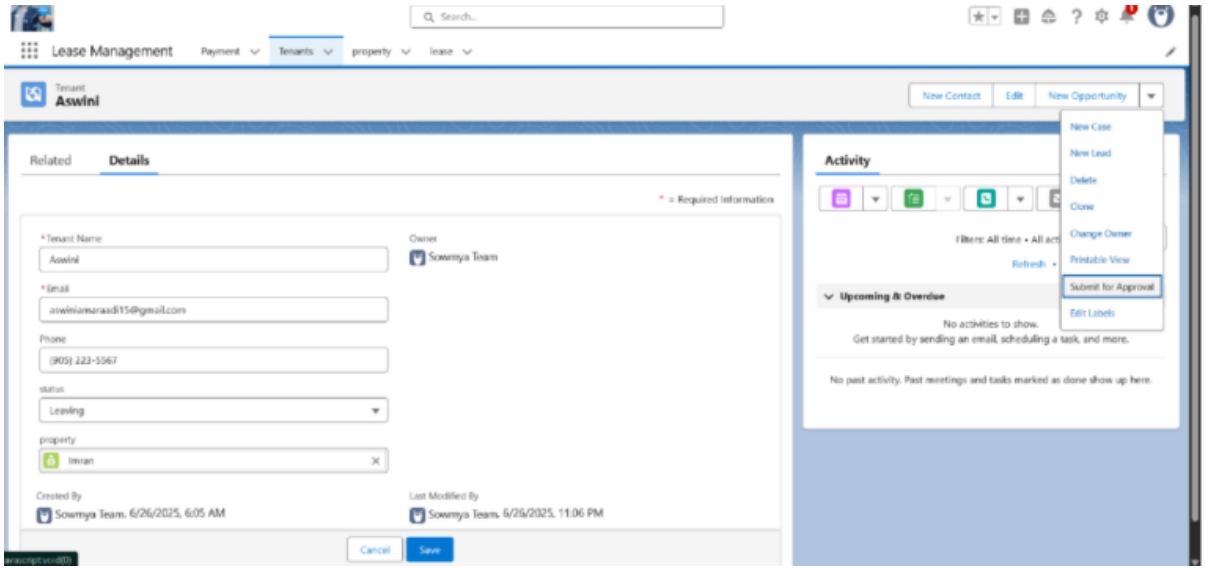
Create an Apex Class

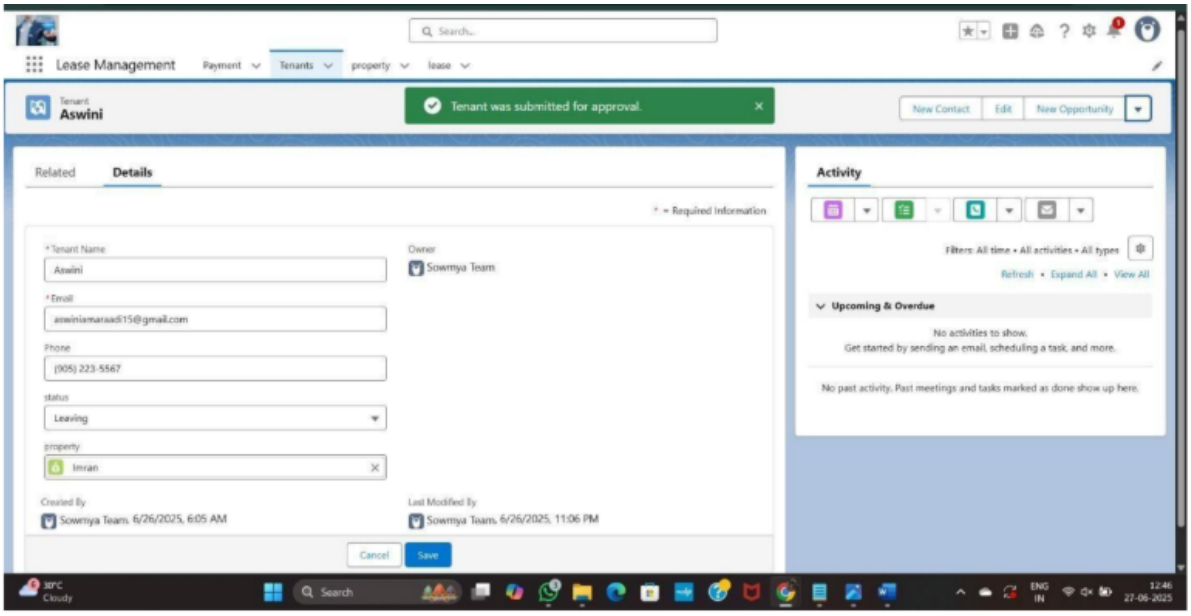


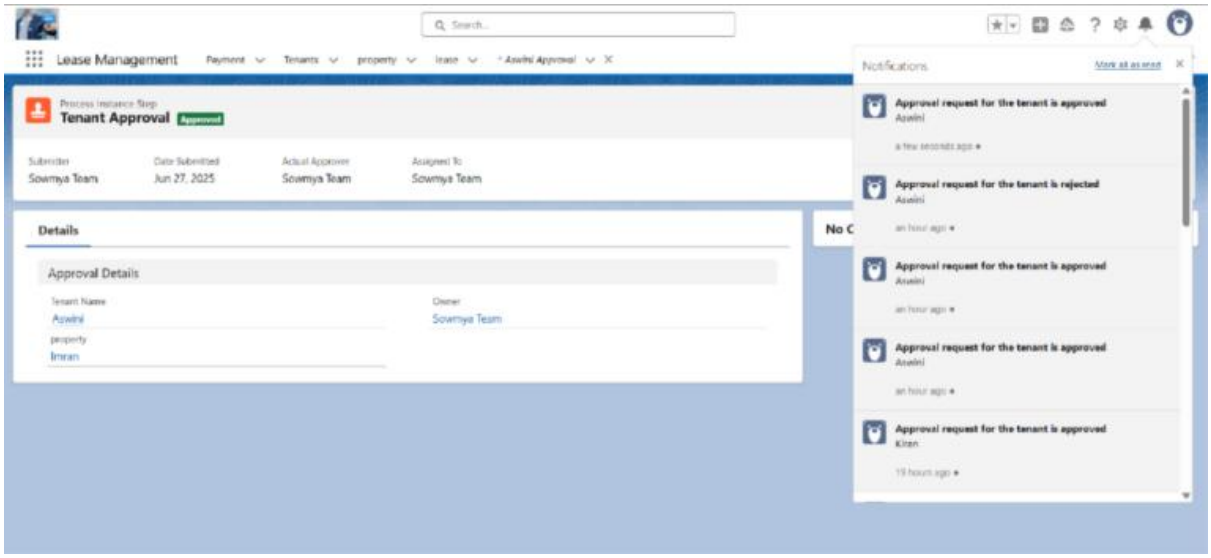


Schedule Apex class


Setup
Home
Object Manager
apex class
Custom Code
Apex Classes
Didn't find what you're looking for?
Try using Global Search.
SETUP
Apex Classes
Apex Classes
Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.
Percent of Apex Used: 0.03%
You are currently using 1,874 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.
Estimate your organization's code coverage
Compile all classes
View: All
Create New View
Developer Console
New
Generate from WSDL
Run All Tests
Schedule Apex
Action
Name
Namespace Prefix
Api Version
Status
Size Without Comments
Last Modified By
Has Trace Flags
Edit | Del | Security
MonthlyEmailScheduler
64.0
Active
1,143
Harini T, 9/5/2025, 9:42 AM
Edit | Del | Security
testHandler
64.0
Active
584
Harini T, 9/5/2025, 7:45 AM
Dynamic Apex Classes
Dynamic Apex extends your programming reach by interacting with Lightning Platform components.
View: All
Create New View
Class Name
Namespace Prefix
Api Version
Created By
Last Modified By
No records to display.
Apex Classes ~ Salesforce - Developer Edition


Lease Management
Payment
Tenants
property
lease
Tenant
Aswini
New Contact
Edit
New Opportunity
New Case
New Lead
Delete
Clone
Change Owner
Printable View
Submit for Approval
Edit Labels
Related
Details
* = Required Information
*Tenant Name
Aswini
Owner
Sowmya Team
*Email
aswiniamaradi15@gmail.com
Phone
(905) 223-5567
status
Leaving
property
Imran
Created By
Sowmya Team, 6/26/2025, 6:05 AM
Last Modified By
Sowmya Team, 6/26/2025, 11:06 PM
Cancel
Save
Activity
Upcoming & Overdue
No activities to show.
Get started by sending an email, scheduling a task, and more.
No past activity. Past meetings and tasks marked as done show up here.

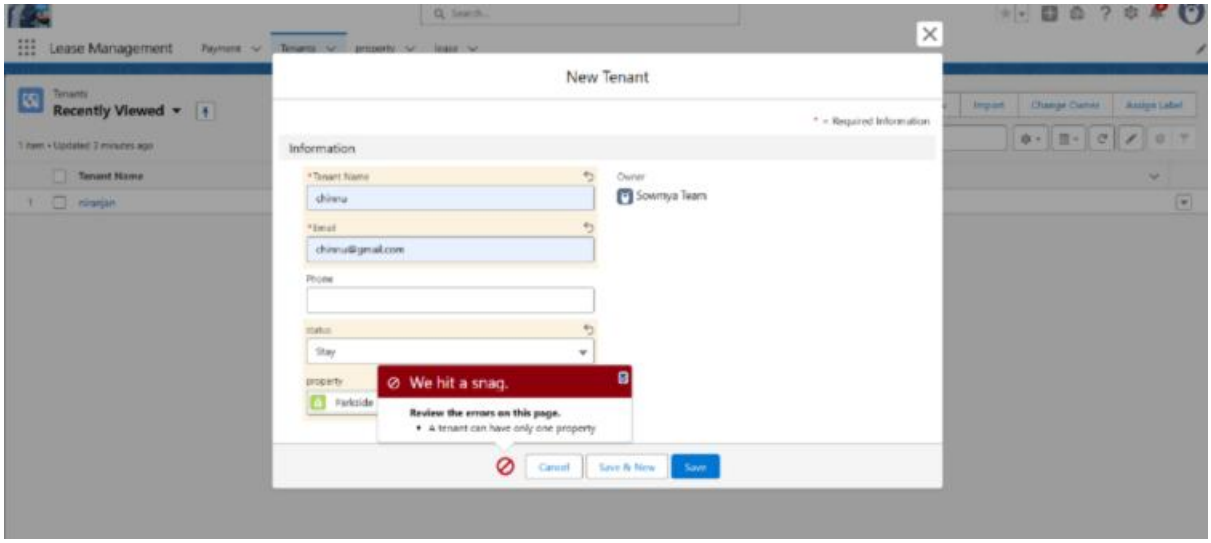


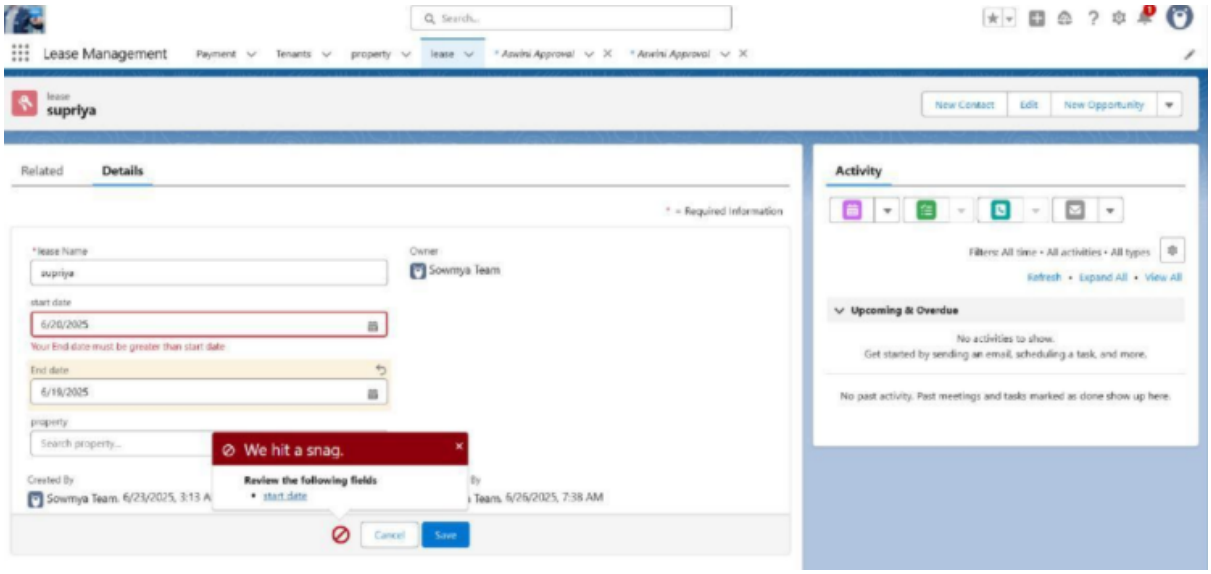


FUNCTIONAL AND PERFORMANCE TESTING

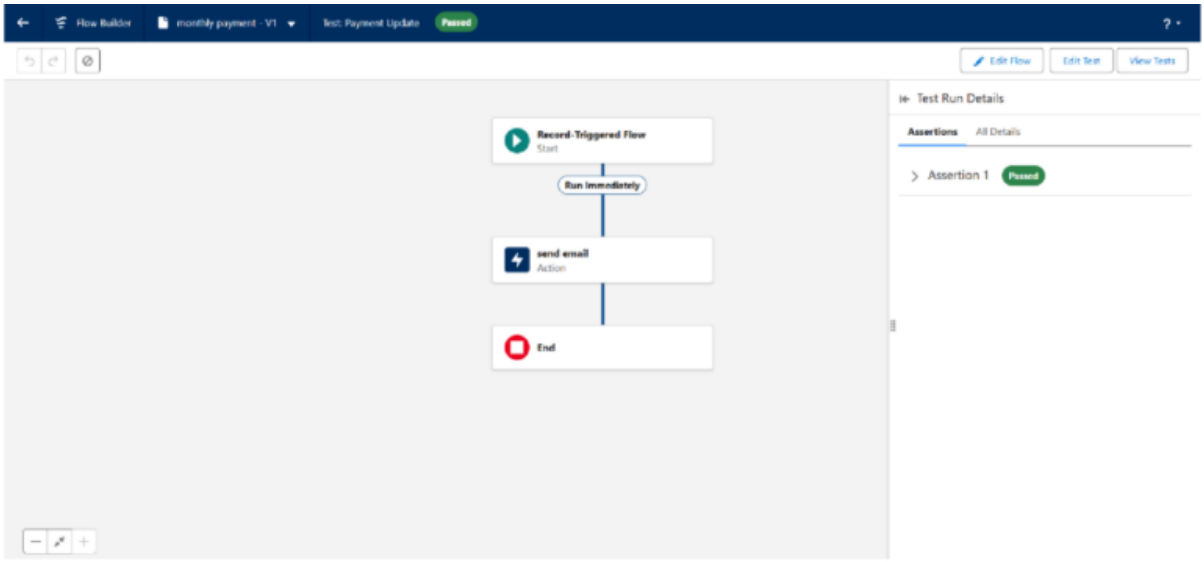
Performance Testing

* Trigger validation by entering duplicate tenant-property records

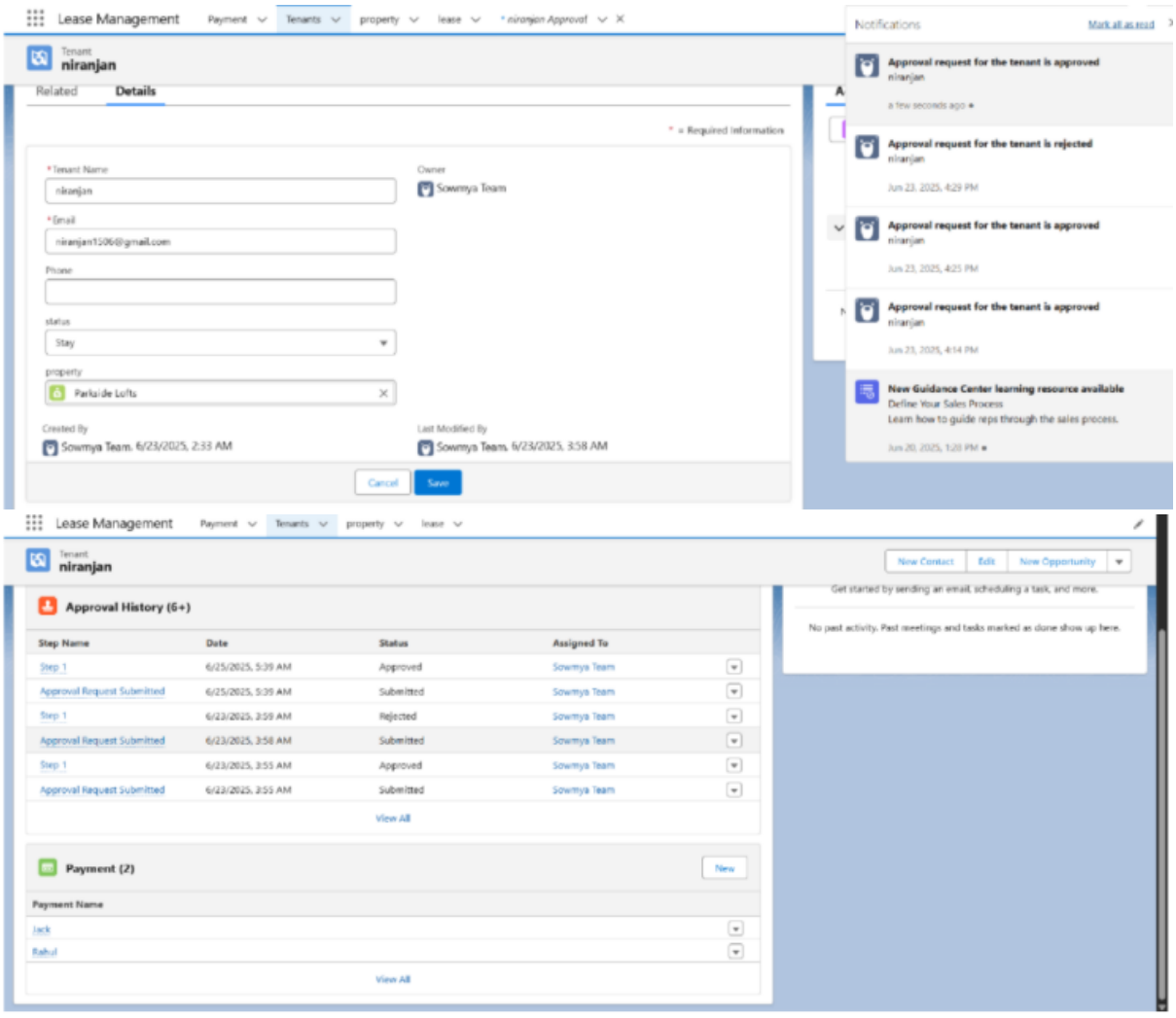




- Test flows on payment update



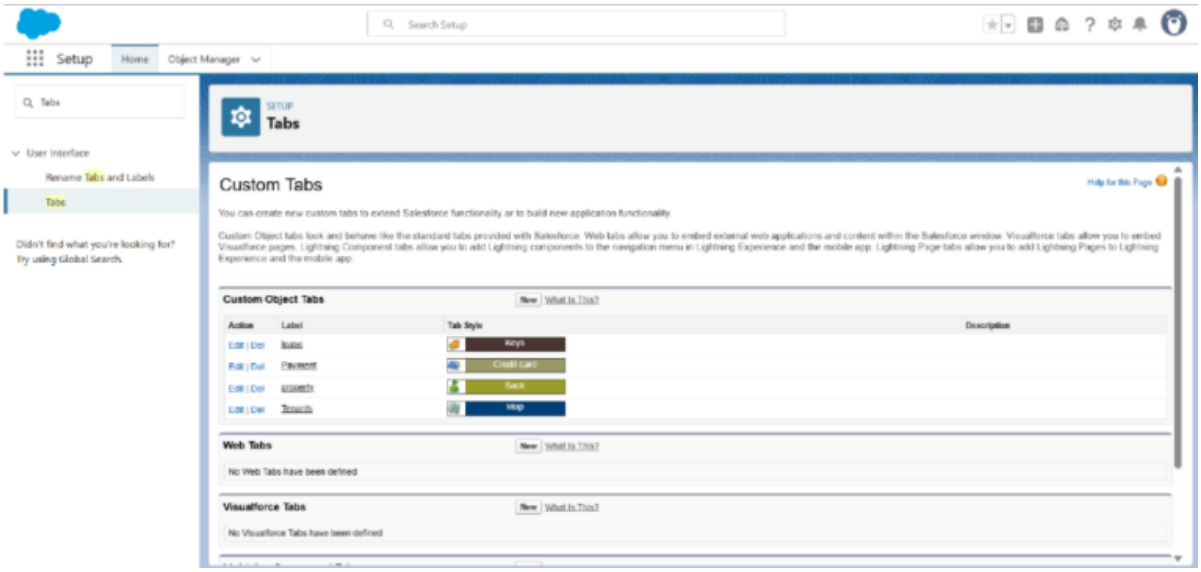
- Approval process validated through email alerts and status updates



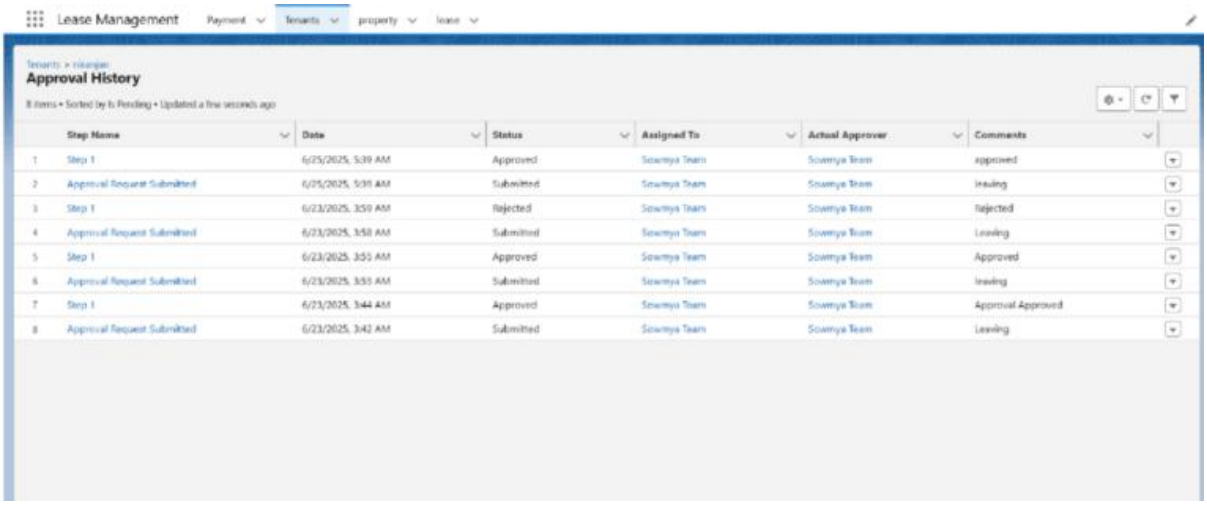
RESULTS

Output Screenshots

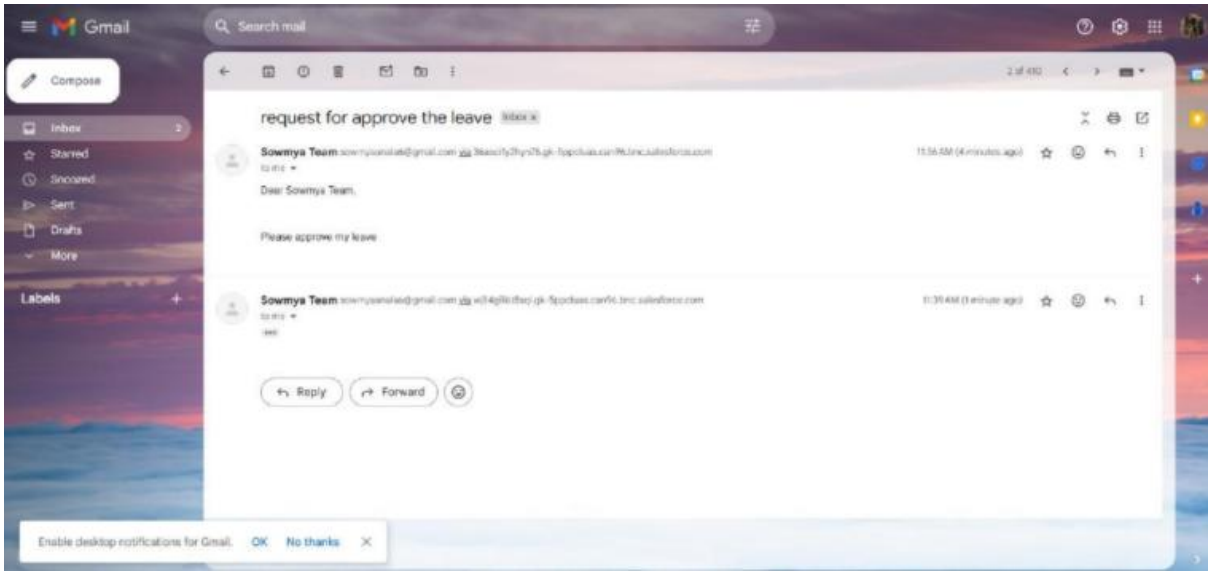
- Tabs for Property, Tenant, Lease, Payment



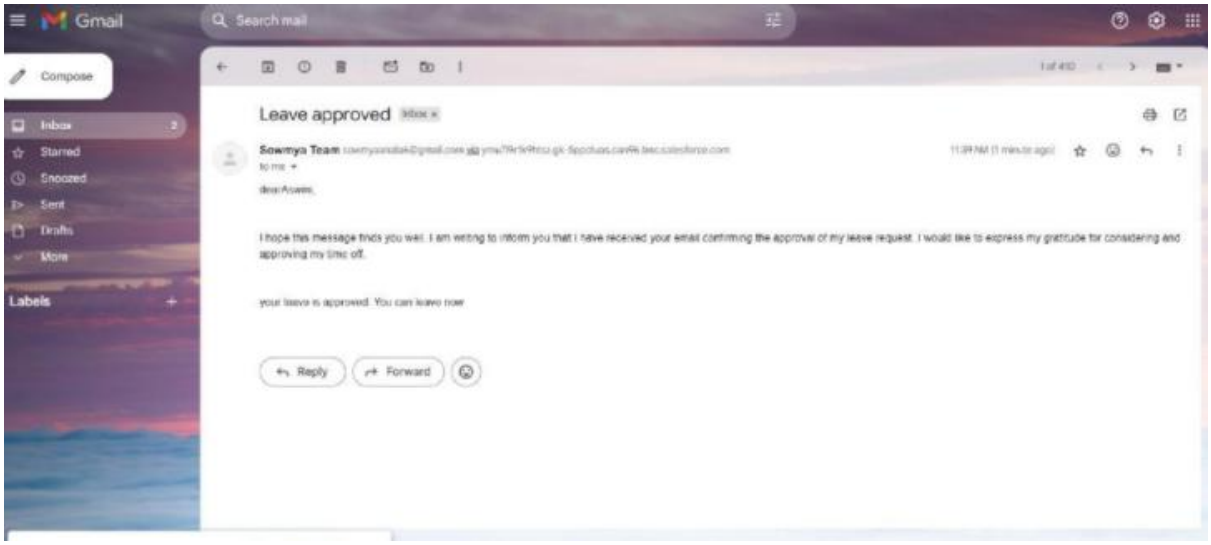
- Email alerts



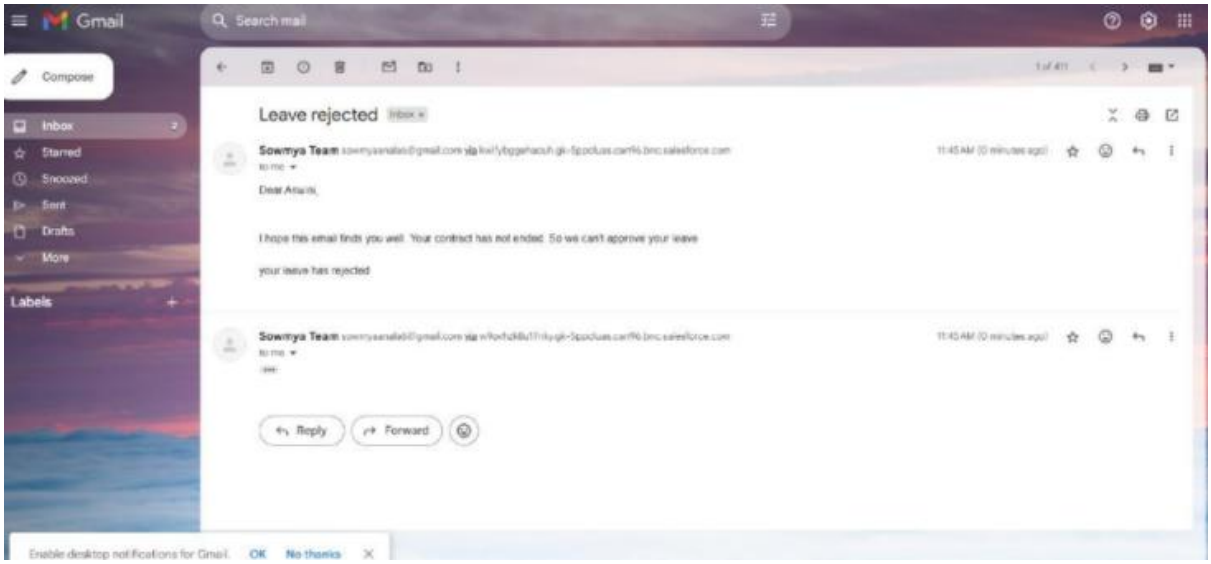
- Request for approve the leave



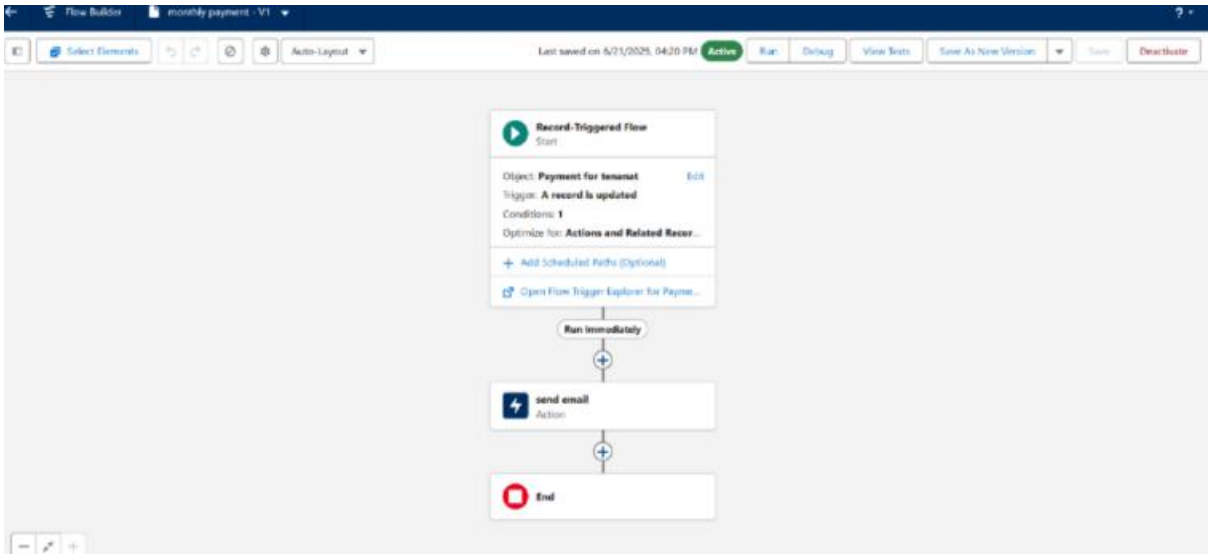
- Leave approved



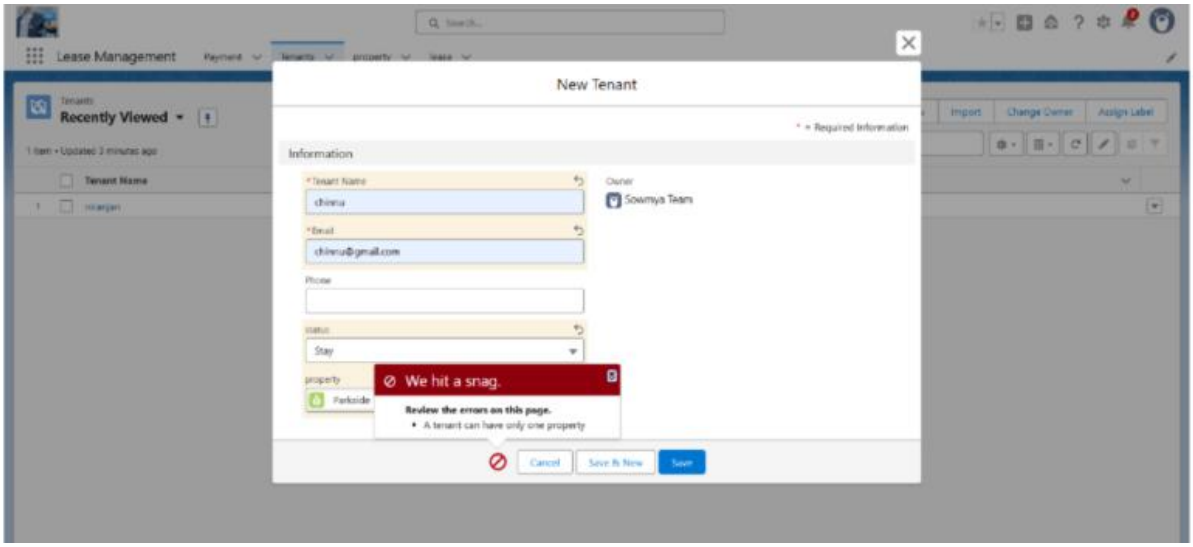
- Leave rejected



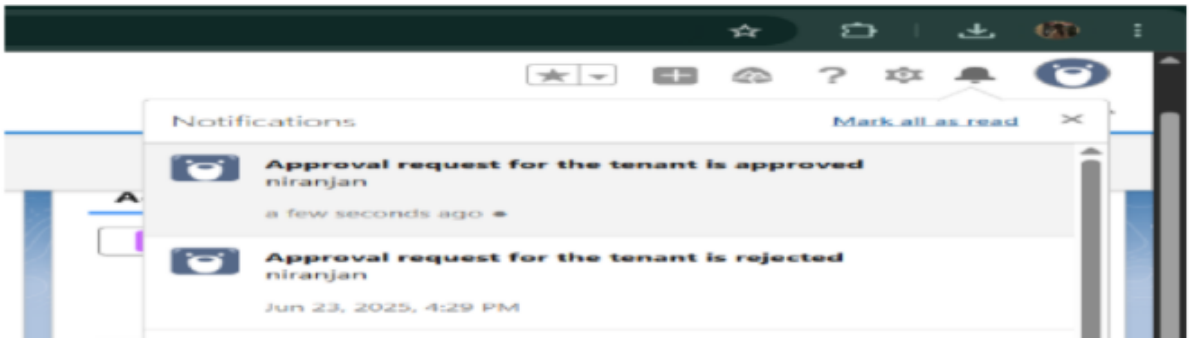
● Flow runs



● Trigger error messages



● Approval process notifications



ADVANTAGES & DISADVANTAGES

Advantages:

- Automates routine tasks like rent reminders, approvals, and payments.

- Reduces manual errors by using validation rules and flows.

- Improves efficiency and saves time for administrators.

- Provides real-time tracking of properties, tenants, and payments.

- Enhances communication through automated email alerts and notifications.

- Centralized data storage ensures easy access and better decision-making.

Disadvantages:

- Requires initial learning effort to understand salesforce setup and features.

- Internet connectivity is mandatory to use salesforce effectively.

- Advanced customization (like Apex coding for triggers/schedulers) may be difficult for

beginners.

- In real-world business, Salesforce licensing costs can be high (though the developer edition is free for projects).

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});
}
}
}
```