LeaseManagement

**CollegeName:SreeNarayanaGuruCollege,Chavadi,Coimbatore. College Code: Bru36**

**TEAMID:NM2025TMID26552**

**TEAMMEMBERS:**

**TeamLeaderName:MUHAMMEDHARSHADM.H Email:**[**achuachuzz479@gmail.com**](mailto:achuachuzz479@gmail.com)

**TeamMember:MANUR**

**Email:** [**manur9963@gmail.com**](mailto:manur9963@gmail.com) **TeamMember:MUHAMMEDAJMALA**

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

### INTRODUCTION

ProjectOverview

TheLeaseManagementSystemisaSalesforce-basedapplicationdesignedtostreamlinetheprocesses 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.



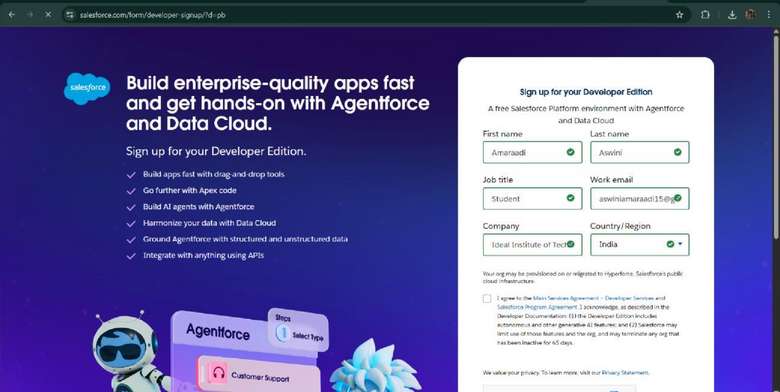
## Purpose

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.

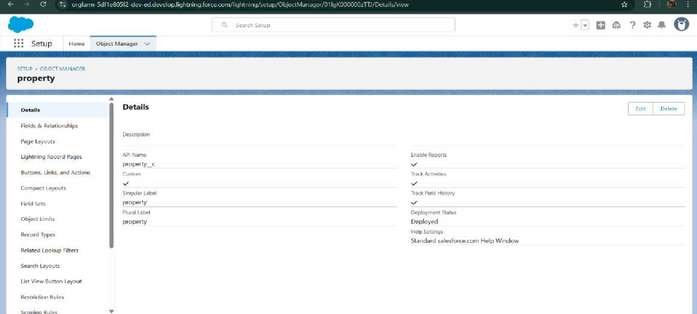
DEVELOPMENTPHASE

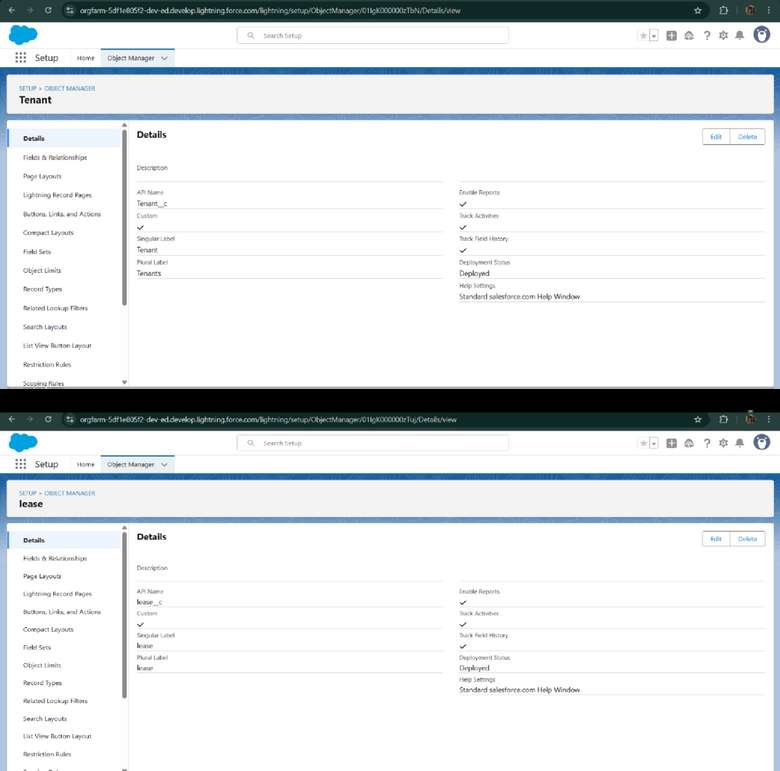
CreatingDeveloperAccount:

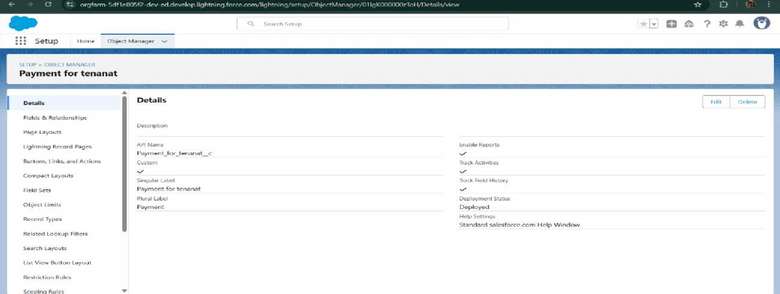
usingthisURL-https://[www.salesforce.com/form/developer-signup/?d=pb](http://www.salesforce.com/form/developer-signup/?d=pb)



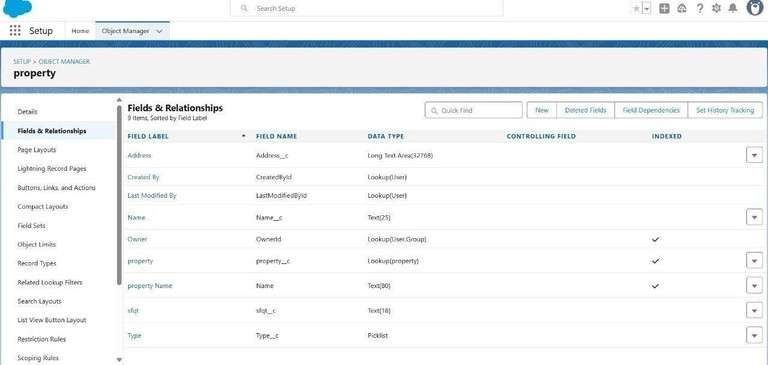
* Createdobjects:Property,Tenant,Lease,Payment

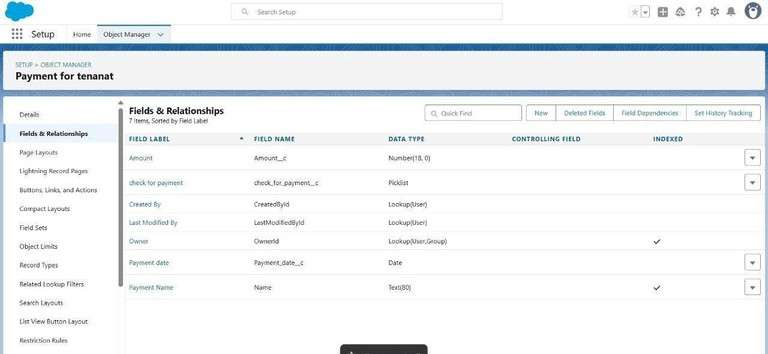


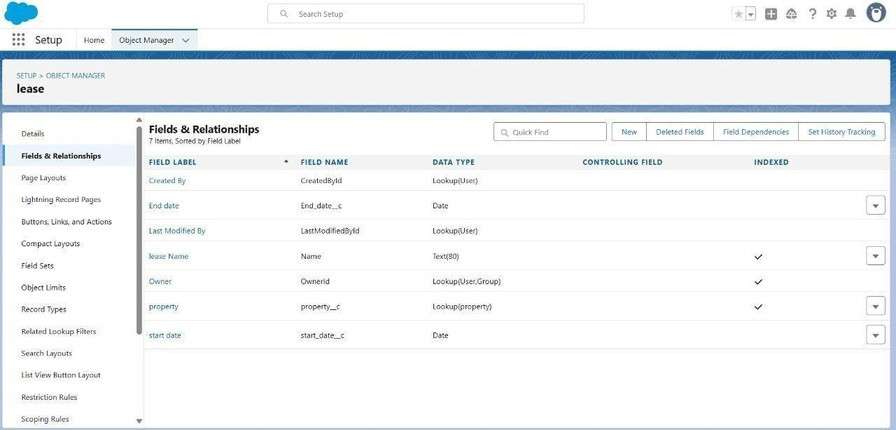


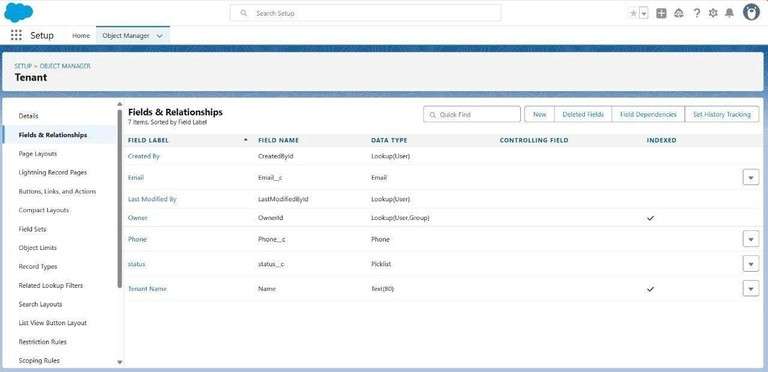


* Configuredfields and relationships

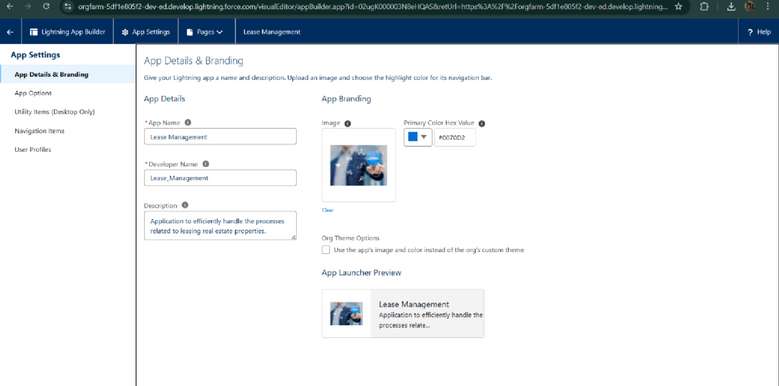


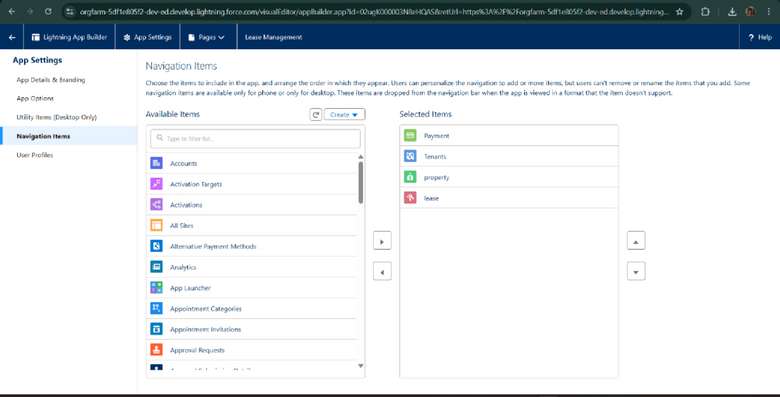


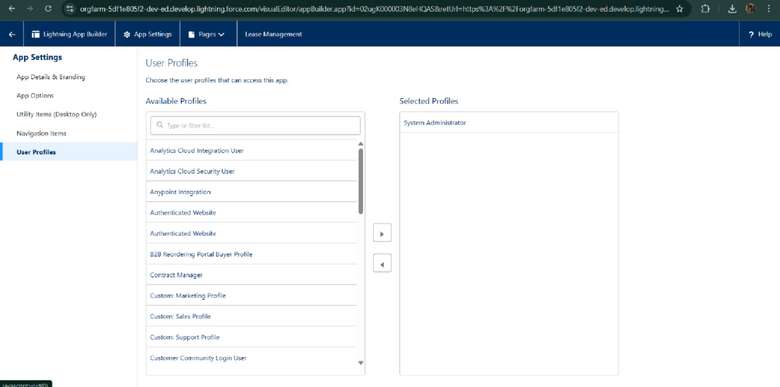


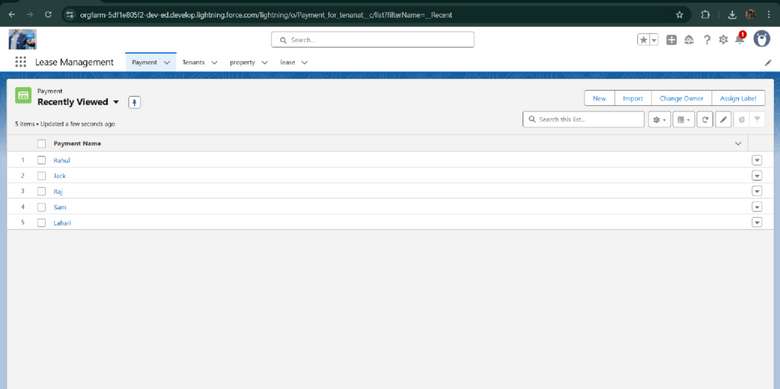


* DevelopedLightningAppwithrelevanttabs

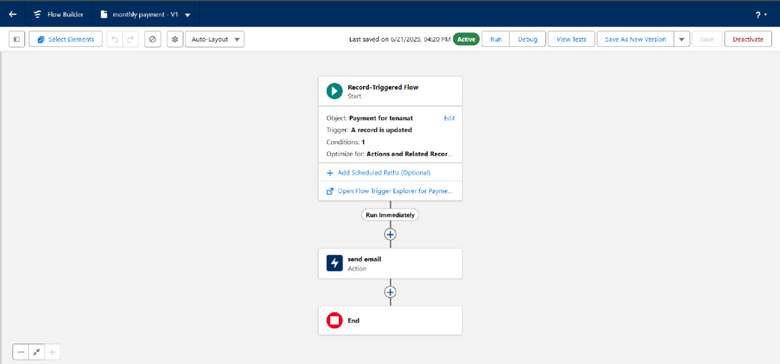




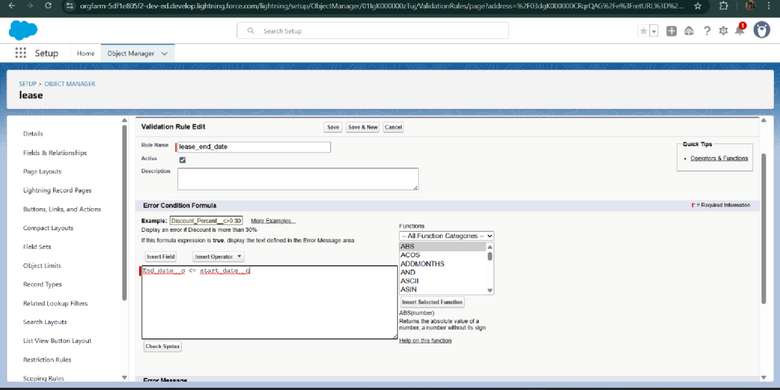


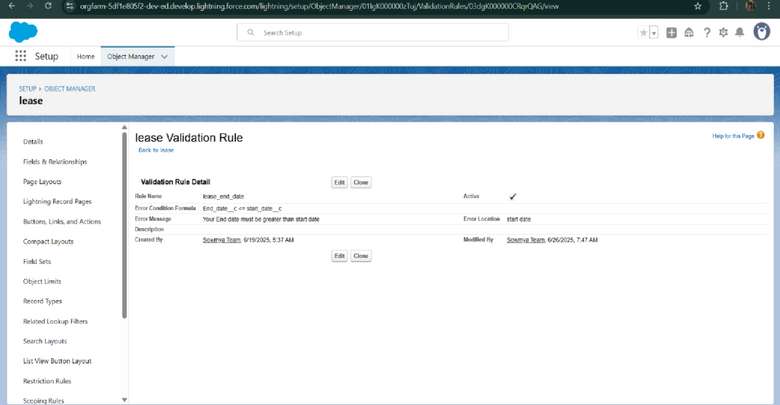


* + Implemented Flows for monthly rent and payment success

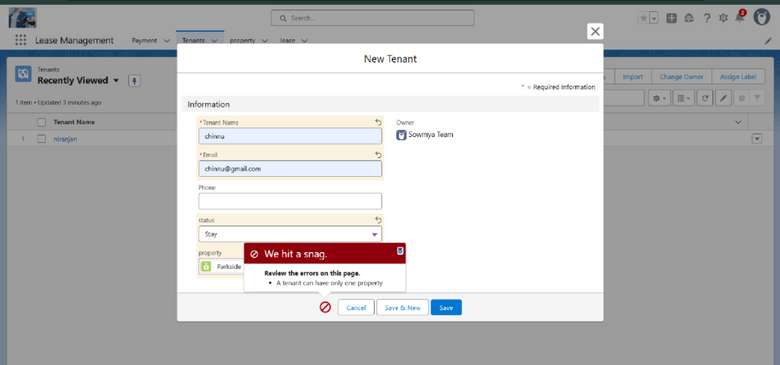


* .TocreateavalidationruletoaLeaseObject

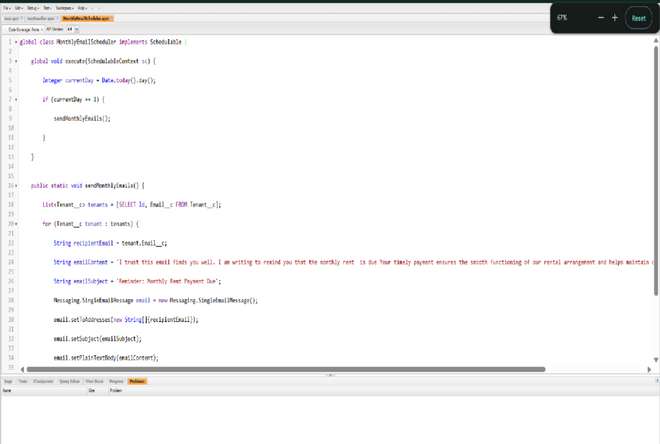




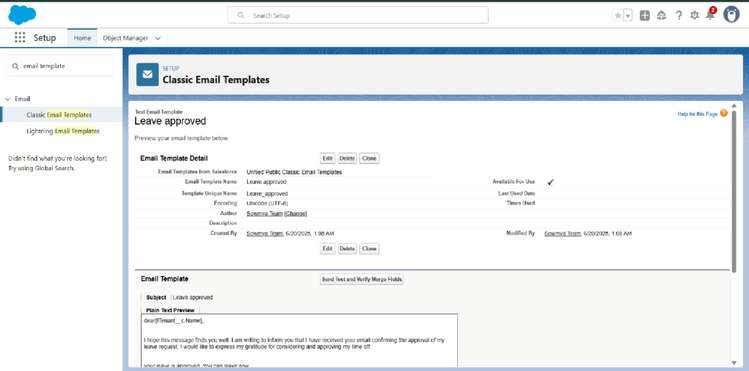
* AddedApextriggertorestrictmultipletenantsperproperty

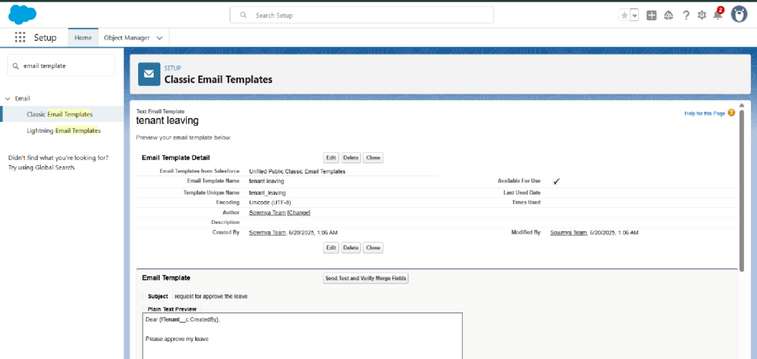


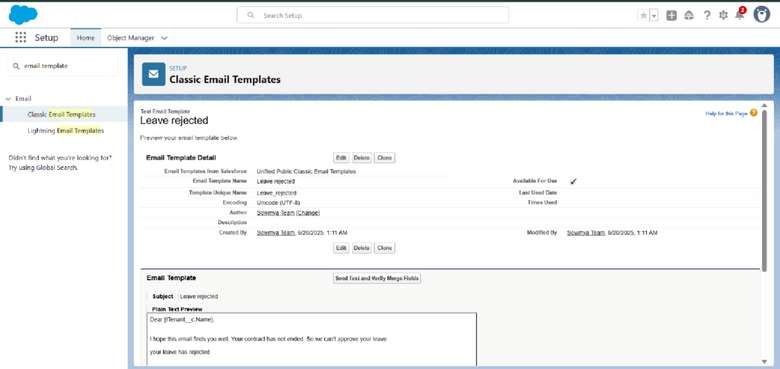
* ScheduledmonthlyreminderemailsusingApexclass

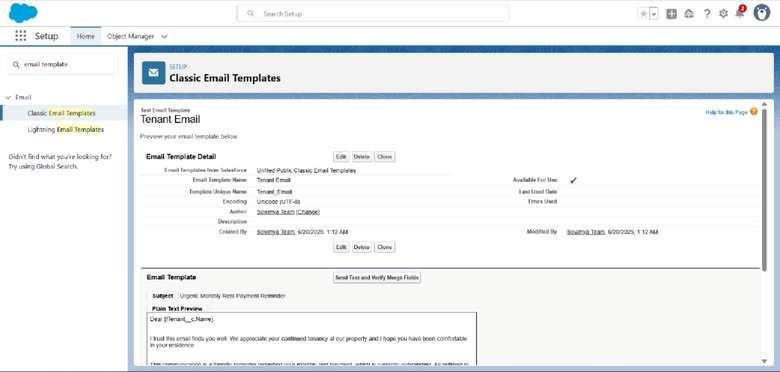


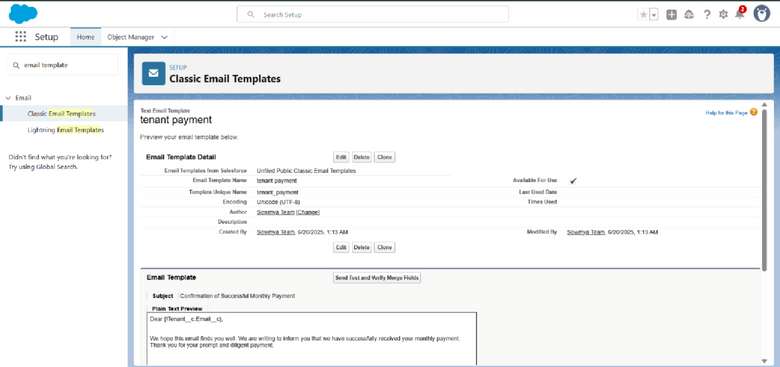
* + Built and tested email templates for leave request, approval, rejection, payment, and reminders





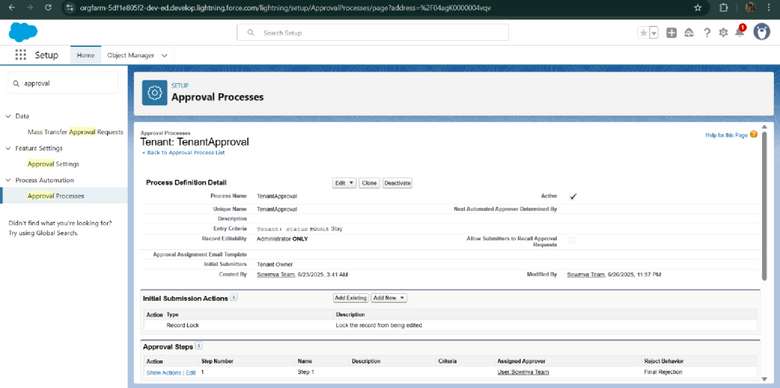




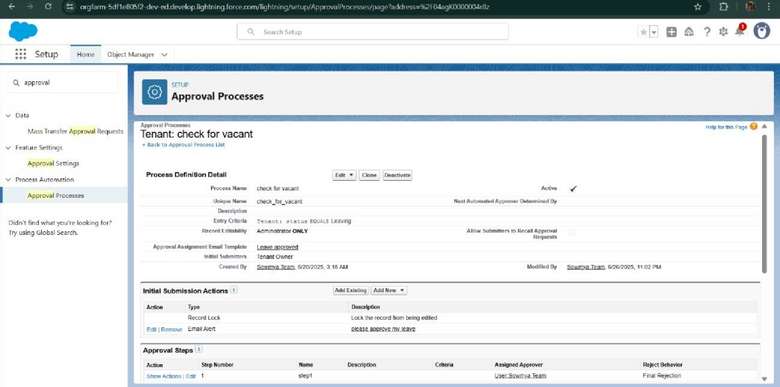


* + ApprovalProcesscreation

ForTenantLeaving:

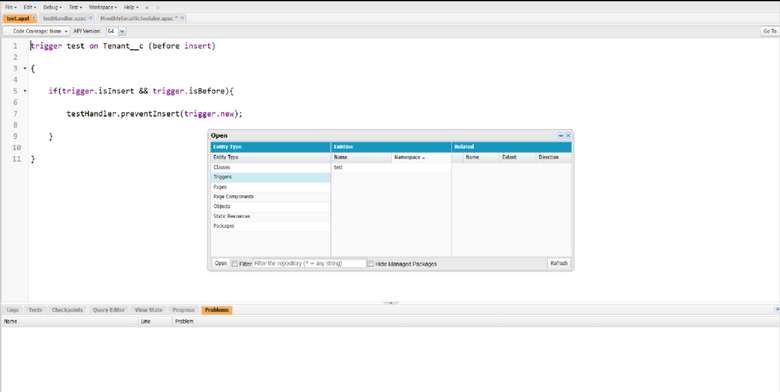


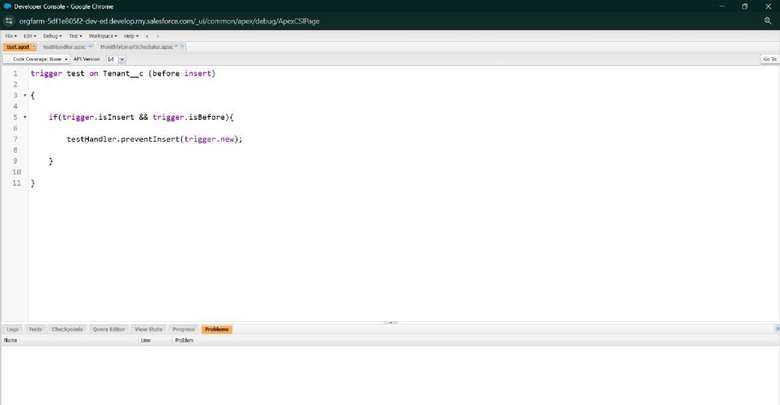
* ForCheckforVacant:



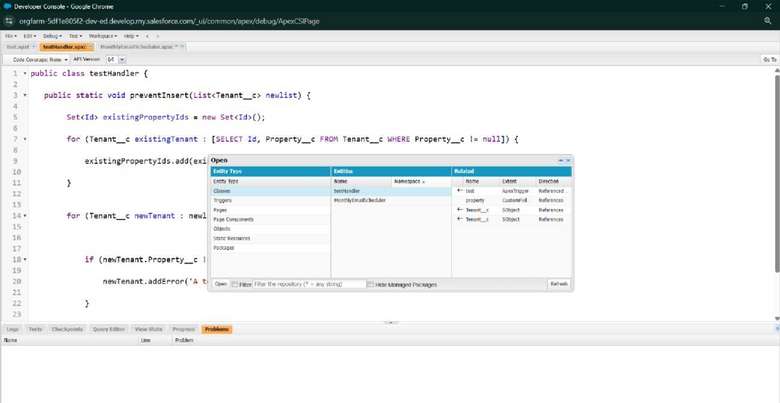
* ApexTrigger

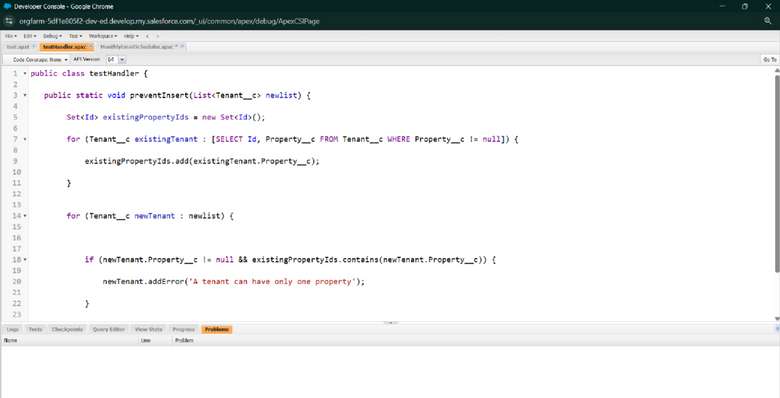
CreateanApexTrigger



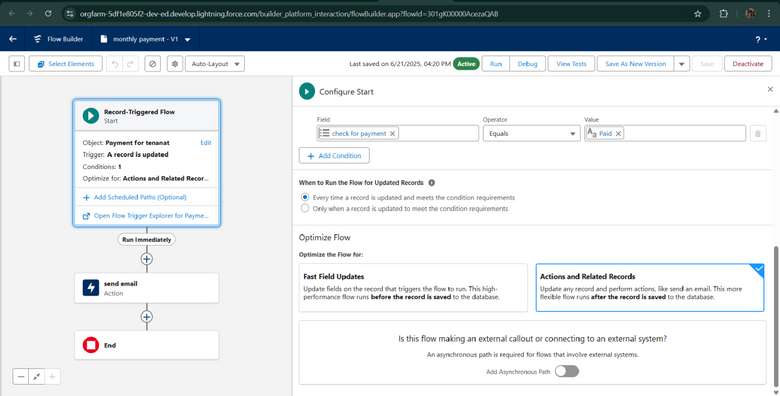


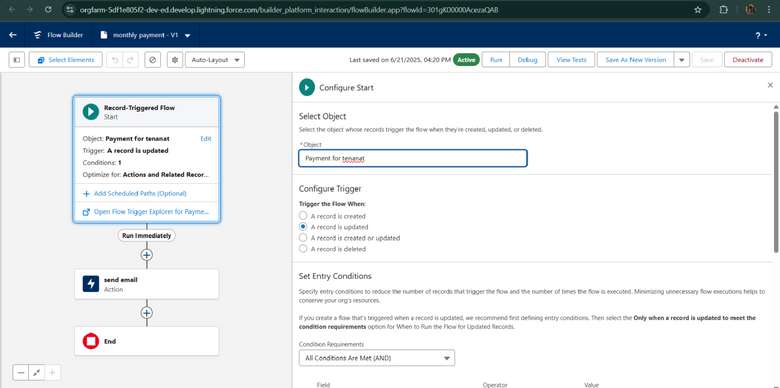
CreateanApexHandlerclass



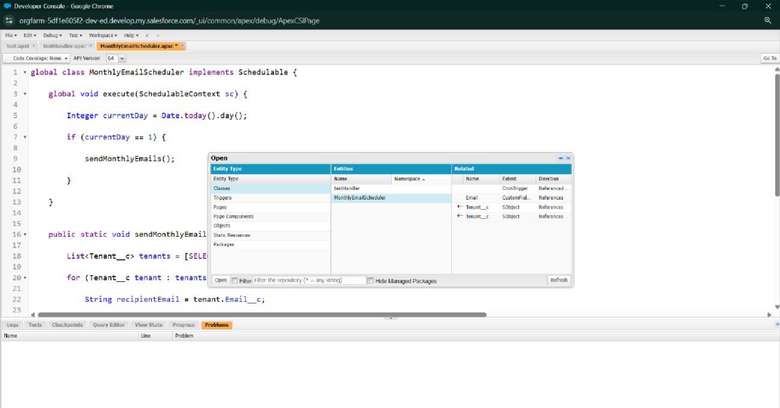


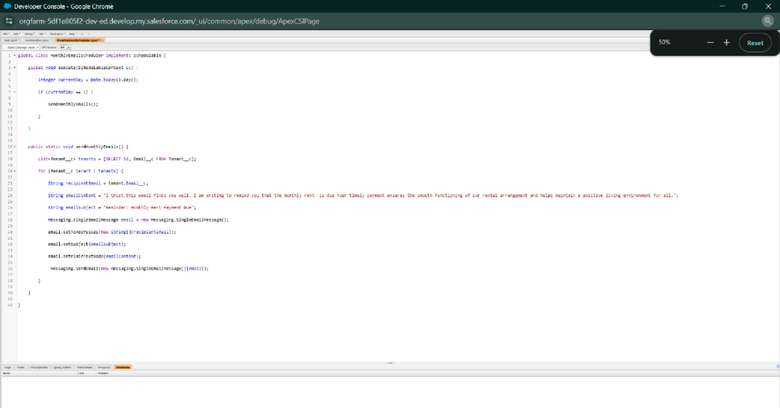
* + FLOWS



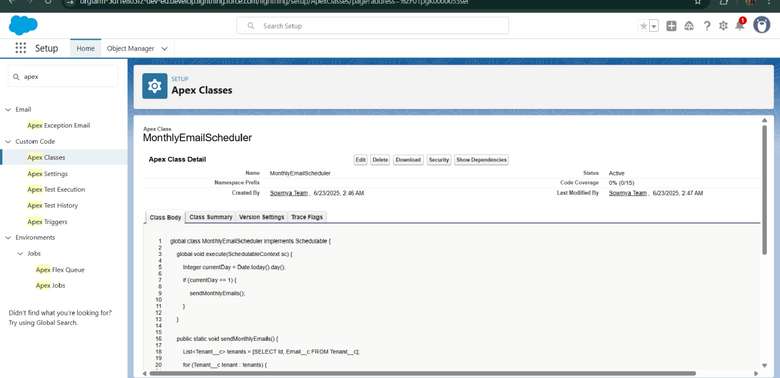


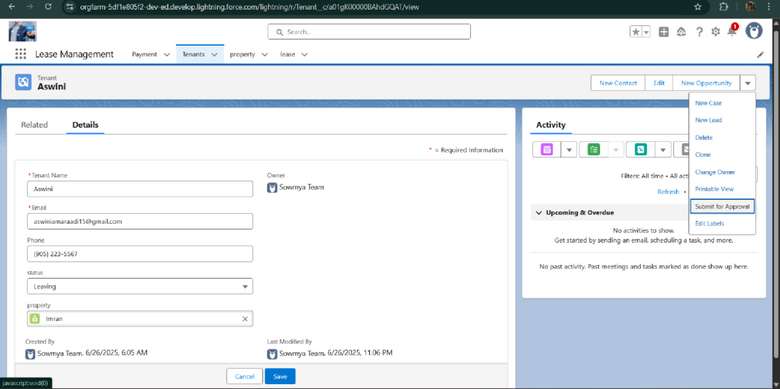
* + Schedule class: CreateanApexClass

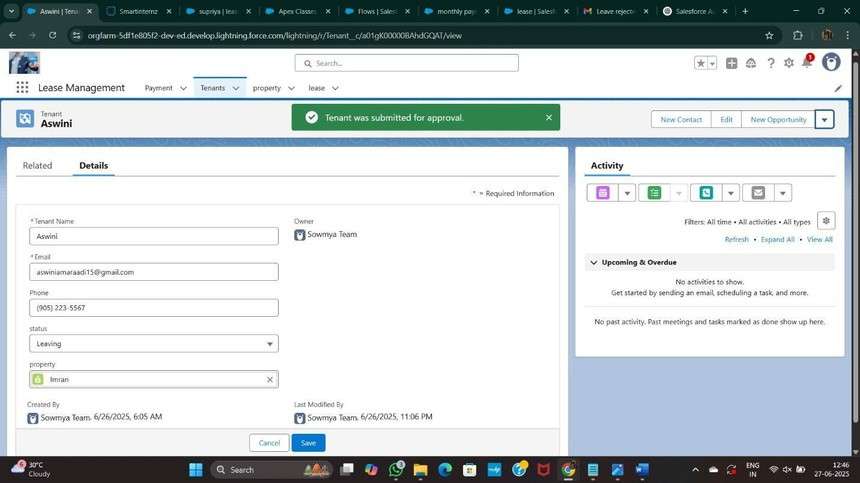


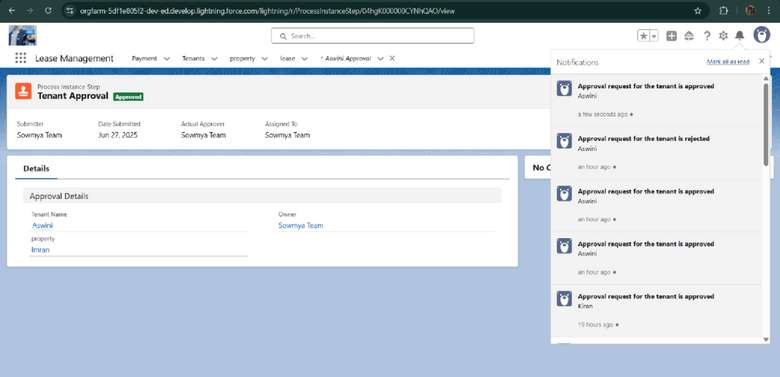


ScheduleApexclass









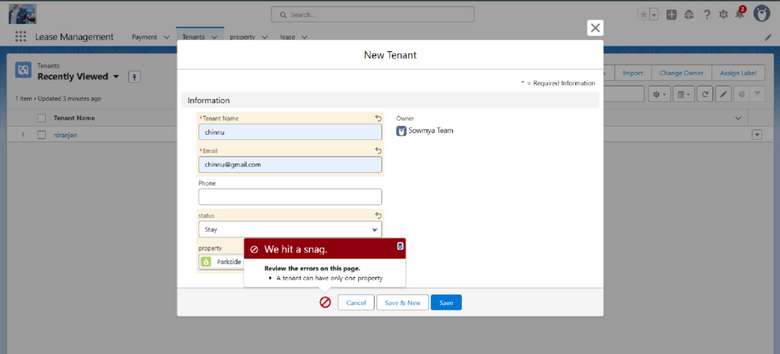


**FUNCTIONALANDPERFORMANCE**

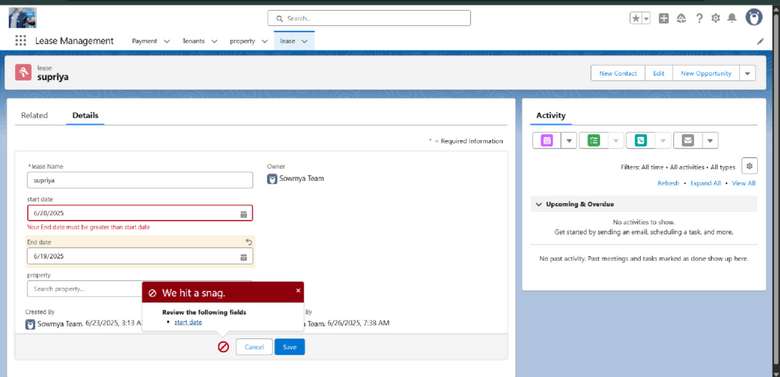
**TESTING**

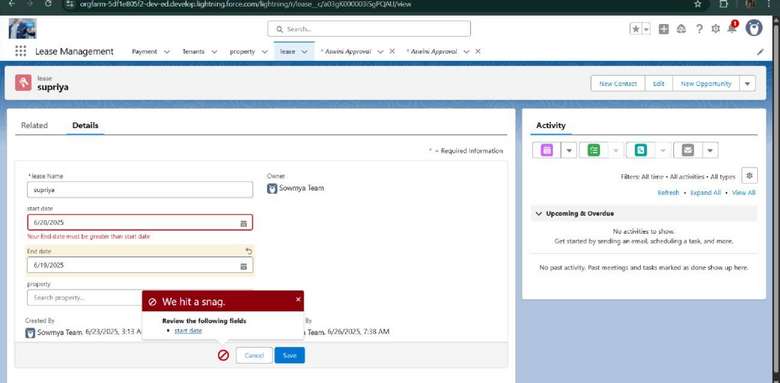
### PerformanceTesting

* Triggervalidationbyenteringduplicatetenant-propertyrecords

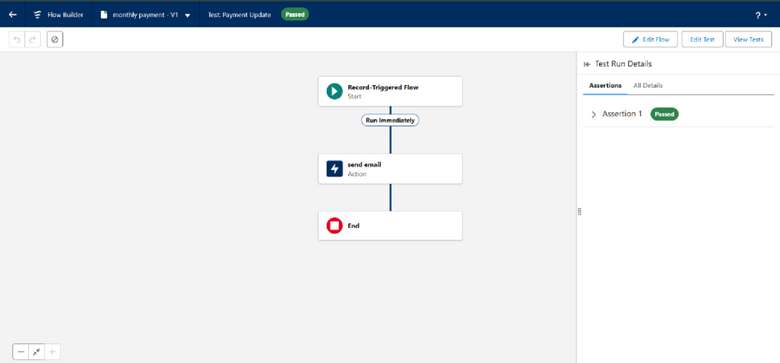


* ValidationRulechecking

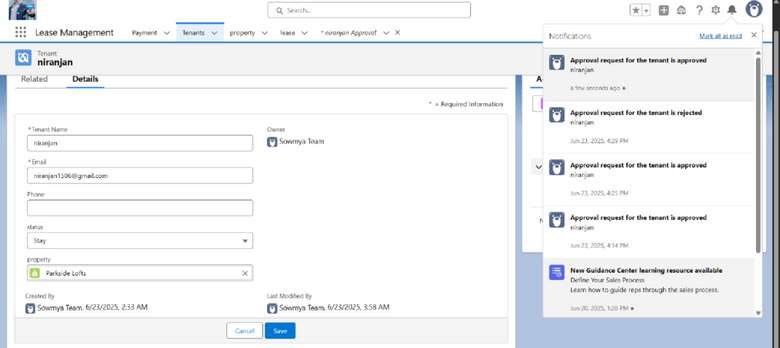


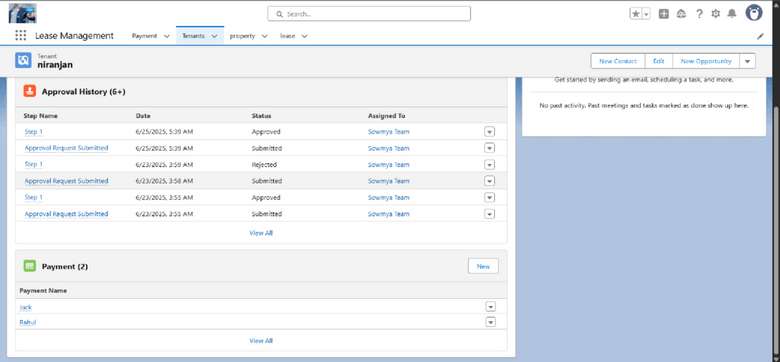


* + Testflowsonpaymentupdate



* + - Approvalprocessvalidatedthroughemailalertsandstatusupdates



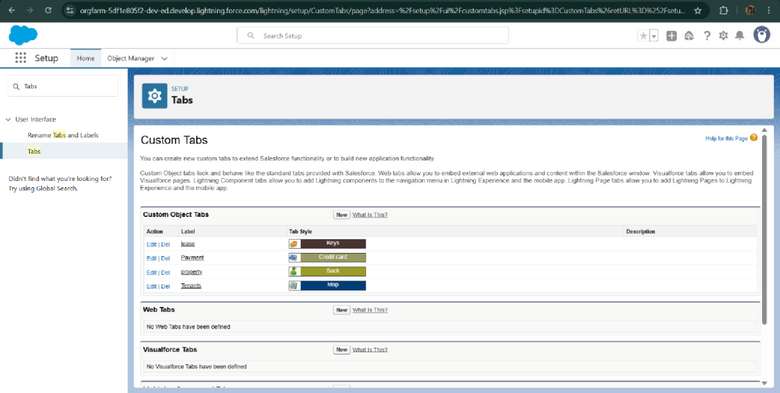




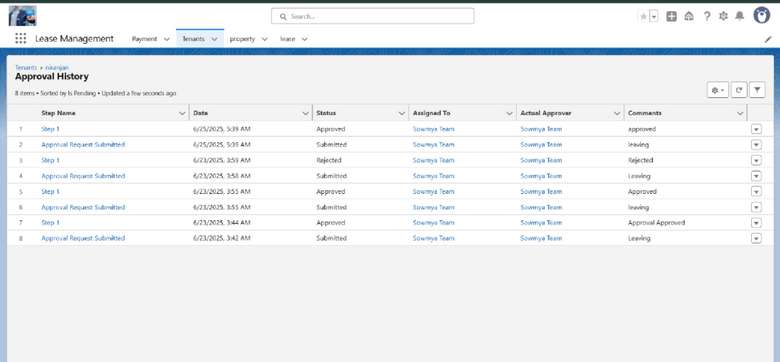
# RESULTS

## Output Screenshots

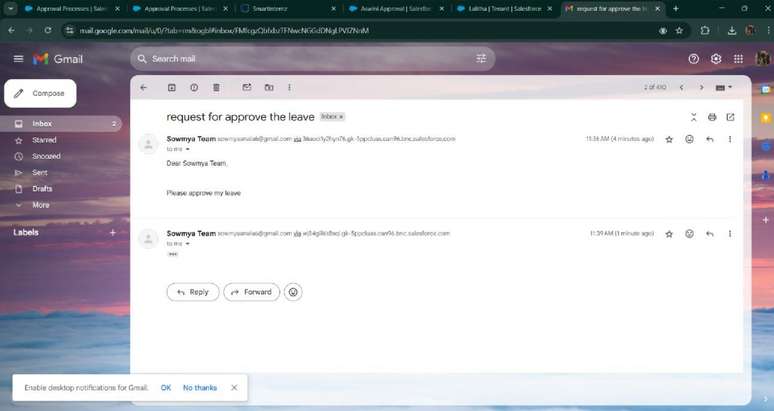
* TabsforProperty,Tenant,Lease,Payment



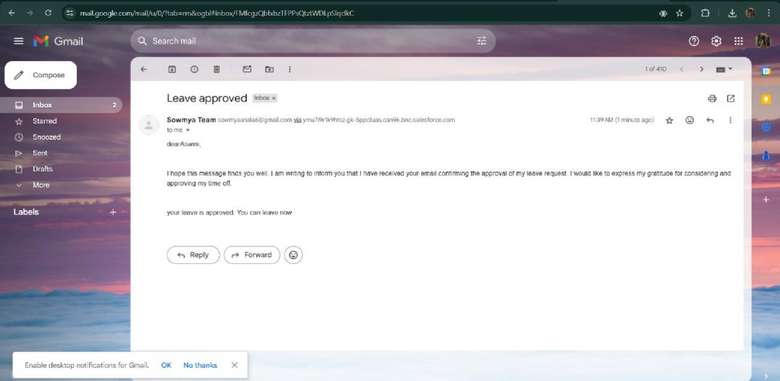
* Emailalerts



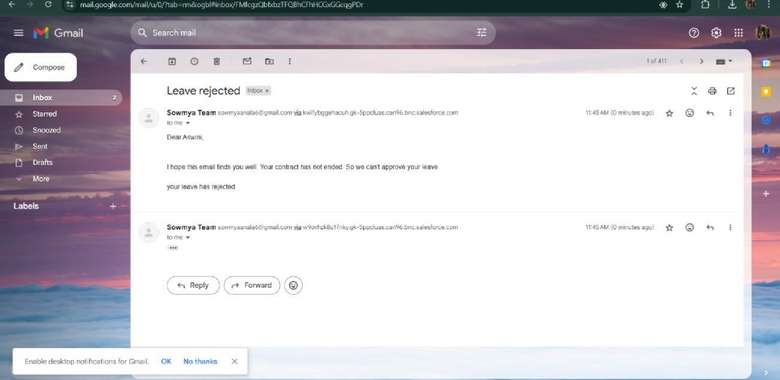
* + Requestforapprovetheleave



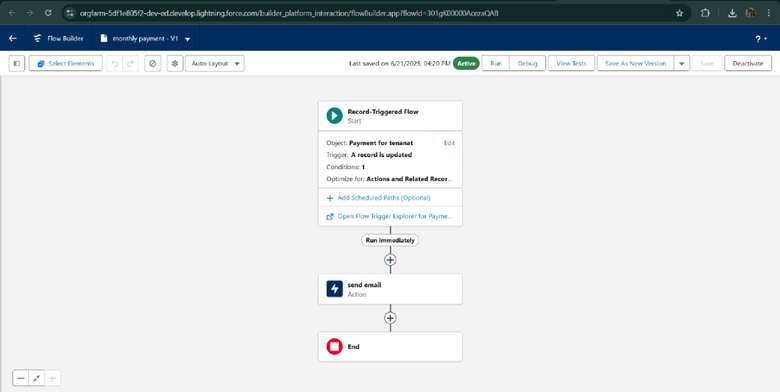
* + Leaveapproved



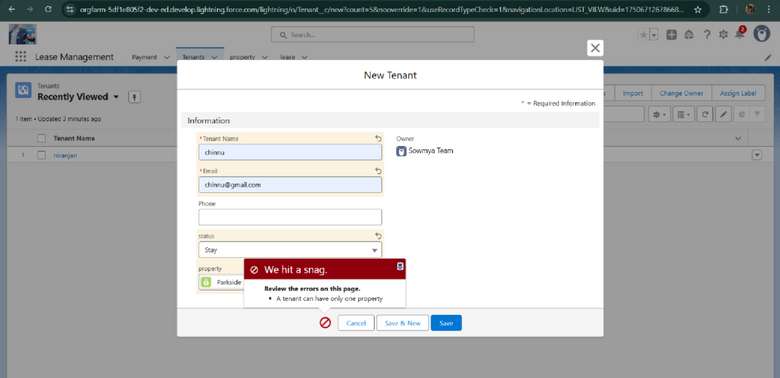
* Leaverejected



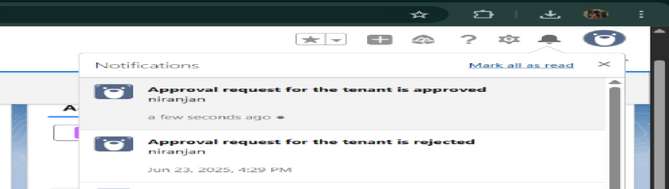
* Flowruns



* Triggererrormessages



* Approvalprocessnotifications



.



# CONCLUSION

TheLeaseManagementSystemsuccessfullystreamlinestheoperationsofleasingthrougha structured, automatedSalesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.



## APPENDIX

* SourceCode:ProvidedinApexClassesand Triggers

Test.apxt:

triggertestonTenantc(beforeinsert){if(trigger.isInsert&& trigger.isBefore){ testHandler.preventInsert(trigger.new);

}}

testHandler.apxc:



public class testHandler { public static voidpreventInsert(List<Tenantc> newlist) Set<Id>

{

existingPropertyIds=newSet<Id>()

for(TenantcexistingTenant:[SELECTId,PropertycFROMTenantc

WHEREPropertyc!=null])

{

existingPropertyIds.add(existingTenant.Propertyc;

}for (Tenantc newTenant

:newlist){

if (newTenant.Propertyc != null && existingPropertyIds.contains(newTenant.Propertyc)) { newTenant.addError('A

tenantcanhaveonlyone

property');

} }

}

}

MothlyEmailScheduler.apxc:

globalclassMonthlyEmailSchedulerimplementsSchedulable

{globalvoidexecute(SchedulableContextsc){Integer currentDay = Date.today().day(); if (currentDay == 1) { sendMonthlyEmails();

}

} public static void sendMonthlyEmails() { List<Tenantc> tenants = [SELECT Id, Emailc FROM Tenantc]; for (Tenantc tenant :

tenants){

StringrecipientEmail=tenant.Emailc;

String emailContent = 'I trust this email finds you well. I am writing to remind you

thatthemonthlyrentisdueYourtimelypaymentensuresthesmoothfunctioningof

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(newString[]

{recipientEmail}); email.setSubject(emailSubject); email.setPlainTextBody(emailContent); Messaging.sendEmail(newMessaging.SingleEmailMessage[]

} {email});

}

}