PROJECT TITLE

College Name: Nandha arts and science college

College Code: BRU4J

TEAM ID:

TEAM MEMBERS:

Team LeaderName: Manimegalai D

Email: manimegalaits3257@gmail.com

Team Member1: Shasmathi B

Email: Shasmathi111@gmail.com

Team Member: Dhanusri P

Email: Dhanusri2830@gmail.com

1.INTRODUCTION

1.1ProjectOverview

TheLease ManagementSystem is aSalesforce-based application designed to streamline the processes associated with leasing real estate properties. It handles tenant management, lease

contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.

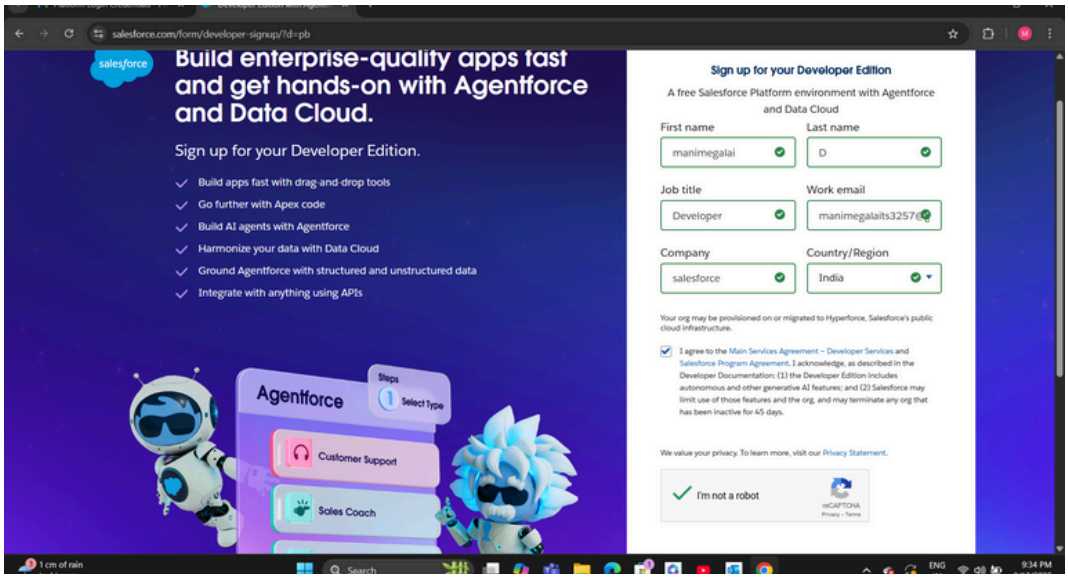
1.2 Purpose

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

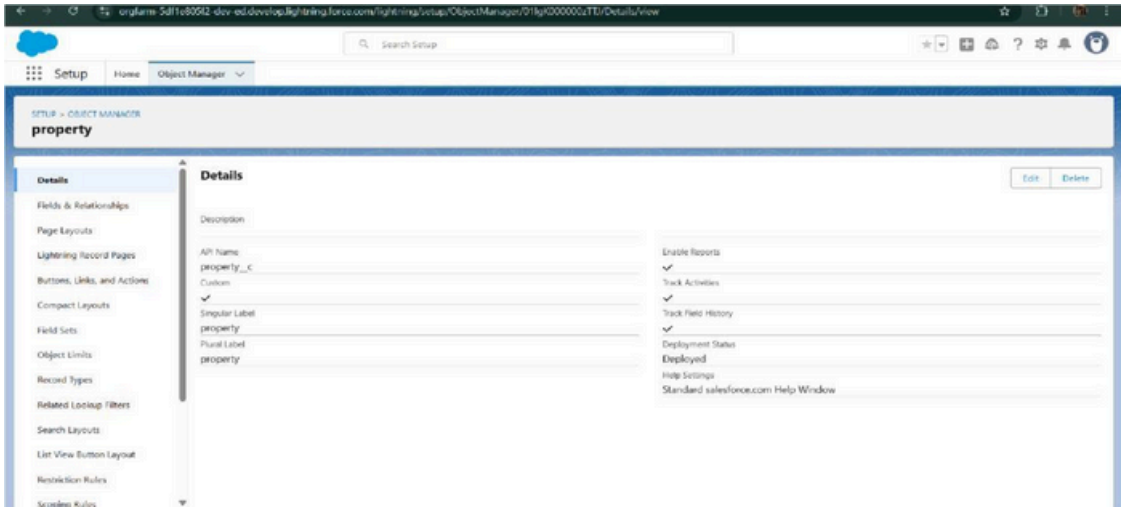
DEVELOPMENT PHASE

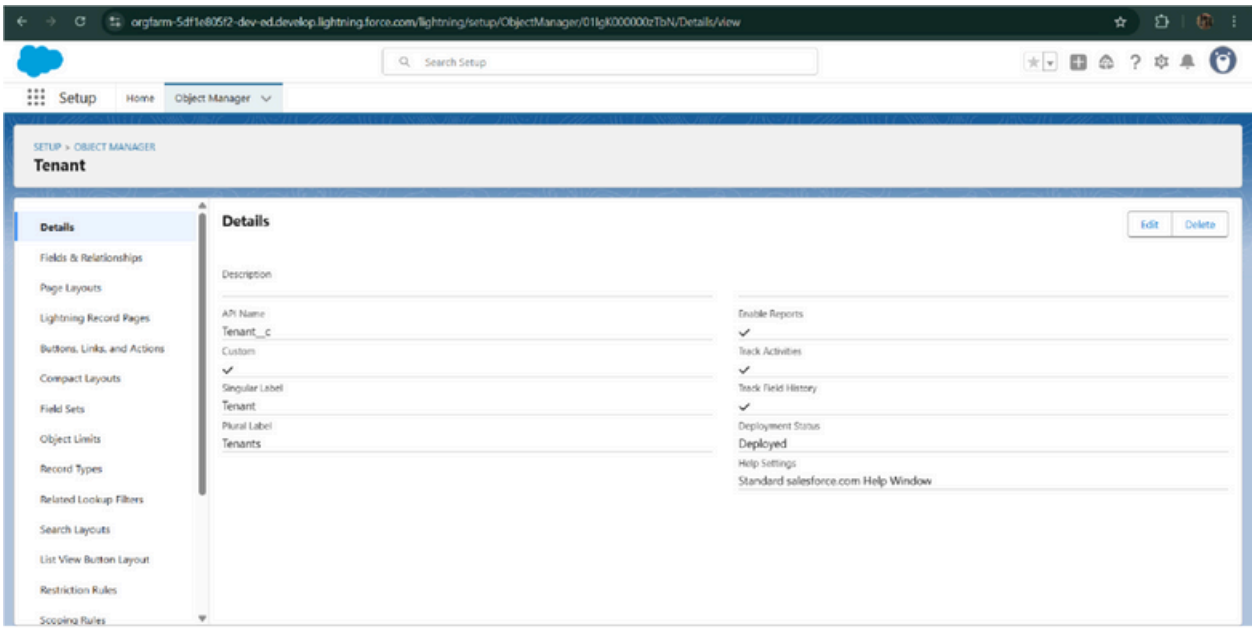
CreatingDeveloper Account:

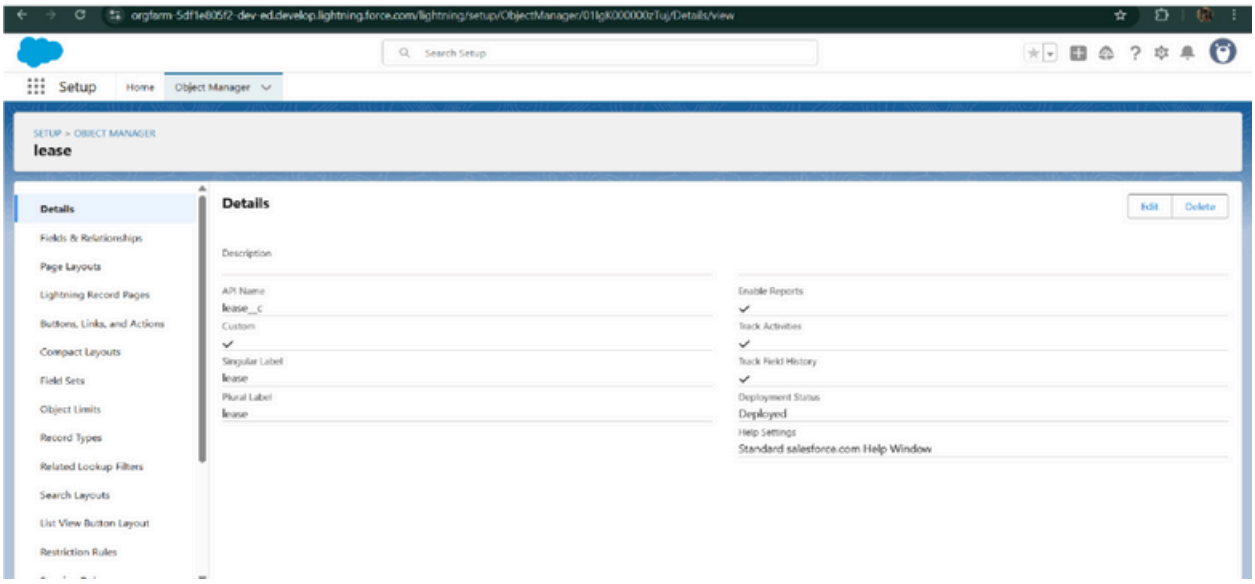
By using this URL - **https://www.salesforce.com/form/developer-signup/?d=pb**

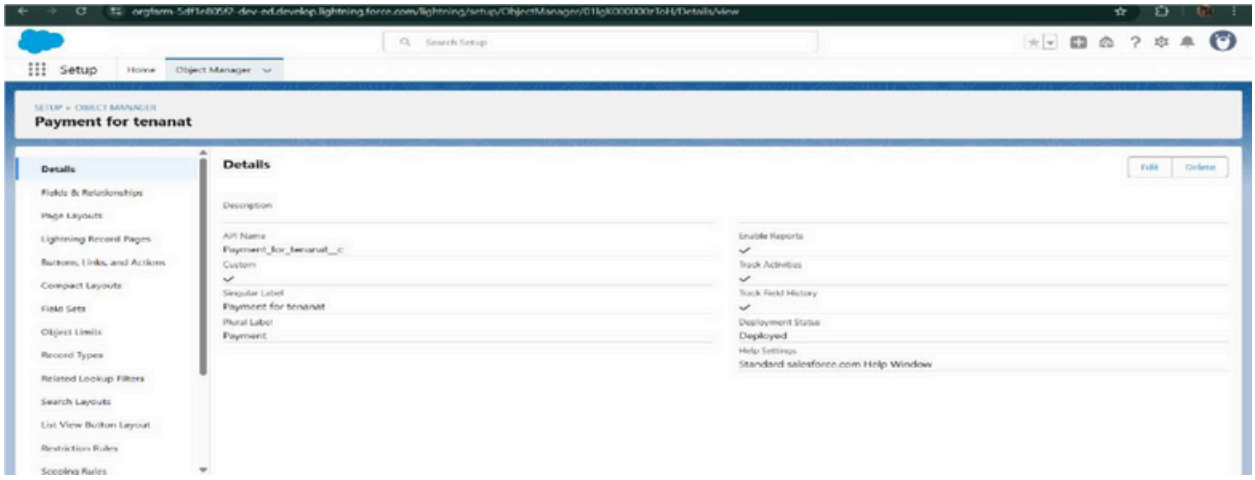


- Created objects: Property, Tenant, Lease, Payment

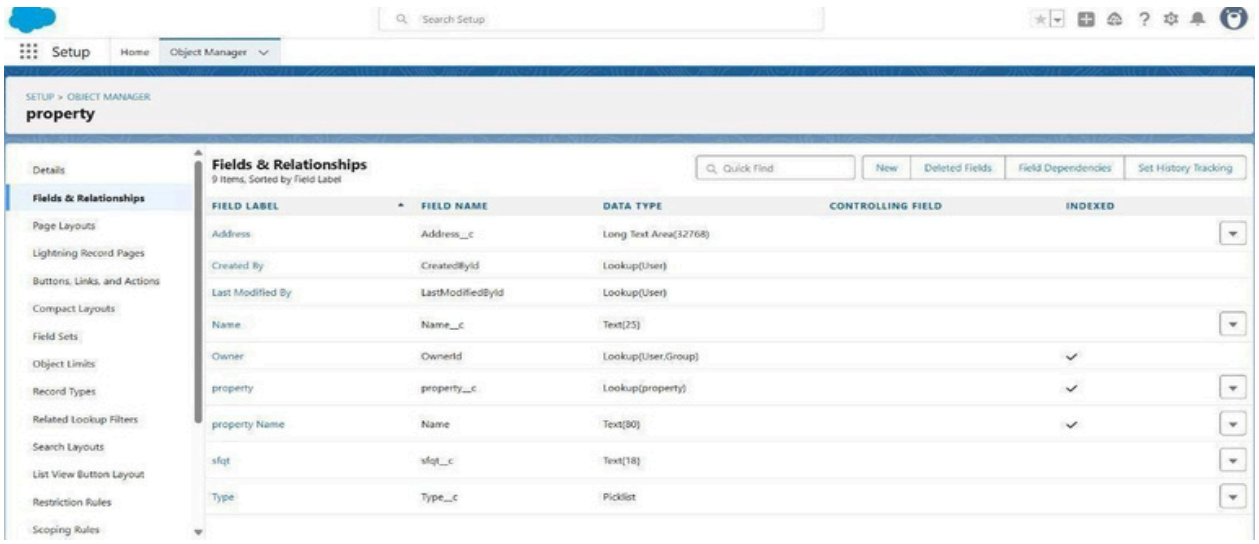


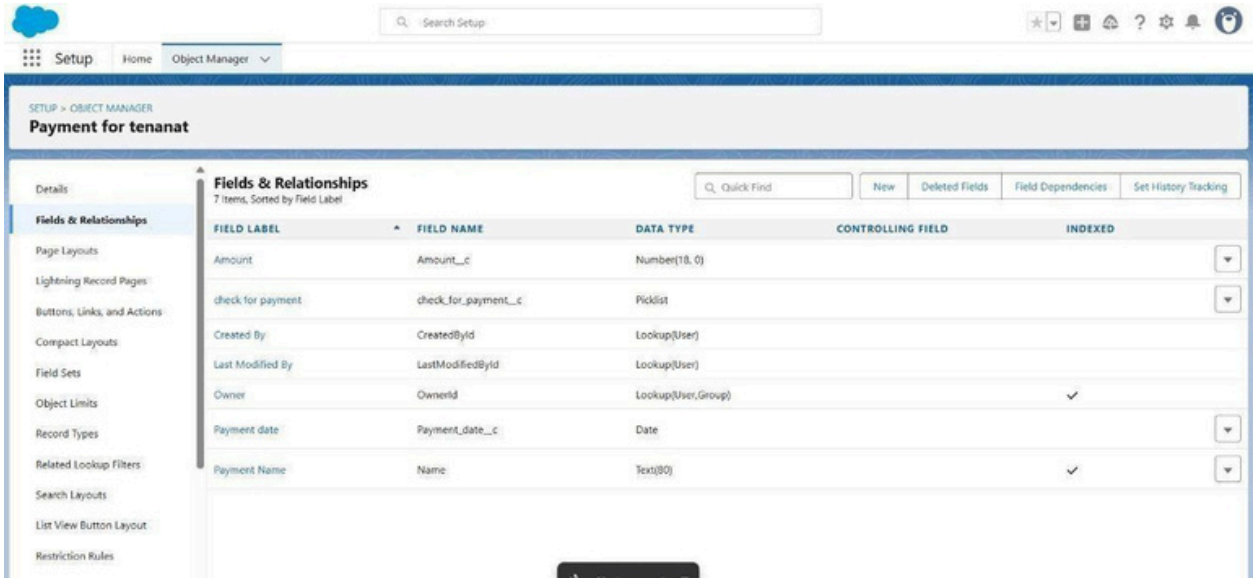

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01Igk000000zTbN/Details/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Tenant
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Details
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

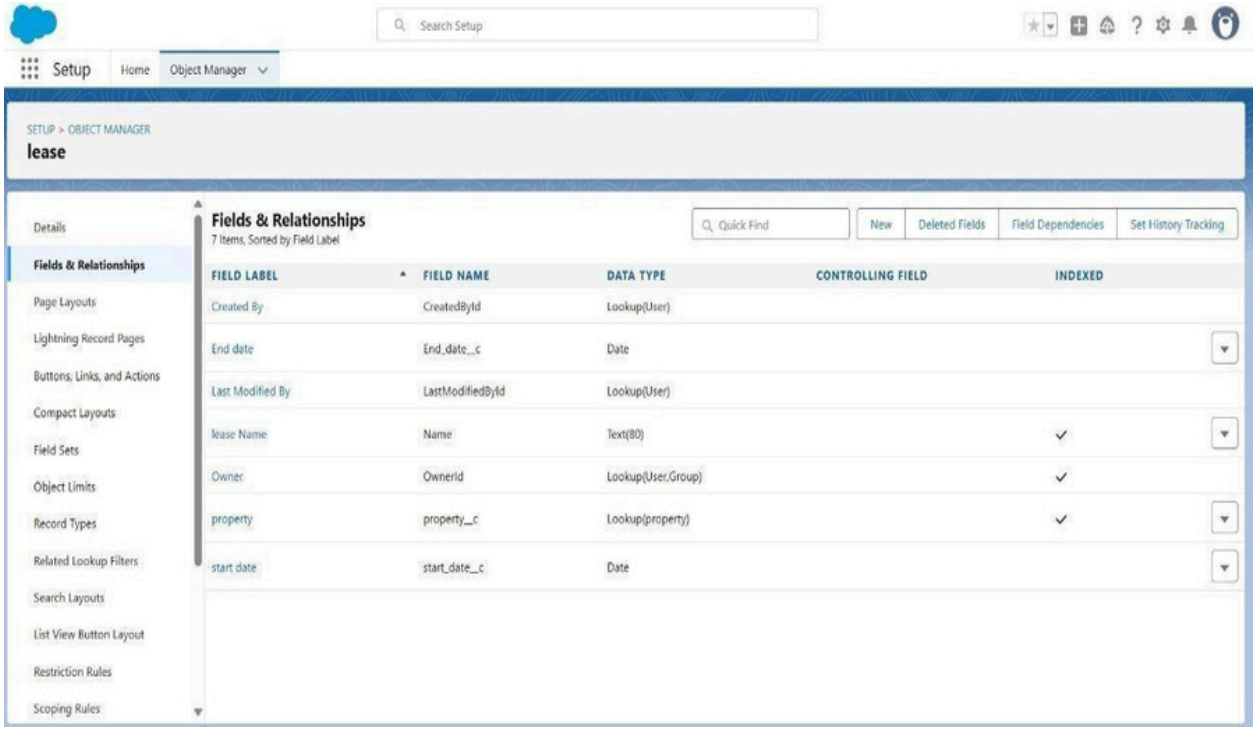

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01Igk000000zTuj/Details/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
lease
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Details
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

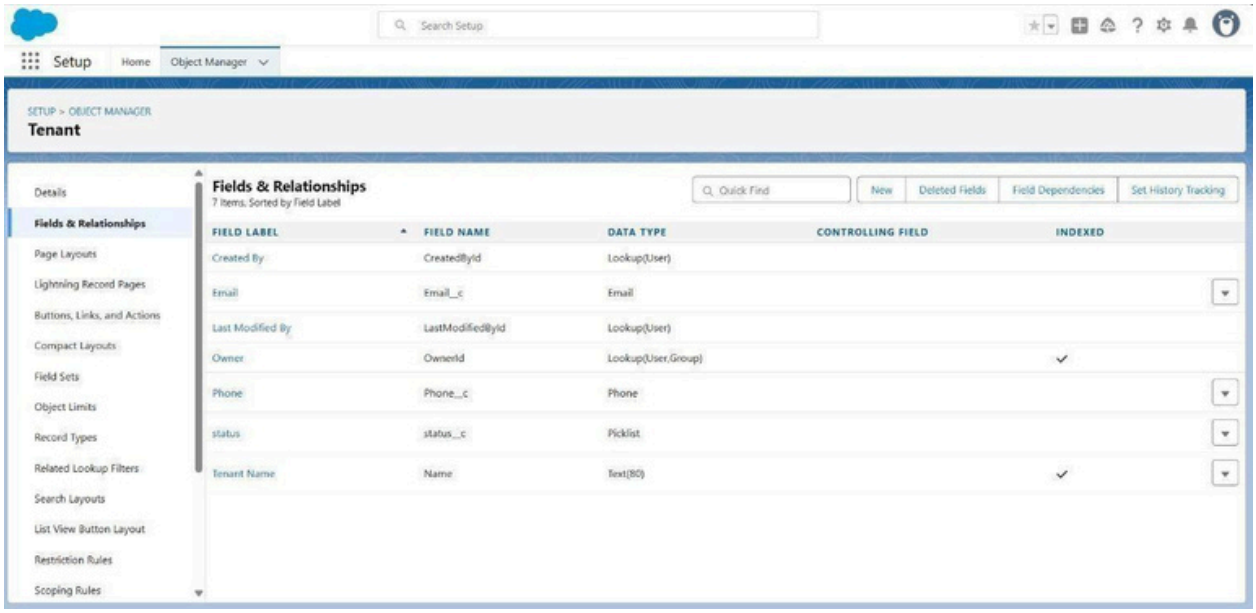


- Configured fields and relationships

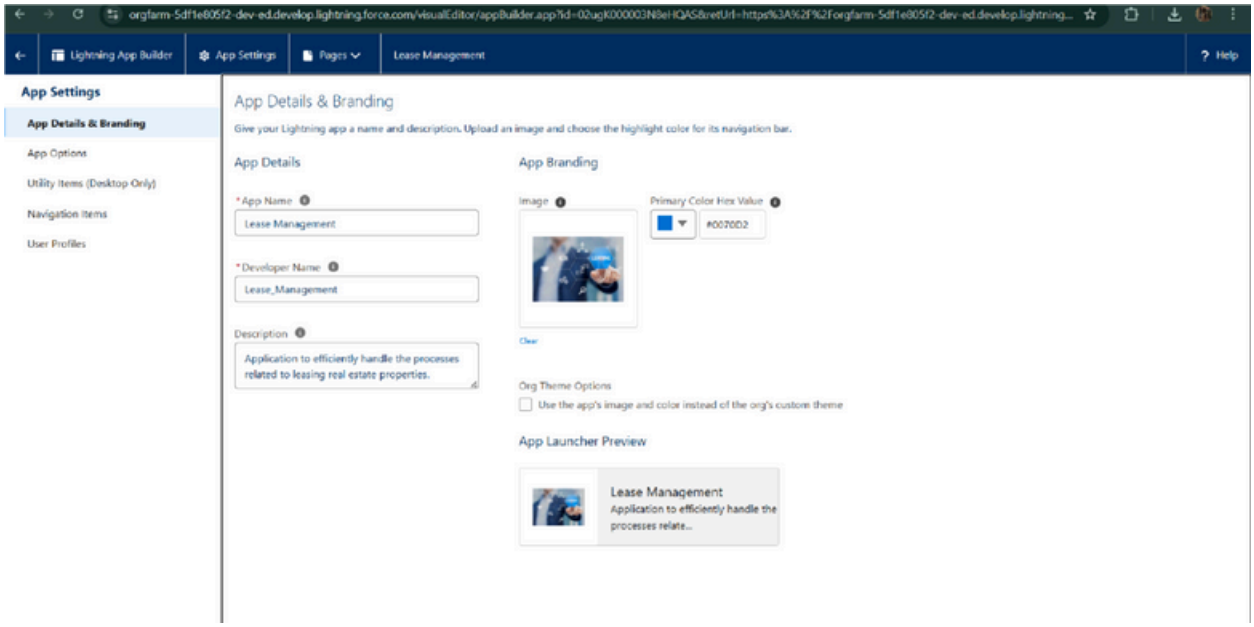


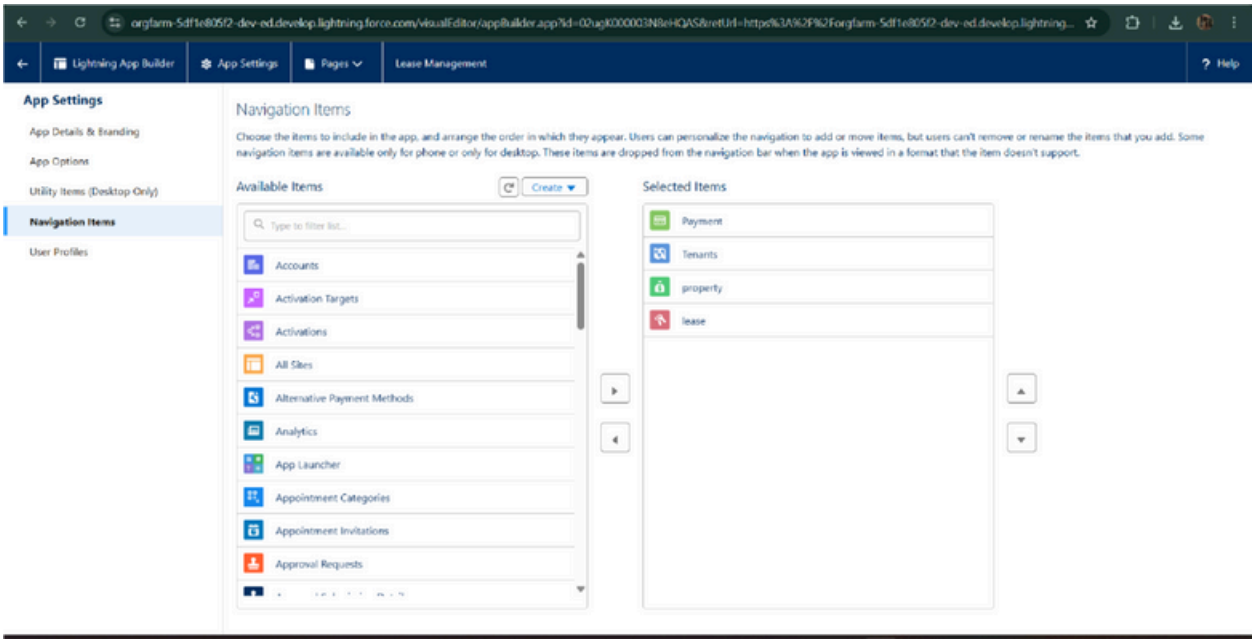

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Payment for tenanat
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction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

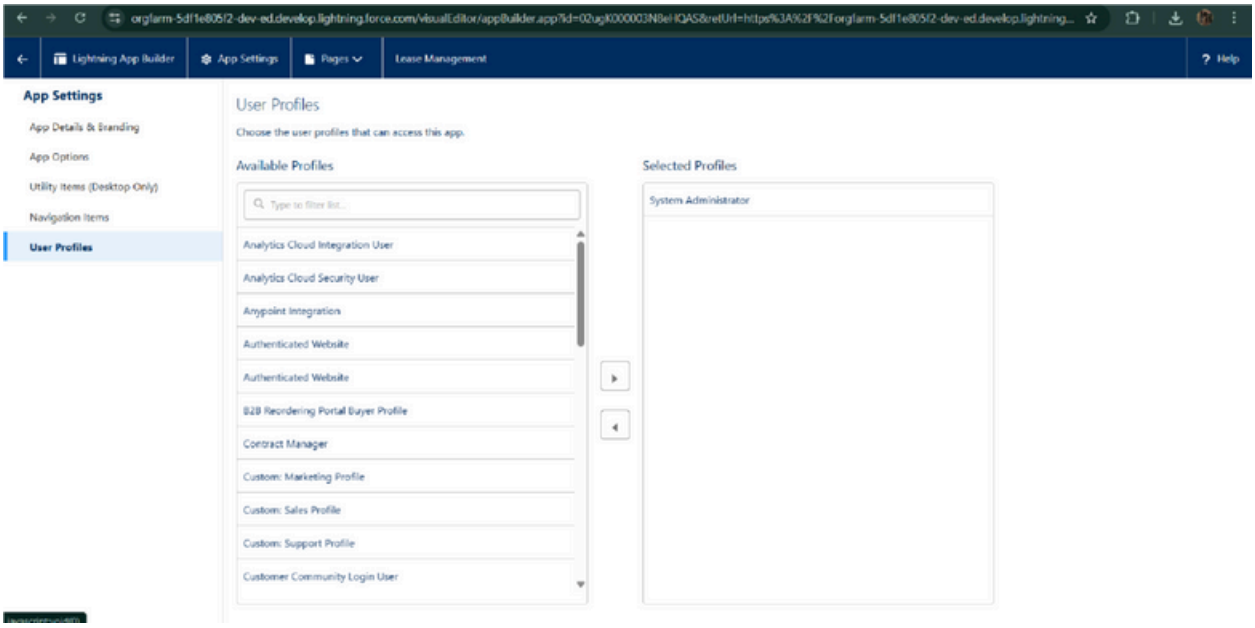

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
lease
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping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

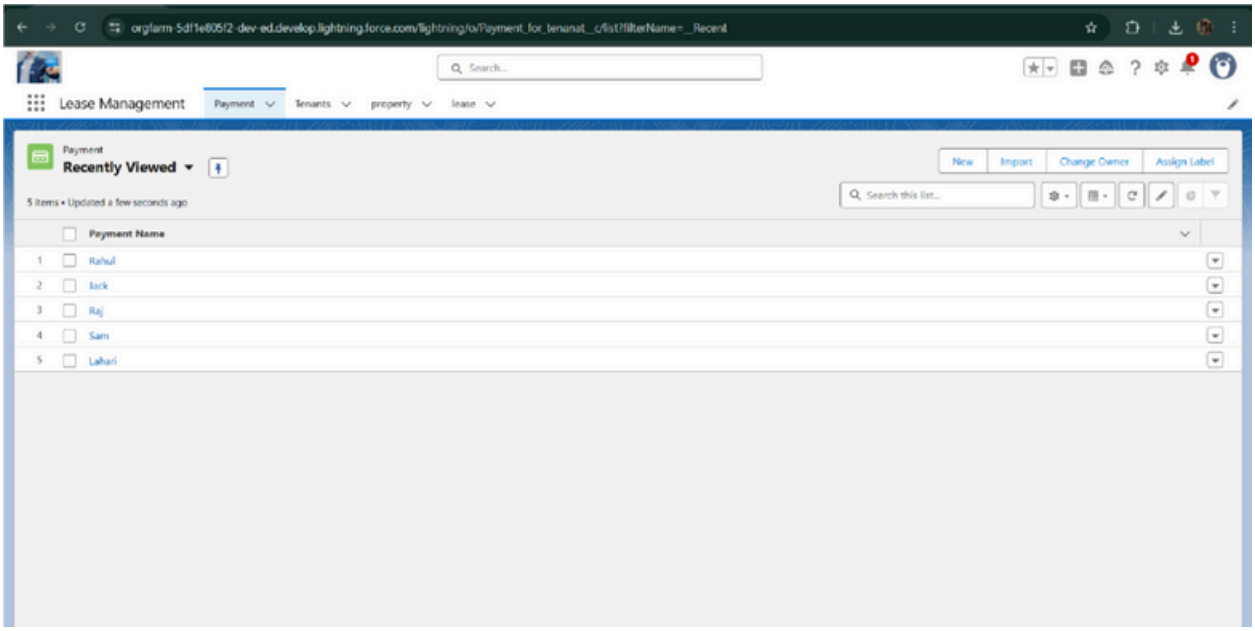


- Developed Lightning App with relevant tabs

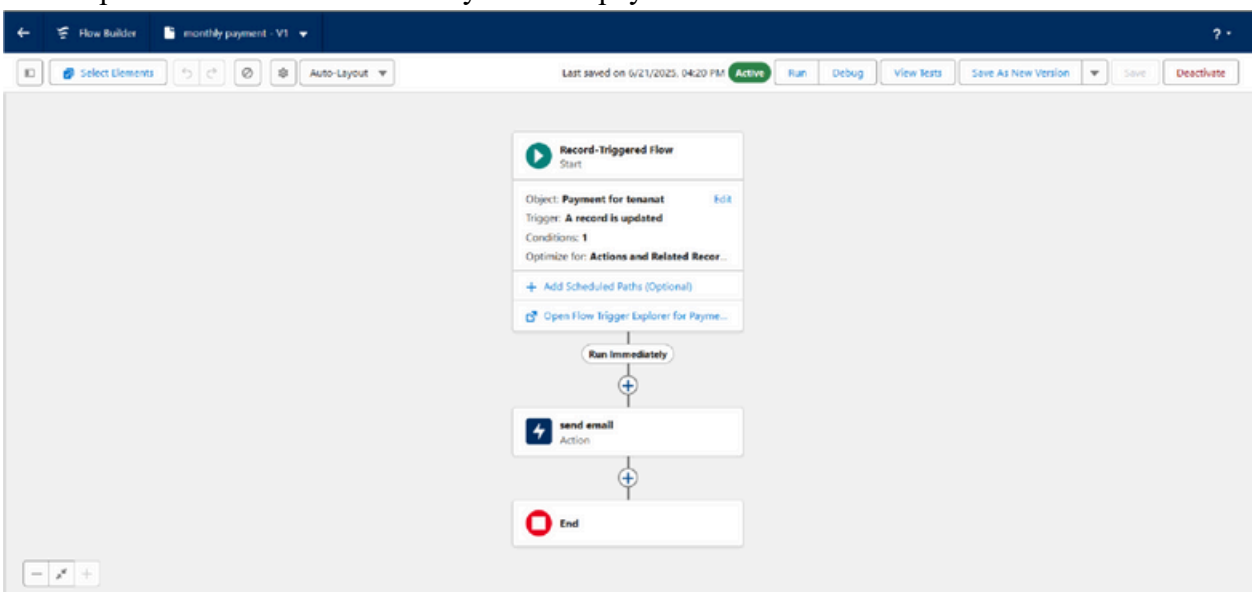



Lightning App Builder
App Settings
Pages
Lease Management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
Navigation Items
Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.
Available Items
Create
Selected Items
Accounts
Activation Targets
Activations
All Sites
Alternative Payment Methods
Analytics
App Launcher
Appointment Categories
Appointment Invitations
Approval Requests
Payment
Tenants
property
lease

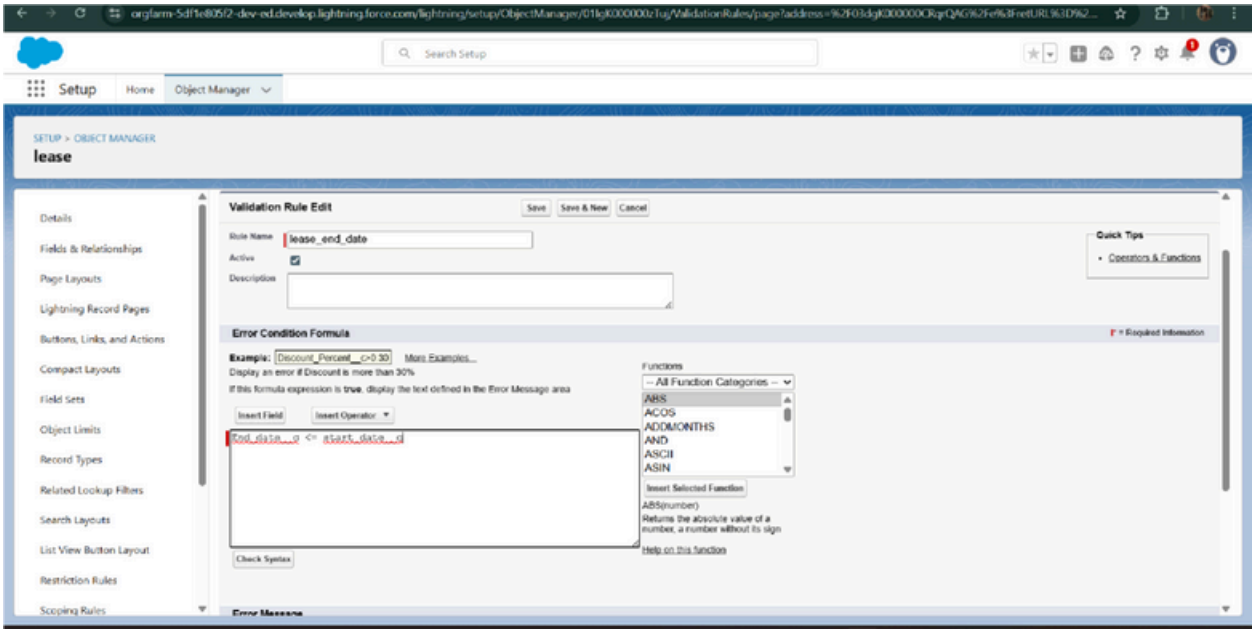

Lightning App Builder
App Settings
Pages
Lease Management
Help
App Settings
App Details & Branding
App Options
Utility Items (Desktop Only)
Navigation Items
User Profiles
User Profiles
Choose the user profiles that can access this app.
Available Profiles
Selected Profiles
Analytics Cloud Integration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

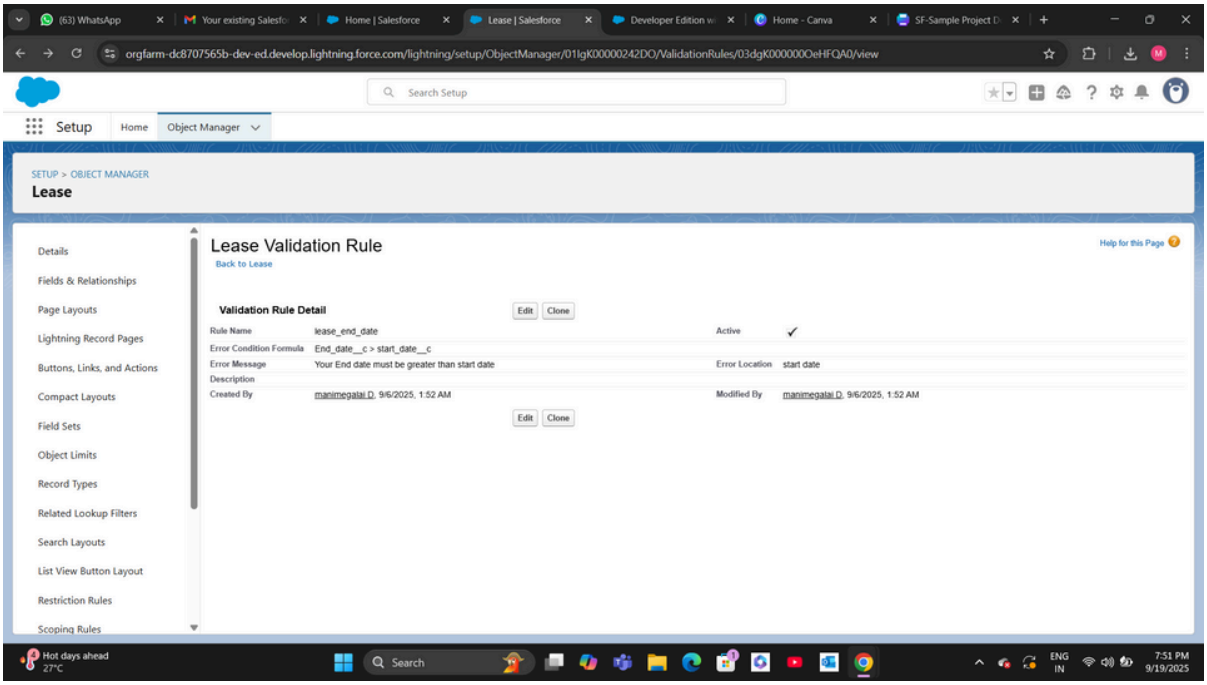


- Implemented Flows for monthly rent and payment success



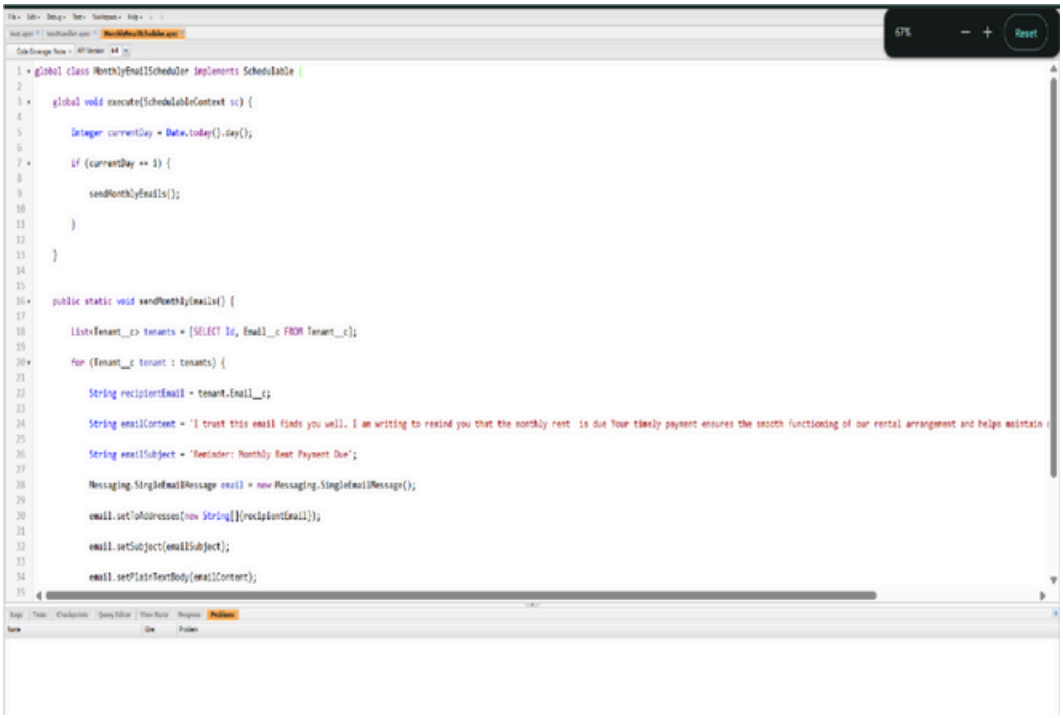
- To create a validation rule to a Lease Object





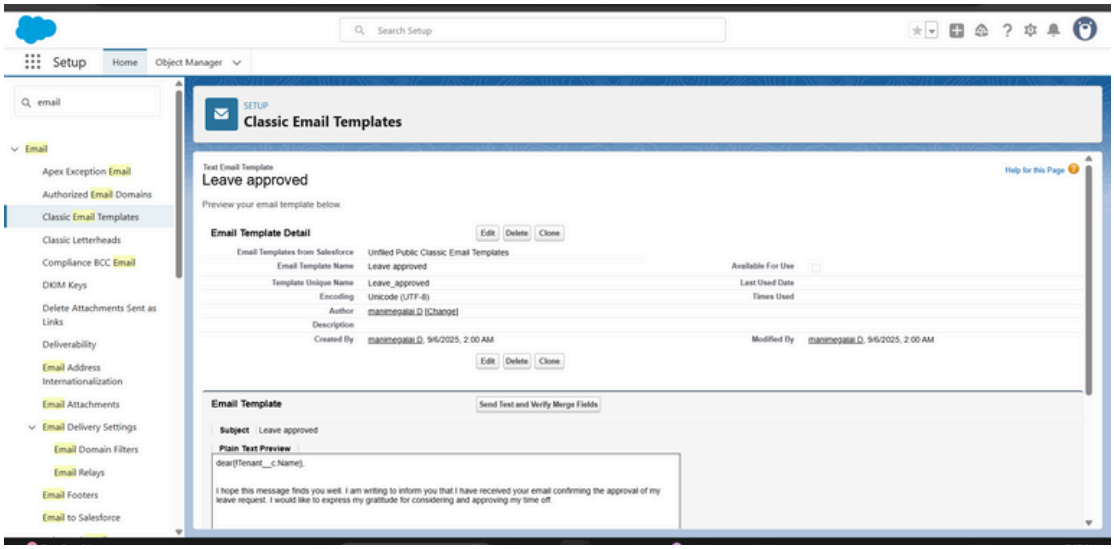
- Added Apex trigger to restrict multiple tenants per property

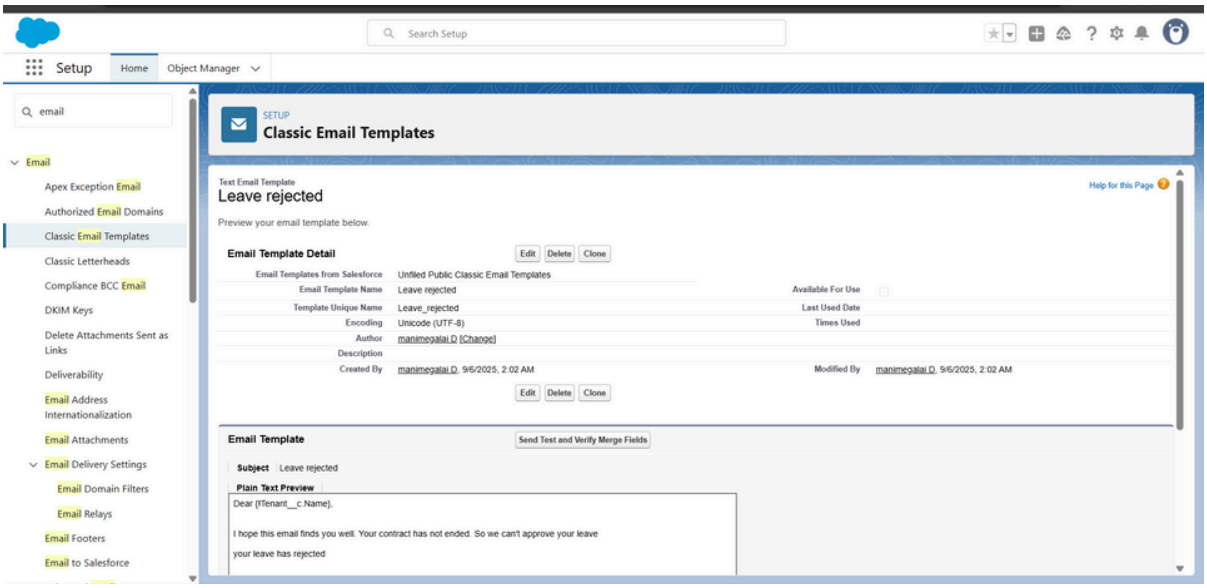
- Scheduled monthly reminder emails using Apex class



```
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);
```

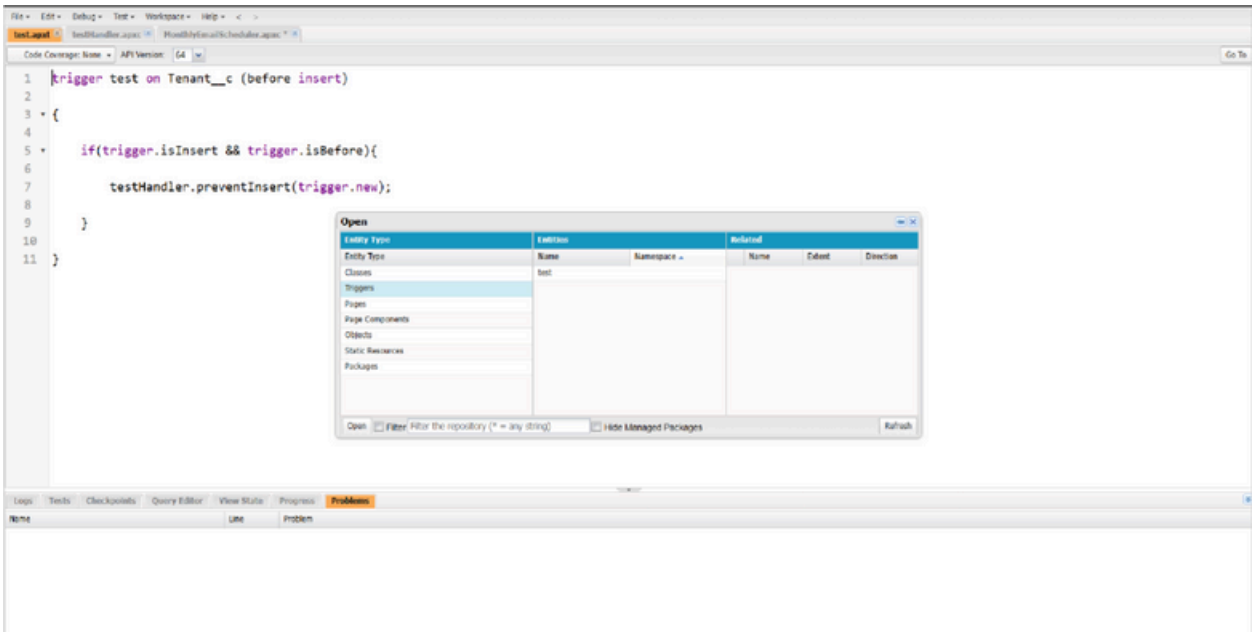
- Built and tested email templates for leave request, approval, rejection, payment, and reminders

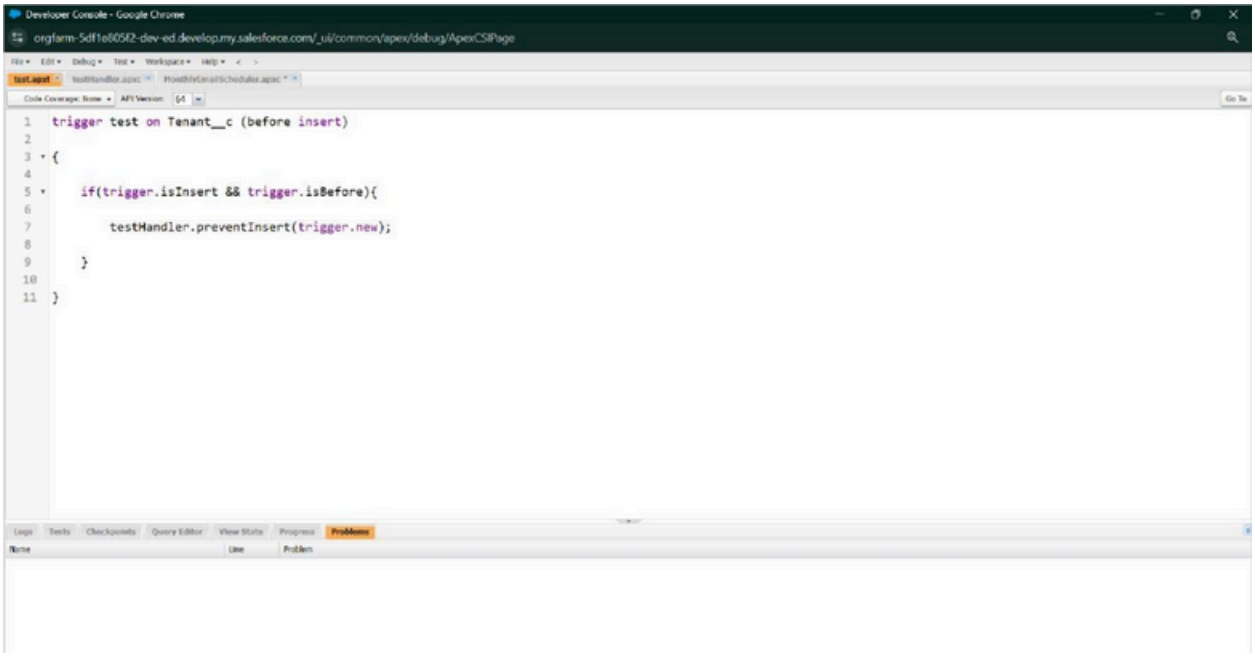




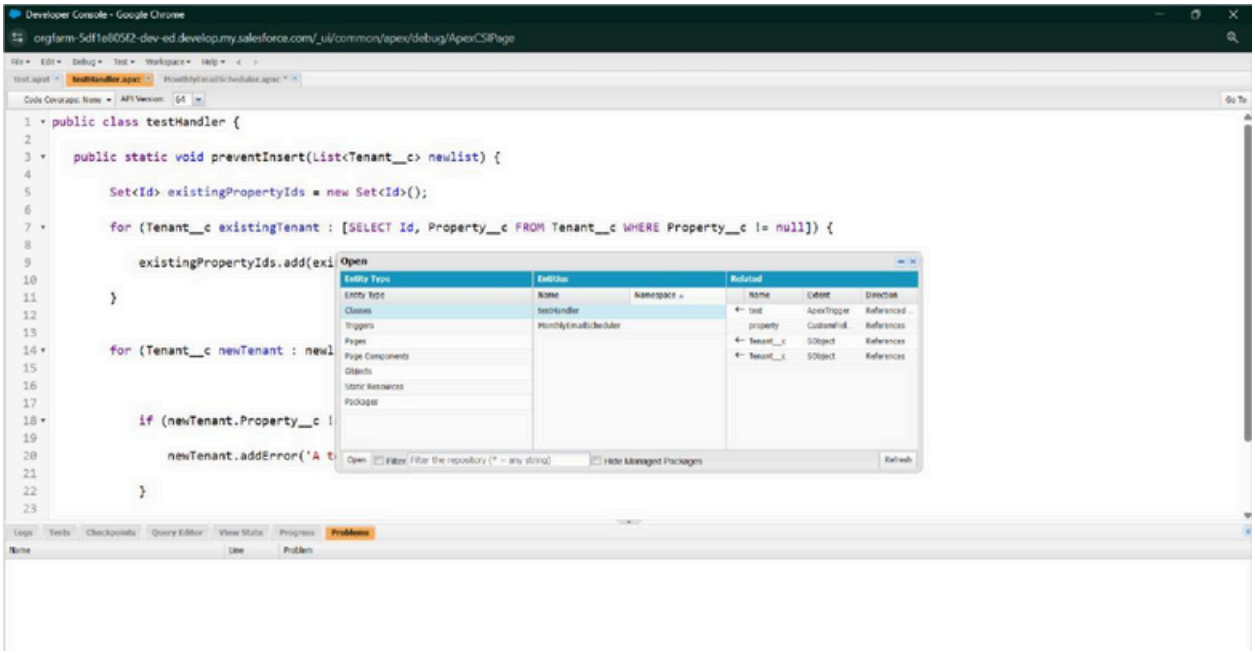
- Apex Trigger

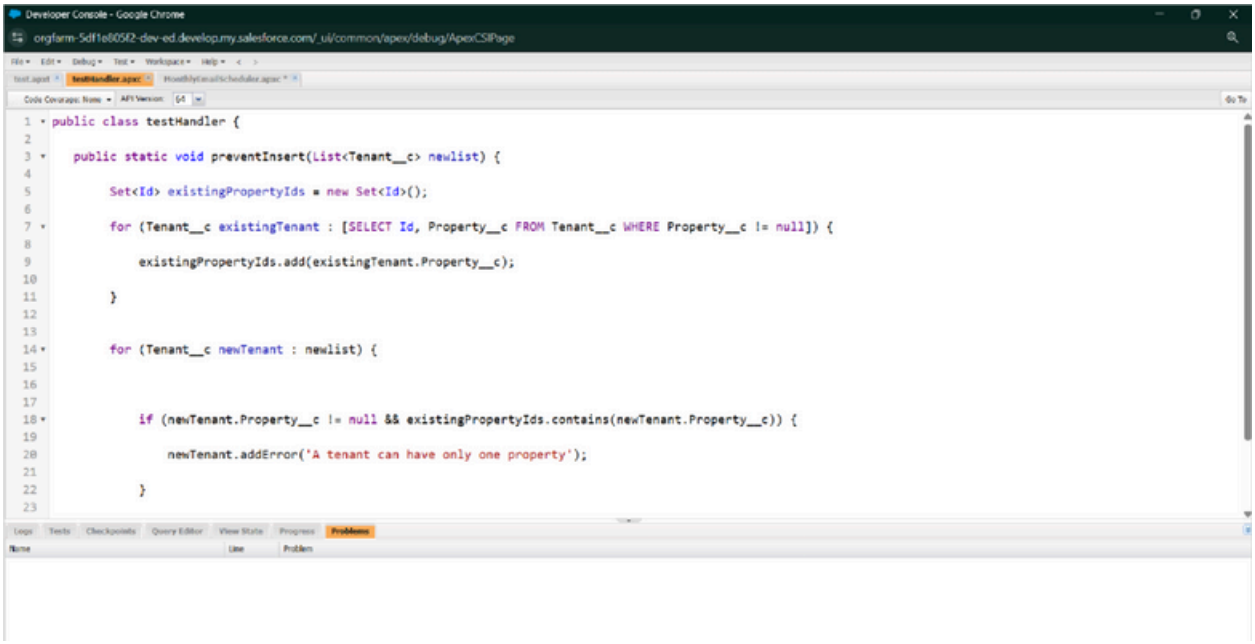
  Create an Apex Trigger



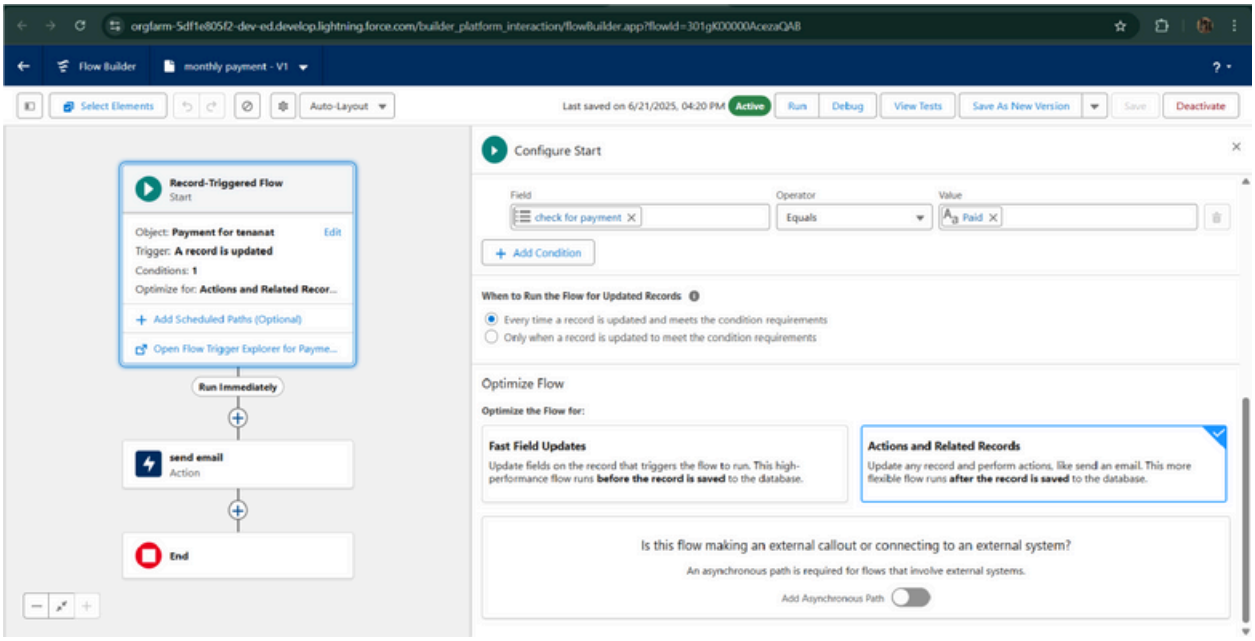


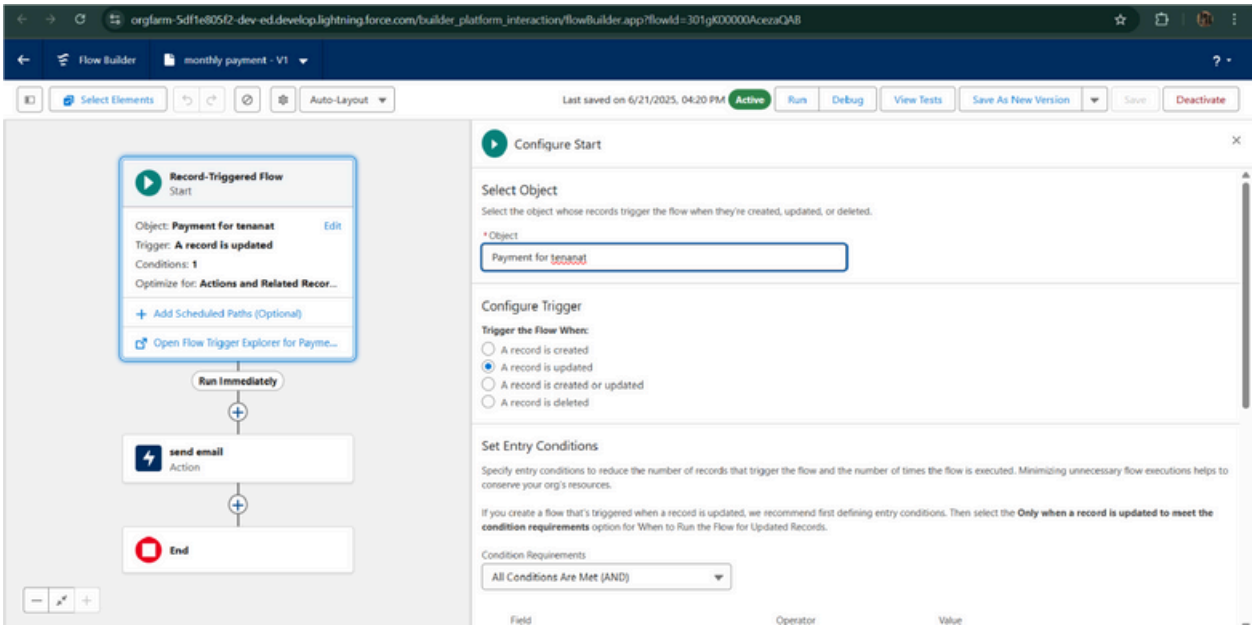
Create an Apex Handler class



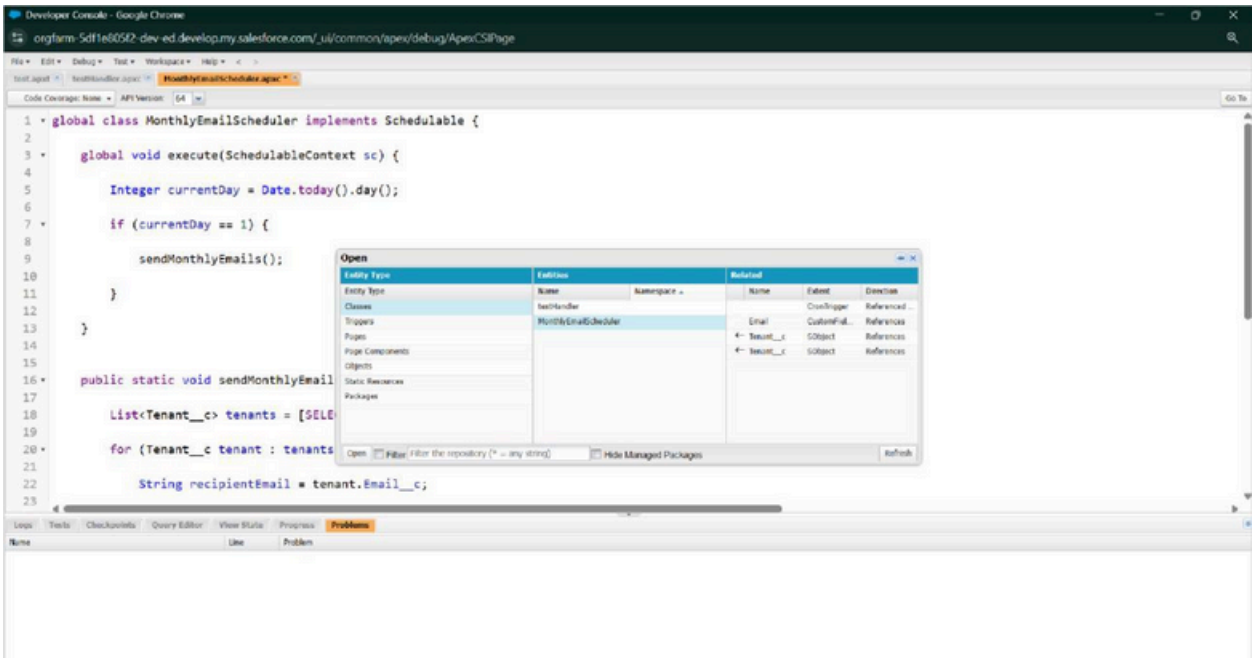

Developer Console - Google Chrome
orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
public class testHandler {
public static void preventInsert(List<Tenant__c> newlist) {
Set<Id> existingPropertyIds = new Set<Id>();
for (Tenant__c existingTenant : [SELECT Id, Property__c 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');
}

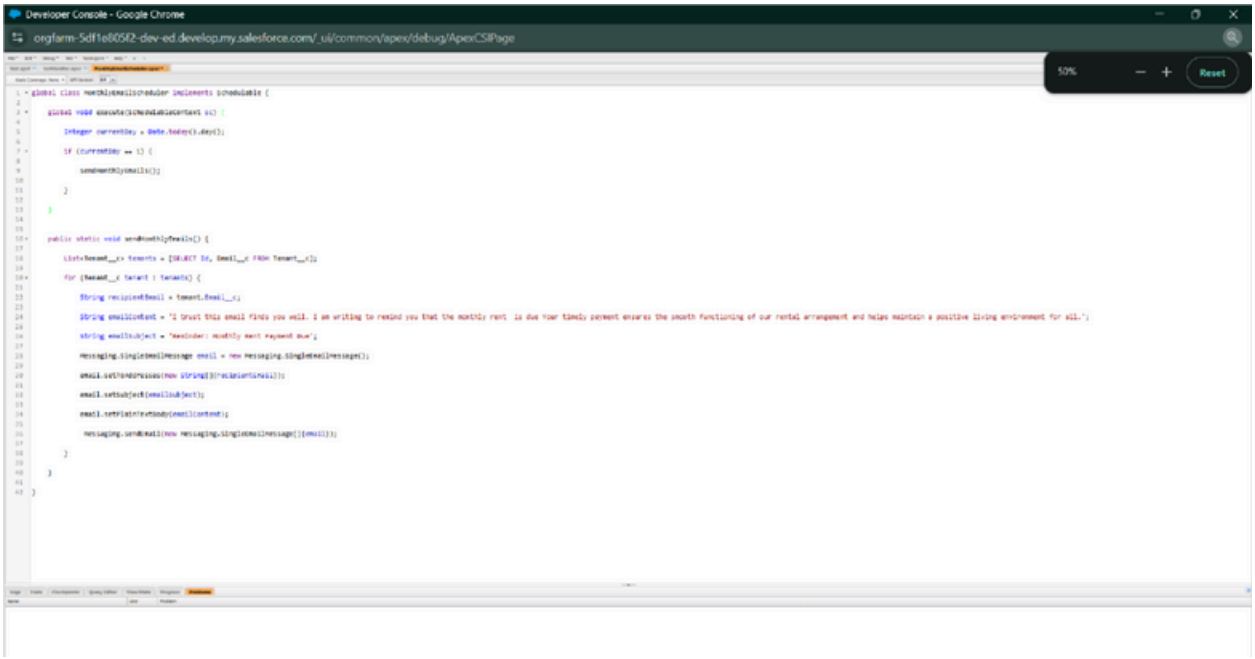
- FLOWS


orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gK00000AcezaQAB
Flow Builder
monthly payment - V1
Select Elements
Auto-Layout
Last saved on 6/21/2025, 04:20 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Object: Payment for tenanat
Edit
Trigger: A record is updated
Conditions: 1
Optimize for: Actions and Related Recor...
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Payme...
Run Immediately
send email
Action
End
Configure Start
Field
Operator
Value
check for payment
Equals
Paid
Add Condition
When to Run the Flow for Updated Records
Every time a record is updated and meets the condition requirements
Only when a record is updated to meet the condition requirements
Optimize Flow
Optimize the Flow for:
Fast Field Updates
Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved to the database.
Actions and Related Records
Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.
Is this flow making an external callout or connecting to an external system?
An asynchronous path is required for flows that involve external systems.
Add Asynchronous Path

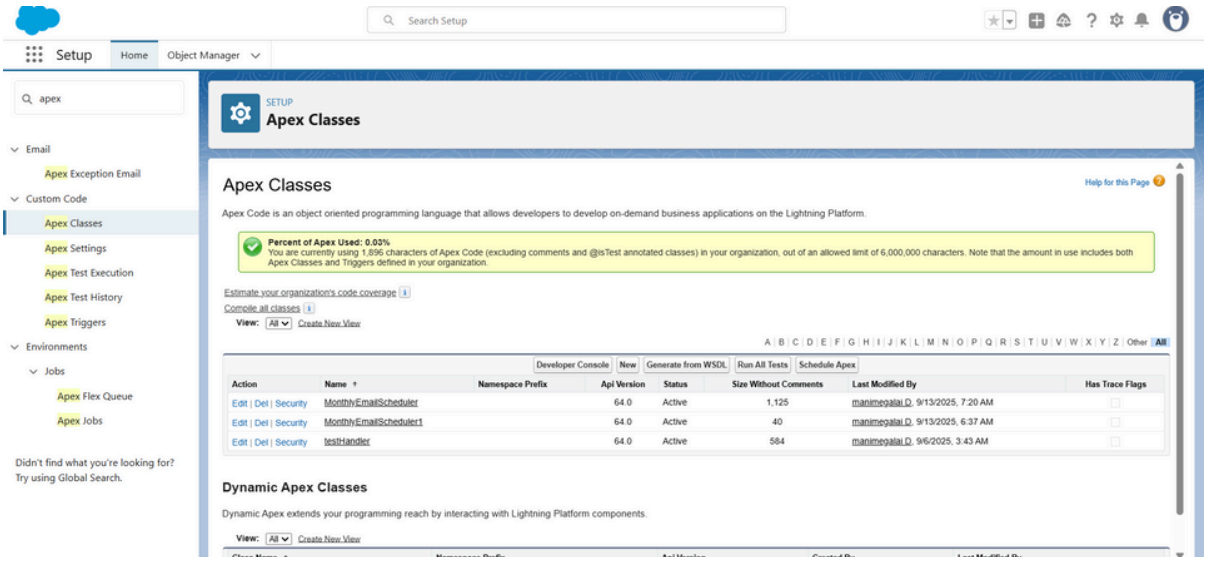


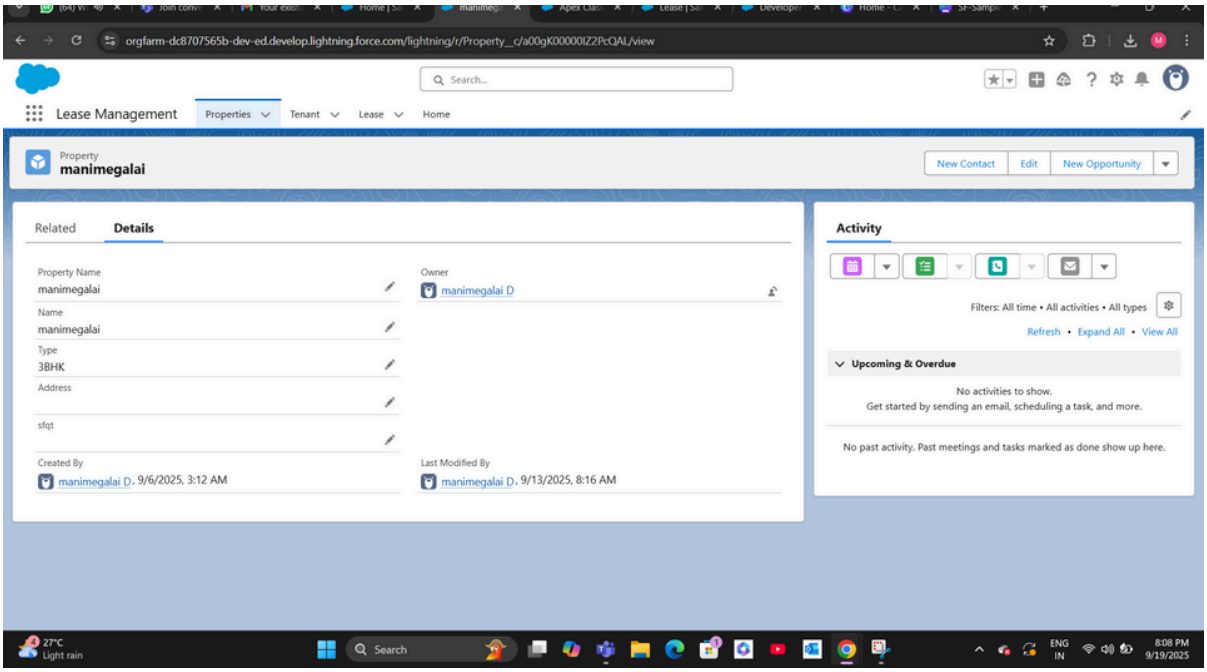
- Schedule class:
  Create an Apex Class





Schedule Apex class

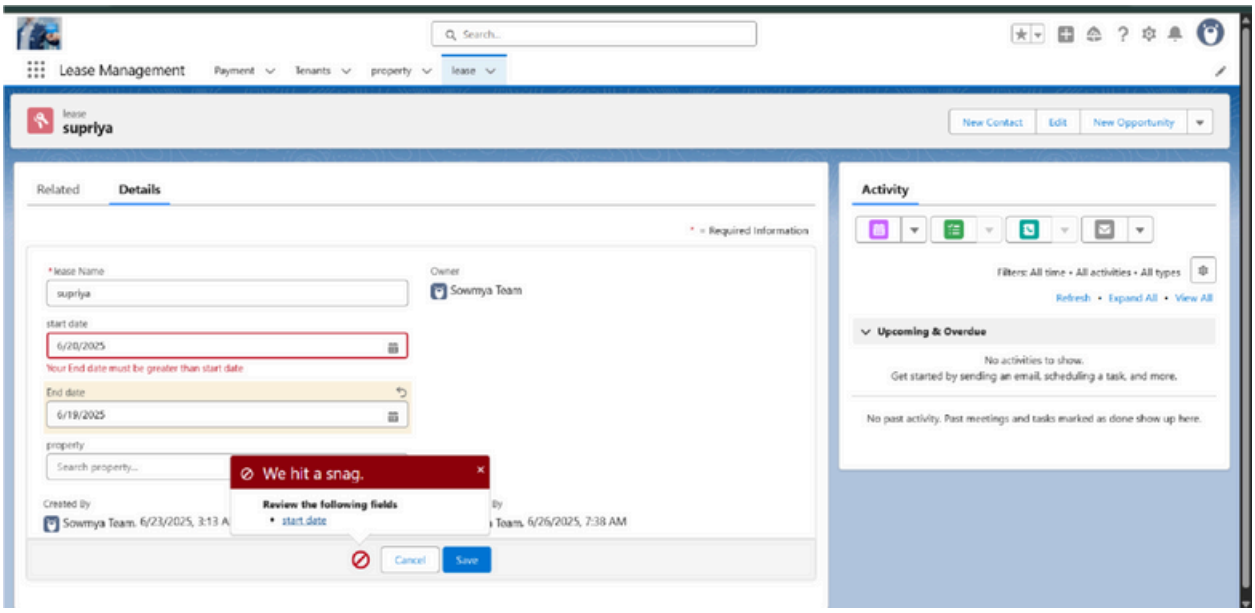


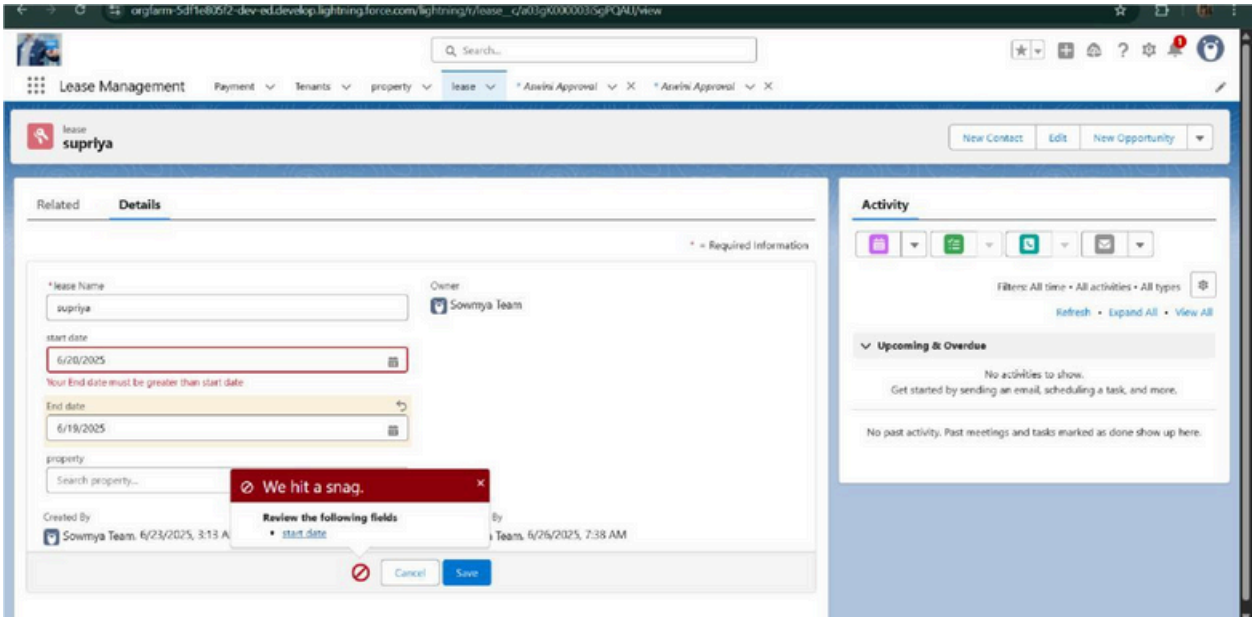


FUNCTIONAL AND PERFORMANCE TESTING

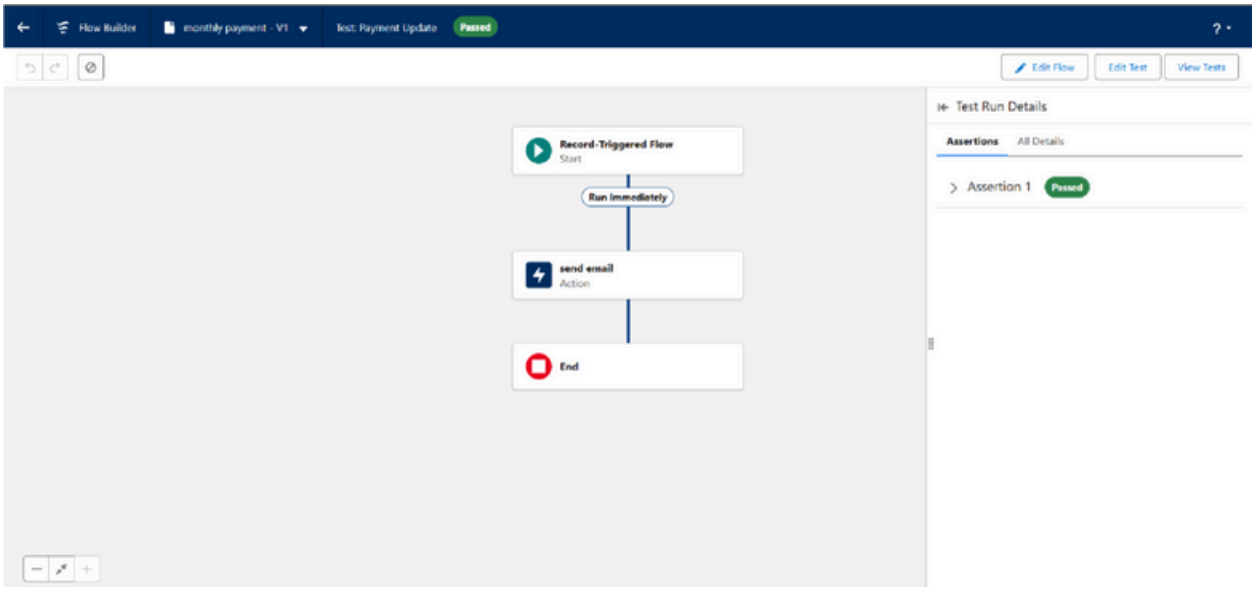
Performance Testing

- Triggervalidationbyentering duplicate tenant-property records
- Validation Rule checking


Lease Management
Payment
Tenants
property
lease
supriya
New Contact
Edit
New Opportunity
Related
Details
* = Required Information
*lease Name
supriya
Owner
Sowmya Team
start date
6/20/2025
Your End date must be greater than start date
End date
6/19/2025
property
Search property...
We hit a snag.
Review the following fields
start date
Created By
Sowmya Team, 6/23/2025, 3:13 A
Team, 6/26/2025, 7:38 AM
Cancel
Save
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
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.


Lease Management
Payment
Tenants
property
lease
* Aswini Approval
* Aswini Approval
supriya
New Contact
Edit
New Opportunity
Related
Details
* = Required Information
*lease Name
supriya
Owner
Sowmya Team
start date
6/20/2025
Your End date must be greater than start date
End date
6/19/2025
property
Search property...
We hit a snag.
Review the following fields
start date
Created By
Sowmya Team, 6/23/2025, 3:13 A
Team, 6/26/2025, 7:38 AM
Cancel
Save
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
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.

- Test flows on payment update

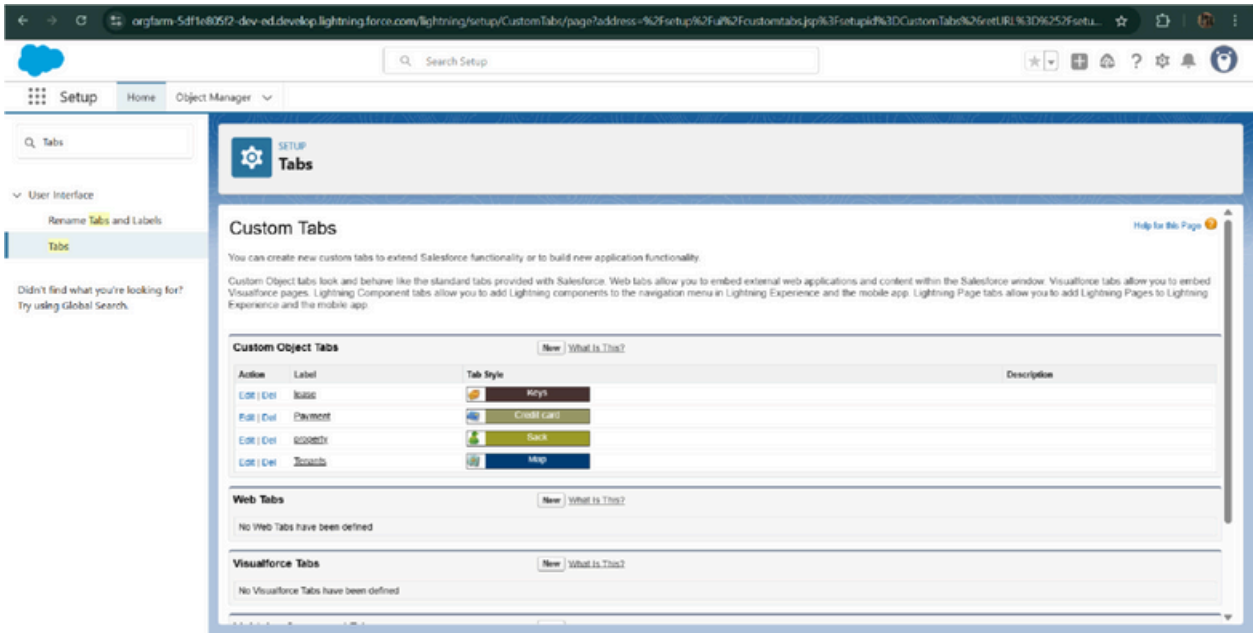


- Approval process validated through email alerts and status updates

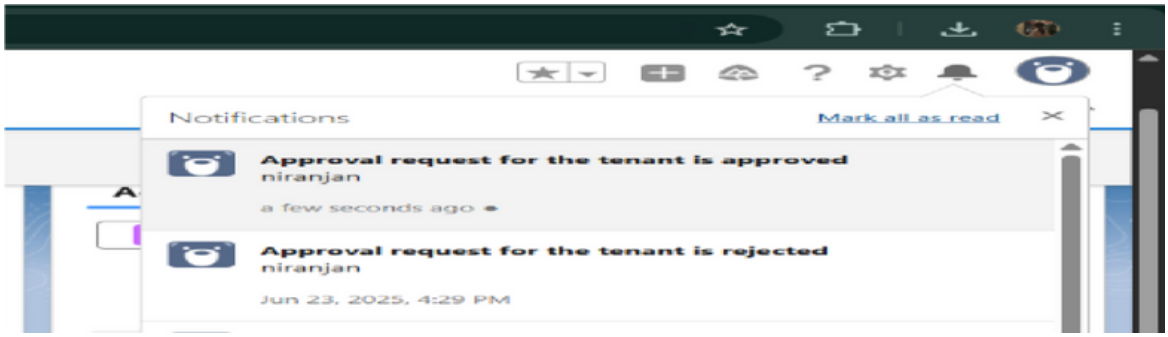
RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment



- Approval process notifications



ADVANTAGES & DISADVANTAGES

CONCLUSION

TheLease Management System successfully streamlines the operations of leasing through a structured, automatedSalesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

APPENDIX

Provided in Apex Classes and Triggers

- **Source Code:**

Test.apxt:

.

```
trigger teston Tenant__c (before insert) { if
(trigger.isInsert && trigger.isBefore){
testHandler.preventInsert(trigger.new);
```

} }

testHandler.apxc:

```
public class
testHandler { public
static void
preventInsert(List<
Tenant__c> newlist)
Set<Id>
{
existingPropertyIds
= new Set<Id>()
for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c
WHERE Property__c != null]) {
existingPropertyIds.add(existingTenant.Property__c;
} for (Tenant__c newTenant :
newlist) {
if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) { newTenant.addError('A
tenant can have only one property');
}
}
```

```
        }

    }
```

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});
            }
        }
}
```