Lease Management

**College Name: Sree Narayana Guru College,Chavadi, Coimbatore. College Code: Bru36**

**TEAM ID:NM2025TMID26552**

**TEAM MEMBERS:**

**Team LeaderName: MUHAMMED HARSHAD M.H Email:** [**achuachuzz479@gmail.com**](mailto:achuachuzz479@gmail.com)

**Team Member: MANU R**

**Email:** [**manur9963@gmail.com**](mailto:manur9963@gmail.com) **Team Member: MUHAMMED AJMAL A**

**Email:** [**ajmal20122005@gmail.com**](mailto:ajmal20122005@gmail.com)

### INTRODUCTION

ProjectOverview

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.



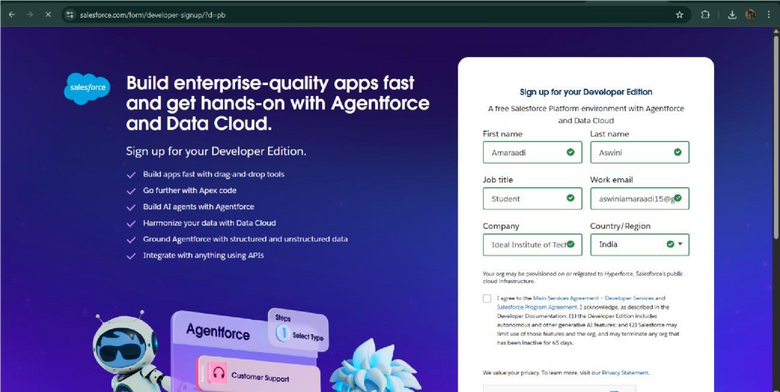
## Purpose

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.

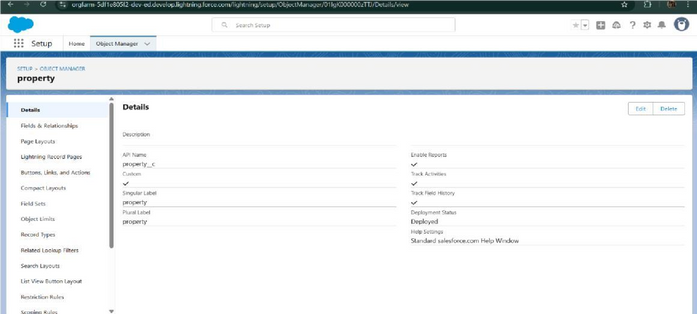
DEVELOPMENT PHASE

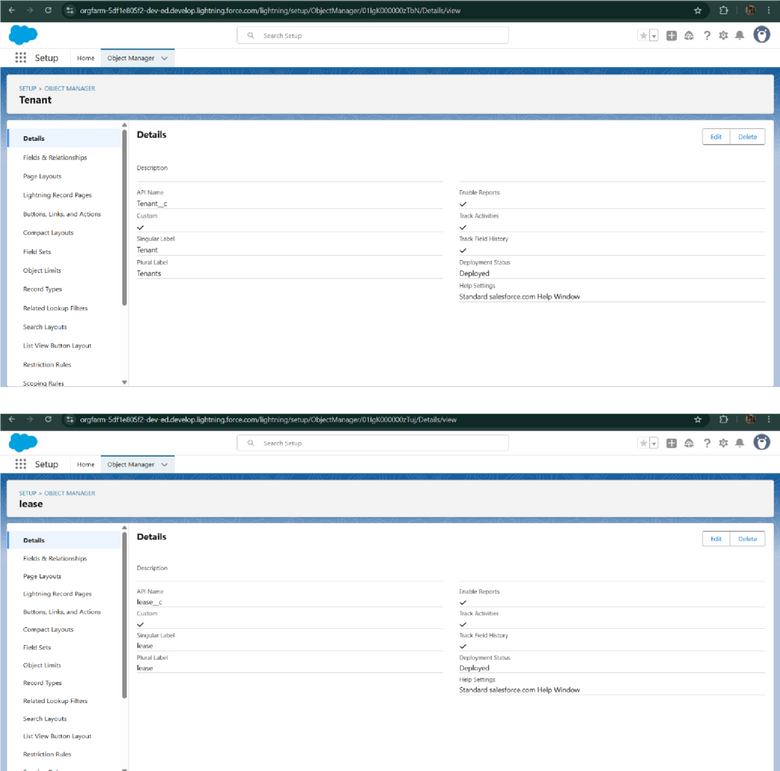
Creating Developer Account:

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



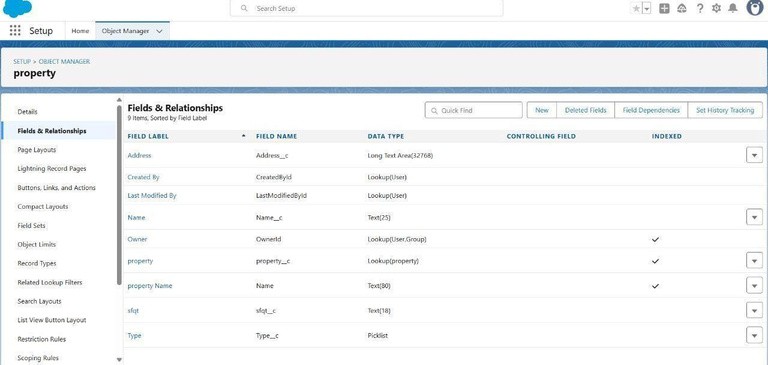
* Created objects: Property, Tenant, Lease, Payment

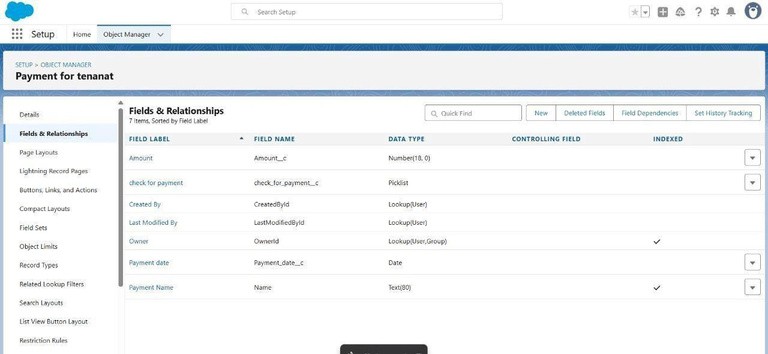


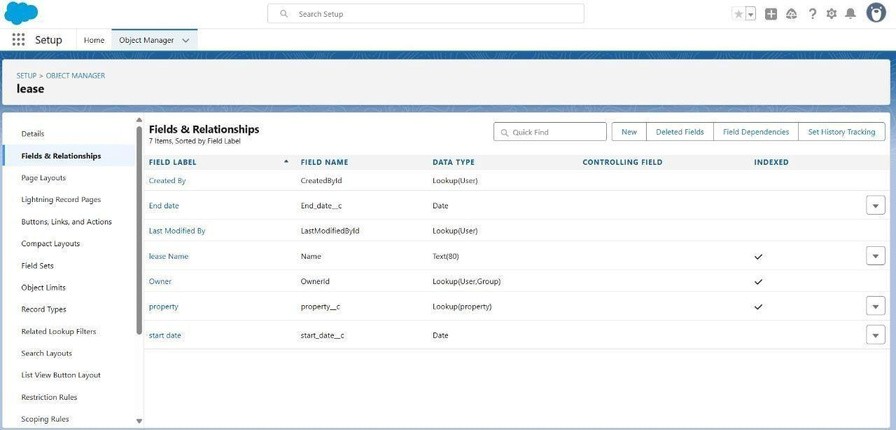


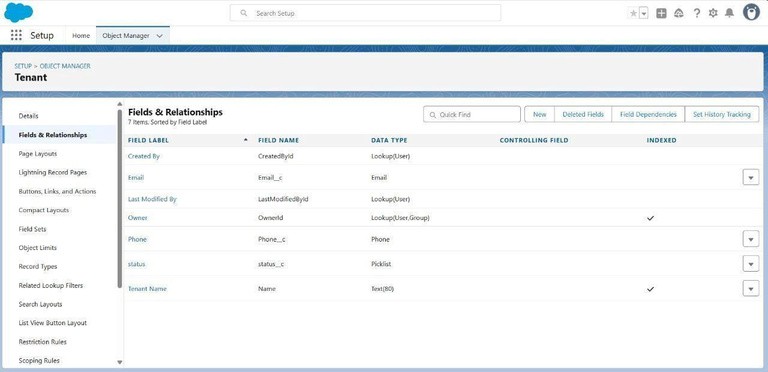


* Configured fields and relationships

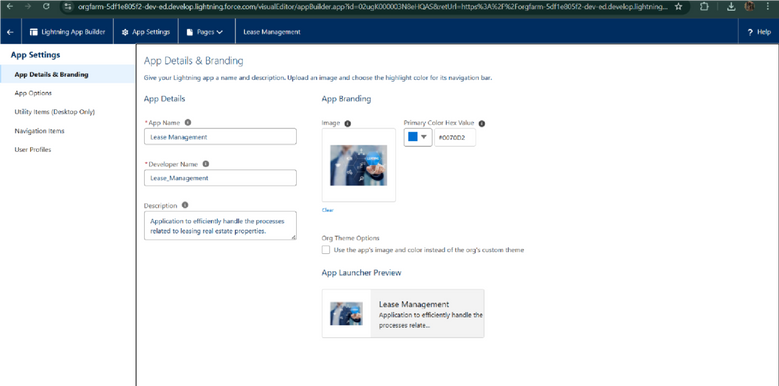


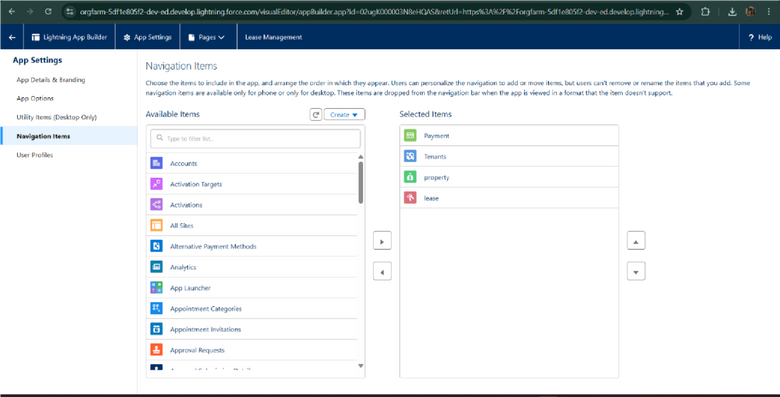


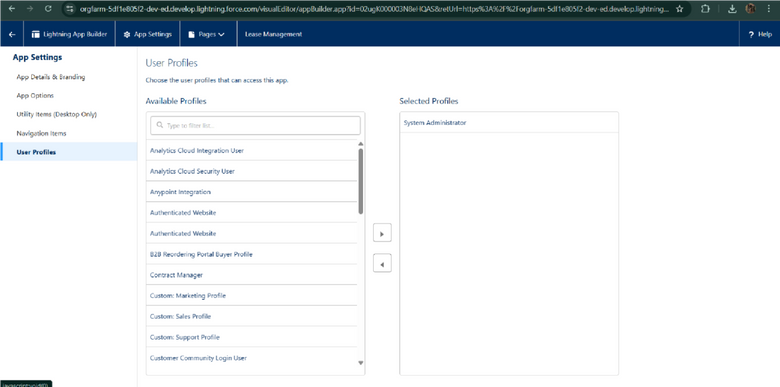


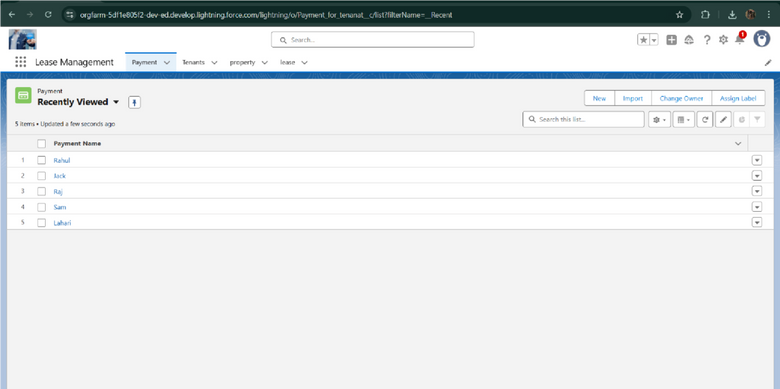


* Developed Lightning App with relevant tabs

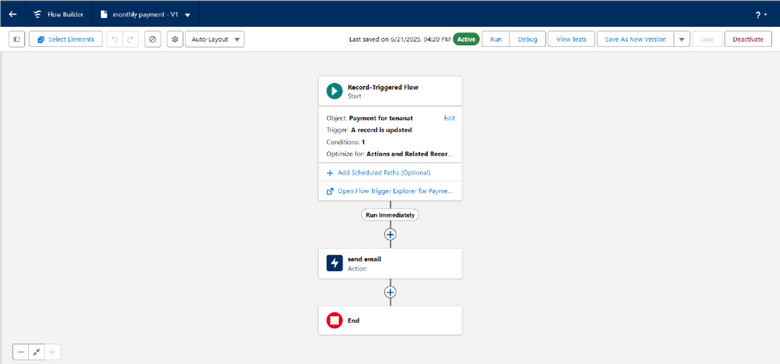




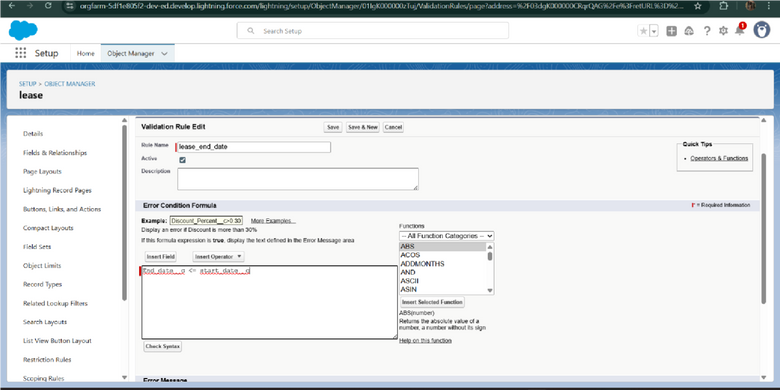


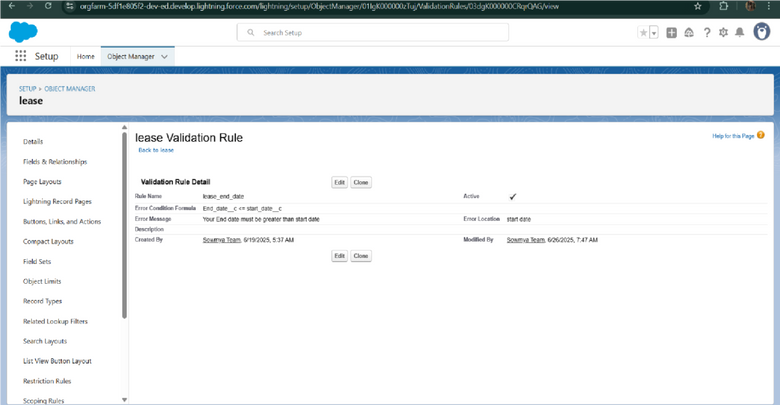


* + Implemented Flows for monthly rent and payment success

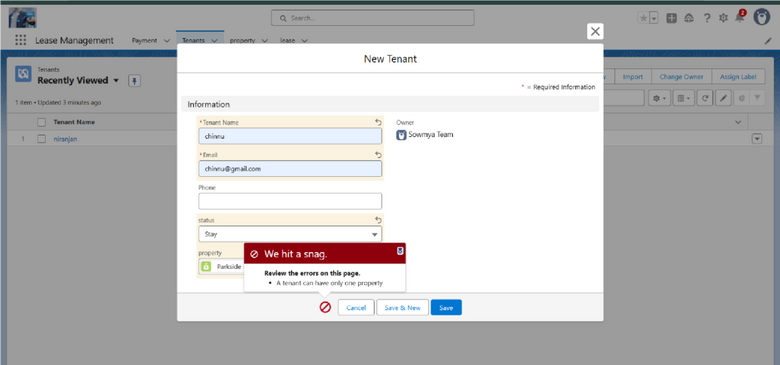


* . To create a validation rule to a Lease Object

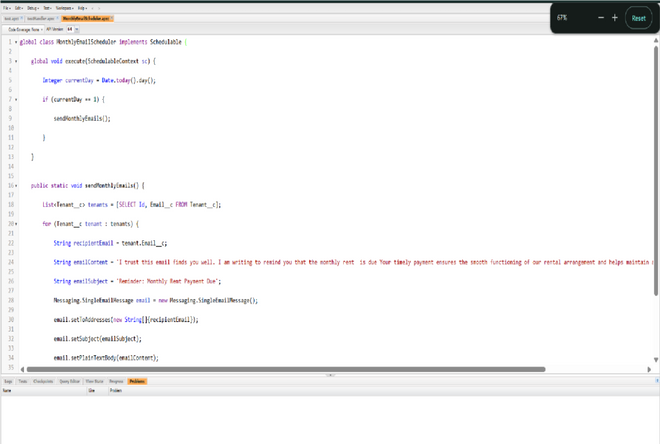




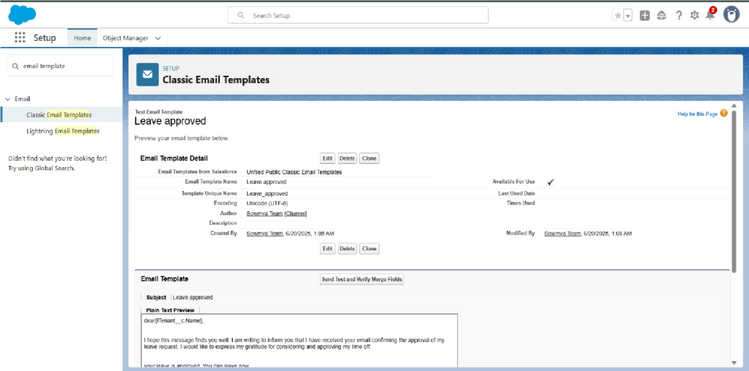
* Added Apex trigger to restrict multiple tenants per property

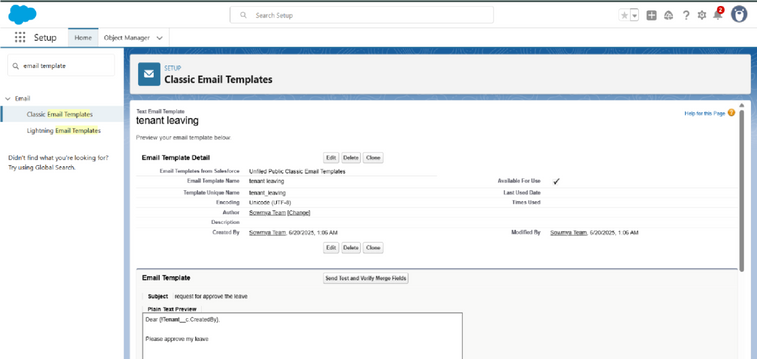


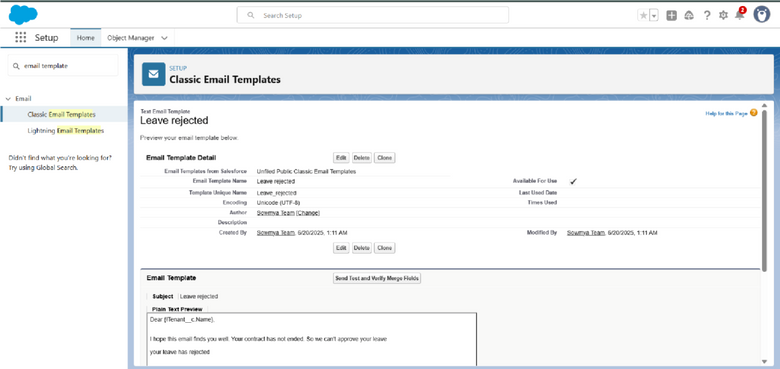
* Scheduled monthly reminder emails using Apex class

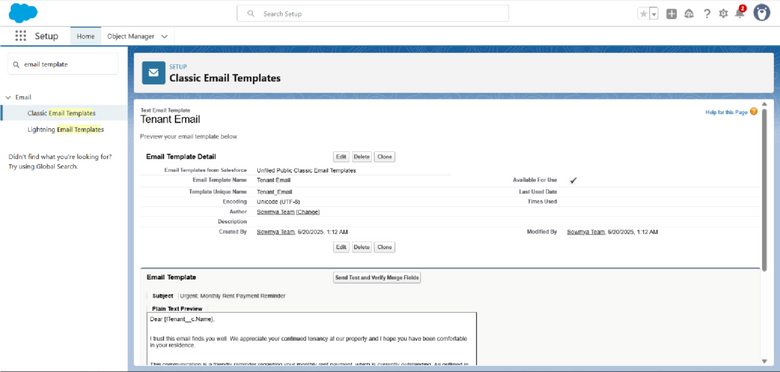


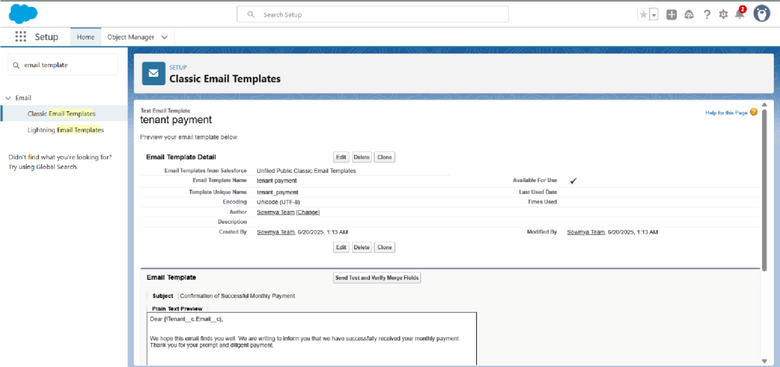
* + Built and tested email templates for leave request, approval, rejection, payment, and reminders





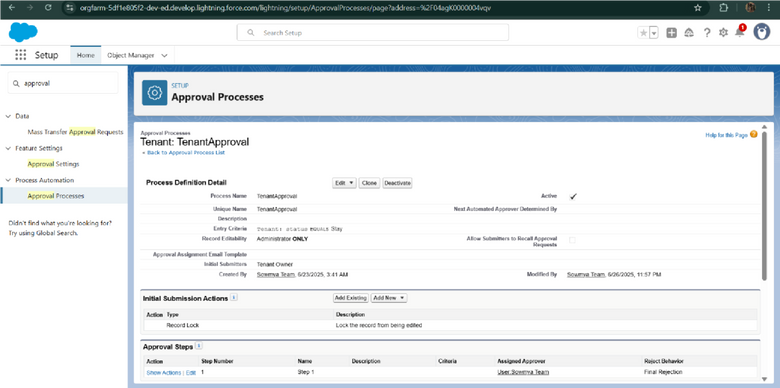




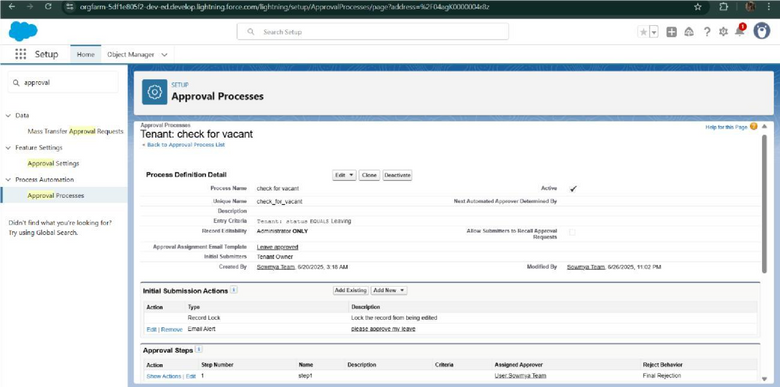


* + Approval Process creation

For Tenant Leaving:

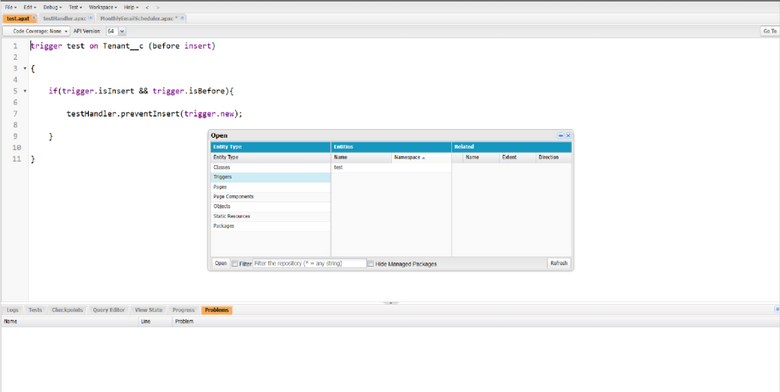


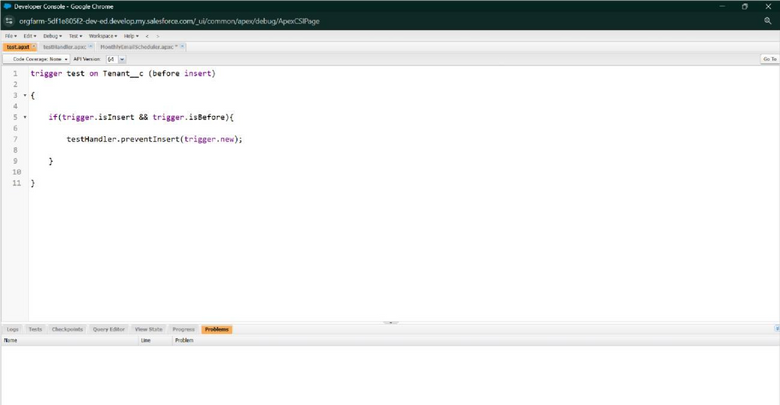
* For Check for Vacant:



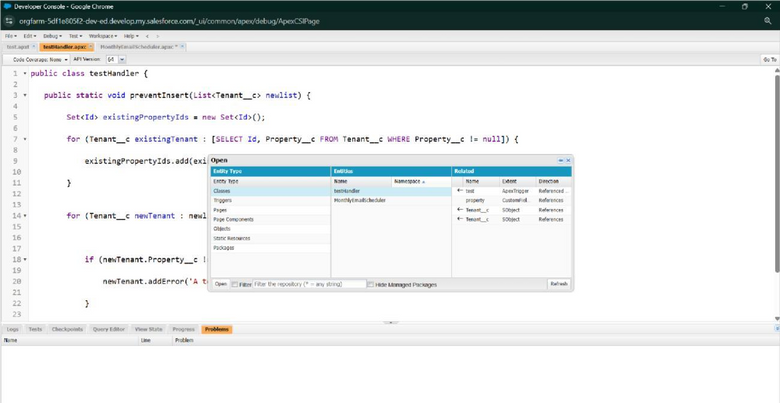
* Apex Trigger

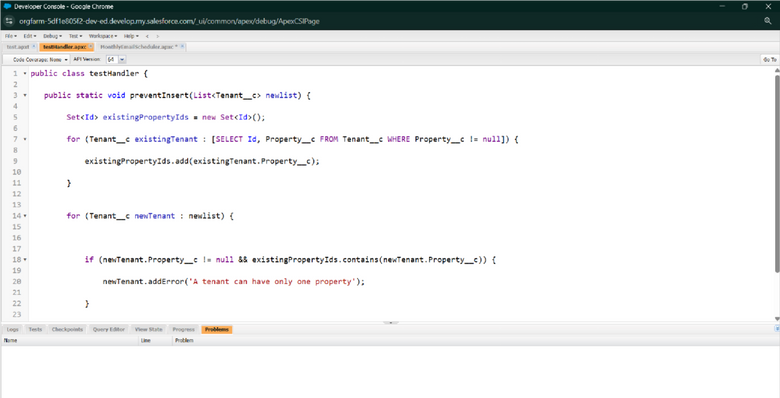
Create an Apex Trigger



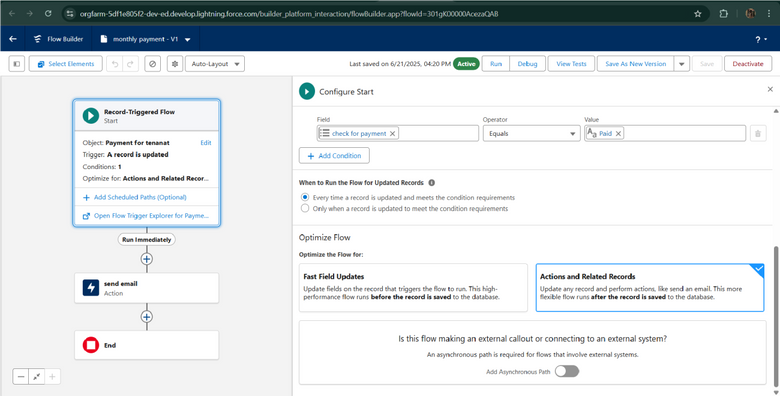


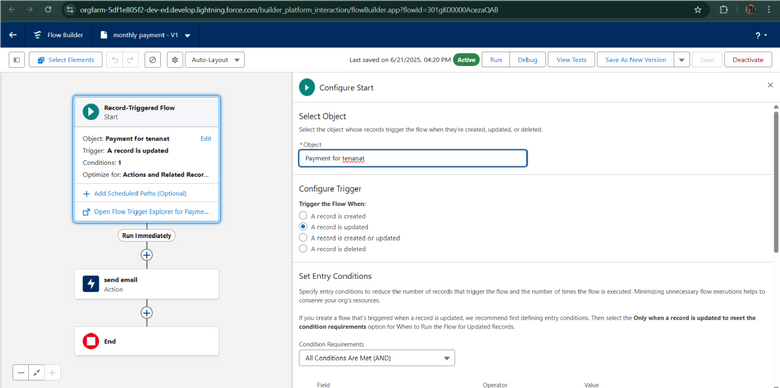
Create an Apex Handler class



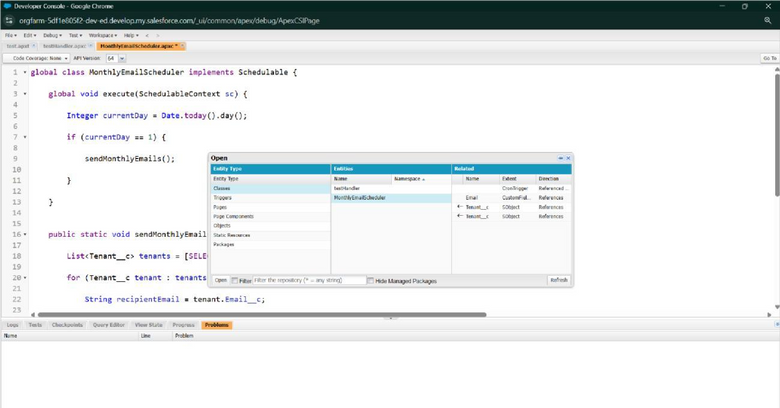


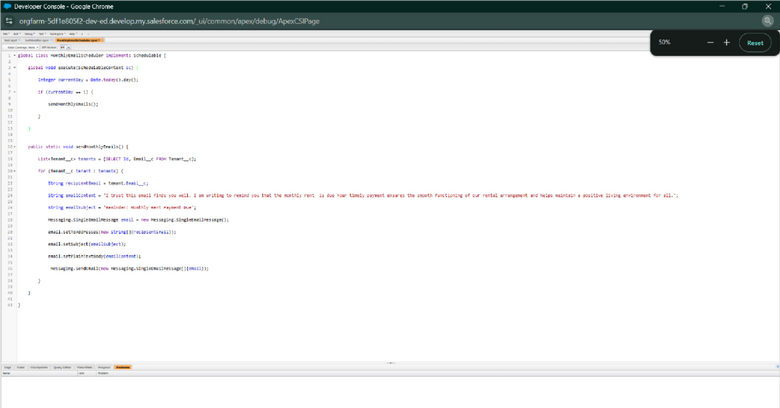
* + FLOWS



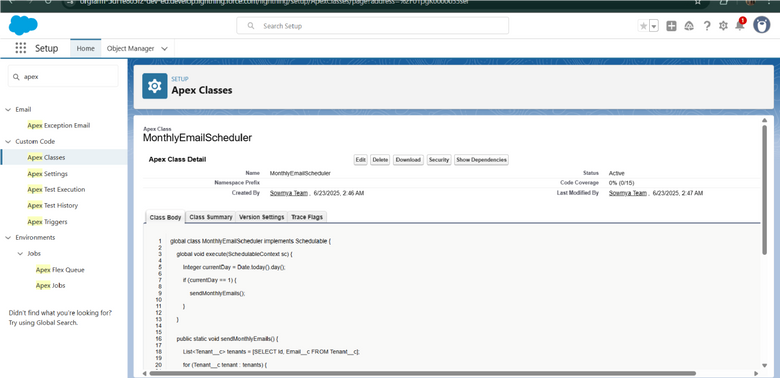


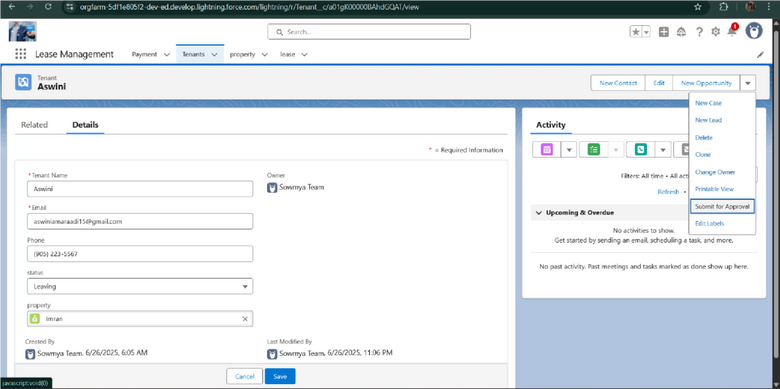
* + Schedule class: Create an Apex Class

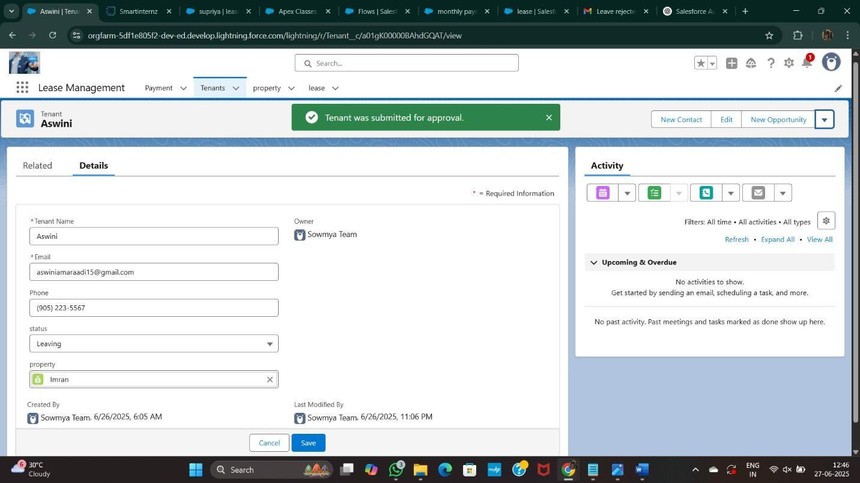


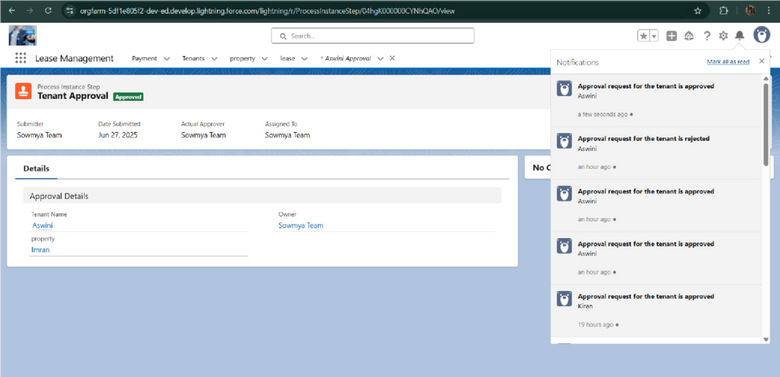


Schedule Apex class









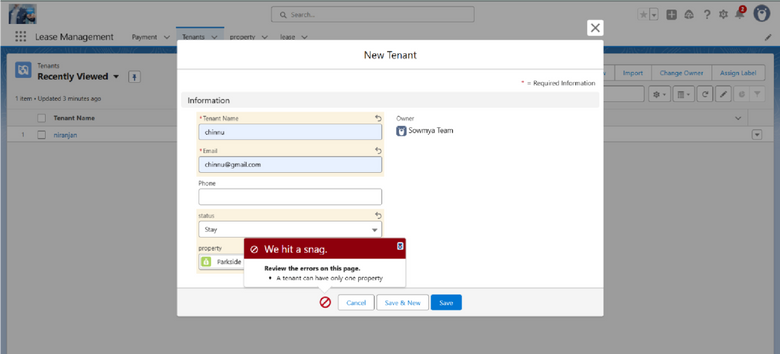


**FUNCTIONAL AND PERFORMANCE**

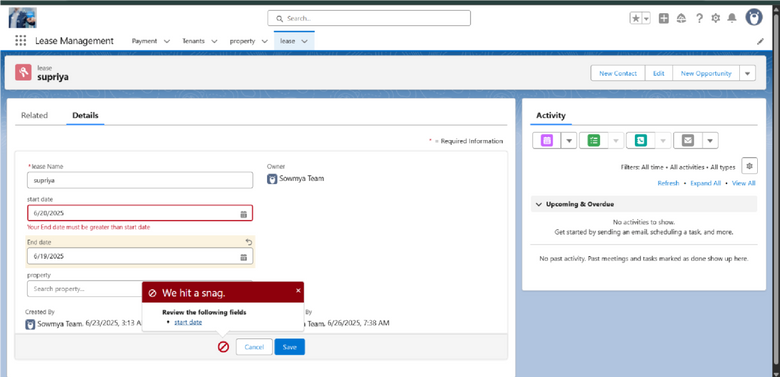
**TESTING**

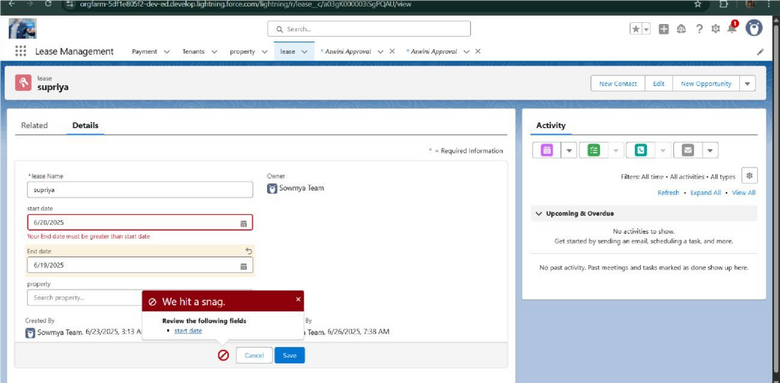
### Performance Testing

* Triggervalidationbyentering duplicate tenant-property records

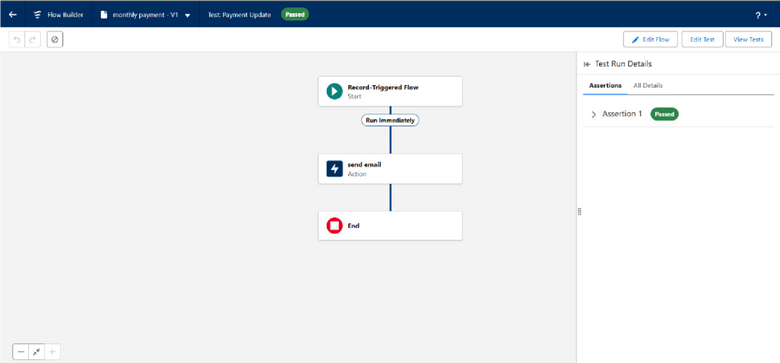


* Validation Rule checking





* + Test flows on payment update



* + - Approval process validated through email alerts and status updates



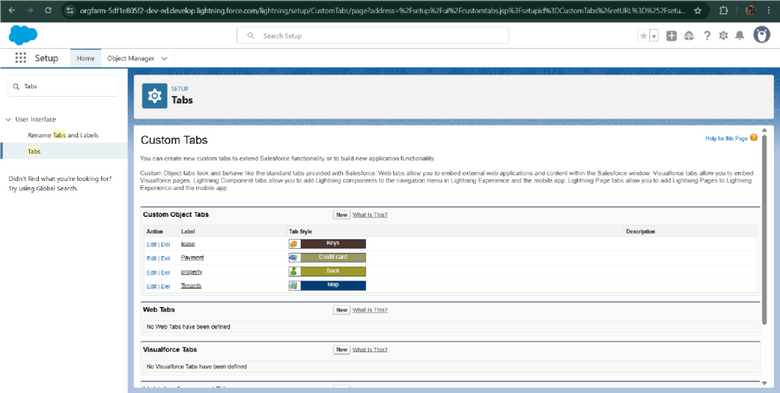




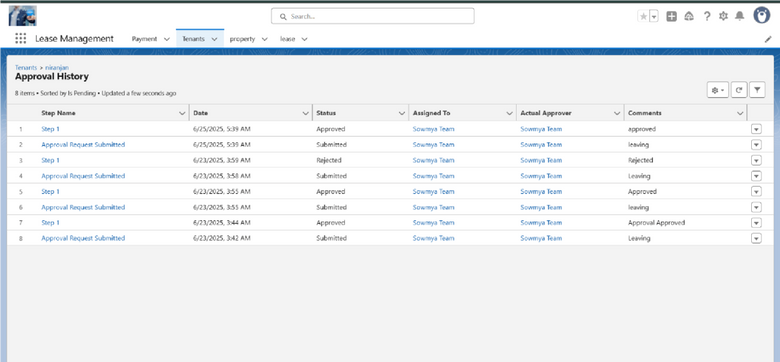
# RESULTS

## Output Screenshots

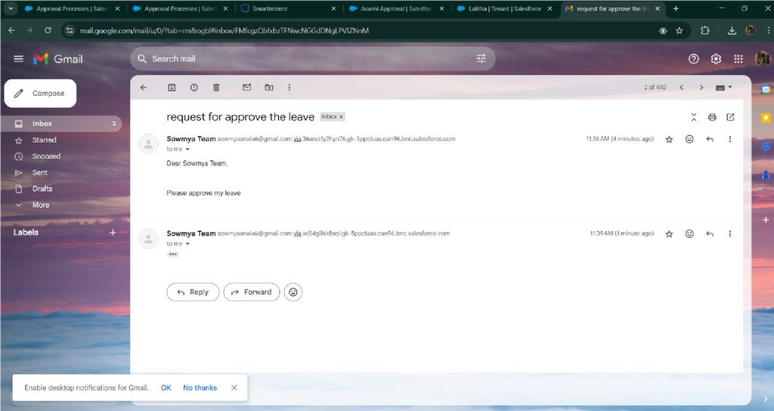
* Tabs for Property, Tenant, Lease, Payment



* Email alerts



* + Request for approve the leave



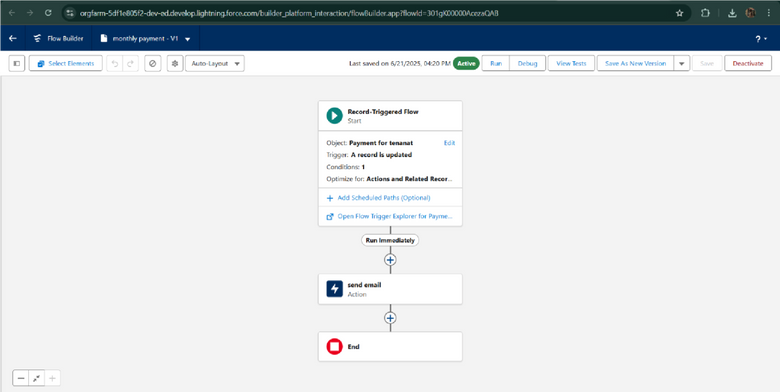
* + Leave approved



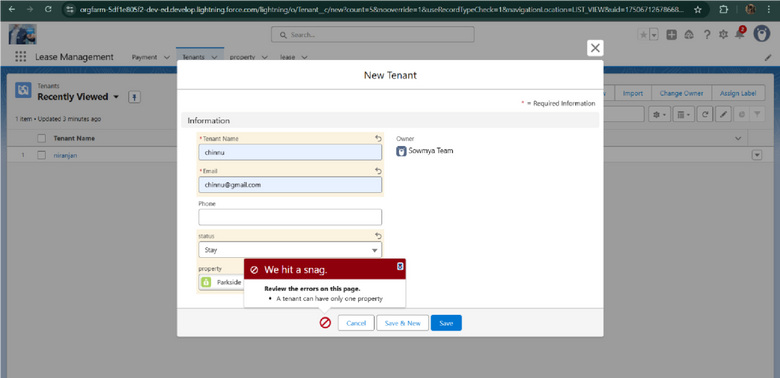
* Leave rejected



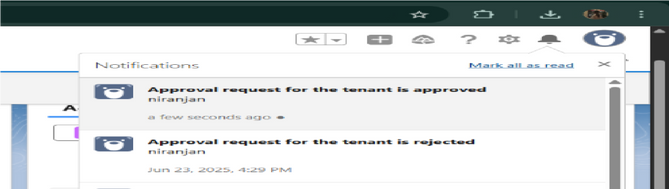
* Flow runs



* Trigger error messages



* Approval process notifications



.



# 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

* Source Code: Provided in Apex Classes and Triggers

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});

}

}