

LEASE MANAGEMENT

College Name: RD NATIONAL COLLEGE OF ARTS AND SCIENCE

College Code: bruba

TEAM ID: NM2025TMID28096

TEAM MEMBERS: 5

Team LeaderName: SAIFA SAMREEN M

Email: saifasamreenm@gmail.com

Team Member1: SHERIN FATHIMA A

Email: sherinkuttyss007@gmail.com

Team Member: SAHANA BANU A

Email: sahanabanusaleem@gmail.com

Team Member: PAVITHRA B

Email: sbaskar4855@gmail.com

Team Member: BARANI S

Email: baranibarani562@gmail.com

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



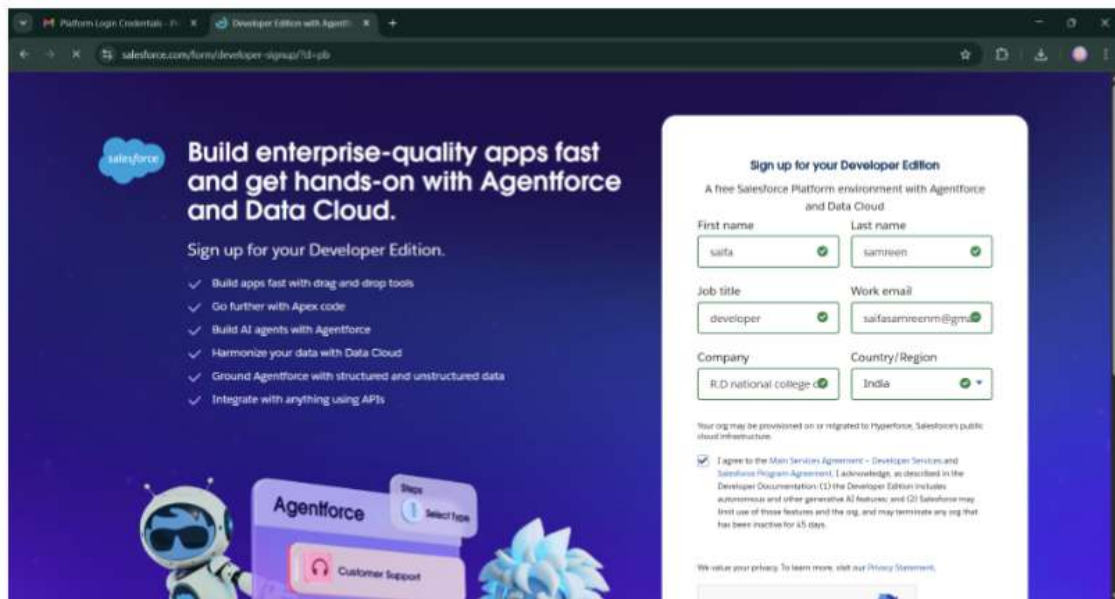
1.2 Purpose

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.

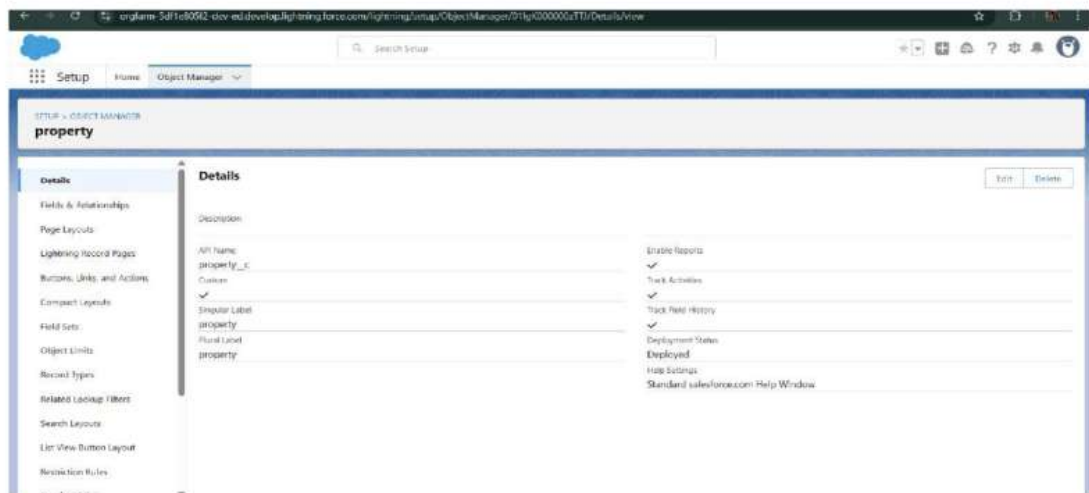
DEVELOPMENT PHASE

Creating Developer Account:

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

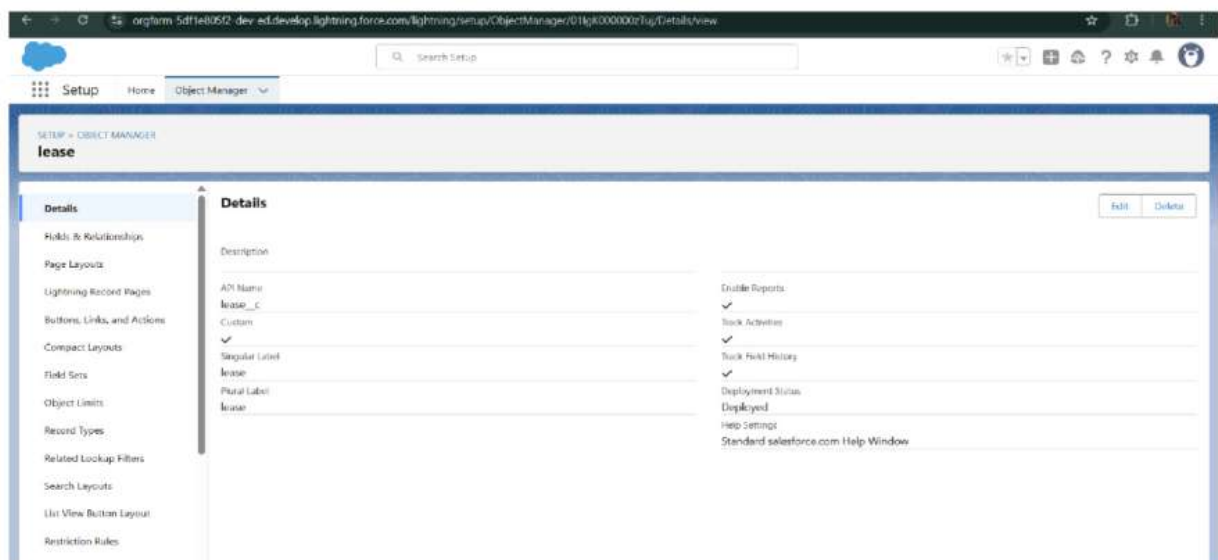
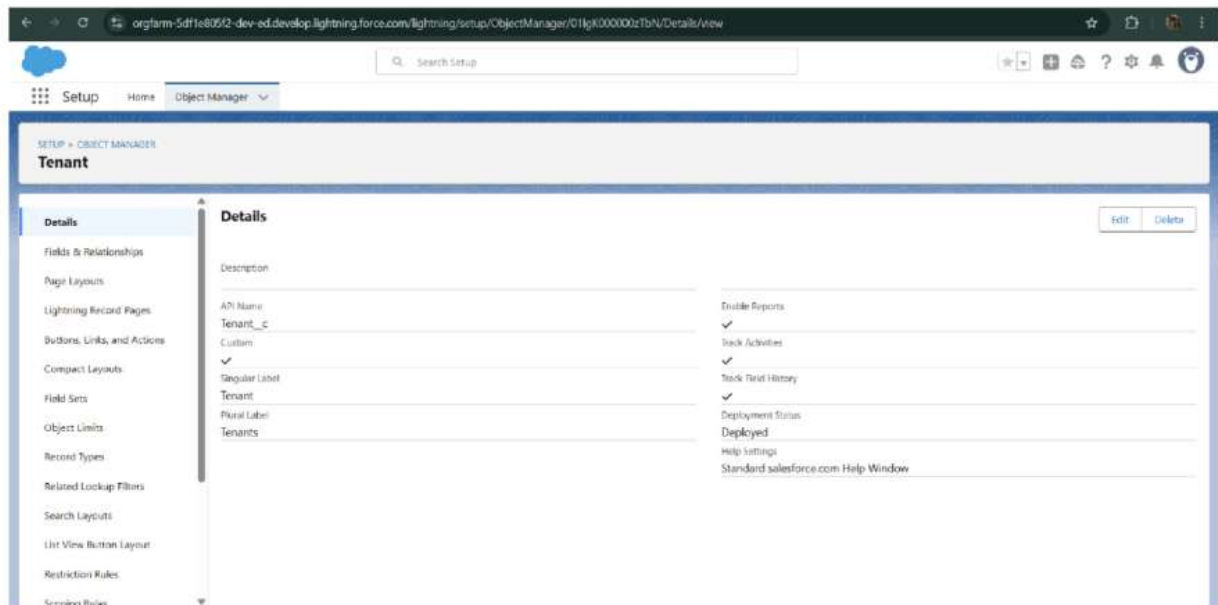


- Created objects: Property, Tenant, Lease, Payment



1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> property
 2. Plural label name >> properties
 3. Enter Record Name Label and Format
Record Name >> property Name
Data Type >> Text
2. Click on Allow reports and Track Field History, Allow Activities, Allow search >> Save.

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> Tenant
 2. Plural label name >> Tenants
 3. Enter Record Name Label and Format



Record Name >> Tenant Name

Data Type >> Text

2. Click on Allow reports and Track Field History, Allow Activities, Allow search >> Save.

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

1. Enter the label name >> lease

2. Plural label name >> lease

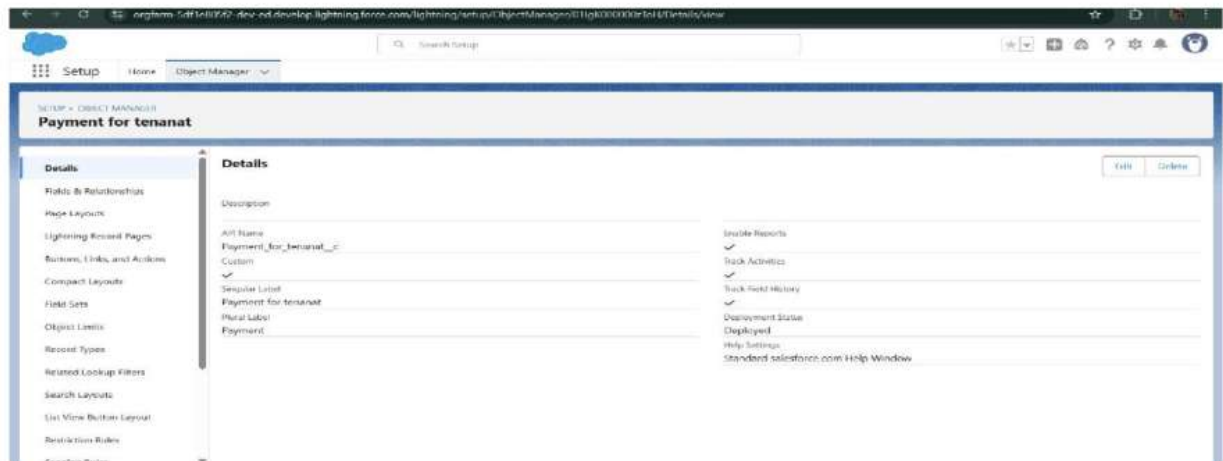
3. Enter Record Name Label and Format

Record Name >> leaseName

Data Type >> Text

2. Click on Allow reports and Track Field History, Allow Activities

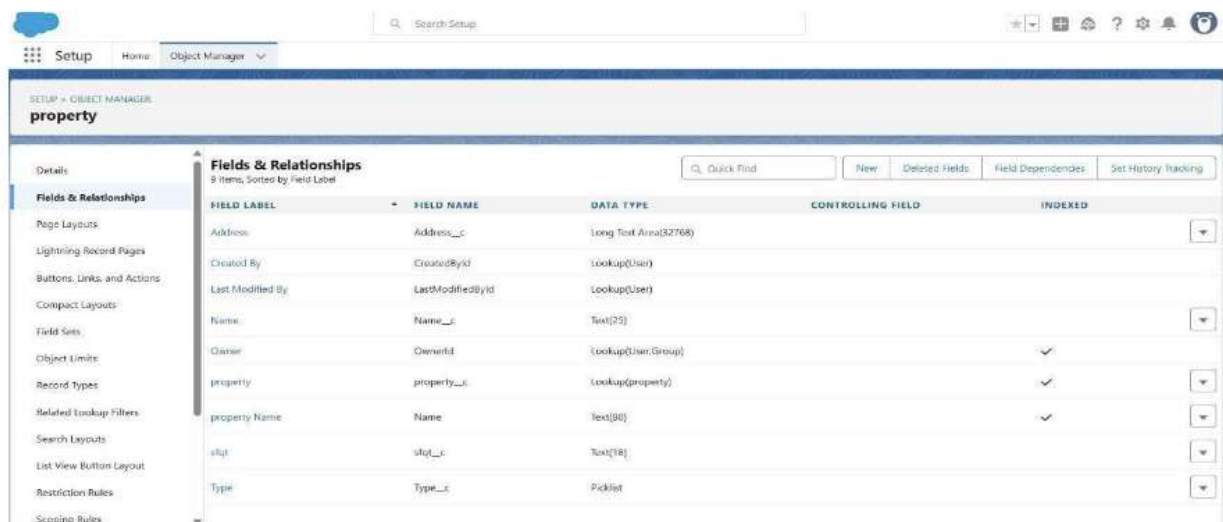
3. Allow search >> Save.



1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> Payment for tenanat
 2. Plural label name >> Payment
 3. Enter Record Name Label and Format
Record Name >> Payment Name
Data Type >> Text
2. Click on Allow reports and Track Field History, Allow Activities
3. Allow search >> Save.

- Configured fields and relationships

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.



Setup

Home

Object Manager

Search Setup

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

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
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)		✓

Setup

Home

Object Manager

Search Setup

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

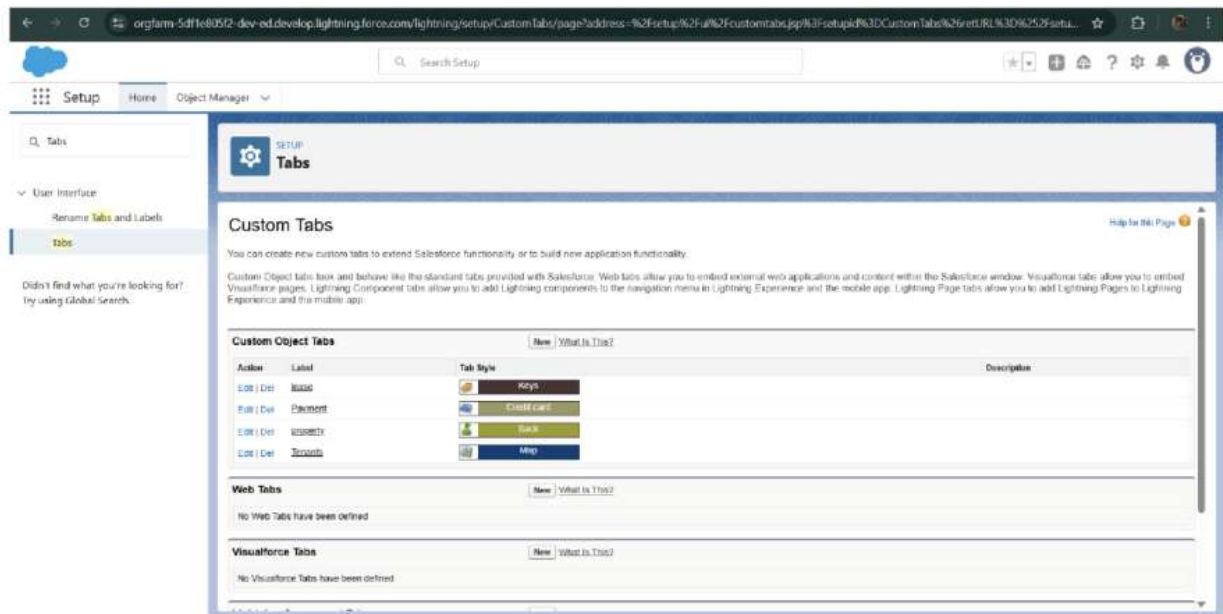
Set History Tracking

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		

Tabs

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment



What is Tab : A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Web Tabs Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4. Lightning Component Tabs

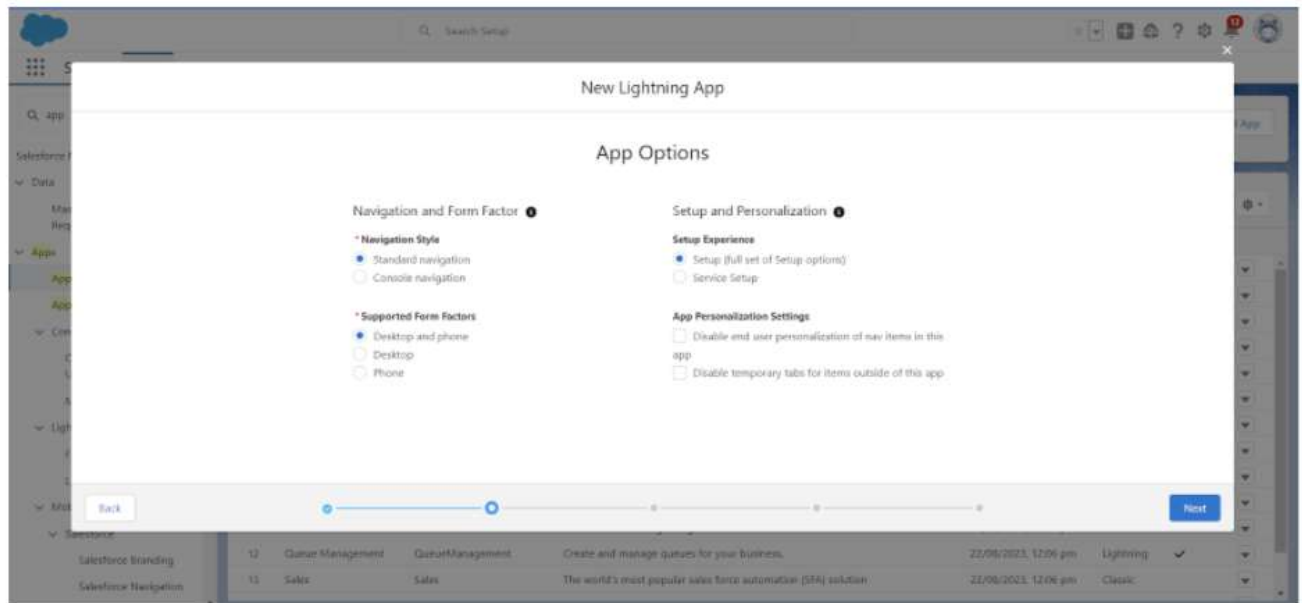
Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

5. Lightning Page Tabs

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

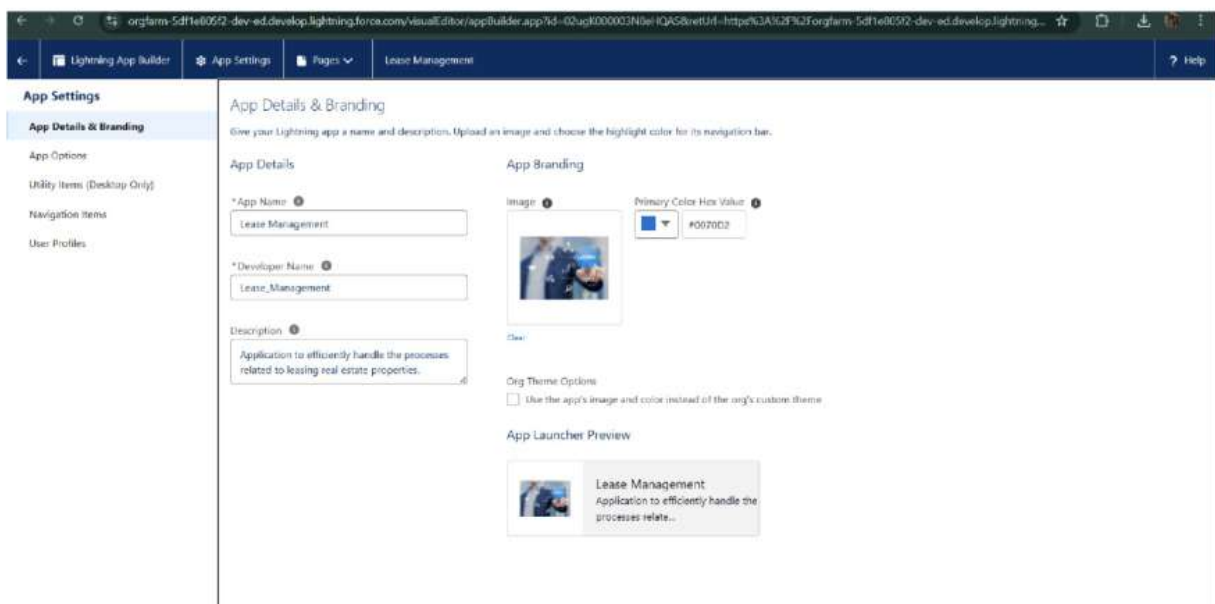
Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

The Lightning App



- Developed Lightning App with relevant tabs

Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app.



← 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 more 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

Create

Type to filter list...

Accounts

Activation Targets

Activations

All Sites

Alternative Payment Methods

Analytics

App Launcher

Appointment Categories

Appointment Invitations

Approval Requests

Selected Items

Payment

Tenants

property

lease

← 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

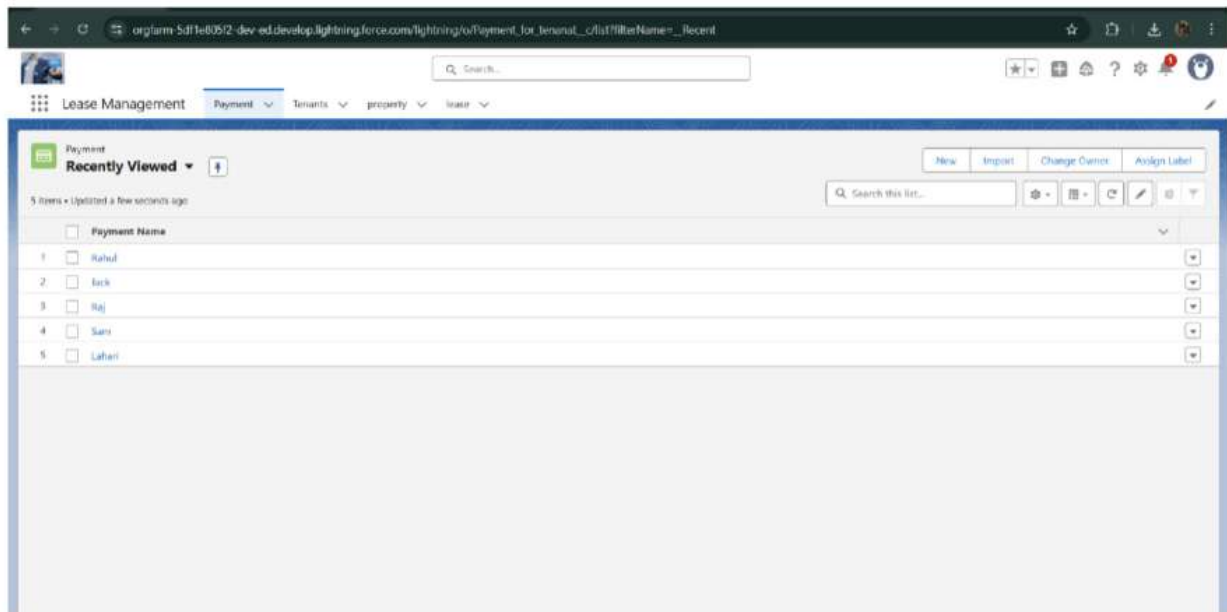
Custom Sales Profile

Custom Support Profile

Customer Community Login User

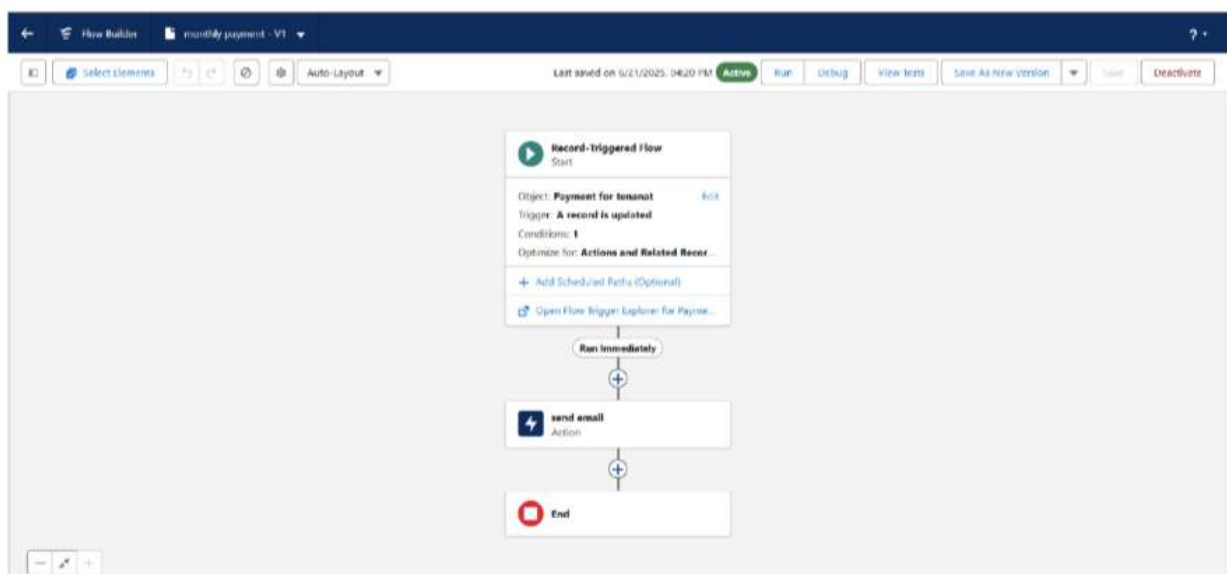
Selected Profiles

System Administrator



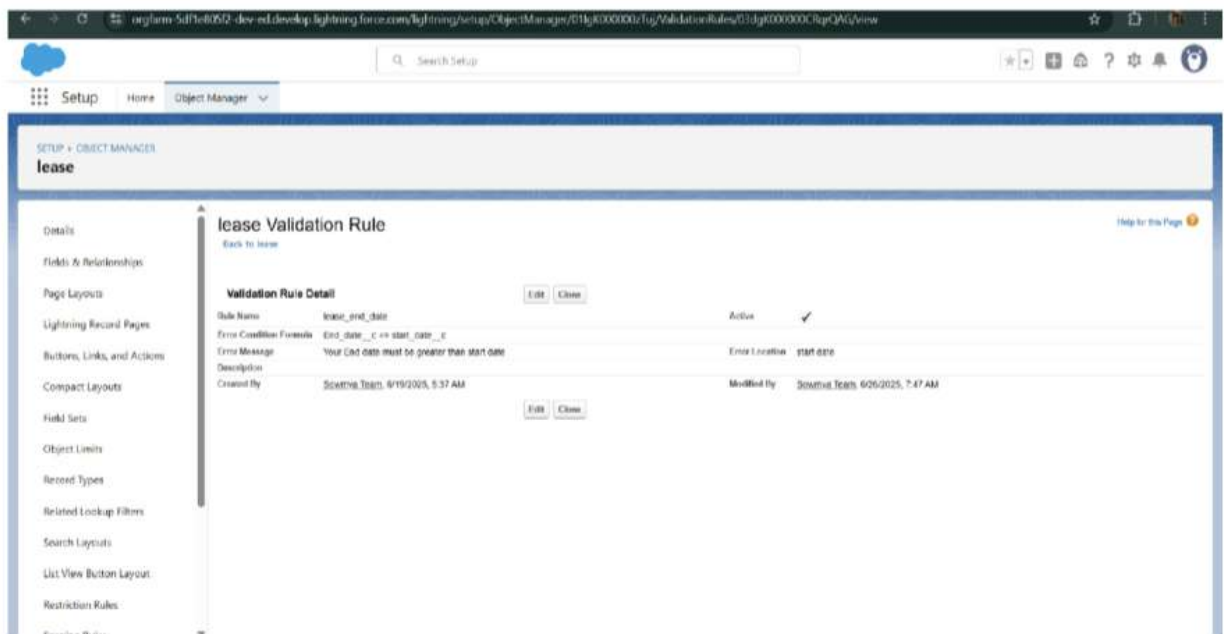
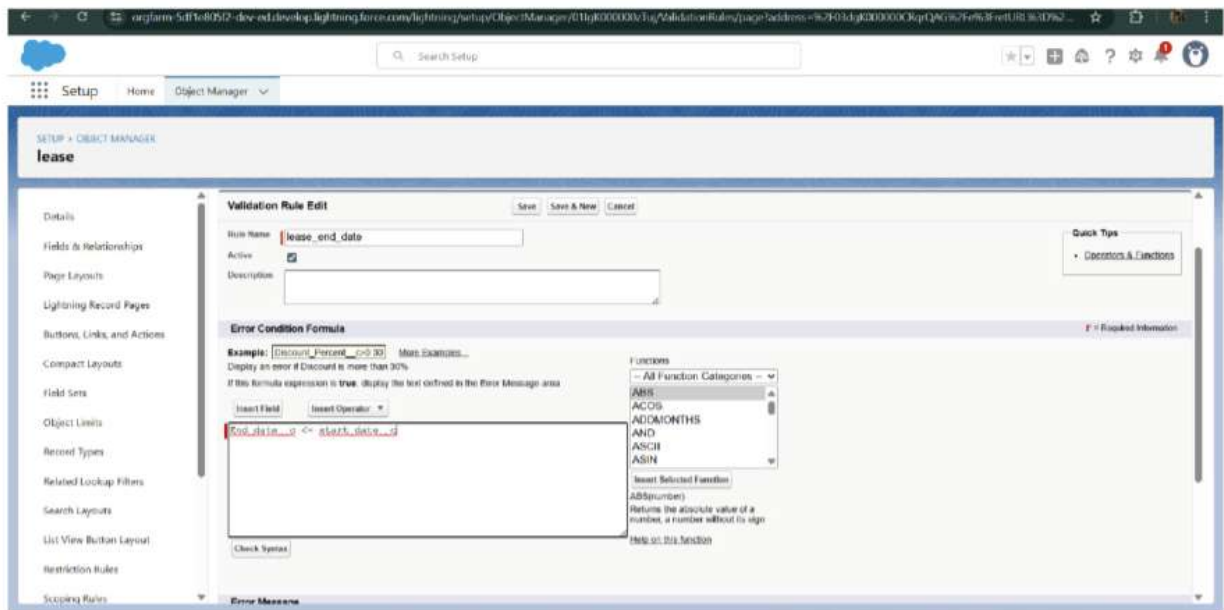
- Implemented Flows for monthly rent and payment success

In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data. Flow Builder is the declarative interface used to build individual flows.



- To create a validation rule to a Lease Object

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.



- Added Apex trigger to restrict multiple tenants per property

The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

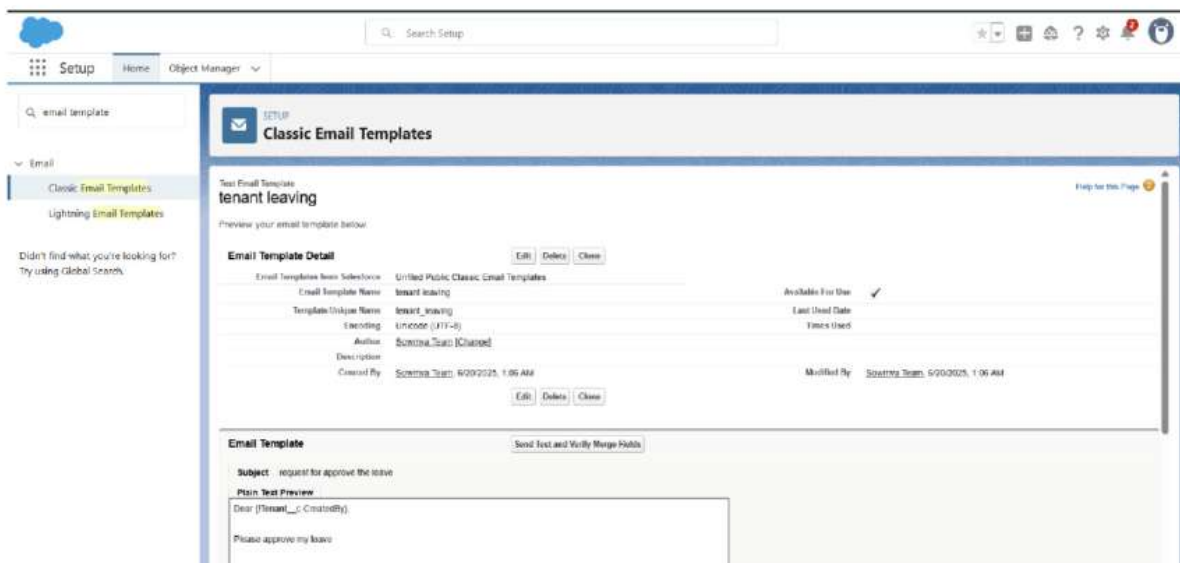
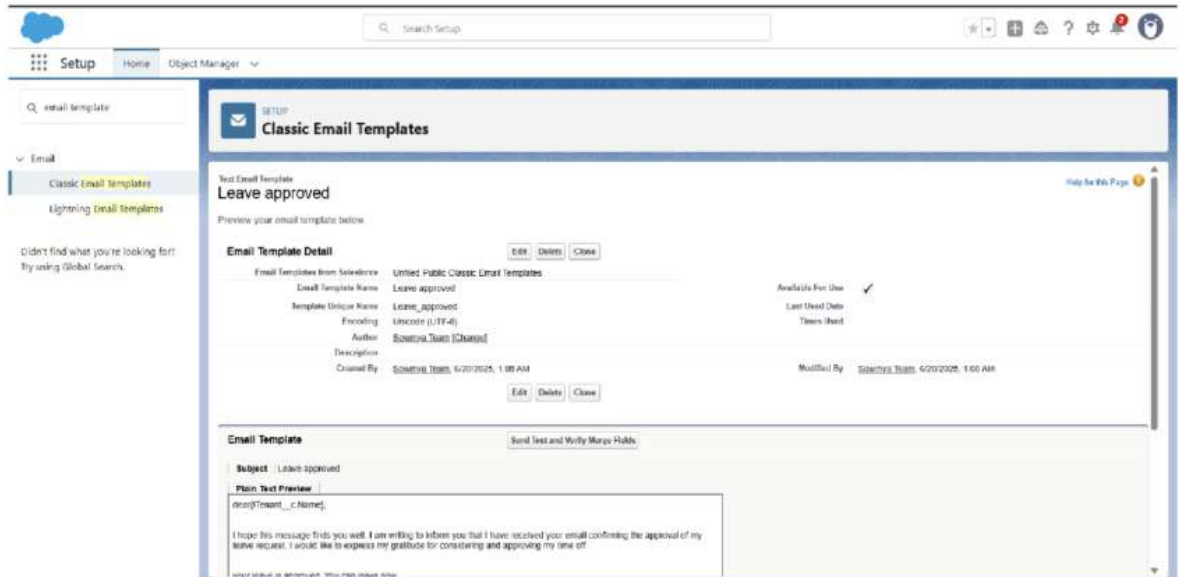
```
testHandler.apex
Code Coverage: None API Version: 59

1 public class testHandler {
2     public static void preventInsert(List<Tenant__c> newList) {
3         Set<Id> existingPropertyIds = new Set<Id>();
4         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
5             existingPropertyIds.add(existingTenant.Property__c);
6         }
7
8         for (Tenant__c newTenant : newList) {
9
10            if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
11                newTenant.addError('A tenant can have only one property');
12            }
13        }
14    }
15 }
```

- Scheduled monthly reminder emails using 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    public static void sendMonthlyEmails() {
14
15        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
16
17        for (Tenant__c tenant : tenants) {
18
19            String recipientEmail = tenant.Email__c;
20
21            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
22
23            String emailSubject = 'Reminder: Monthly Rent Payment Due';
24
25            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
26
27            email.setToAddresses(new String[] {recipientEmail});
28
29            email.setSubject(emailSubject);
30
31            email.setPlainTextBody(emailContent);
32        }
33    }
34 }
```

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



We use email templates to increase productivity and ensure consistent messaging. Email templates with merge fields let you quickly send emails that include field data from Salesforce records like contacts, leads, or opportunities. You can use email templates when emailing groups of people—with list email or mass email—or just one person.

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.

Classic Email Templates

Test Email Template

Leave rejected

Help for this Page

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 Team (Change)

Description

Created By

System Team

6/20/2025, 1:11 AM

Modified By

System 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 lease.
your lease 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.

Classic Email Templates

Test Email Template

Tenant Email

Help for this Page

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 Team (Change)

Description

Created By

System Team

6/20/2025, 1:12 AM

Modified By

System Team

6/20/2025, 1:12 AM

Email Template

Send Test and Verify Merge Fields

Subject

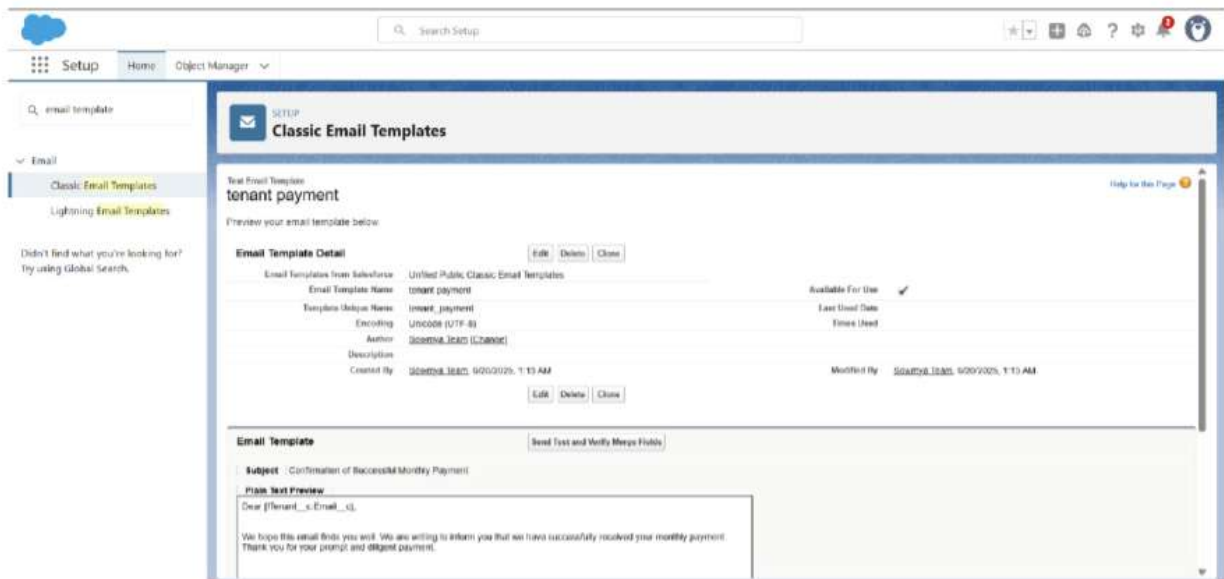
Urgent: Monthly Rent Payment Reminder

Plain Text Preview

Dear {Tenant__c.Name},

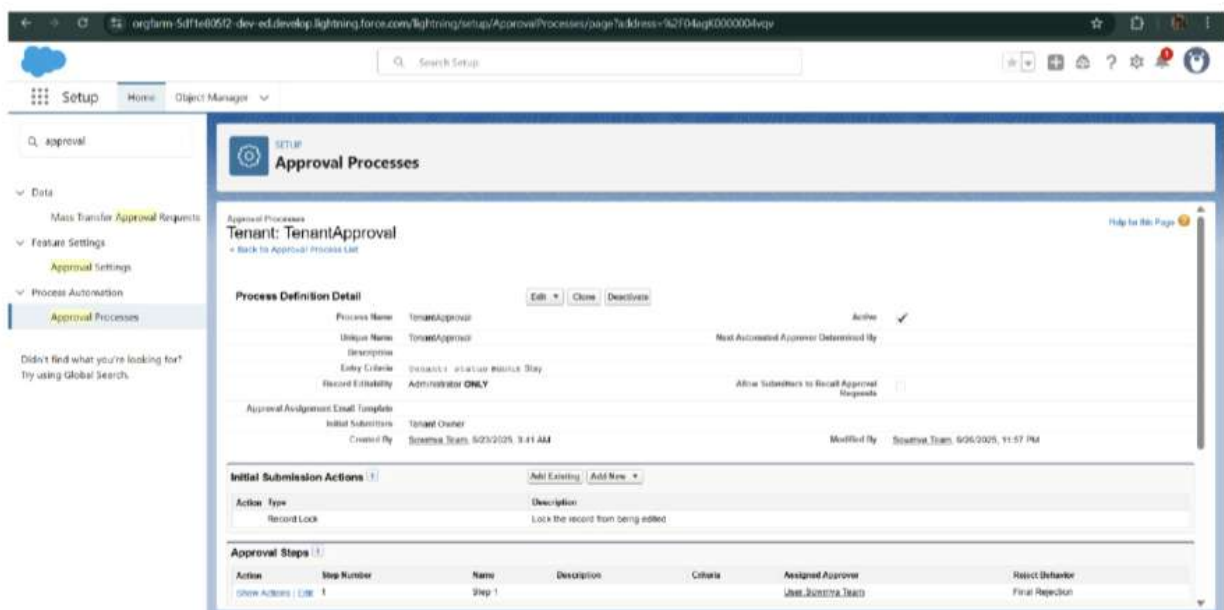
I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

This communication is a friendly reminder to please pay your monthly rent on time. Please contact us if you have any questions or need assistance.



● Approval Process creation

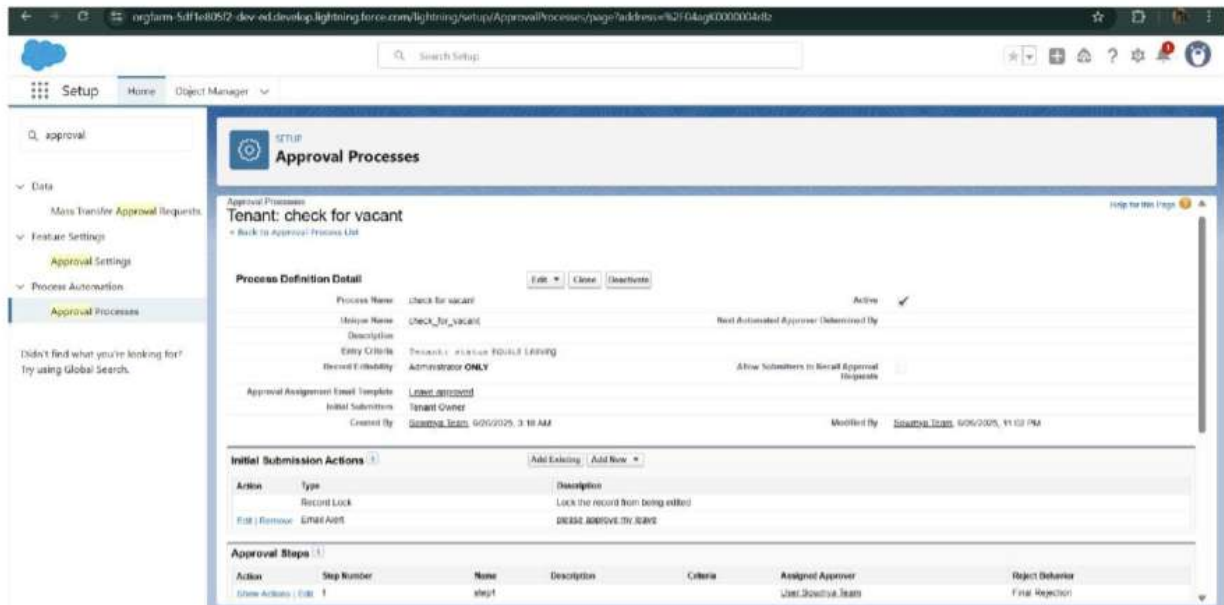
For Tenant Leaving:



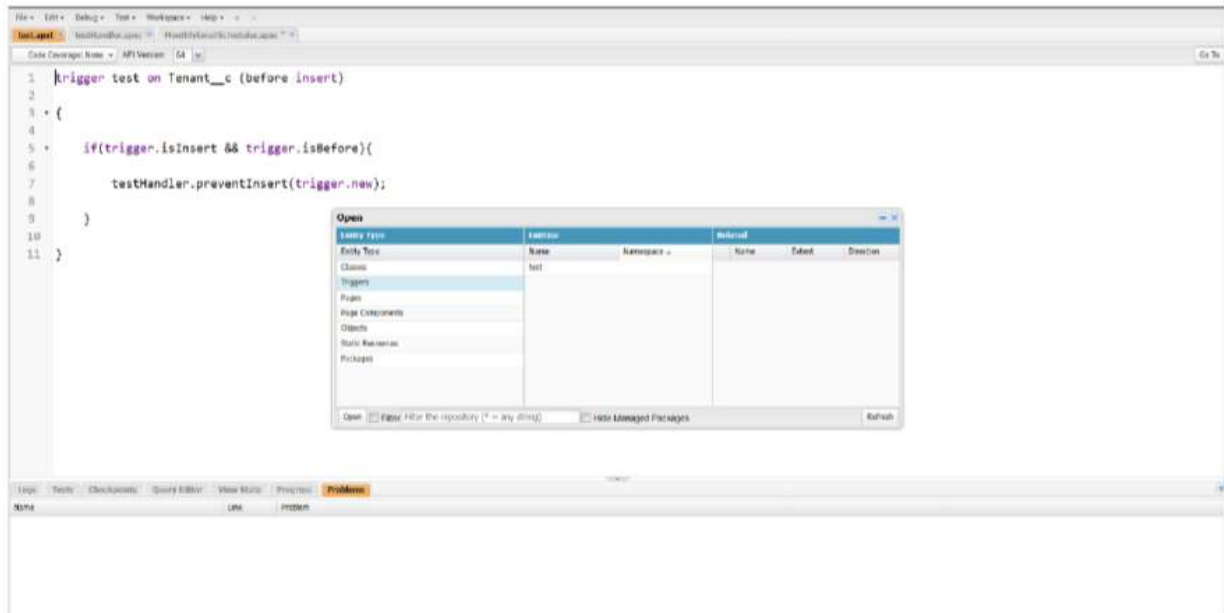
Approval Process In Salesforce:

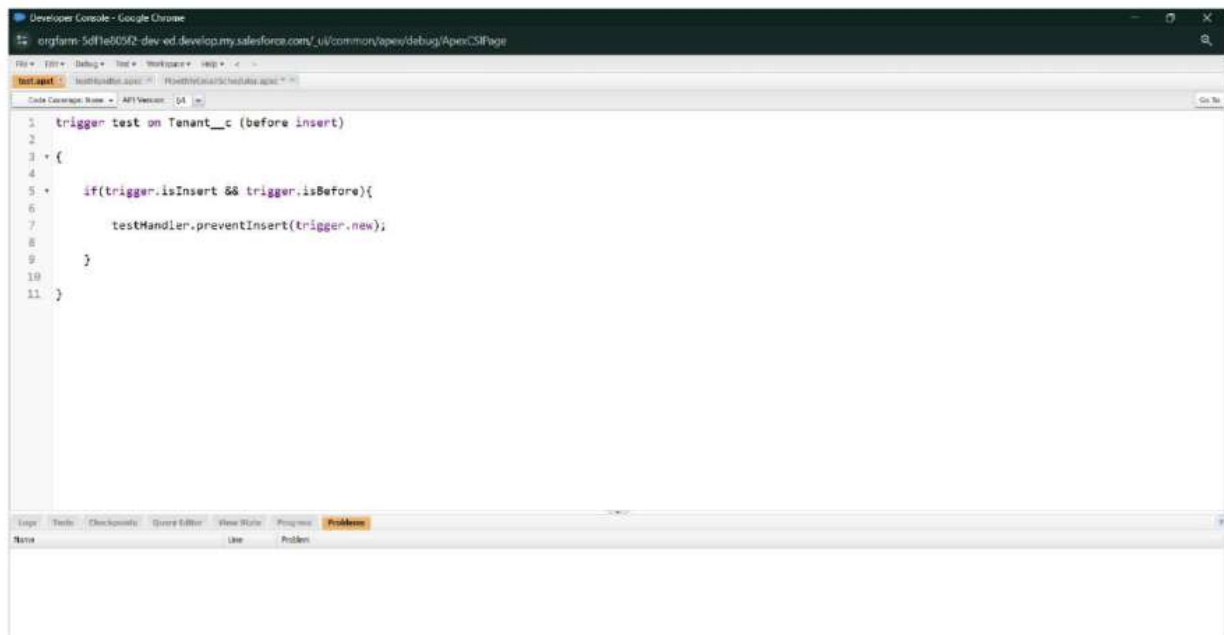
The Approval Process is an automated process that an org uses to approve records in Salesforce. For example, When In the organization, someone is not able to decide a particular thing then he can ask someone else for approval. So, for such frequent cases or situations, one can define the approval process. So, Users can take benefit of such an approval process whenever needed.

For Check for Vacant:

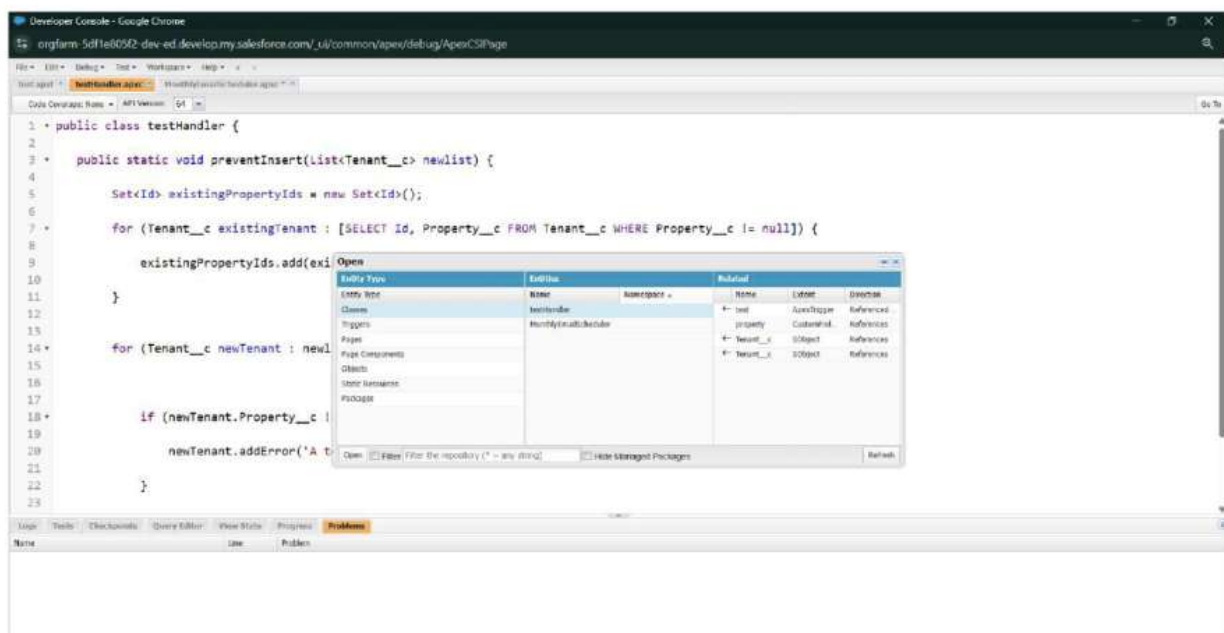


- Apex Trigger
- ## Create an Apex Trigger



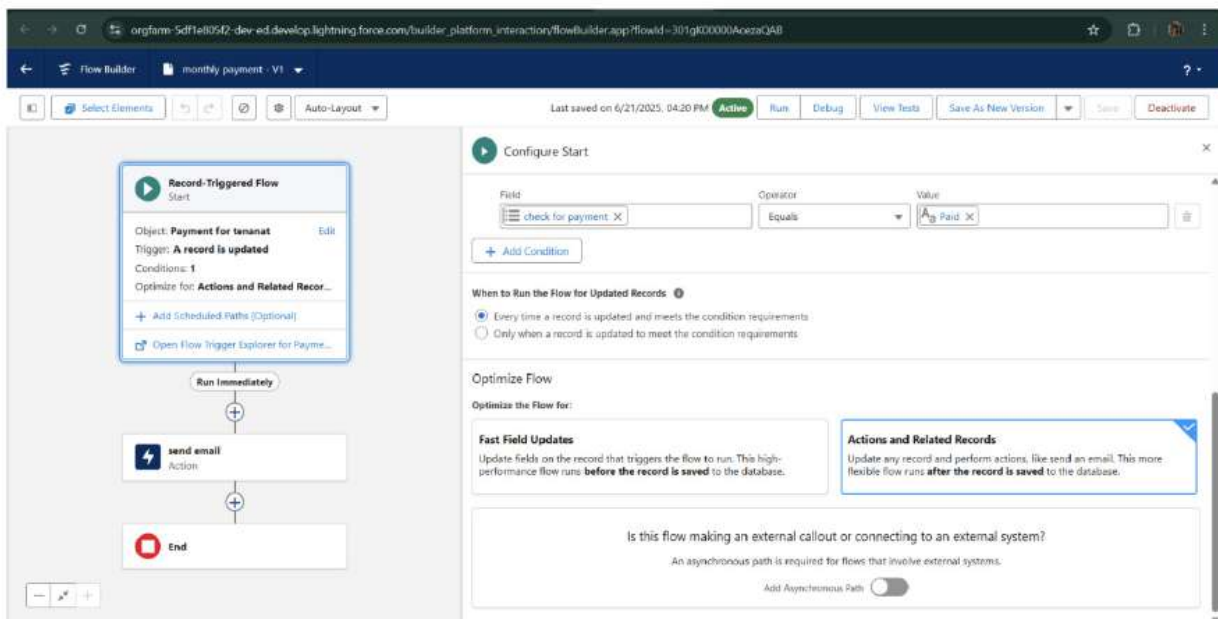


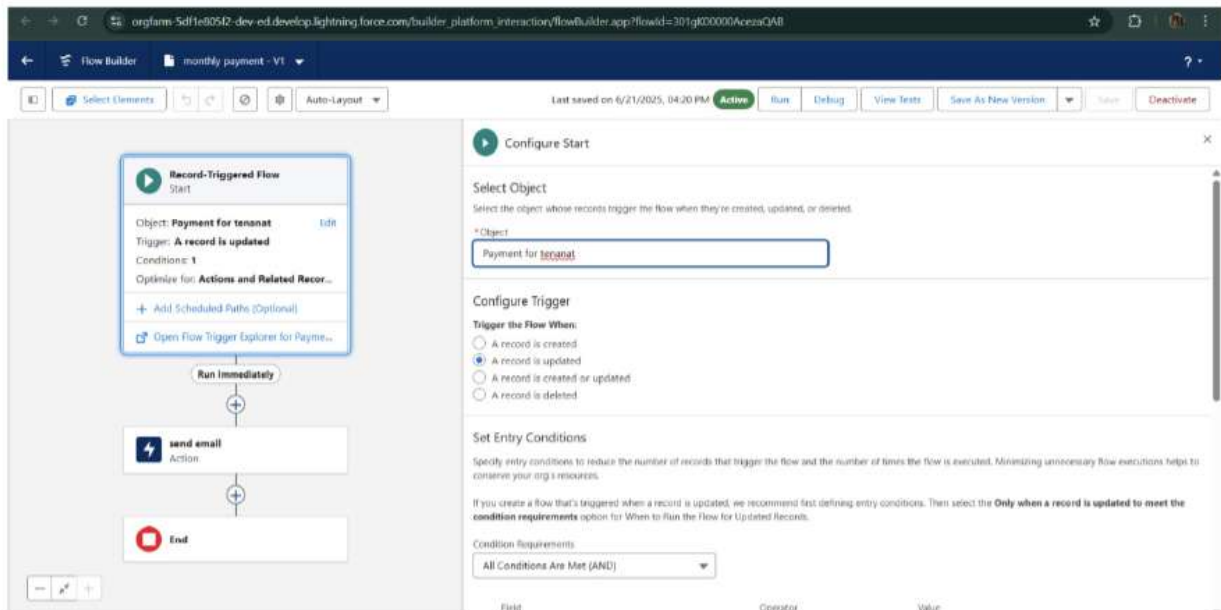
Create an Apex Handler class



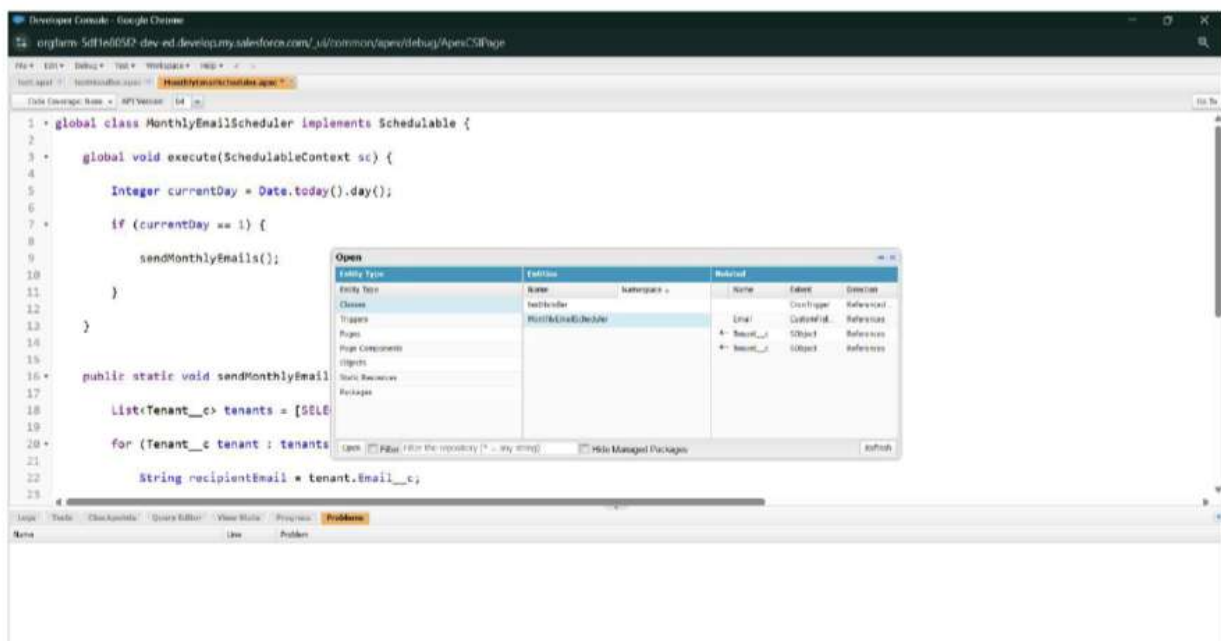
```
1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17
18             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
19
20                 newTenant.addError('A tenant can have only one property');
21
22             }
23
24         }
25     }
26 }
```

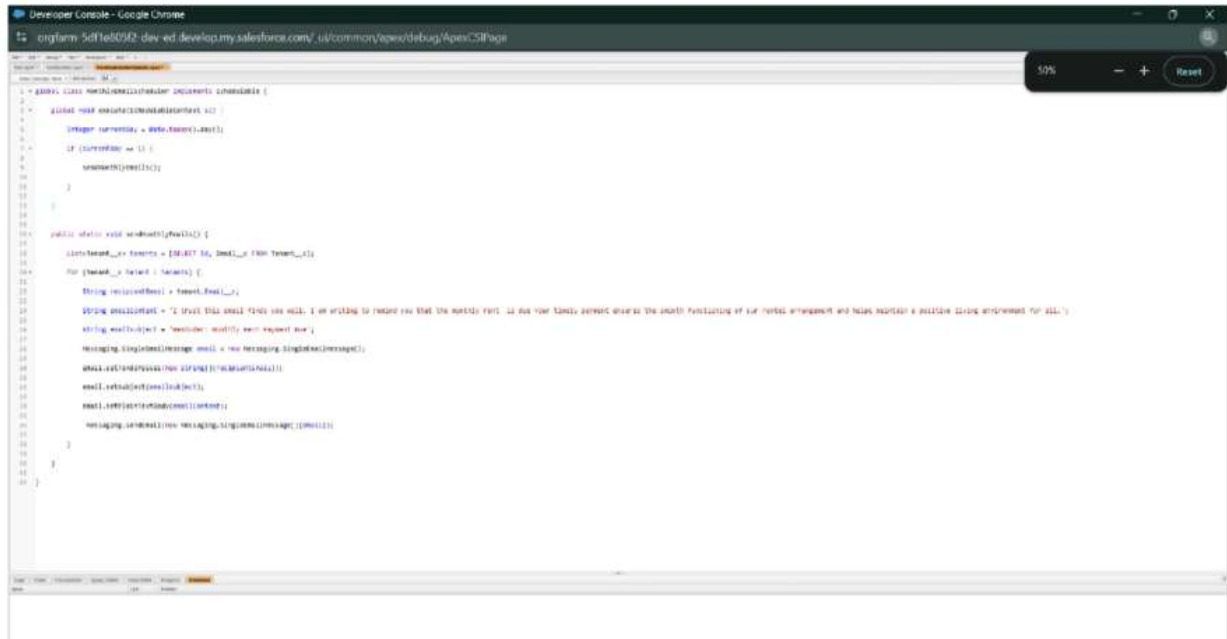
• FLOWS





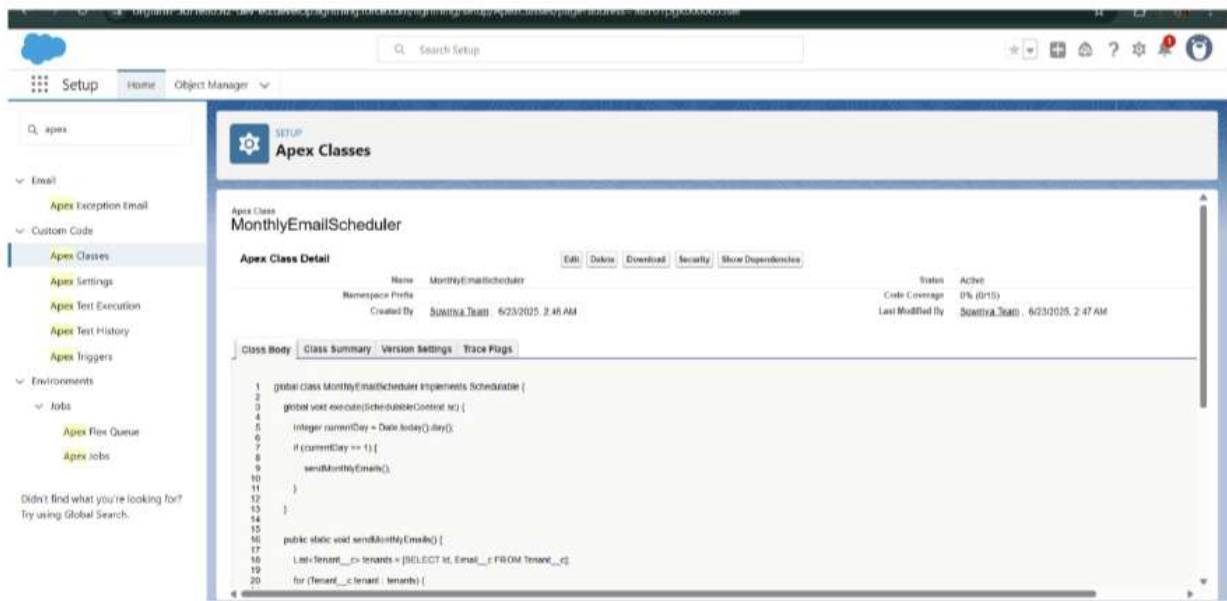
- Schedule class:
Create an Apex Class





```
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             sendMonthlyEmail();
9         }
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            // TODO: Replace this email body with real content. I am writing to remind you that the monthly rent is due. You have 10 days to pay it. If you don't, we will have to start legal action against you. We are sorry for the inconvenience and hope you will pay it as soon as possible.
24            String emailBody = "Monthly rent is due. Please pay it as soon as possible. If you don't, we will have to start legal action against you. We are sorry for the inconvenience and hope you will pay it as soon as possible.";
25
26            Messaging.SingleTextMessage email = new Messaging.SingleTextMessage();
27
28            email.setFromAddressFromSystemEmail("Salesforce");
29
30            email.setToAddresses(new List<String> {recipientEmail});
31
32            email.setSubject("Monthly rent is due");
33
34            Messaging.sendEmail(new Messaging.SingleTextMessage(email));
35
36        }
37    }
38 }
```

Schedule Apex class



Setup

Search Setup

Setup

Apex Classes

Apex Class

MonthlyEmailScheduler

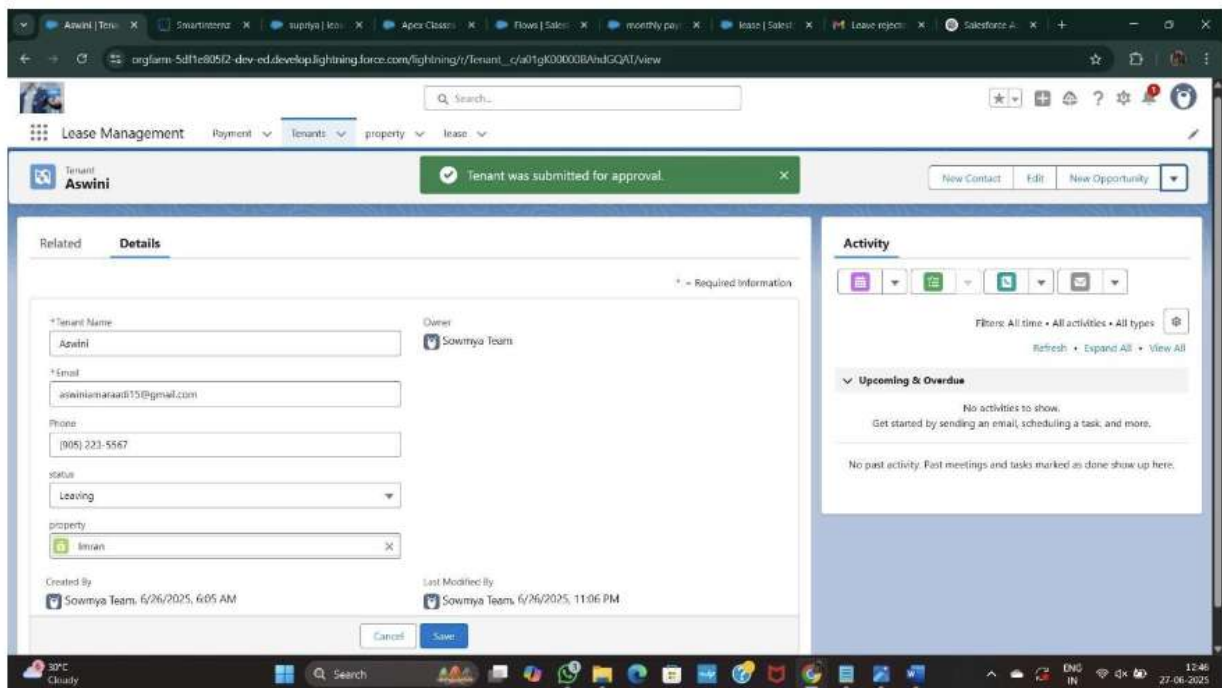
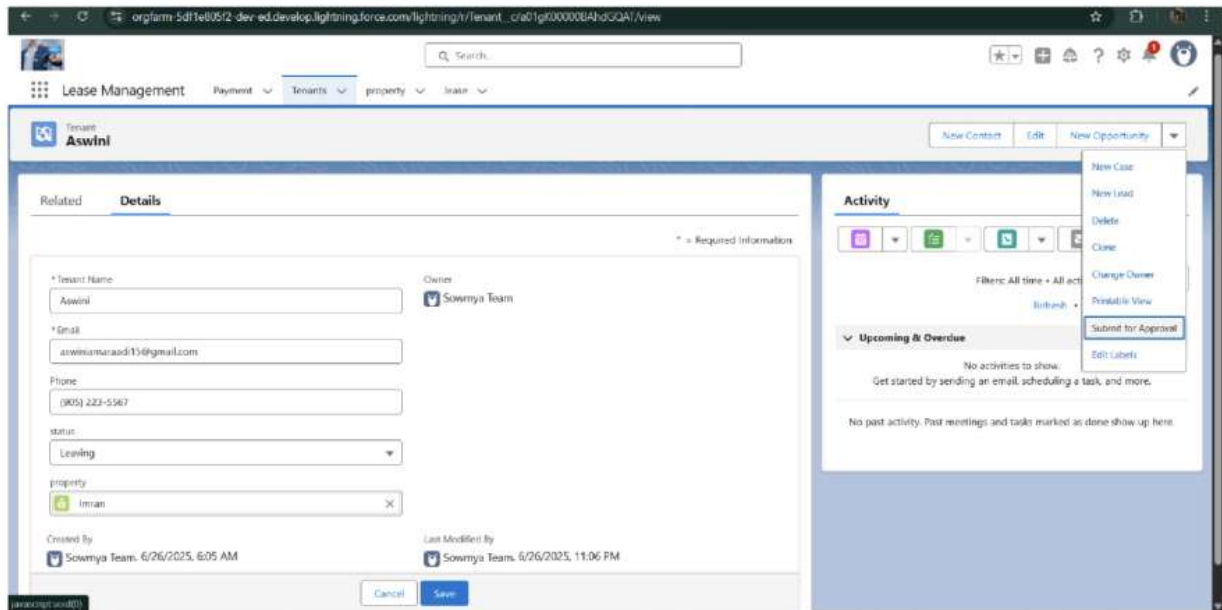
Apex Class Detail

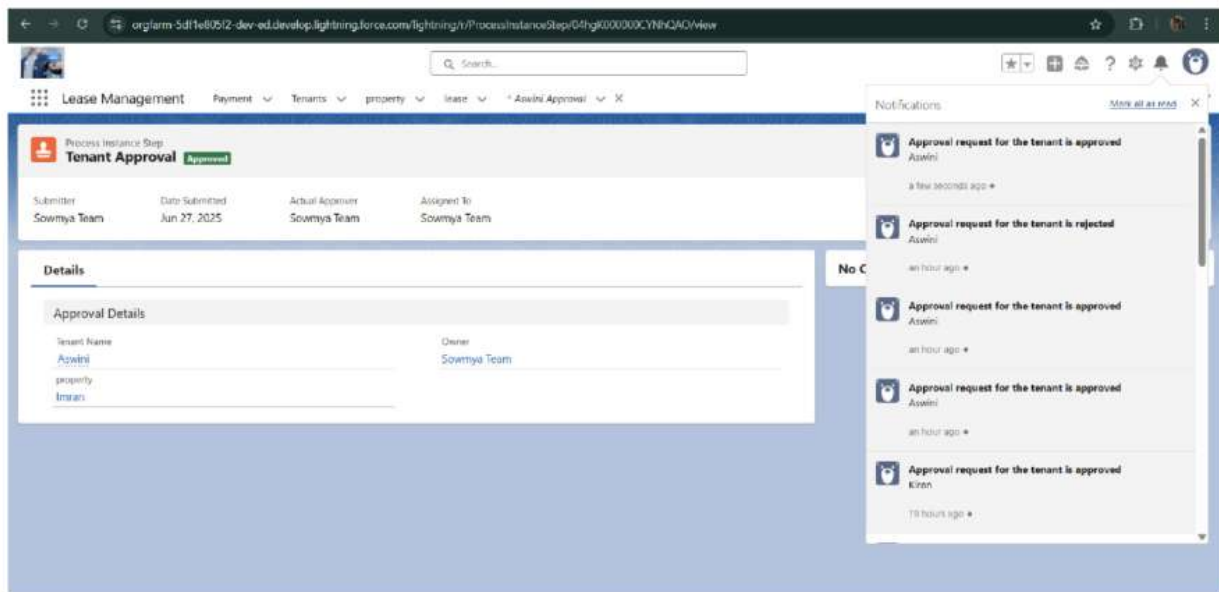
Edit | Delete | Download | Security | Show Dependencies

Name	MonthlyEmailScheduler	Status	Active
Namespace Prefix		Code Coverage	0% (0/15)
Created By	Sudhita Thakur	Last Modified By	Sudhita Thakur
	6/23/2025, 2:45 AM		6/23/2025, 2:47 AM

Class Body | Class Summary | Version Settings | Trace Logs

```
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             sendMonthlyEmail();
9         }
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            // TODO: Replace this email body with real content. I am writing to remind you that the monthly rent is due. You have 10 days to pay it. If you don't, we will have to start legal action against you. We are sorry for the inconvenience and hope you will pay it as soon as possible.
24            String emailBody = "Monthly rent is due. Please pay it as soon as possible. If you don't, we will have to start legal action against you. We are sorry for the inconvenience and hope you will pay it as soon as possible.";
25
26            Messaging.SingleTextMessage email = new Messaging.SingleTextMessage();
27
28            email.setFromAddressFromSystemEmail("Salesforce");
29
30            email.setToAddresses(new List<String> {recipientEmail});
31
32            email.setSubject("Monthly rent is due");
33
34            Messaging.sendEmail(new Messaging.SingleTextMessage(email));
35
36        }
37    }
38 }
```

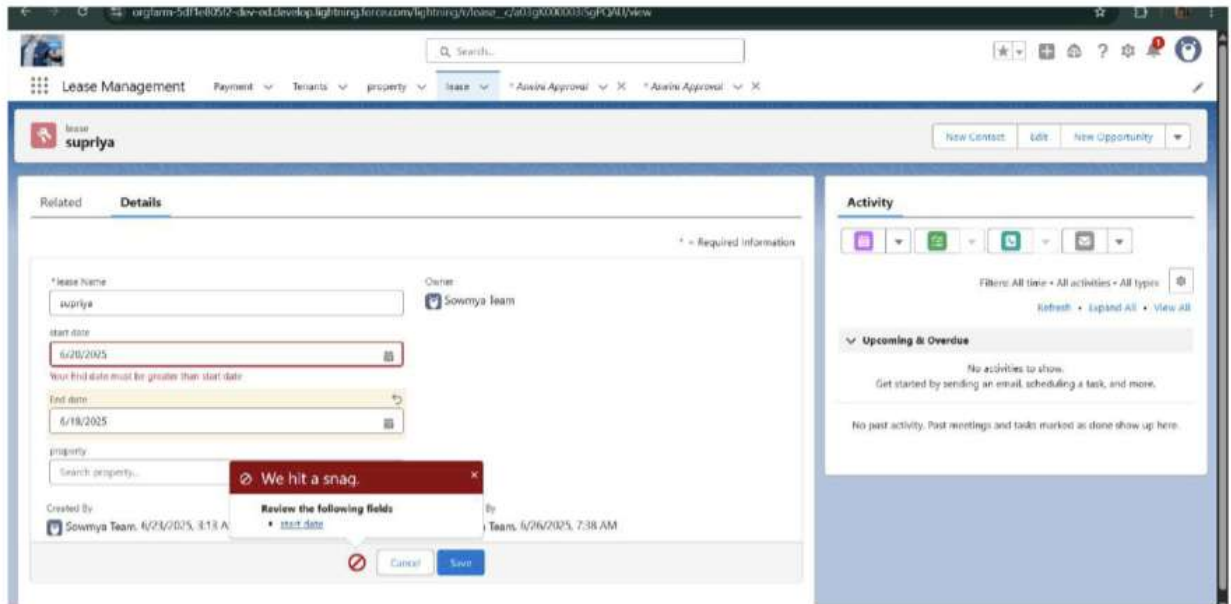




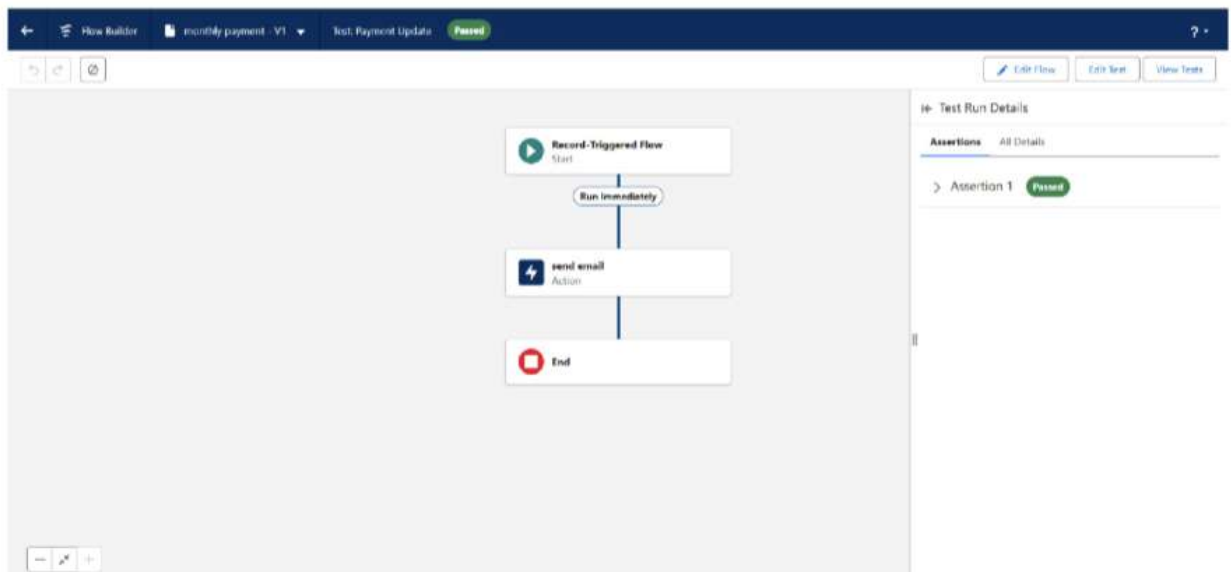
FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Trigger validation by entering duplicate tenant-property records



- Test flows on payment update



- Approval process validated through email alerts and status updates

Lease Management

Payment

Tenants

property

lease

niranjan Approval

Search...

Tenant

niranjan

Related

Details

* Tenant Name

niranjan

Owner

Sowmya Team

* Email

niranjan1200@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, 3:58 AM

Cancel

Save

Notifications

Mark all as read

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, 1:01 PM

Lease Management

Payment

Tenants

property

lease

Search...

Tenant

niranjan

New Contact

Edit

New Opportunity

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

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

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, 5:39 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, 2:55 AM	submitted	Sowmya Team

View All

Payment (2)

New

Payment Name

Jack

Bahul

View All

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

Save Cancel

Job Name MonthlyEmailScheduler

Apex Class MonthlyEmailScheduler

Schedule Apex Execution

Frequency ☐ Weekly ☒ Monthly

☒ On day 1 of every month ☐ On the 1st Sunday of every month

Start 04/12/2023 [04/12/2023]

End 04/01/2024 [04/12/2023]

Preferred Start Time 9:00 am

Exact start time will depend on job queue activity.

Save Cancel

- Testing the approval process

Submit for Approval

Comments

Leaving

Cancel Submit

Manne Niranjana Reddy, 05/12/2023, 10:18 am

- Approval For Leave

Approval Request

Tenant Approval

Pending

Approve

Reject

Reassign

Submitter

Manne Niranjan Reddy

Date Submitted

05-Dec-2023

Actual Approver

Manne Niranjan Reddy

Assigned To

Manne Niranjan Reddy

Details

Approval Details

Tenant Name

Niranjan

Submitter Comments

Manne Niranjan Reddy

Leaving

05-Dec-2023, 10:25:01 am

★ ▼


+

🔒

?

⚙️


🔔



Notifications

[Mark all as read](#)

✕



Approval request for the tenant is approved

Niranjan

a few seconds ago ●

- Leave Approval Notification

Advantages

1. Centralized Data

All lease details (tenants, contracts, payment schedules, renewals, assets) are stored in one system.

Reduces dependency on multiple spreadsheets or siloed systems.

2. Automation

Automated reminders for rent due dates, lease renewals, and compliance deadlines.

Minimizes manual errors and missed follow-ups.

3. Customization & Flexibility

Salesforce can be tailored with custom objects, workflows, and approval processes to match specific lease business needs.

Integration with accounting or property management tools is possible.

4. Reporting & Analytics

Real-time dashboards for revenue, occupancy rates, lease expirations, and performance tracking.

Better forecasting for renewals and payments.

5. Improved Collaboration

Teams (sales, finance, operations) can access the same data with role-based security.

Enhances communication across departments.

Disadvantages

1. High Implementation Cost

Custom development, consulting, and licensing can be expensive compared to standalone lease software.

2. Complex Setup

Requires proper planning and skilled Salesforce developers/administrators.

Not plug-and-play—needs customization for industry-specific processes.

3. Training Requirement

End-users need training to use Salesforce efficiently.

Non-technical users may initially find it overwhelming.

4. Ongoing Maintenance

Customizations require continuous updates and testing during Salesforce seasonal releases.

May need dedicated admin support.

5. Integration Challenges

If leases need to connect with ERP, accounting, or external property systems, integration can be complex and costly.

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');

        }

    }

}

```

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});
}
}
}
```