# LEASE MANAGEMENT

**College Name: NANDHA ARTS AND SCIENCE COLLEGE ERODE-52**

**College Code: bru4j**

**TEAM ID:** **NM2025TMID22930**

***TEAM MEMBERS:*** *5*

**Team LeaderName:** Varshini v

**Email:** vairavelvarshini@gmail.com

**Team Member:** Karthika E

**Email:** karthikaeswaran2005@gmail.com

**Team Member:** Antony P

**Email:** antony2005anto123@gmail.com

**Team Member:** Arulkumaran T

**Email:** arulkumaran0311@gmail.com

**Team Member:** Madhumitha M

**Email:** madhumitha05236@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.

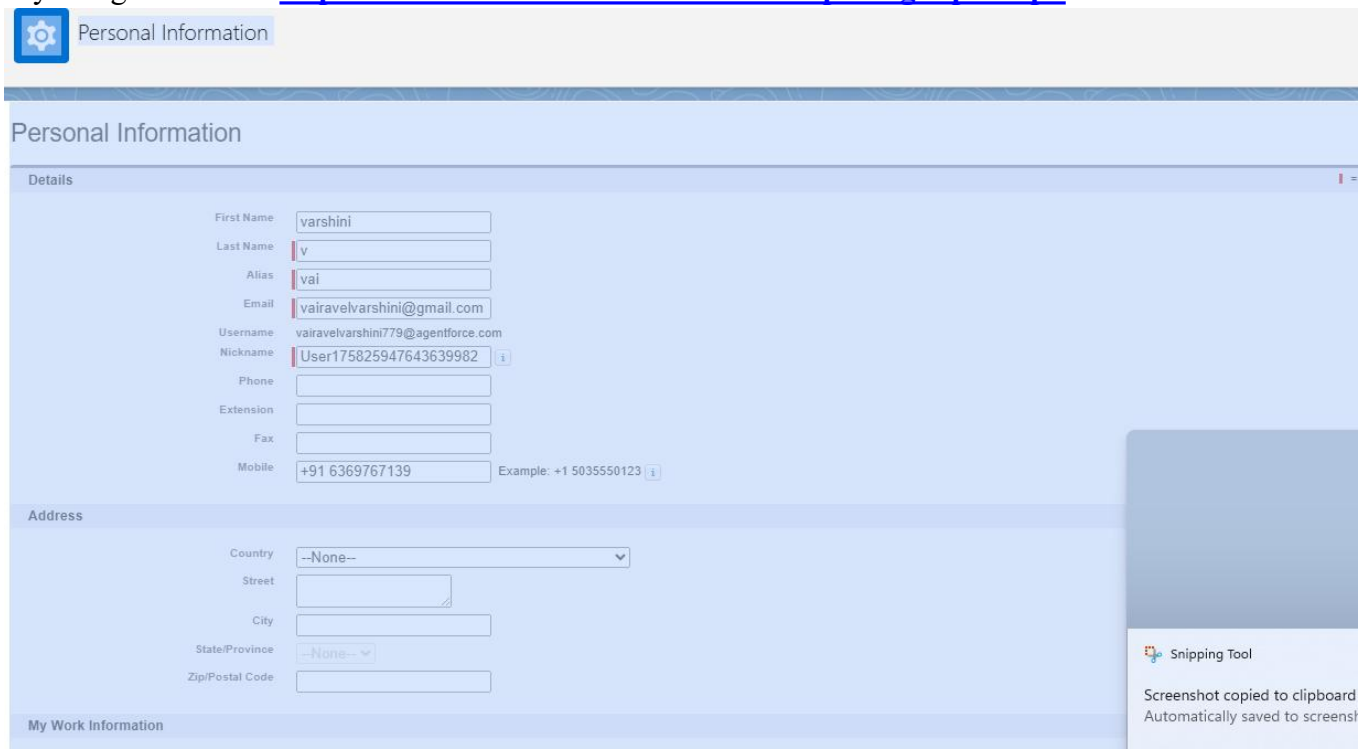
## 1.2 Purpose

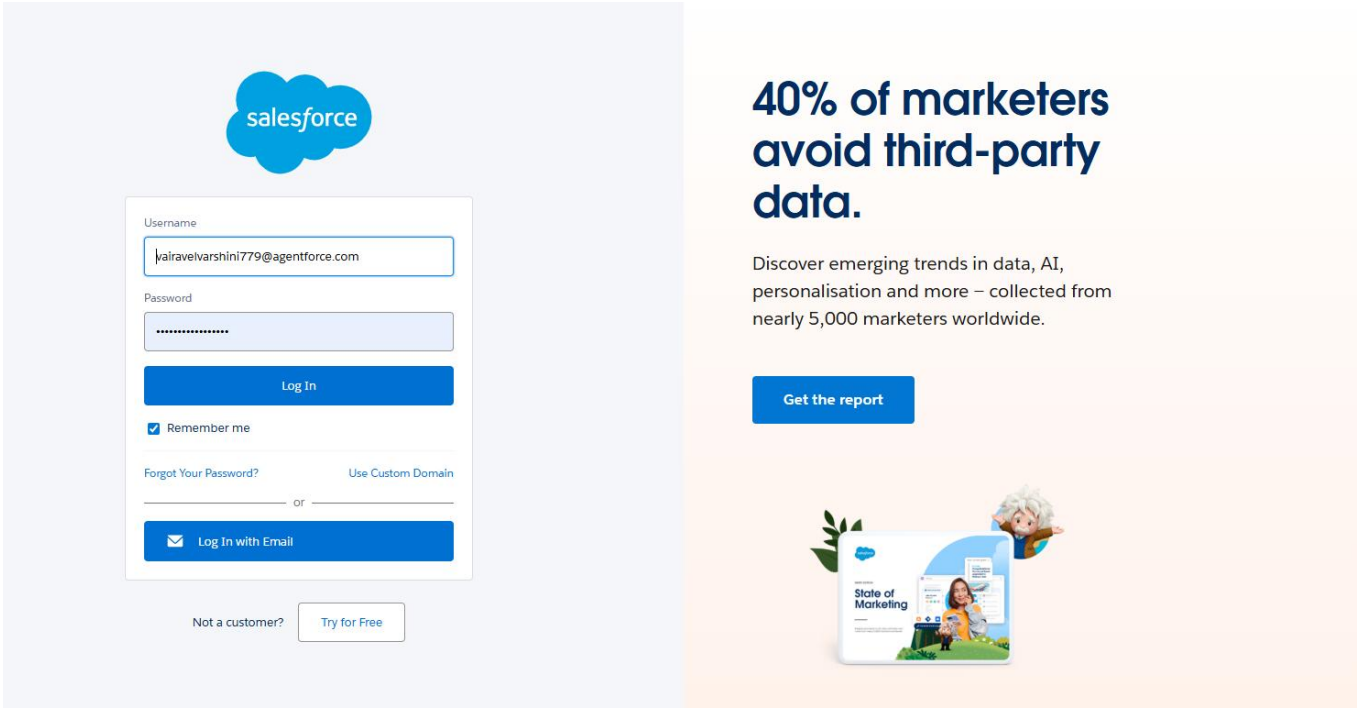
The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

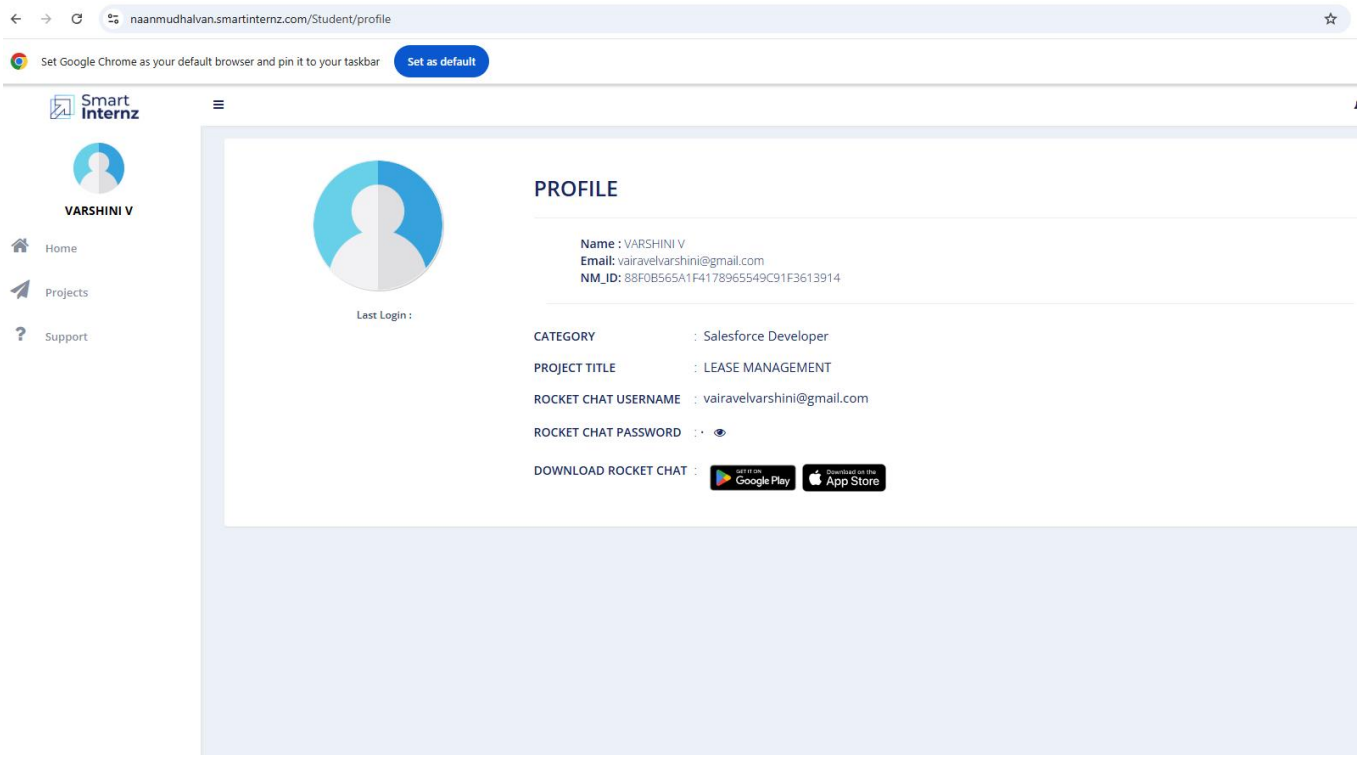
# DEVELOPMENT PHASE

**Creating Developer Account:**

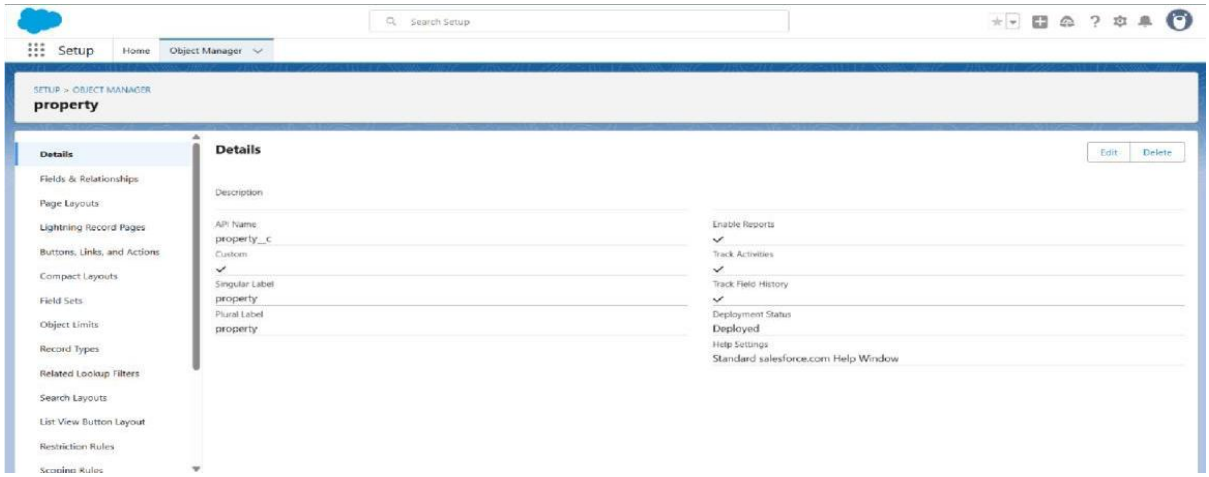
By using this URL - **https://www.salesforce.com/form/developer-signup/?d=pb**

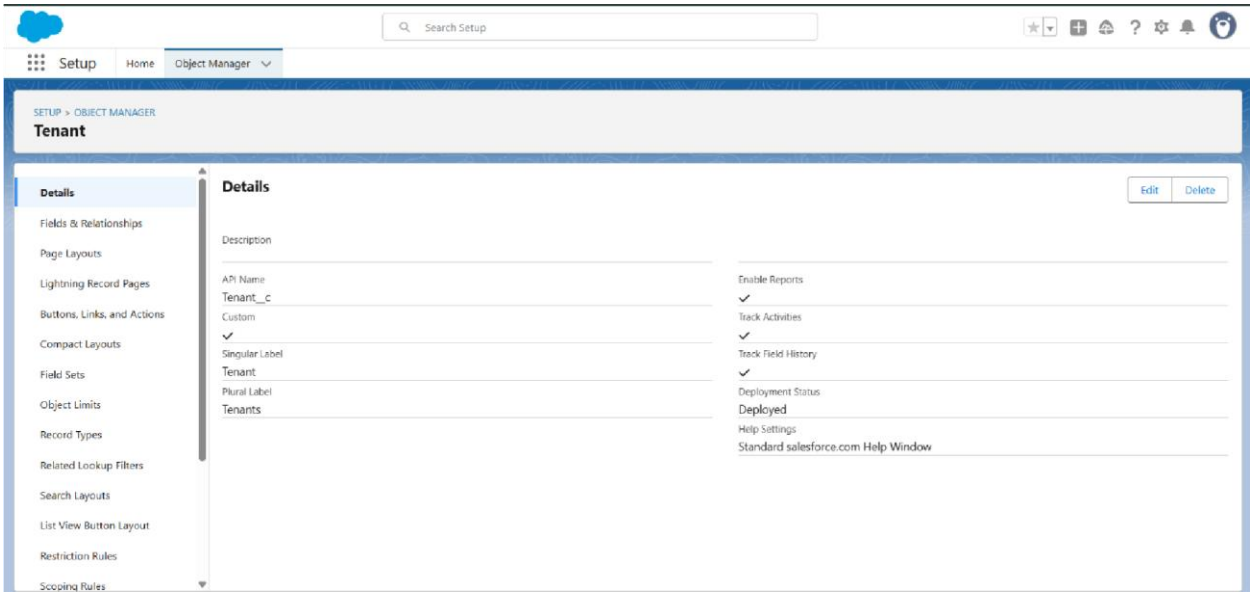


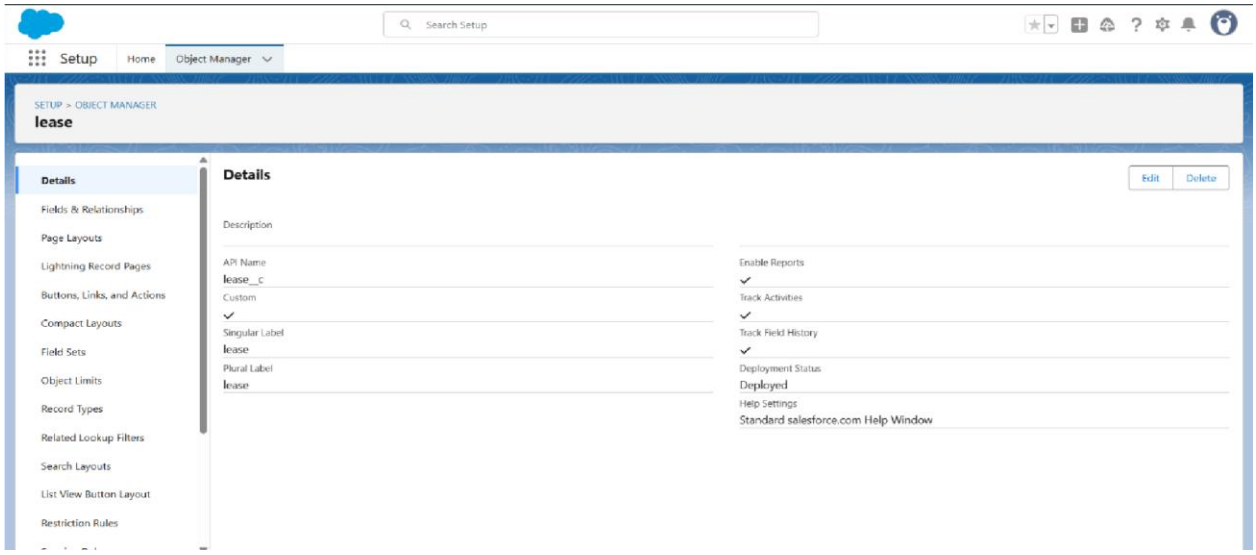

salesforce
Username
vairavelvarshini779@agentforce.com
Password
Log In
Remember me
Forgot Your Password?
Use Custom Domain
or
Log In with Email
Not a customer?
Try for Free
40% of marketers avoid third-party data.
Discover emerging trends in data, AI, personalisation and more – collected from nearly 5,000 marketers worldwide.
Get the report
State of Marketing

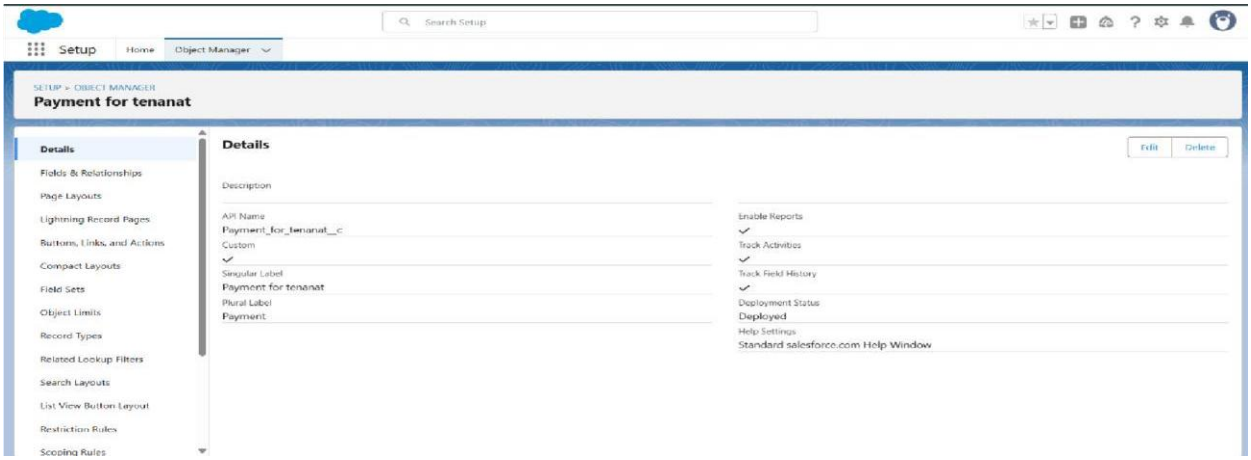

naanmudhalvan.smartinternz.com/Student/profile
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Smart Internz
VARSHINI V
Home
Projects
Support
Last Login :
PROFILE
Name : VARSHINI V
Email: vairavelvarshini@gmail.com
NM_ID: 88F0B565A1F4178965549C91F3613914
CATEGORY : Salesforce Developer
PROJECT TITLE : LEASE MANAGEMENT
ROCKET CHAT USERNAME : vairavelvarshini@gmail.com
ROCKET CHAT PASSWORD :
DOWNLOAD ROCKET CHAT :
Google Play
App Store

- Created objects: Property, Tenant, Lease, Payment

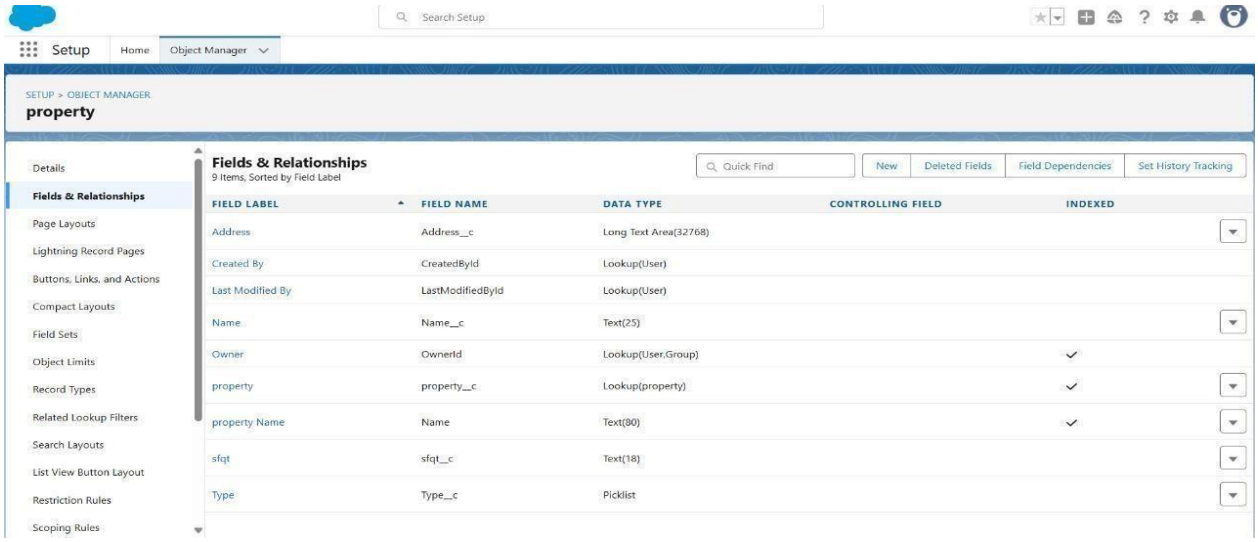


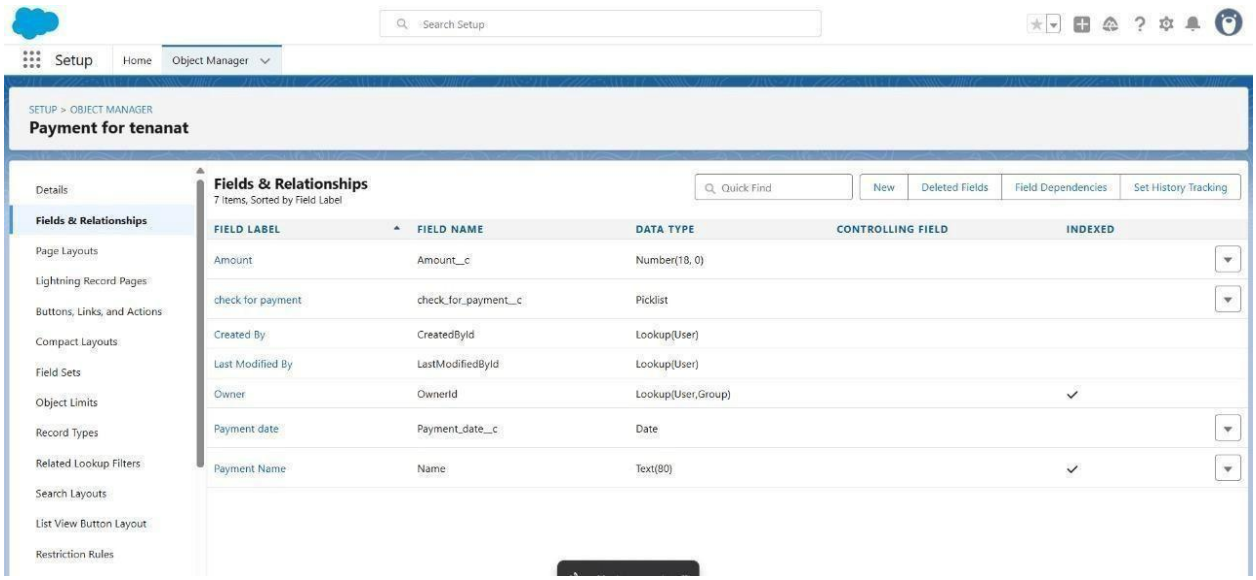



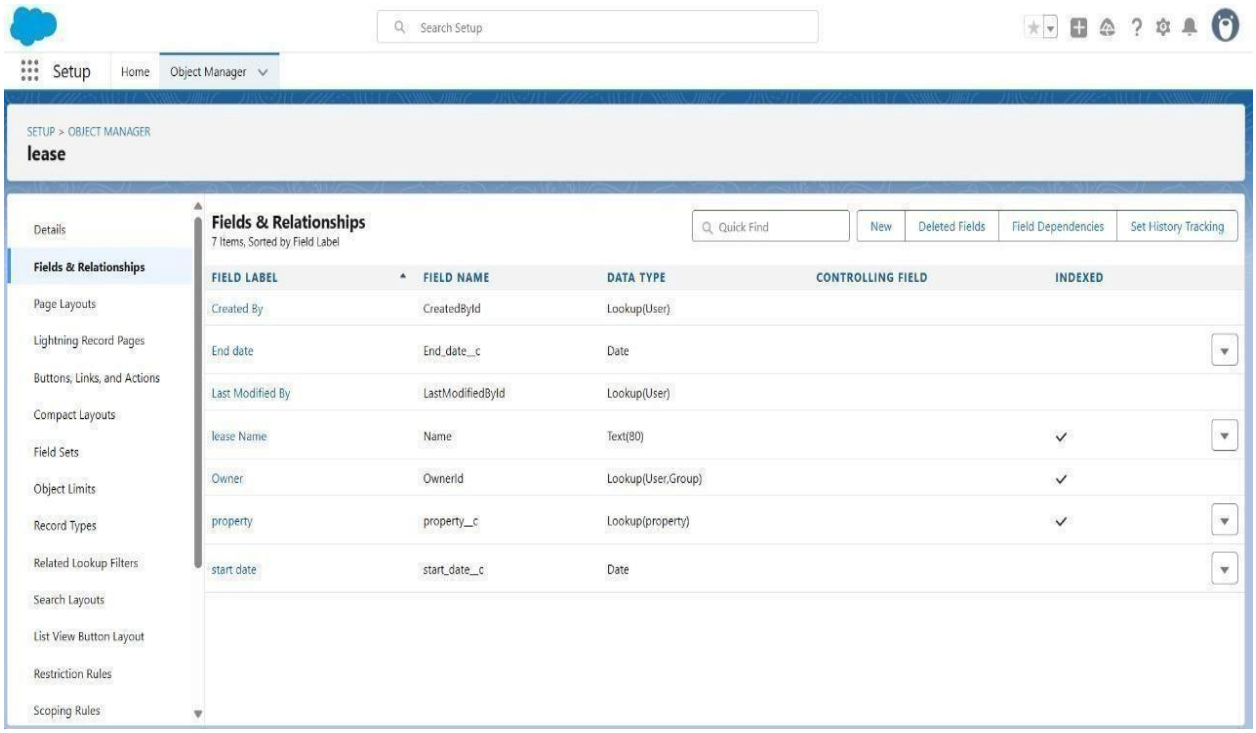

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
lease
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Details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

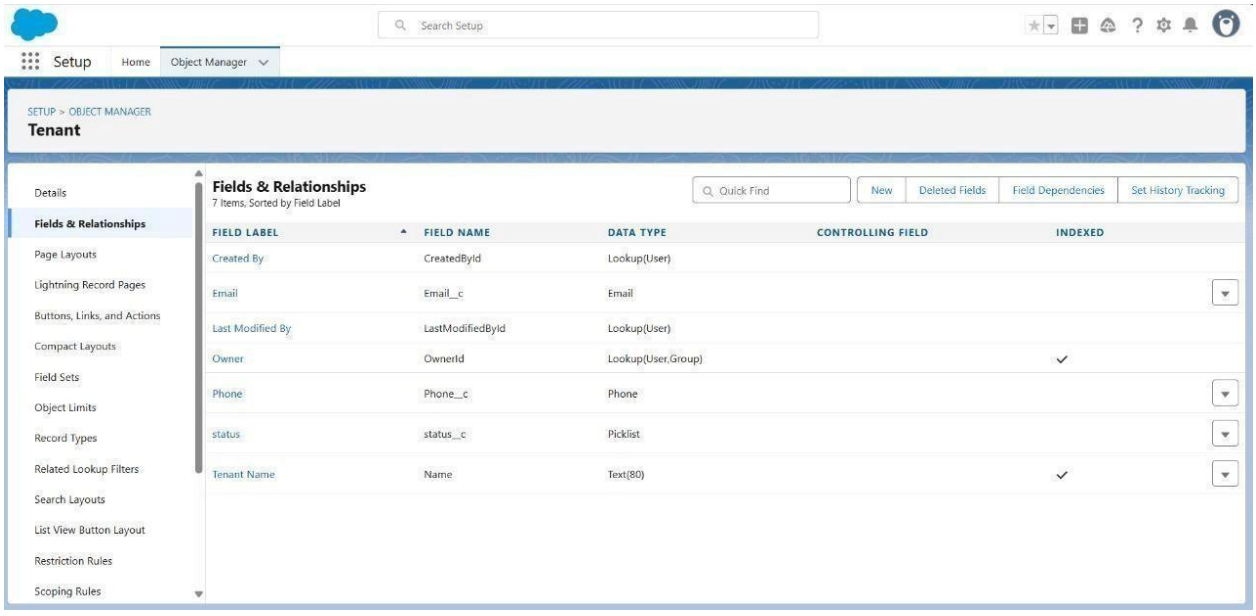

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Payment for tenanat
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Details
Edit
Delete
Description
API Name
Payment_for_tenanat__c
Custom
Singular Label
Payment for tenanat
Plural Label
Payment
Enable Reports
Track Activities
Track Field History
Deployment Status
Deployed
Help Settings
Standard salesforce.com Help Window

- Configured fields and relationships

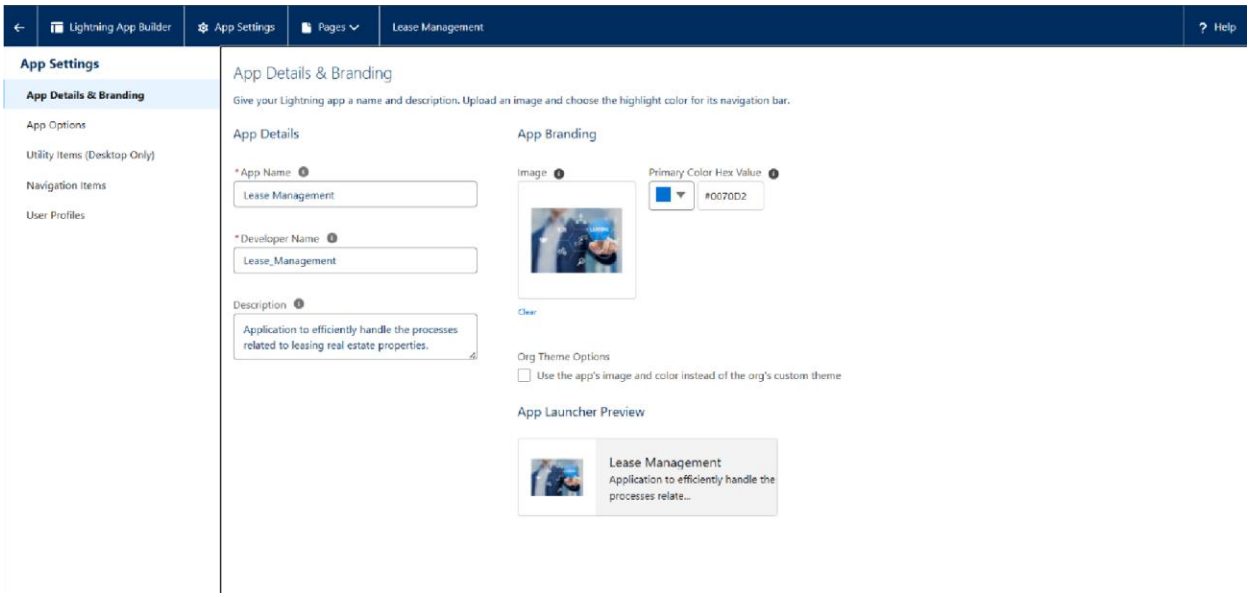


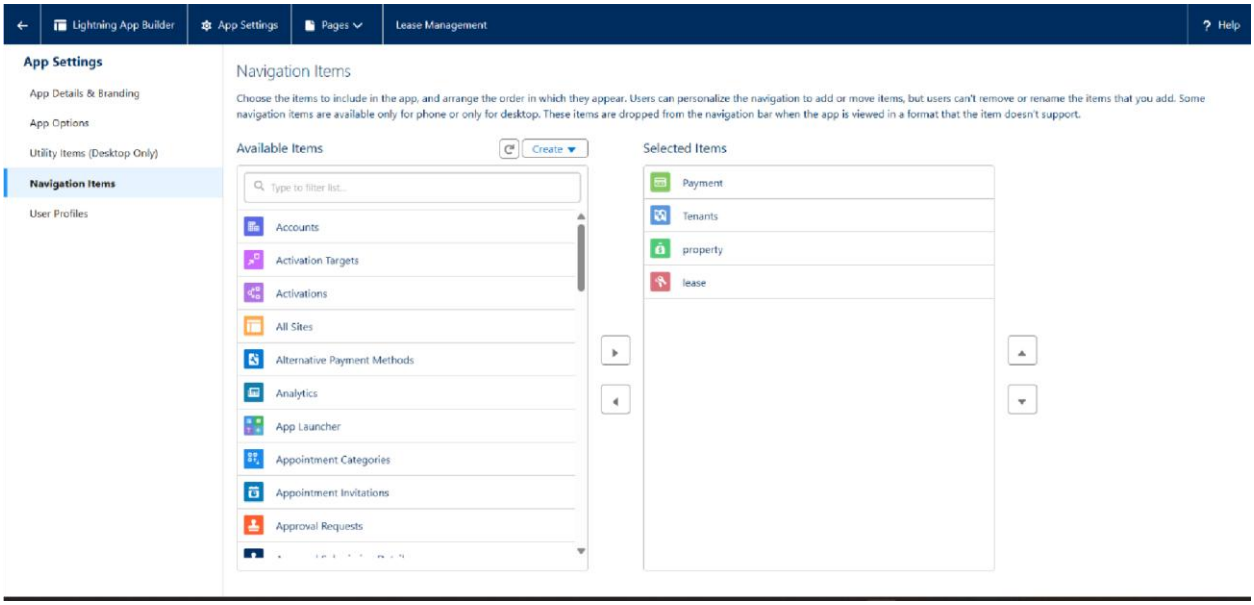



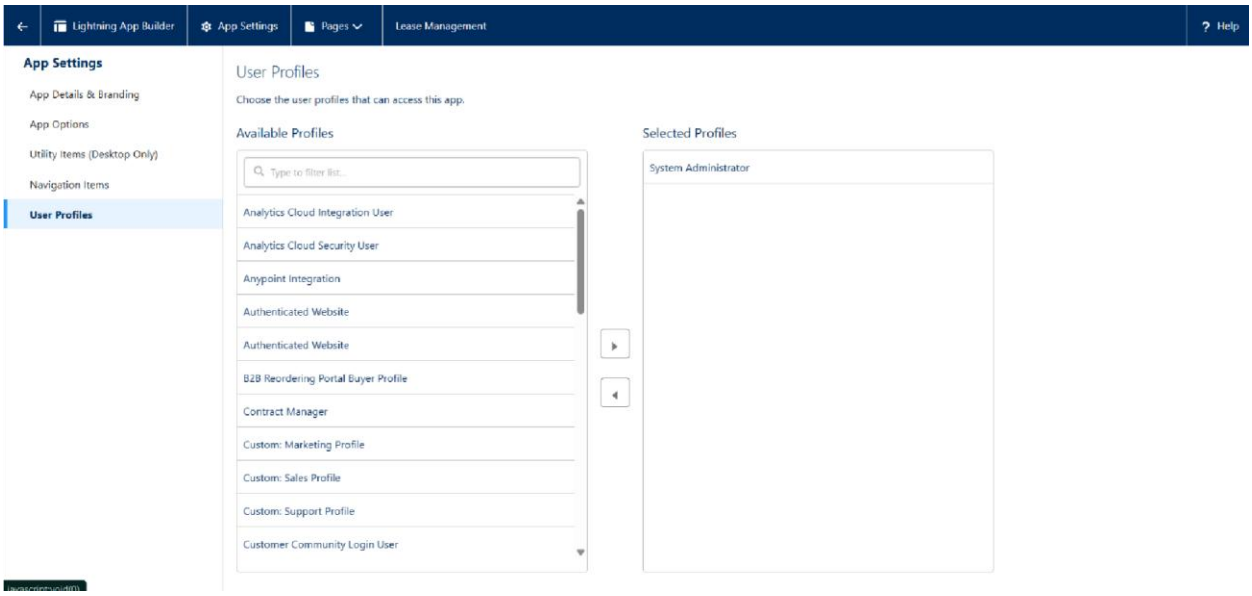

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
lease
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Fields & Relationships
7 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set 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

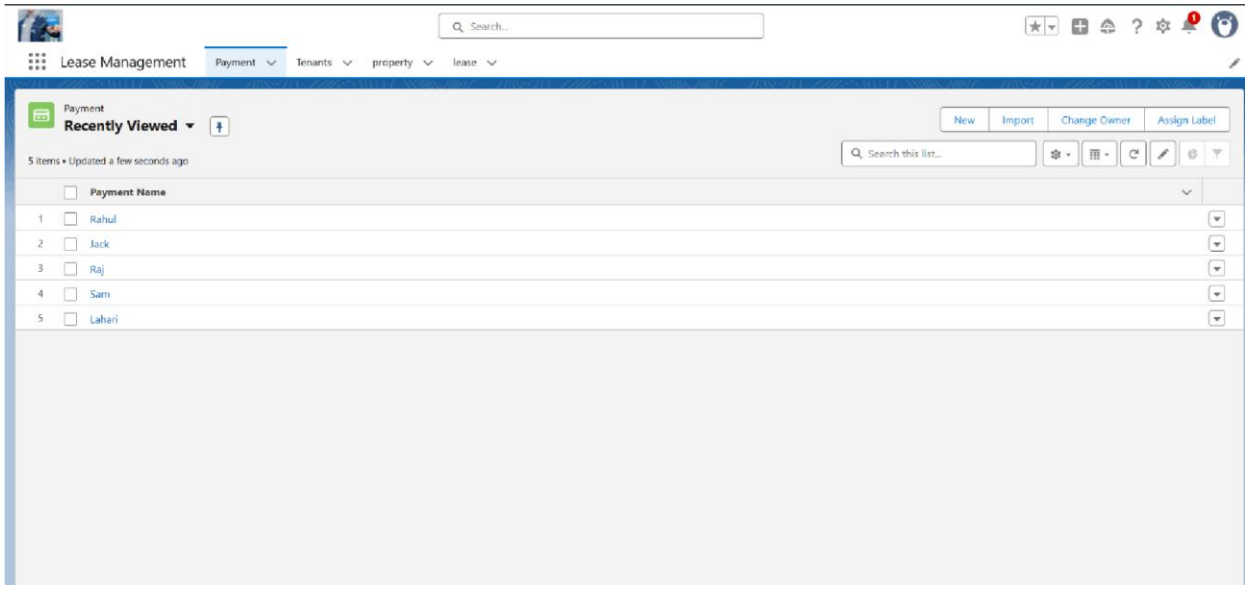

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Tenant
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Fields & Relationships
7 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Created By
CreatedById
Lookup(User)
Email
Email__c
Email
Last Modified By
LastModifiedById
Lookup(User)
Owner
OwnerId
Lookup(User,Group)
Phone
Phone__c
Phone
status
status__c
Picklist
Tenant Name
Name
Text(80)

- Developed Lightning App with relevant tabs

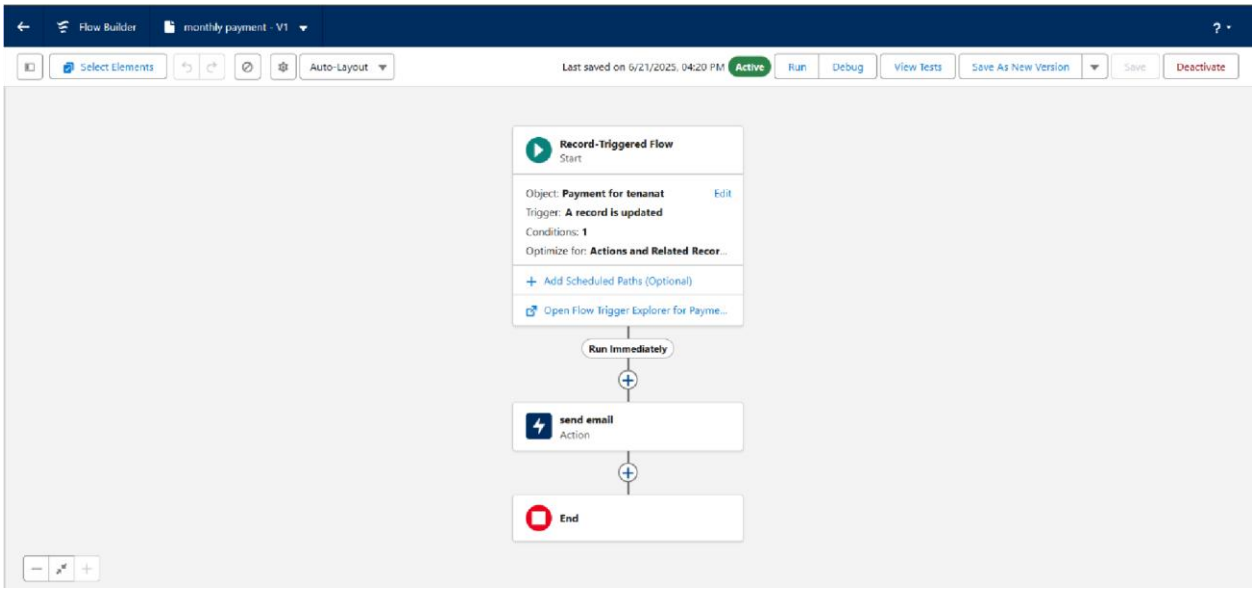


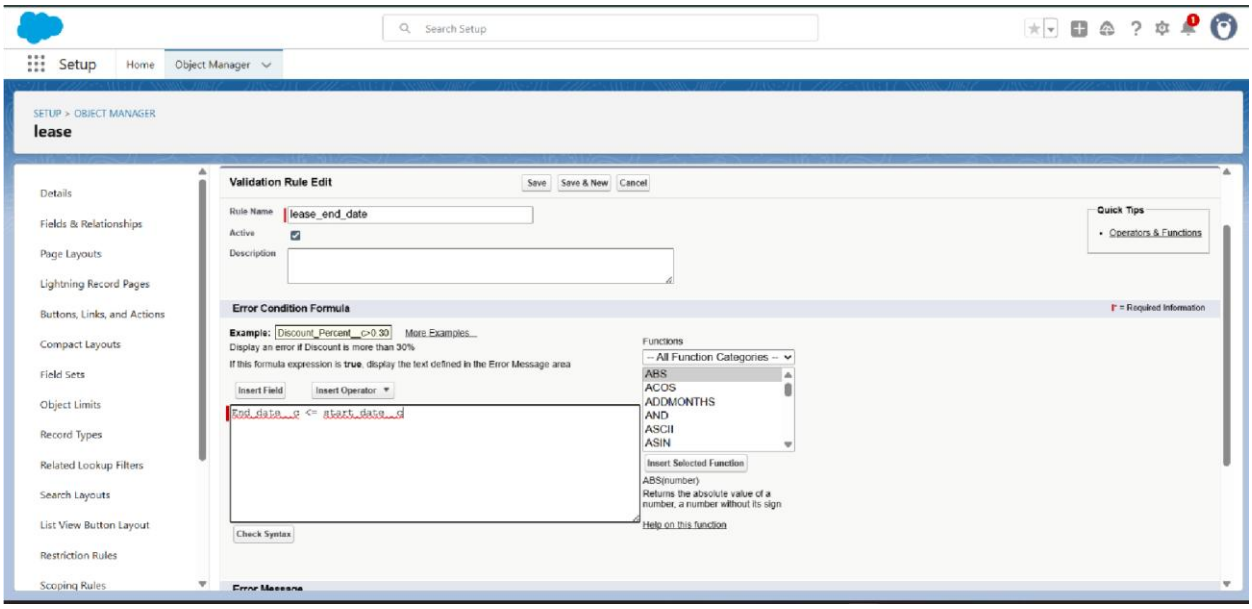

Lightning App Builder
App Settings
Pages
Lease Management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
User Profiles
Choose the user profiles that can access this app.
Available Profiles
Type to filter list...
Analytics Cloud Integration 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


Search...
Lease Management
Payment
Tenants
property
lease
Payment
Recently Viewed
5 items • Updated a few seconds ago
New
Import
Change Owner
Assign Label
Search this list...
Payment Name
1 Rahul
2 Jack
3 Raj
4 Sam
5 Lahari

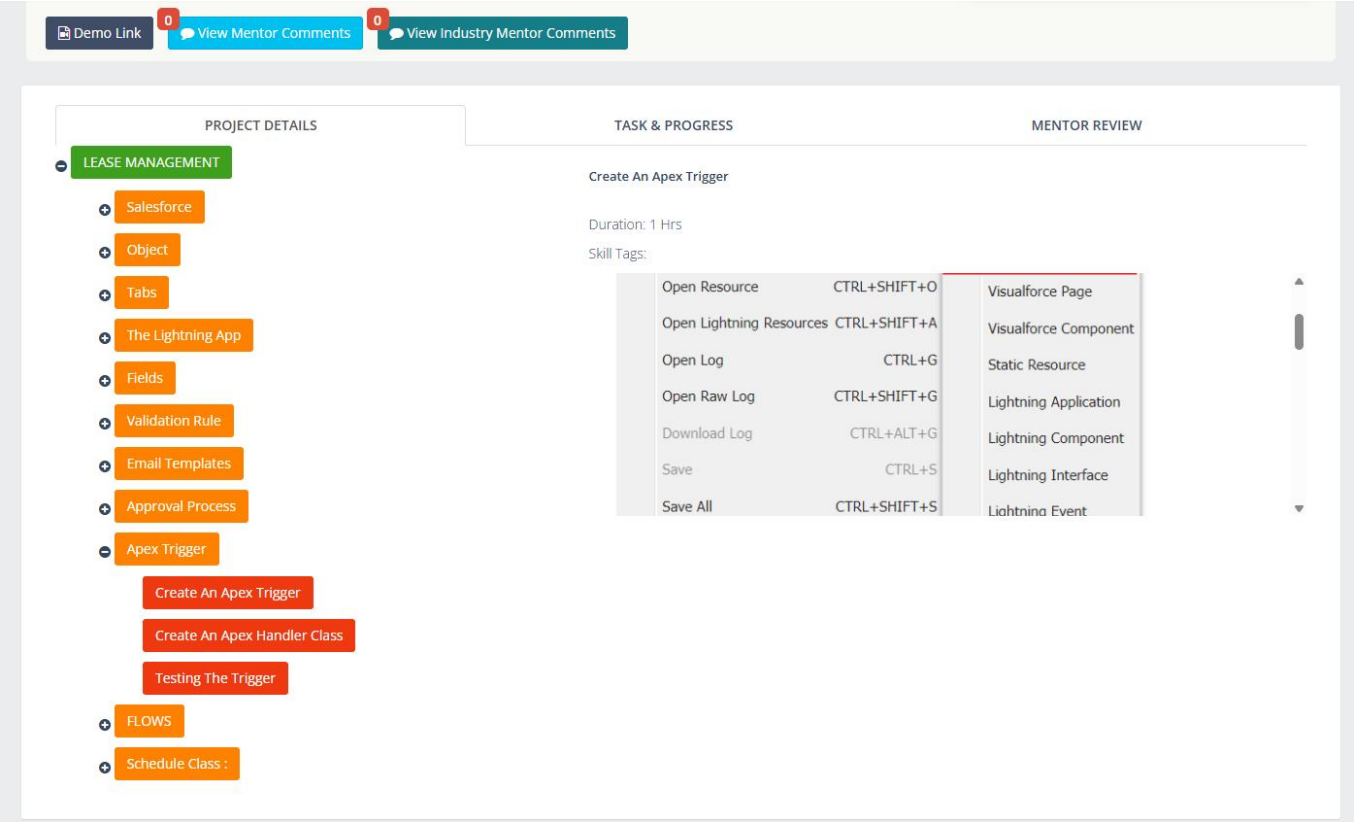
- Implemented Flows for monthly rent and payment success



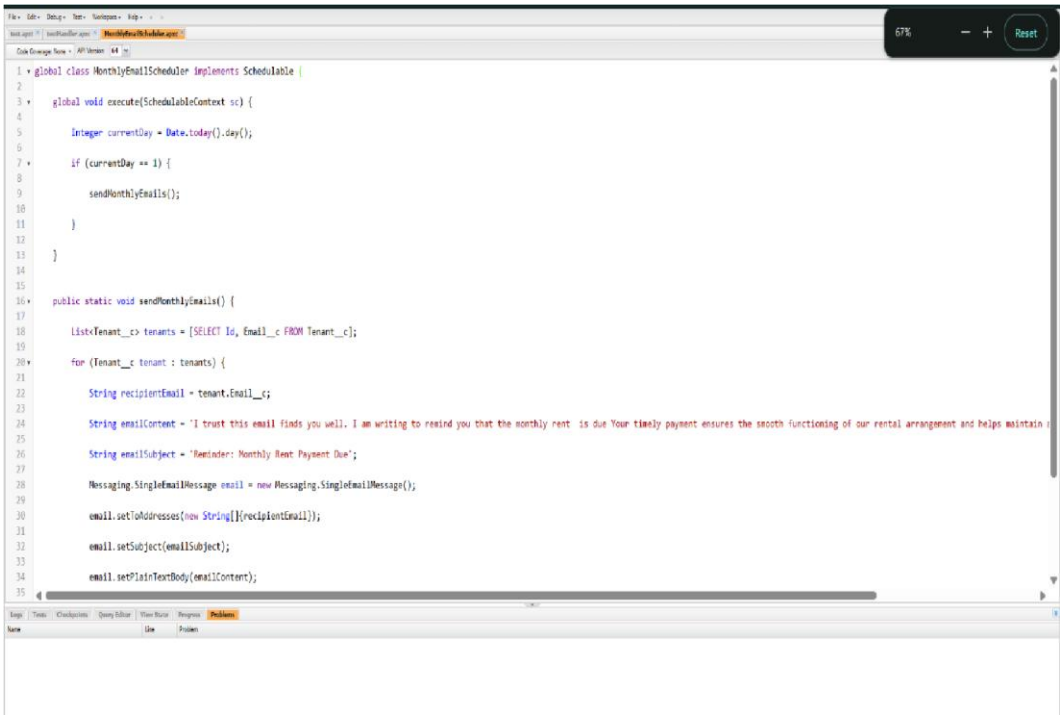

- To create a validation rule to a Lease Object



- Added Apex trigger to restrict multiple tenants per property

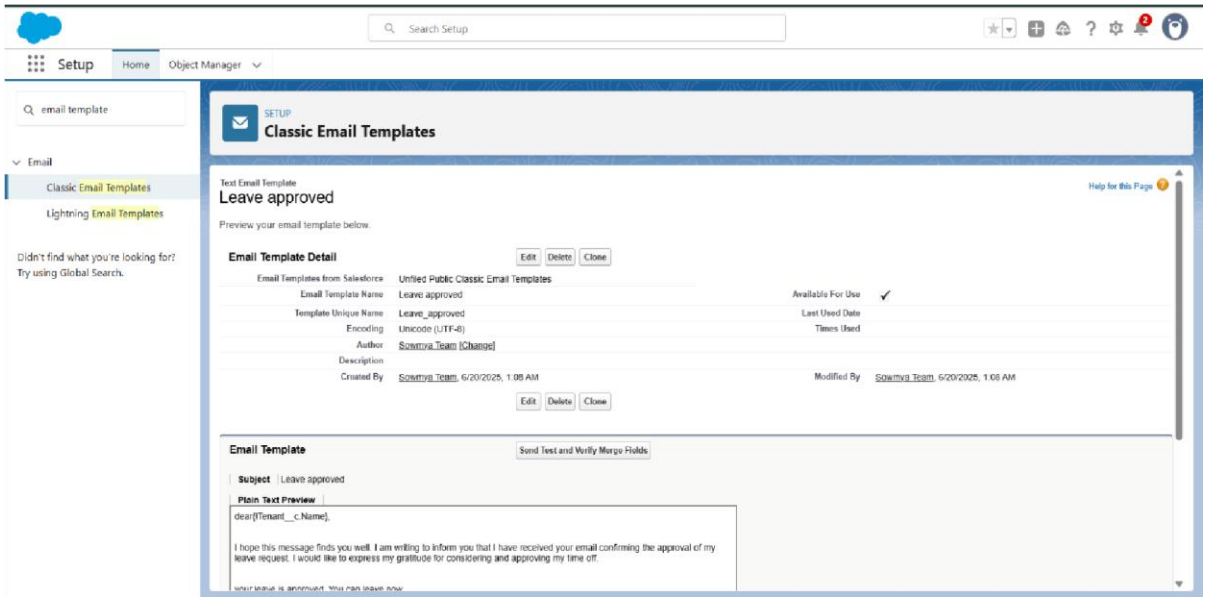


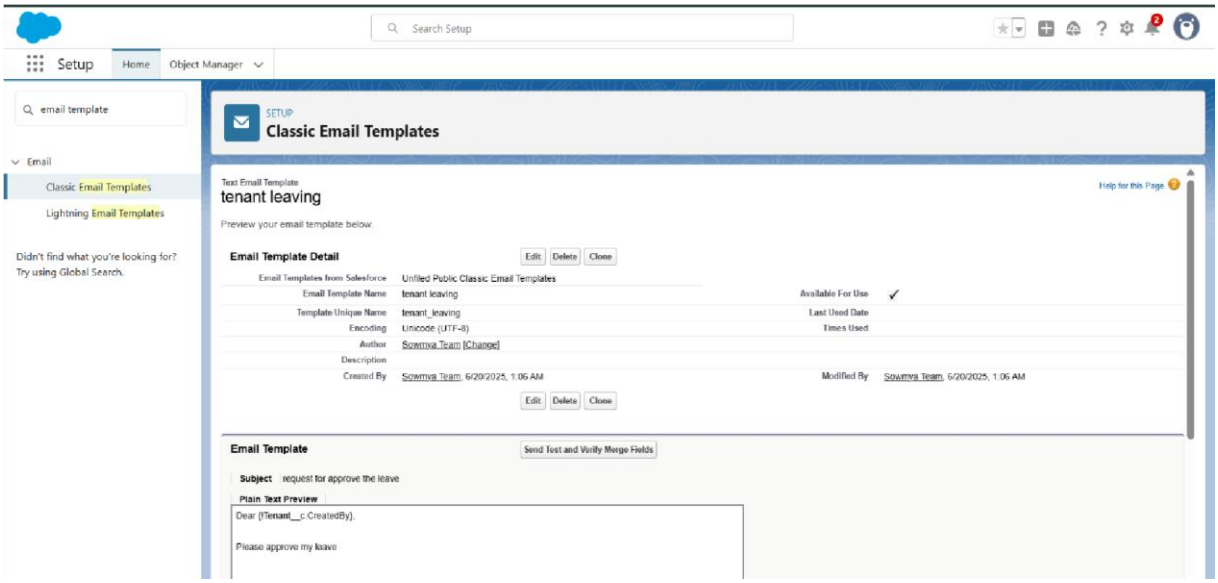

- Scheduled monthly reminder emails using Apex class

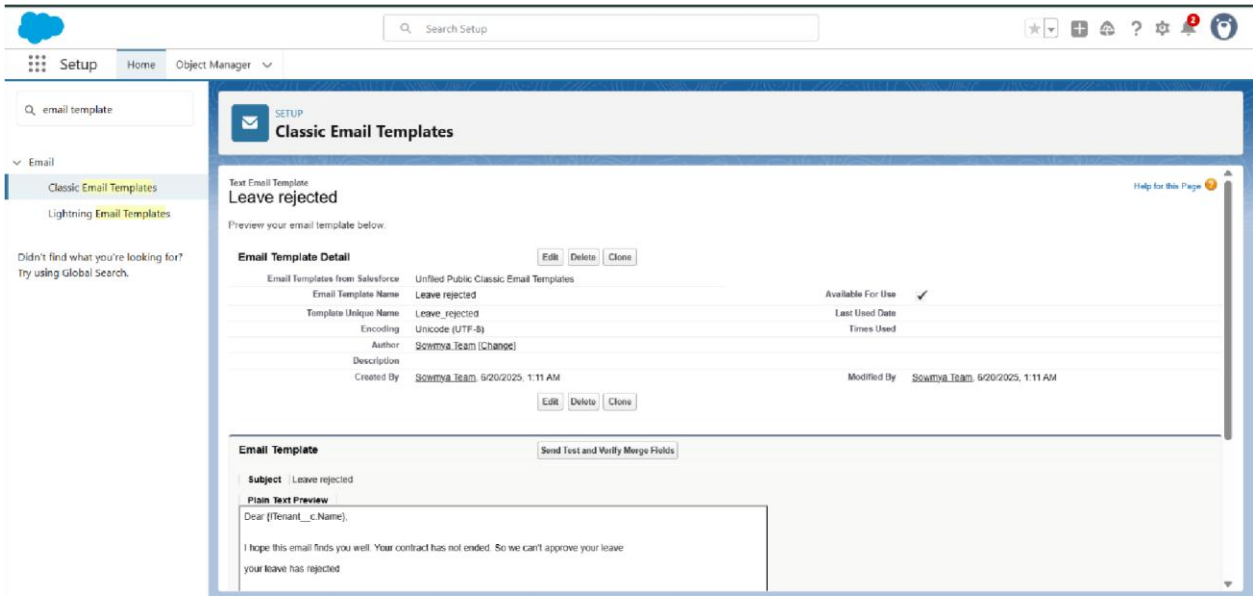


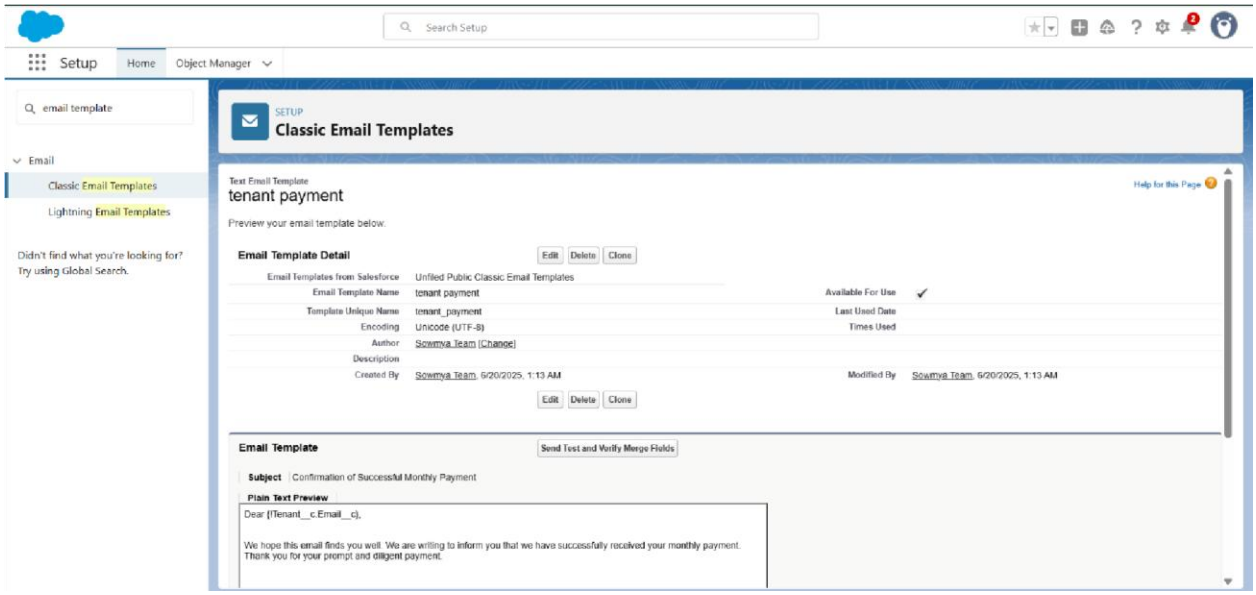
```
global class MonthlyEmailScheduler implements Schedulable {

    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



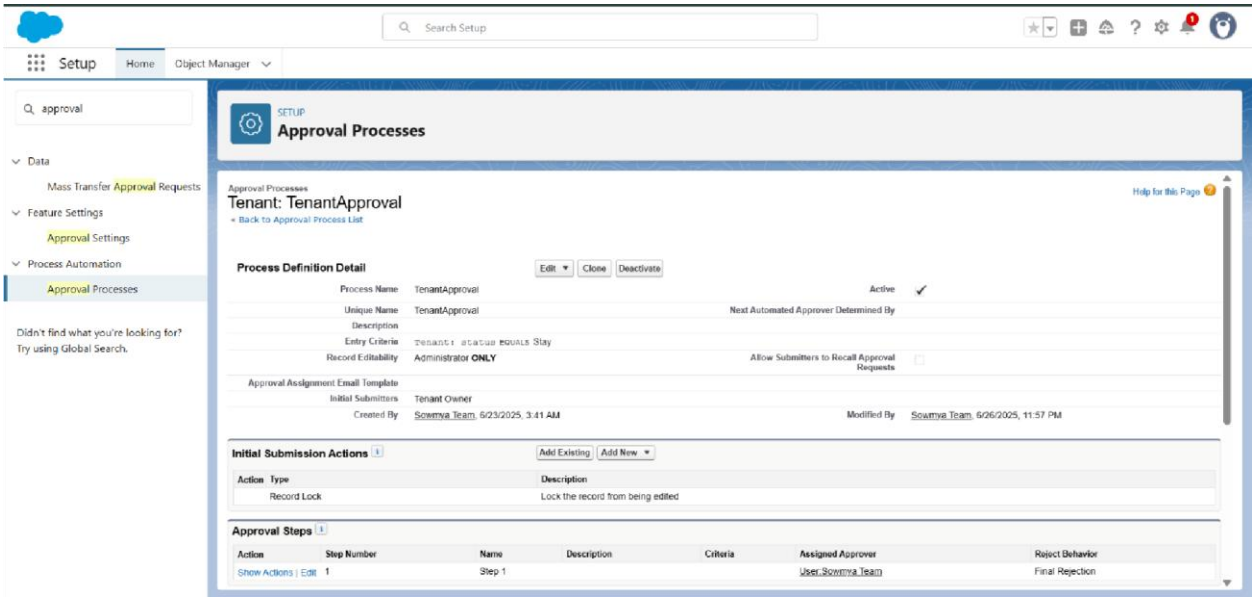

Search Setup
Setup
Home
Object Manager
email template
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
tenant leaving
Help for this Page
Preview your email template below.
Email Template 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
Sowmya Team [Change]
Description
Created By
Sowmya Team, 6/20/2025, 1:06 AM
Modified By
Sowmya Team, 6/20/2025, 1:06 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


Search Setup
Setup
Home
Object Manager
email template
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
Sowmya Team [Change]
Description
Created By
Sowmya Team, 6/20/2025, 1:11 AM
Modified By
Sowmya Team, 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

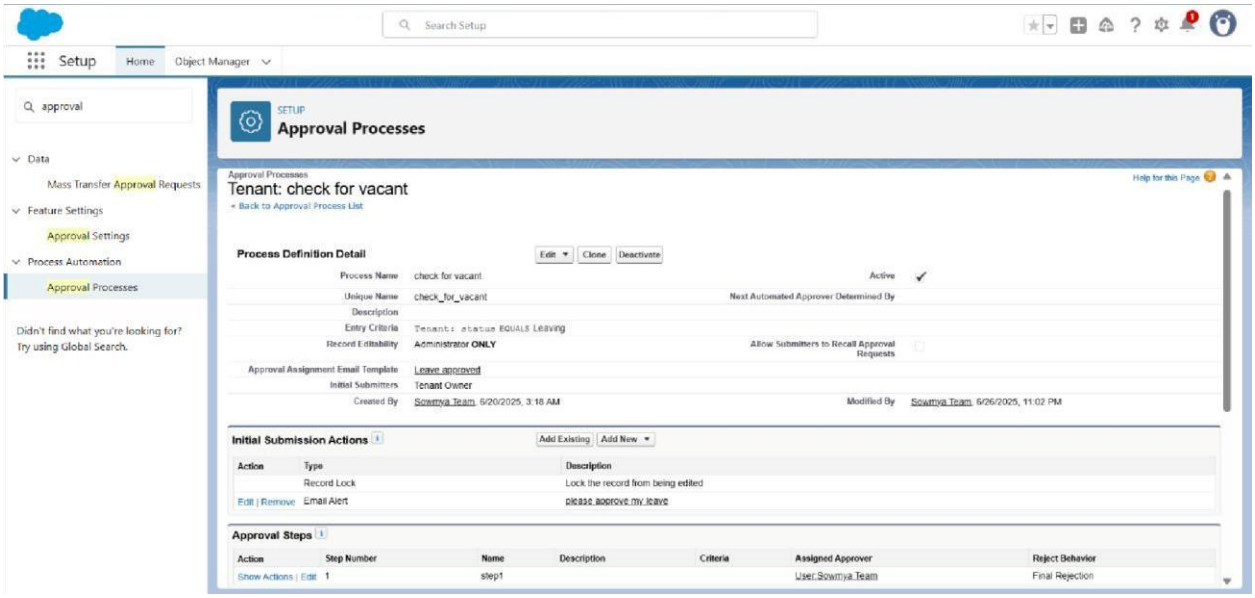



- Approval Process creation

For Tenant Leaving:

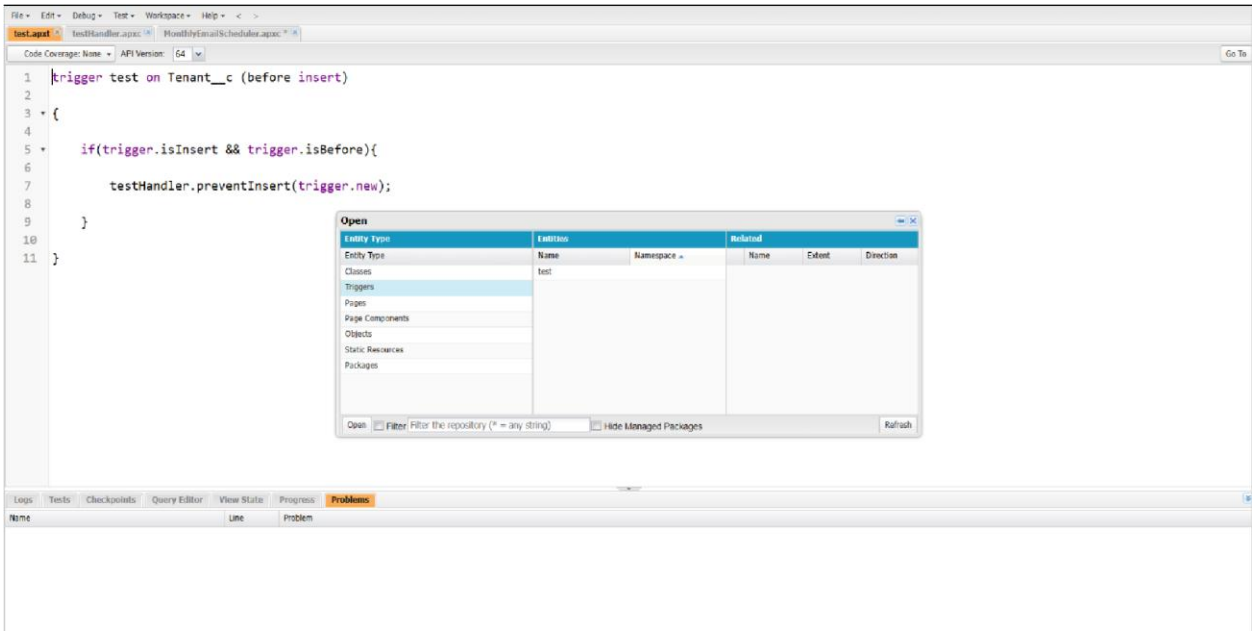


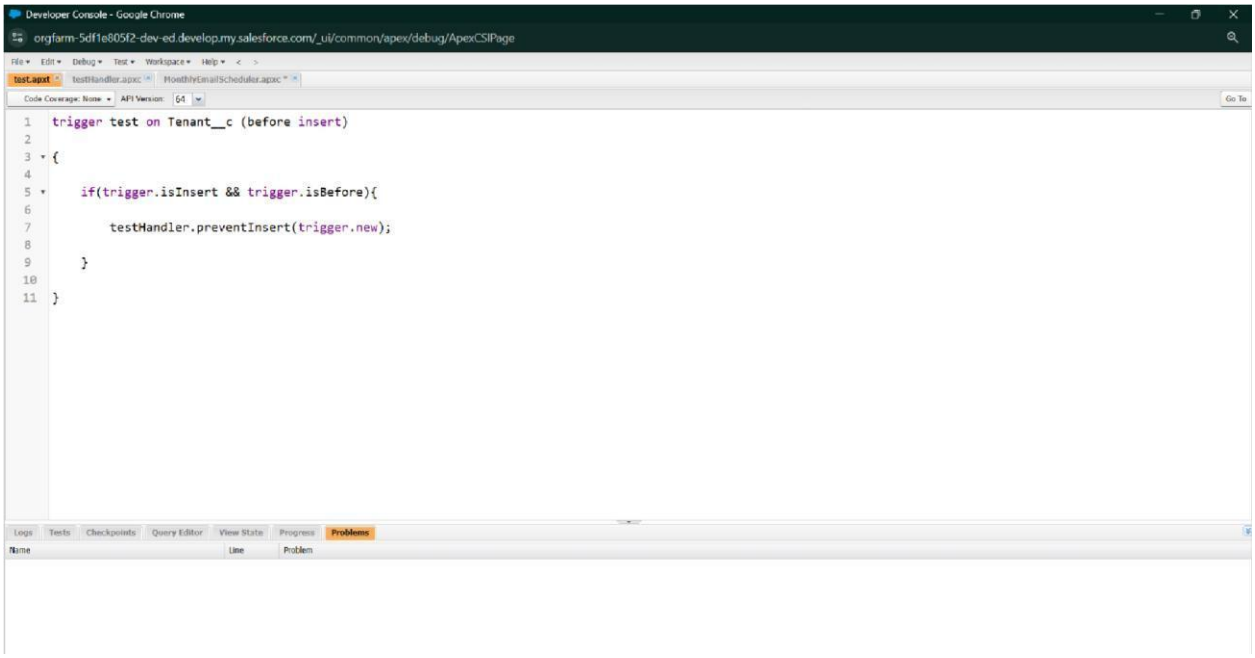
For Check for Vacant:



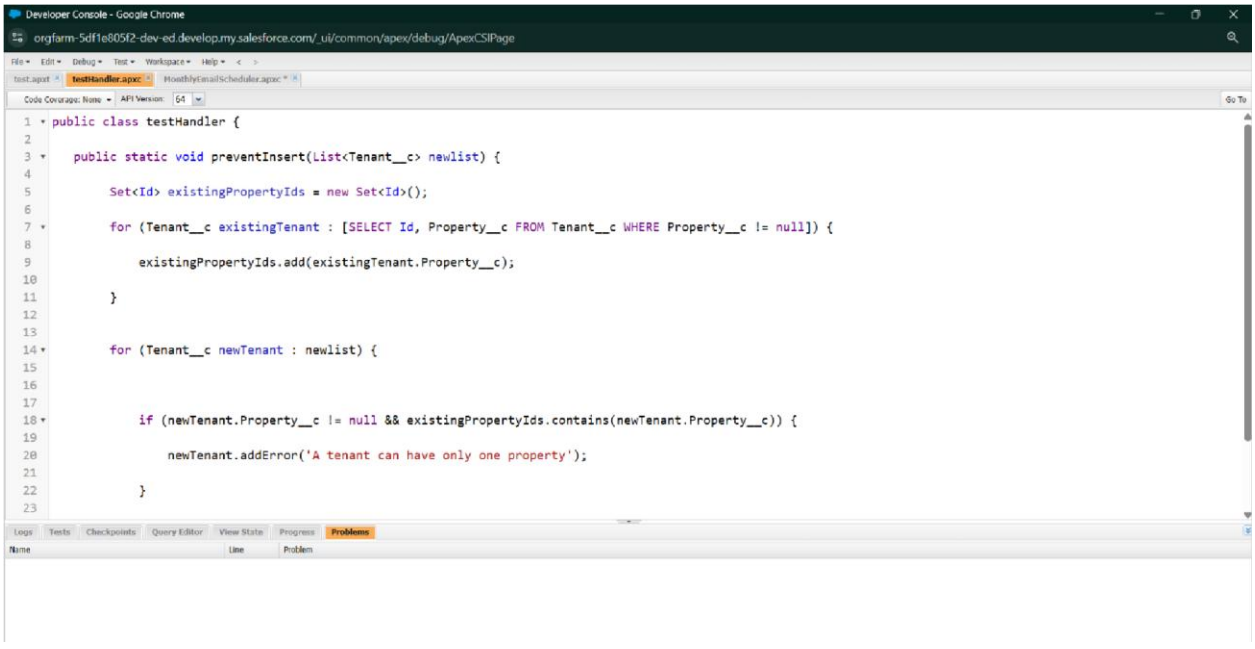

- Apex Trigger

Create an Apex Trigger

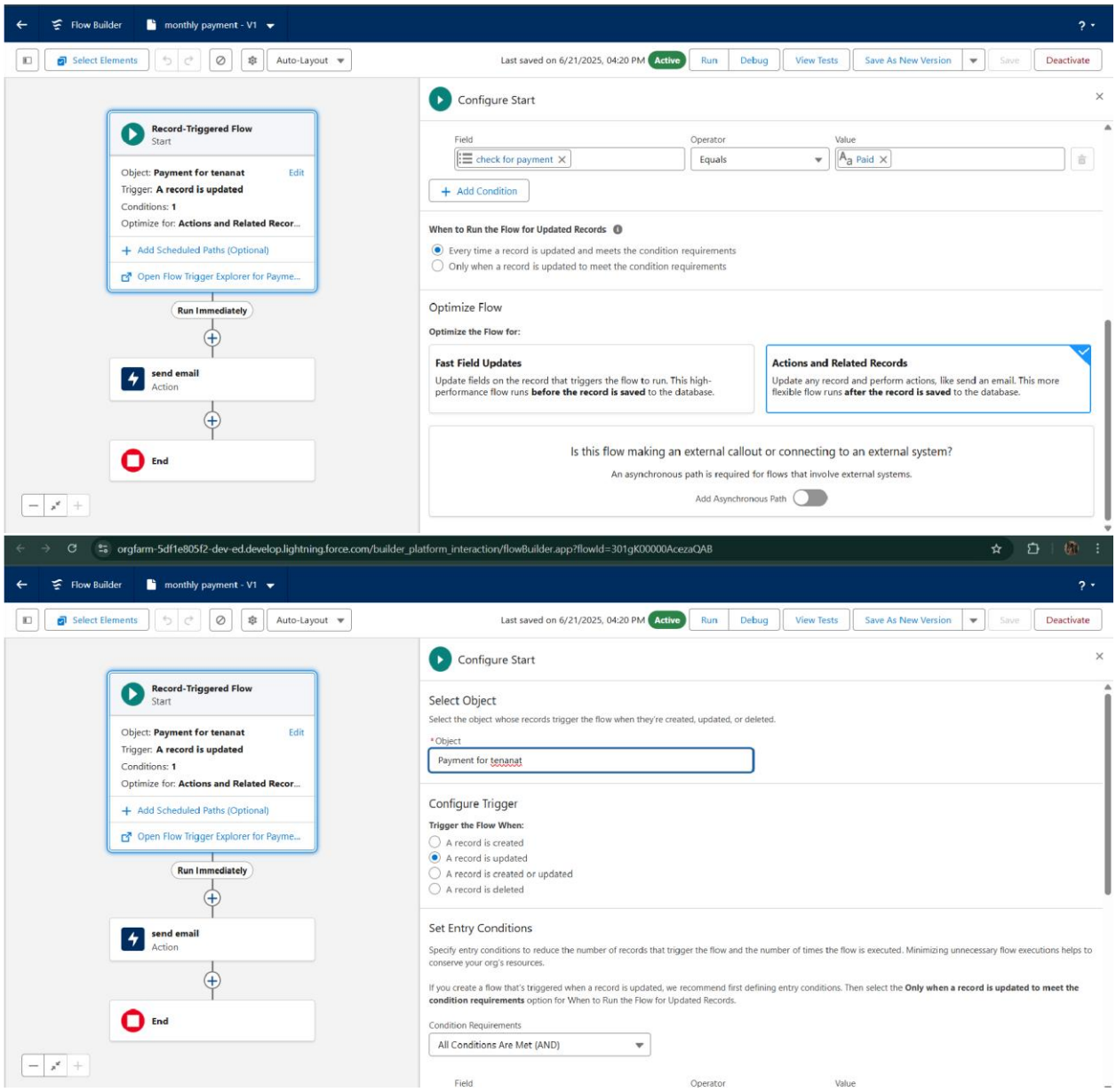




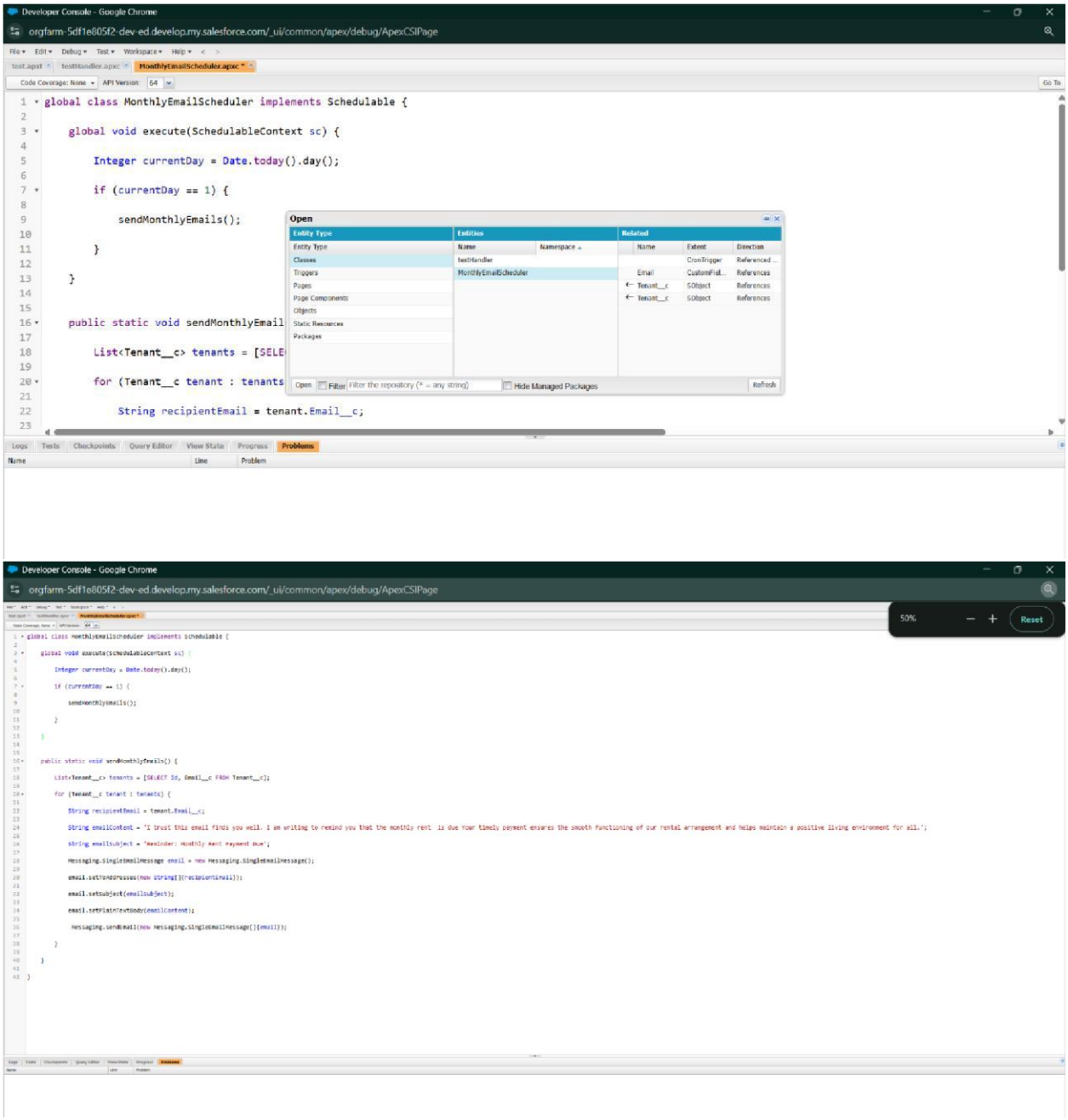

Create an Apex Handler class




- FLOWS



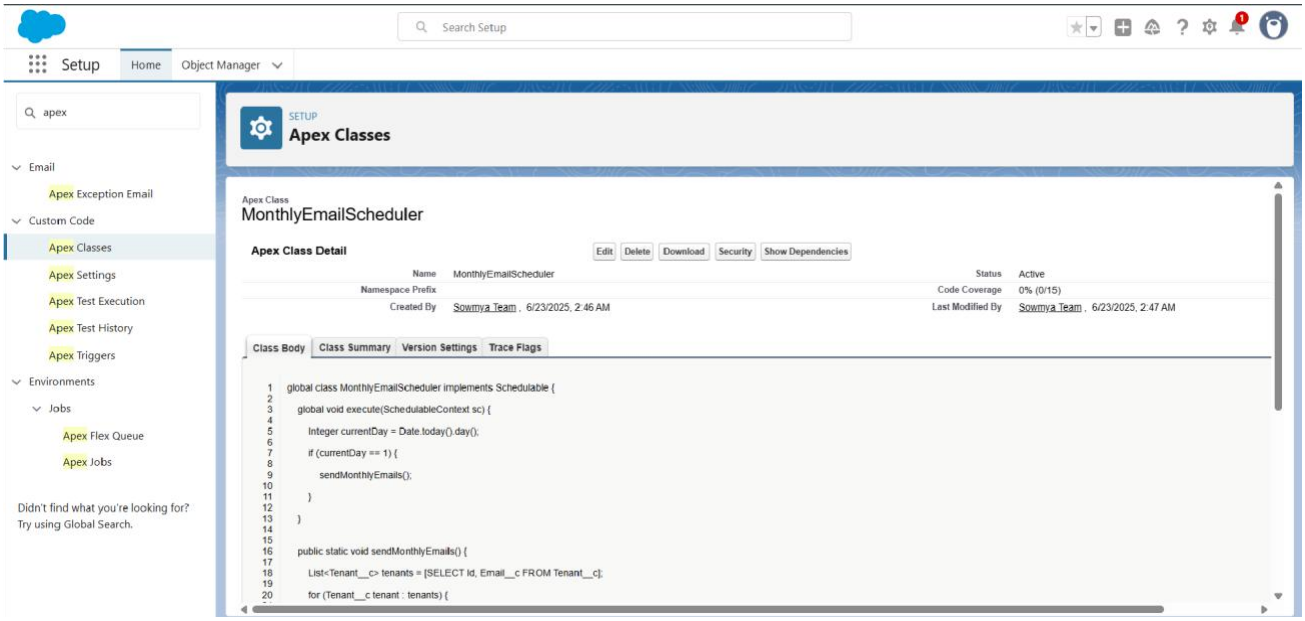

- Schedule class:
  Create an Apex Class







Schedule Apex class

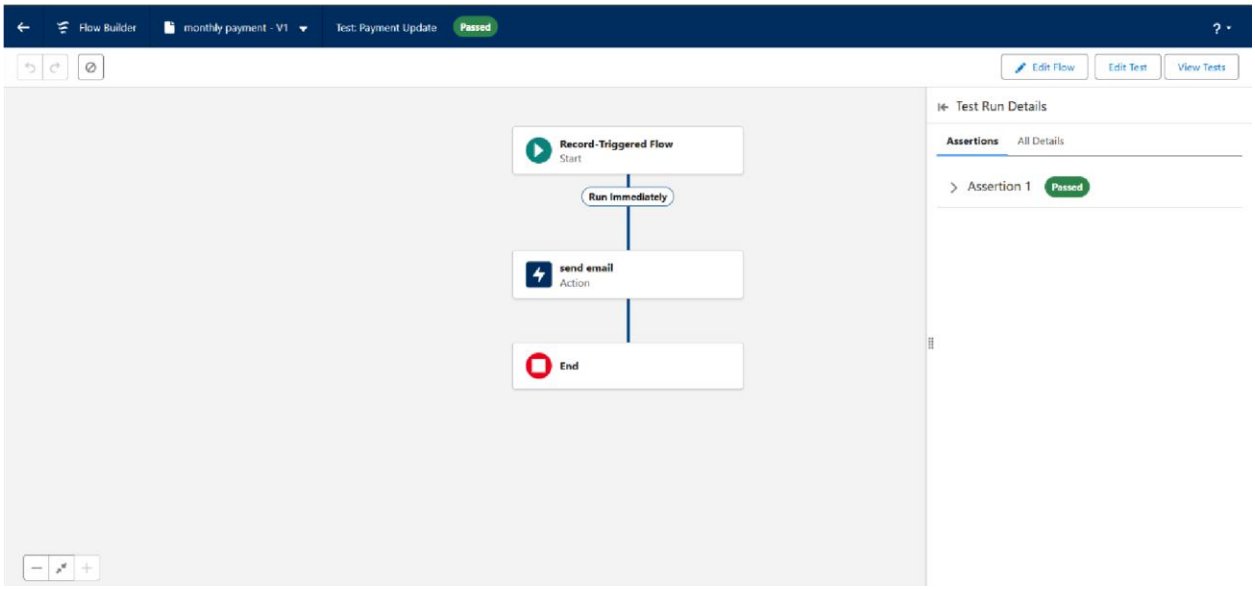



---

# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

- Trigger validation by entering duplicate tenant-property records



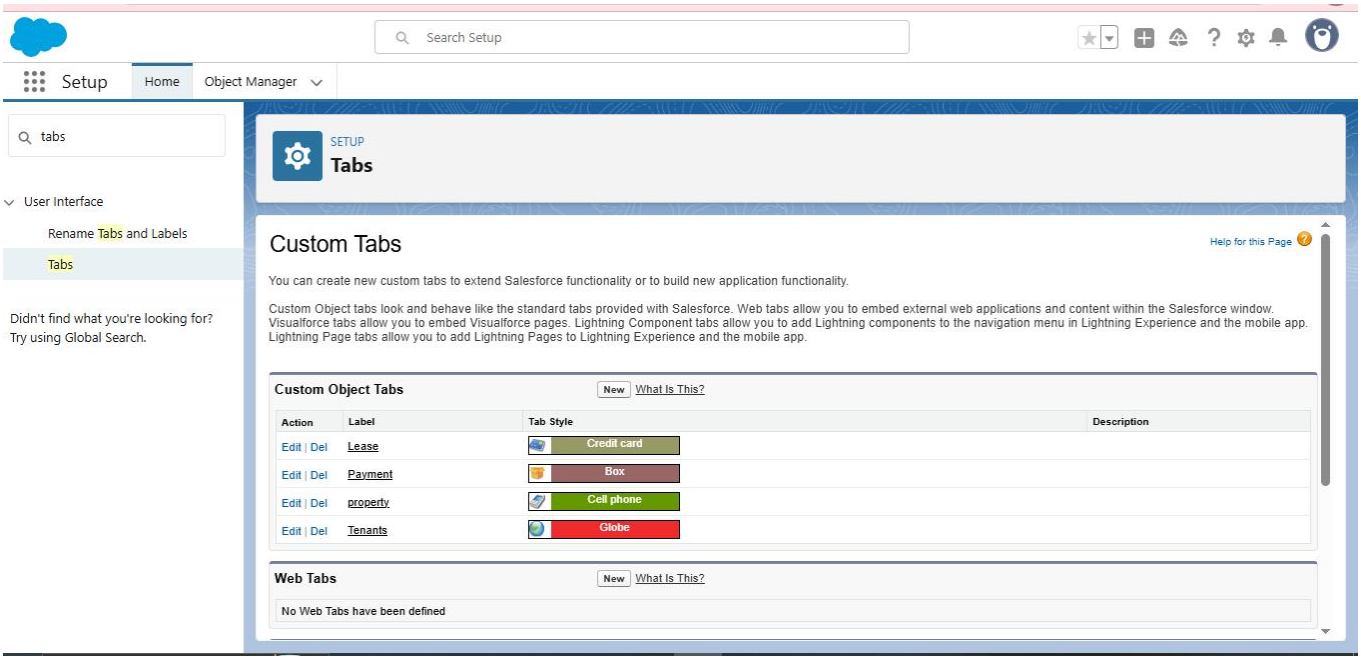

- Approval process validated through email alerts and status updates

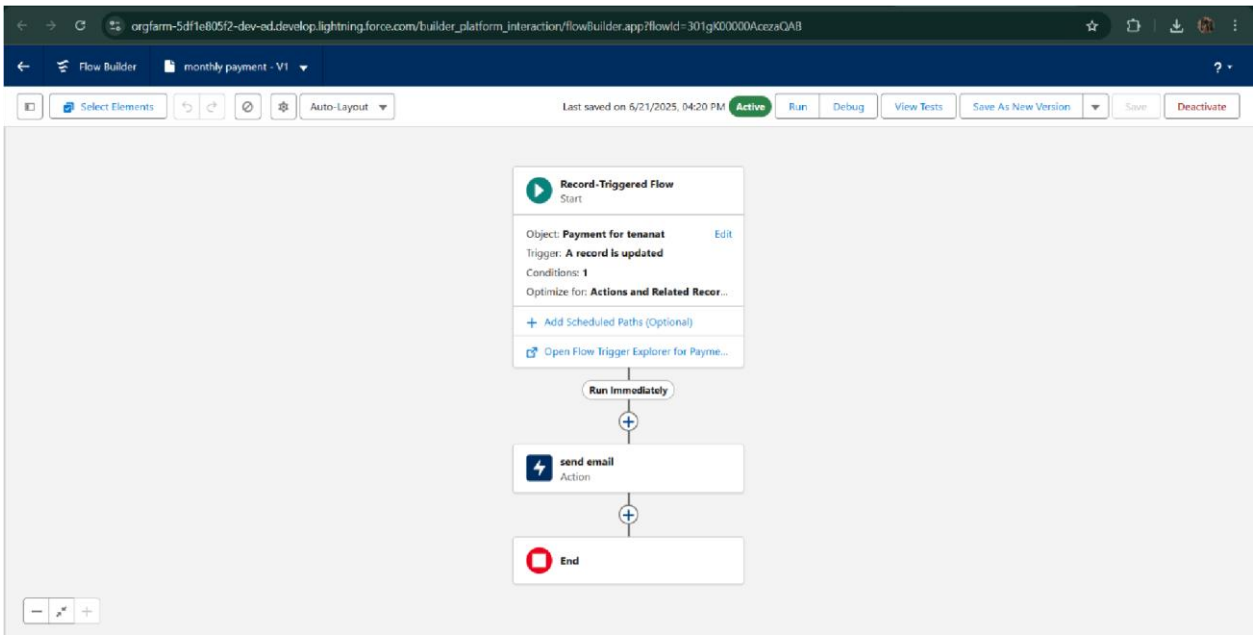
---

# RESULTS

## Output Screenshots

- Tabs for Property, Tenant, Lease, Payment




- Flow runs
- Trigger error messages

# 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});

        }

    }

}
```