LEASE MANAGEMENT

**College name**: Sri Ramakrishna College Of Arts and Science for Women

**College code**: Bru28

**Team id:** NM2025TMID24283

TEAM MEMBERS:

**Team Leader :** HARINISRI R

**Email id**: srcw2326j118@srcw.ac.in

**Team member 1 :** BHAVANA DHARSHINI S

**Email id**: srcw2326j110@srcw.ac.in

**Team member 2 :** DHATCHAYINI P

**Email id**: srcw2326j159@srcw.ac.in

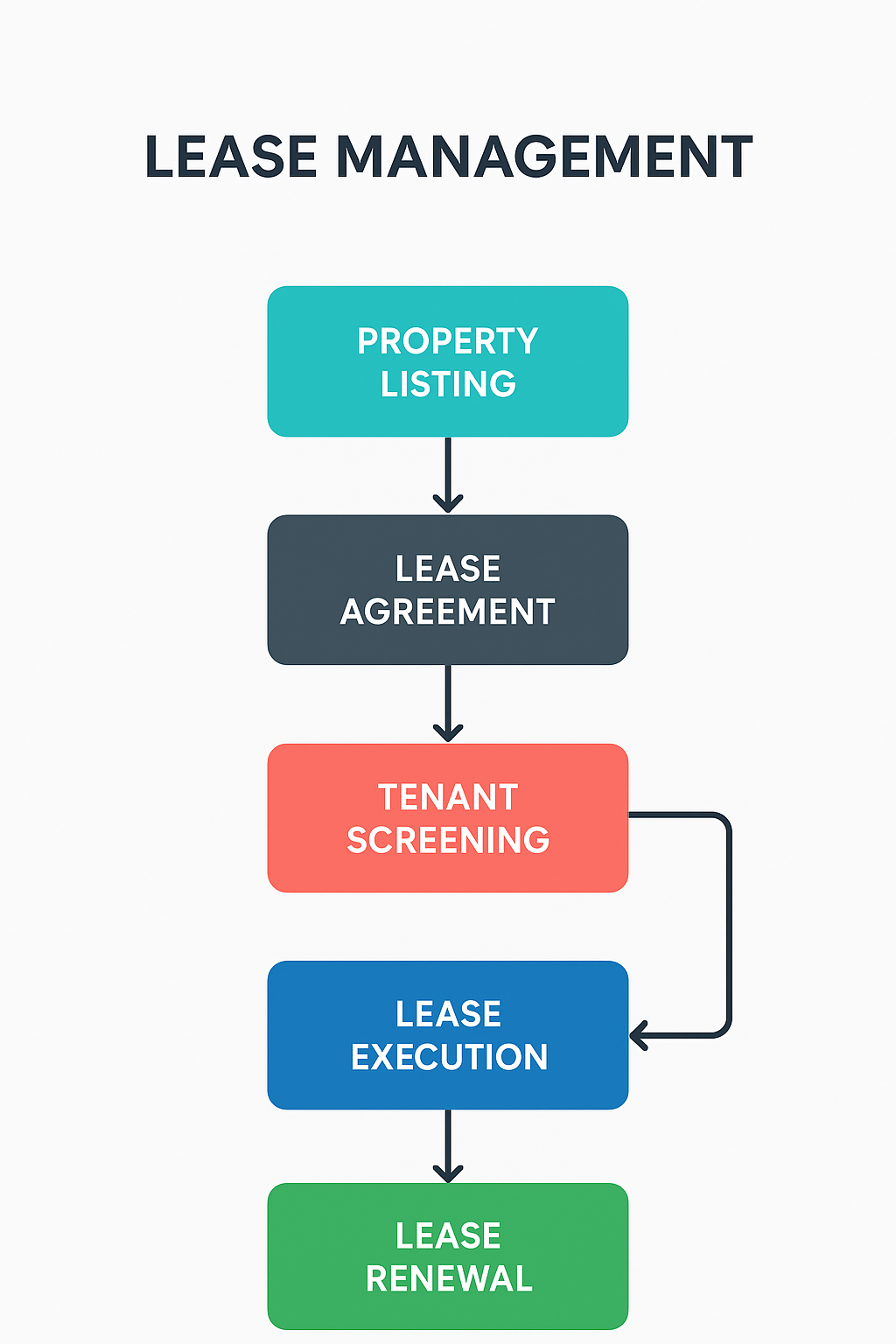
**Team member 3 :** ILAKKIYA R

**Email id**: srcw2326j121@srcw.ac.in

**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 associate

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

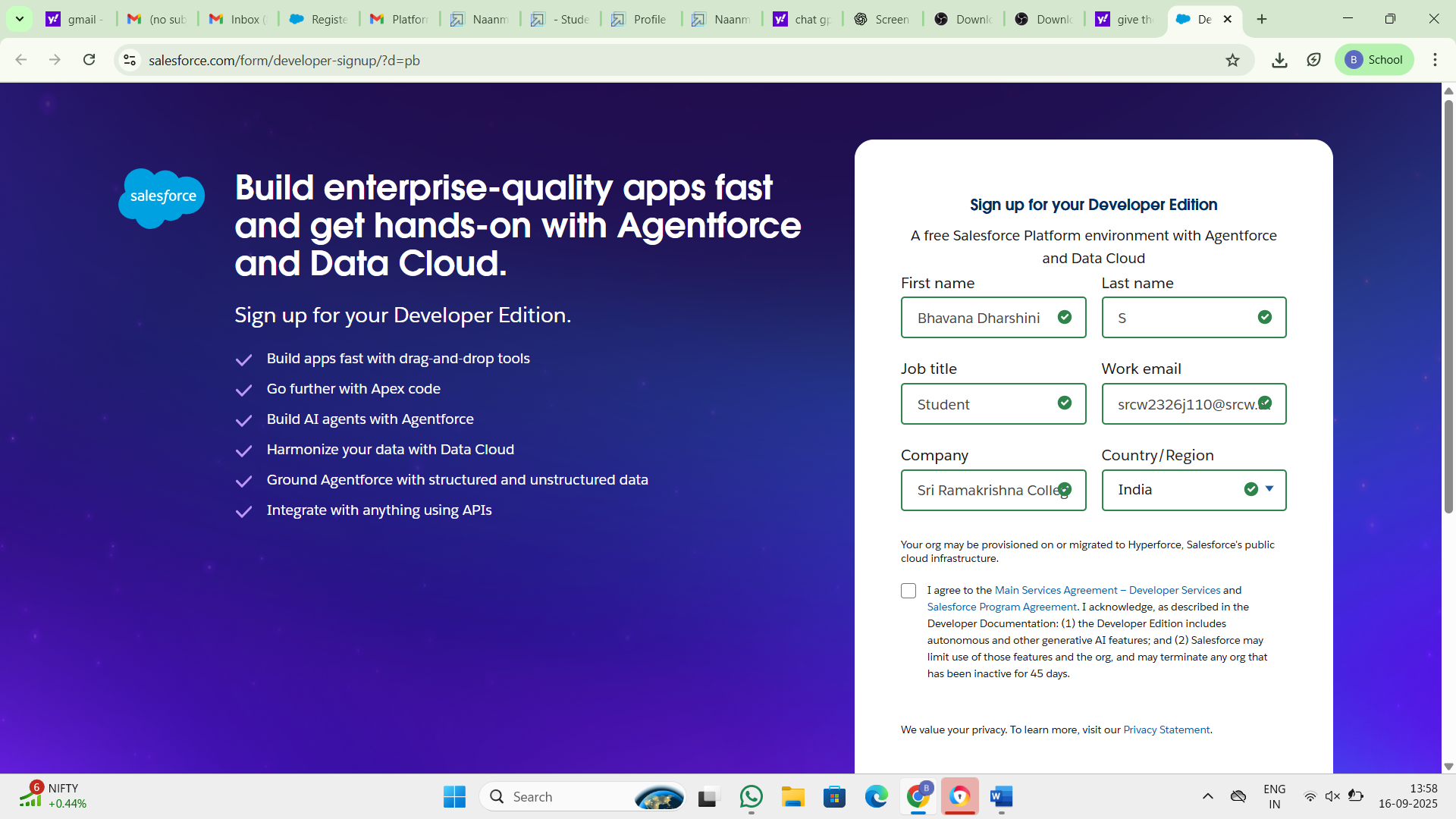
**1.2 PURPOSE:**

The purpose of the Lease Management project is to streamline and automate the end-to-end process of handling property leases. It helps in systematically managing property listings, creating accurate lease agreements, performing proper tenant screening, executing leases efficiently, and ensuring timely lease renewals. By following this structured workflow, the project aims to reduce manual errors, save time, and provide a clear and organized approach to managing properties, tenants, and payments within a single platform.

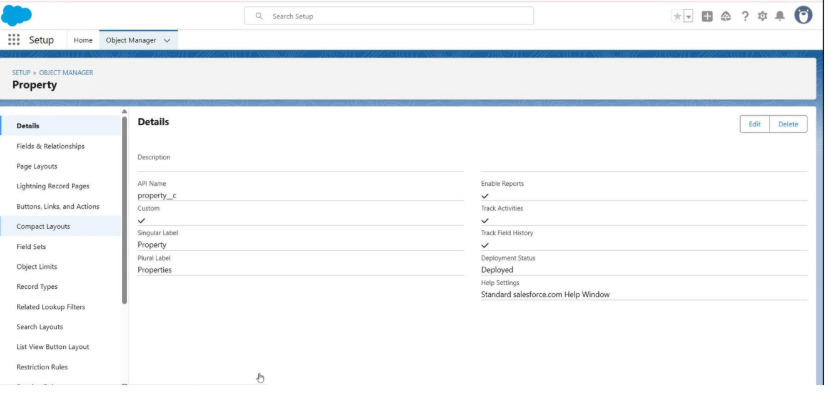
**DEVELOPMENT PHASE:**

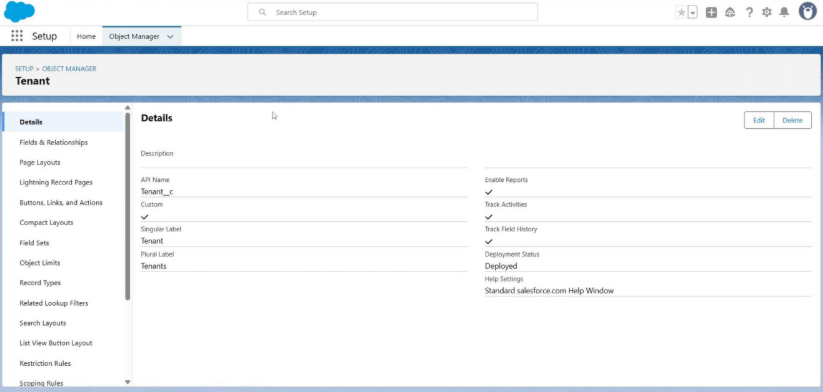
**CREATING A DEVELOPER ACCOUNT**:

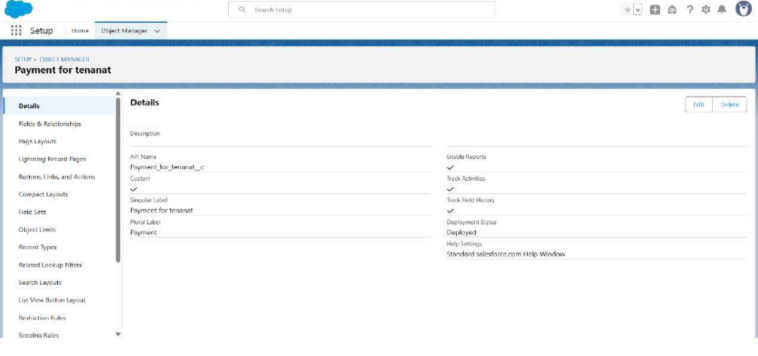
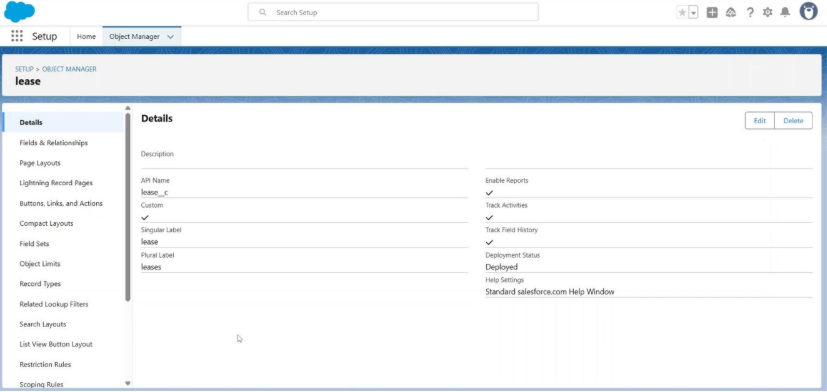
BY USING THE URL: https://www.salesforce.com/form/developer-signup/?d=pb



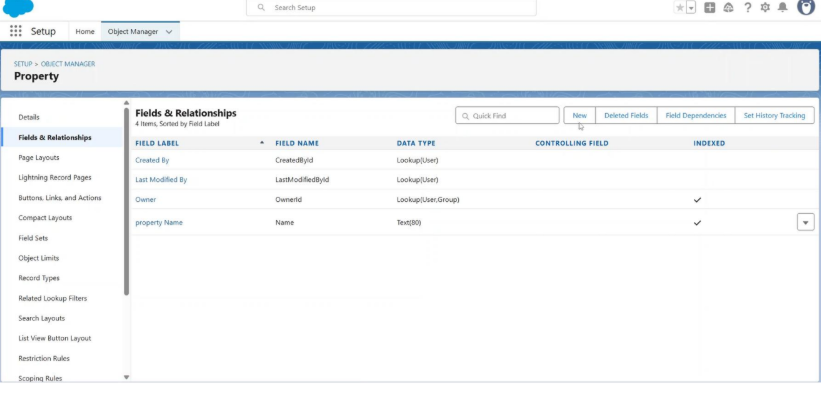
* Created objects : Property, Tenant,Lease,Payment

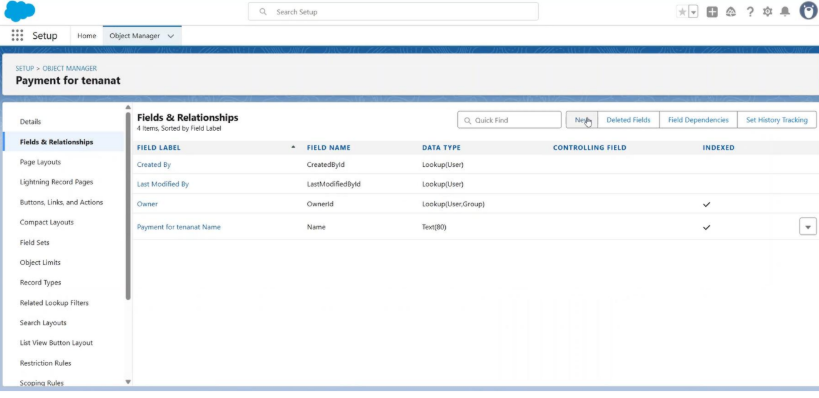


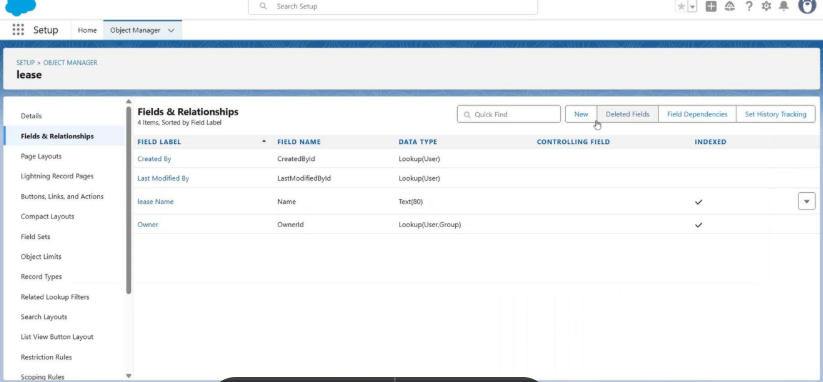


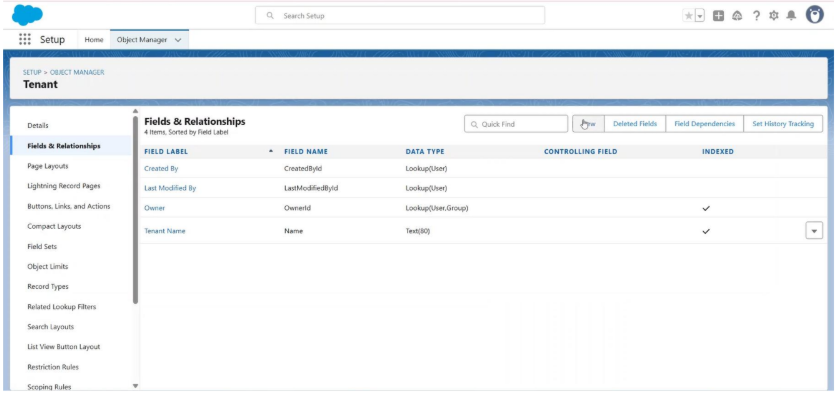


* Configured fields and relationships.

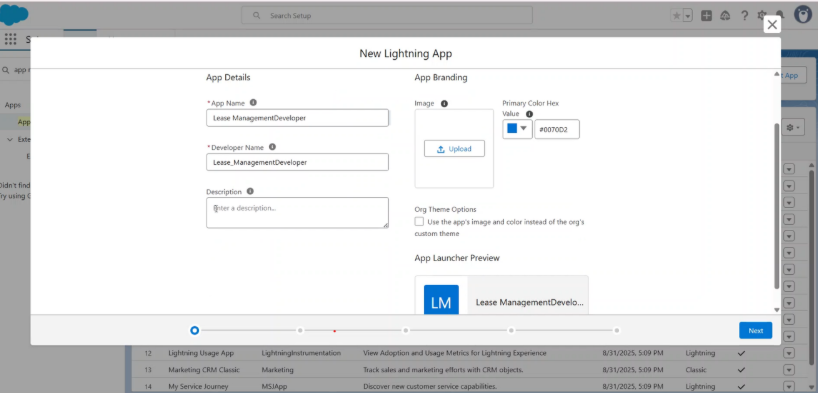


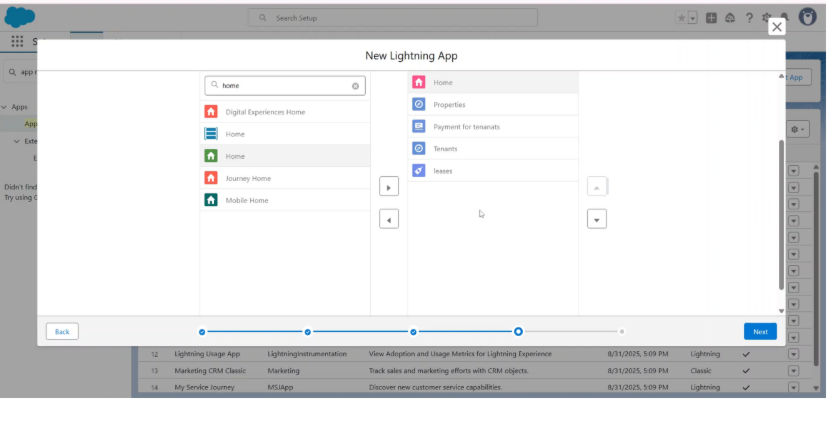


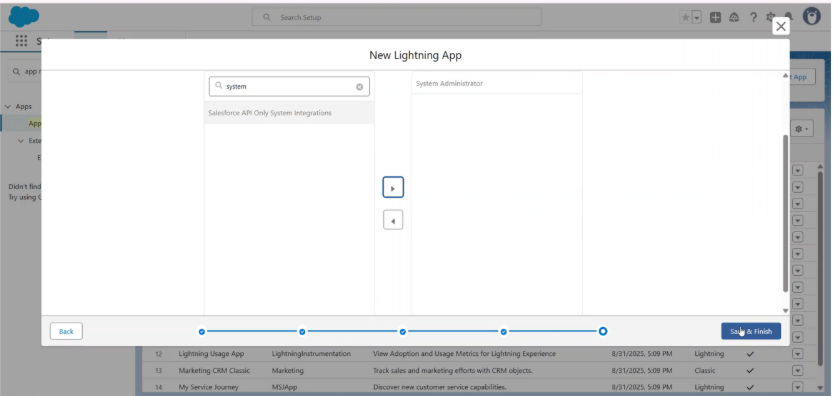


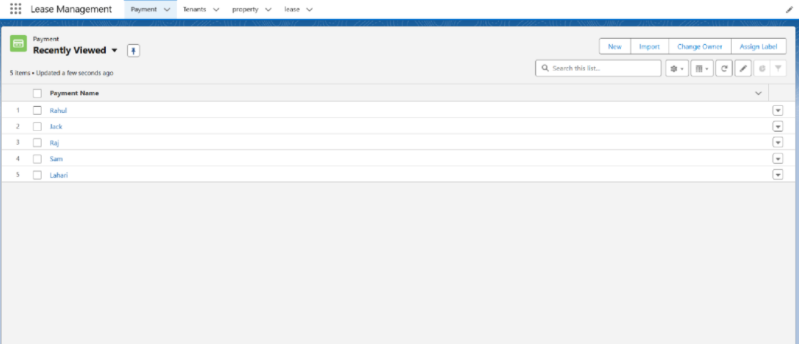


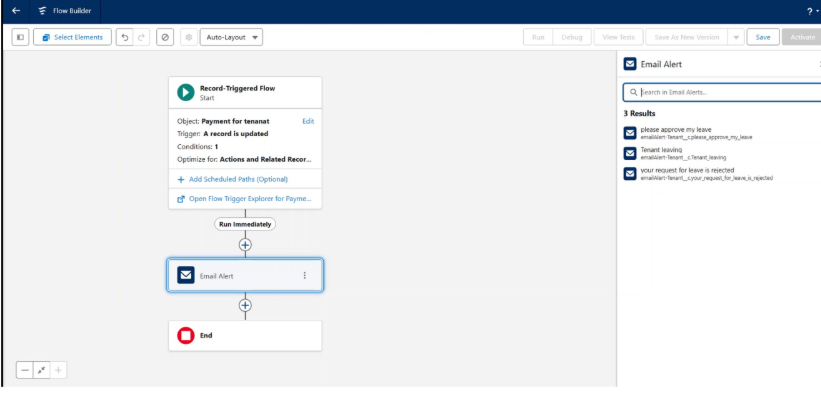
• Developed Lightning App with relevant tabs



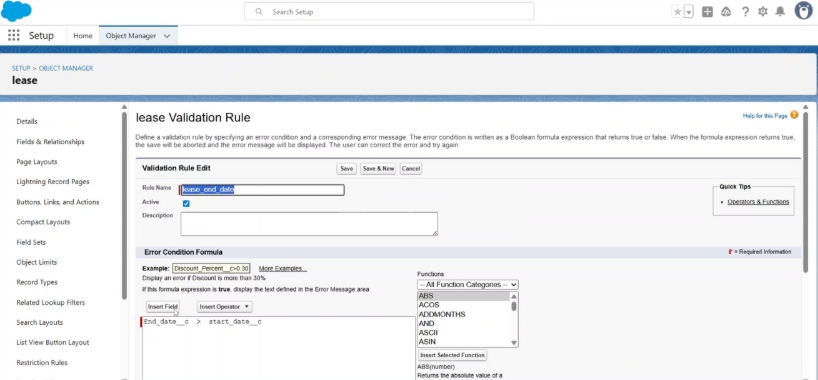


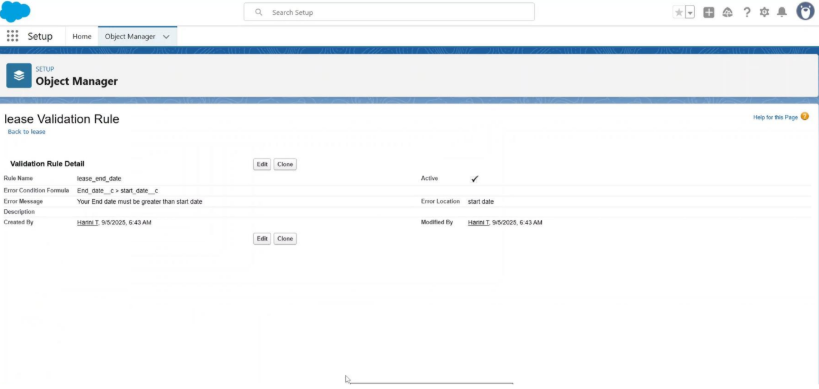




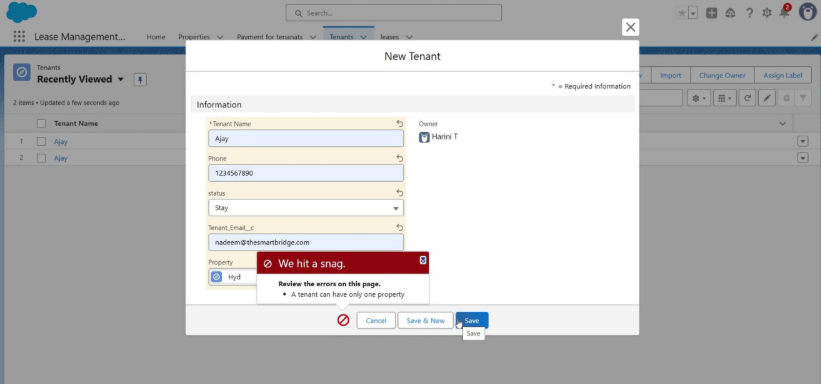
● Implemented Flows for monthly rent and payment success

● To create a validation rule to a Lease Object

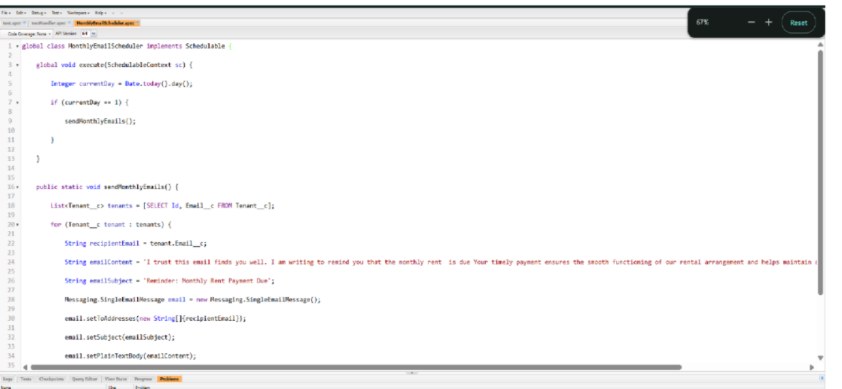




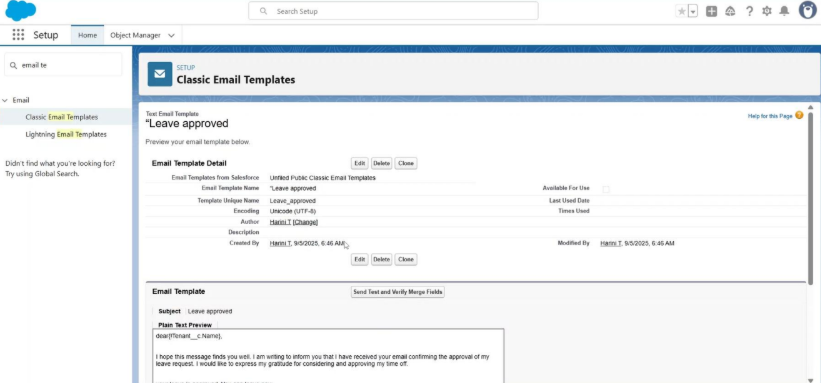
● Added Apex trigger to restrict multiple tenants per property

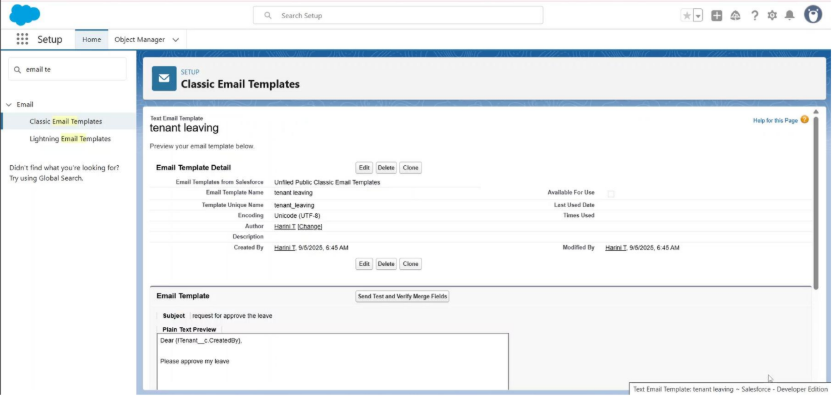


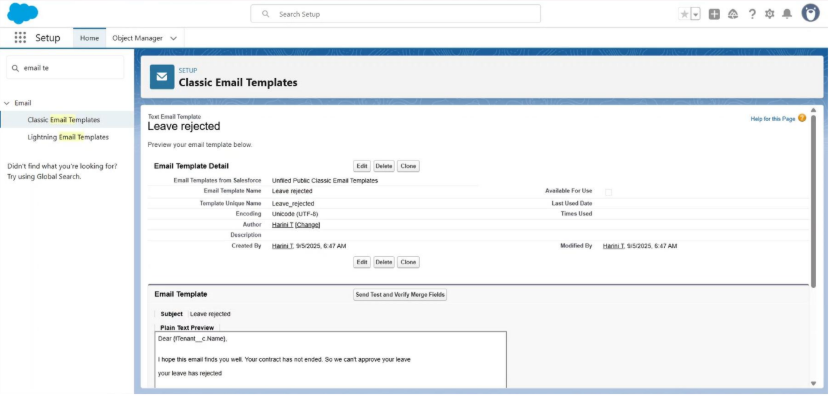
● Scheduled monthly reminder emails using Apex class



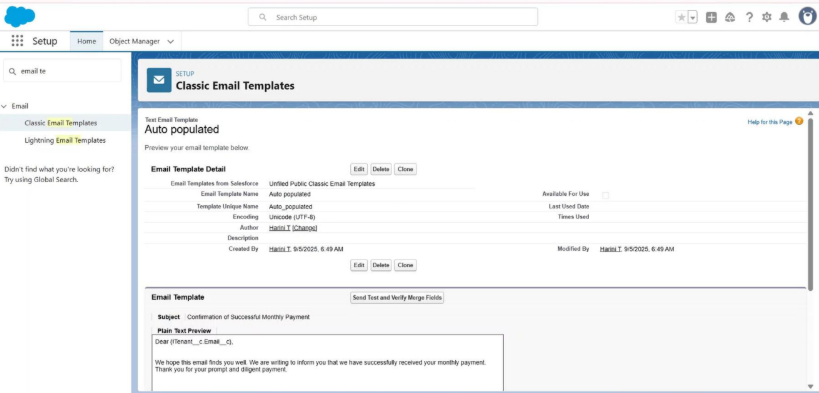
● Built and tested email templates for leave request, approval, rejection, payment, and reminders









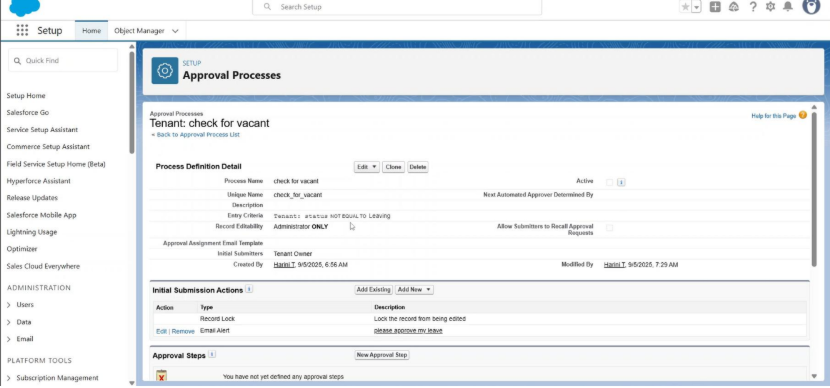


● Approval Process creation

For Tenant Leaving:

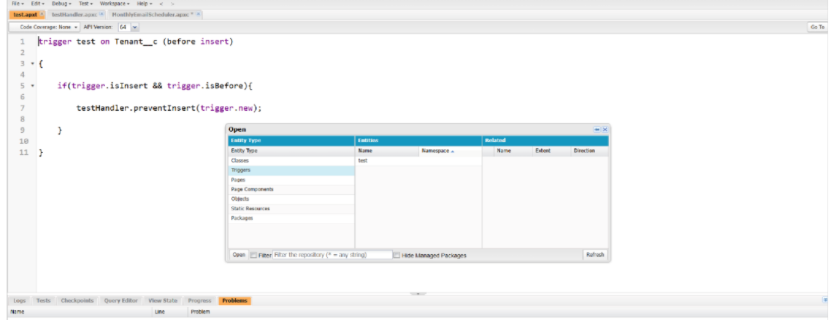


For Check for Vacant:

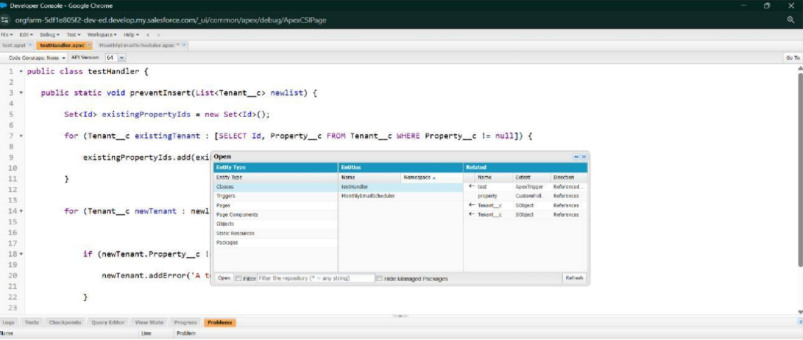


● Apex Trigger

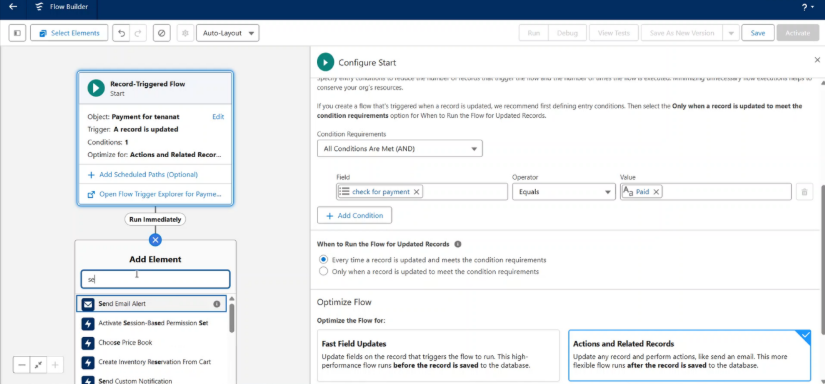
Create an Apex Trigger

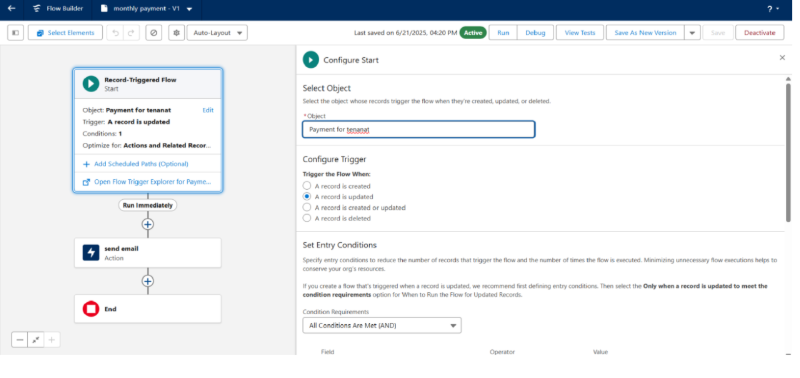


Create an Apex Handler class:



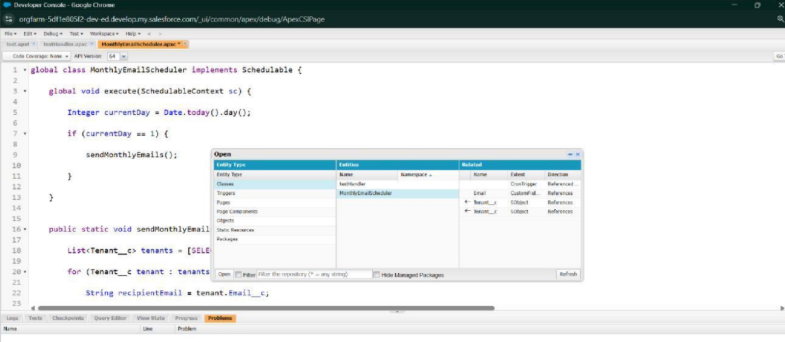
● FLOWS

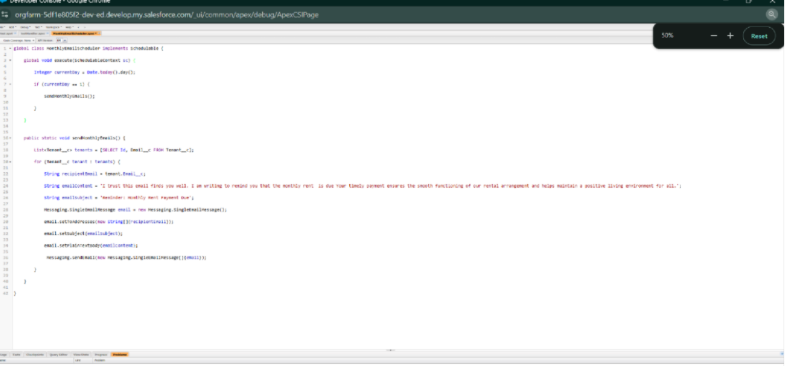




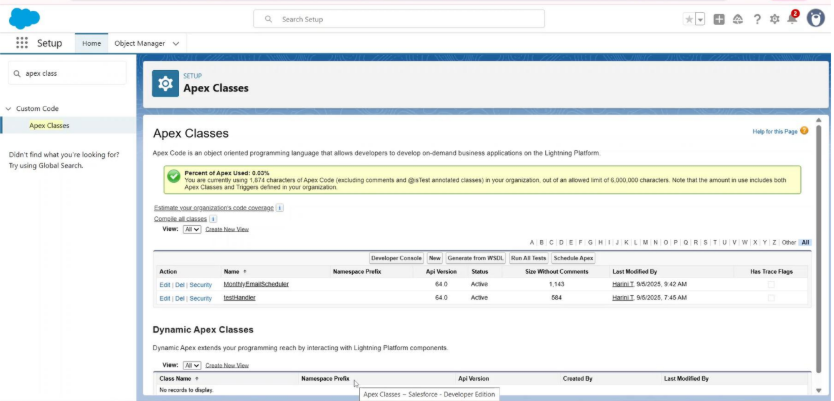
● Schedule class:

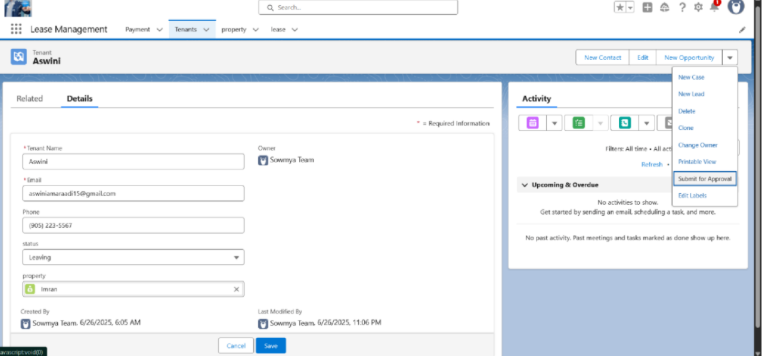
Create an Apex Class

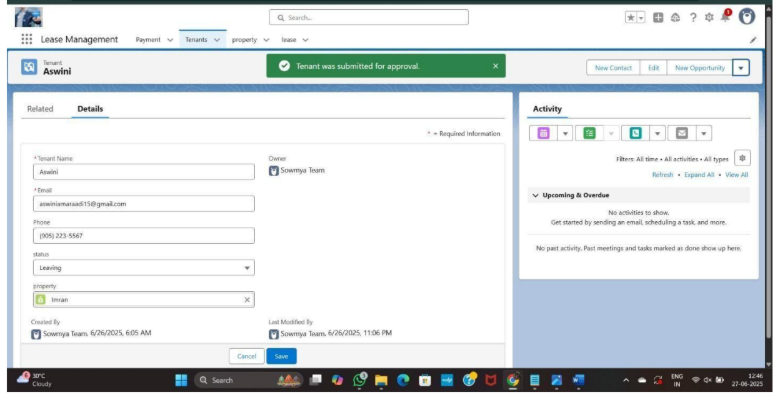


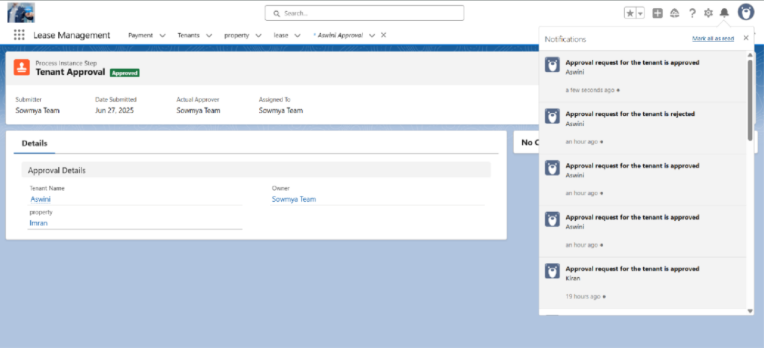


Schedule Apex class





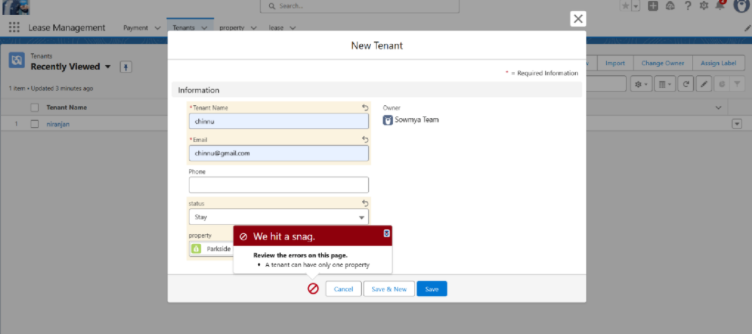


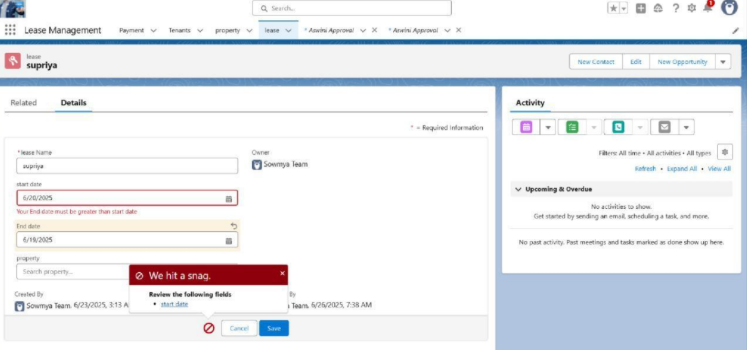


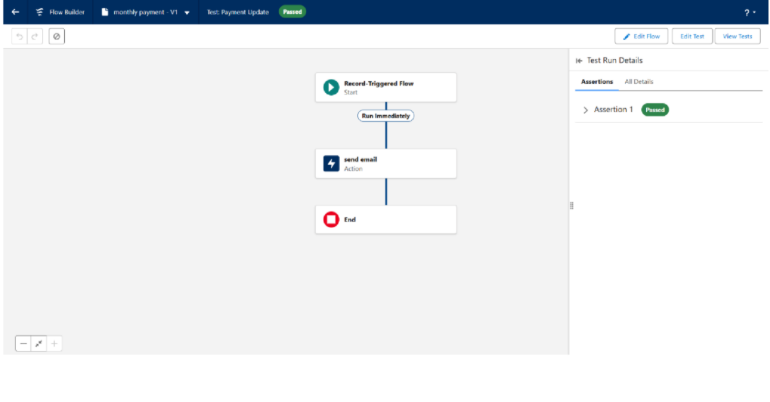
FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

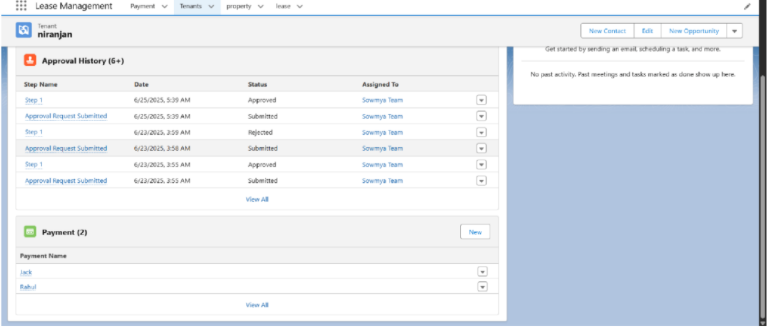
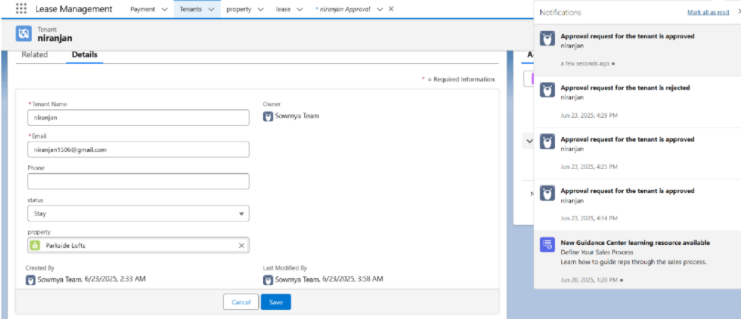
\* Trigger validation by entering duplicate tenant-property records



● Test flows on payment update



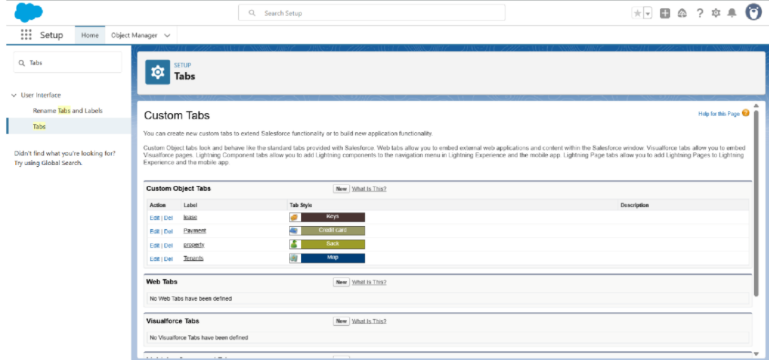
● Approval process validated through email alerts and status updates



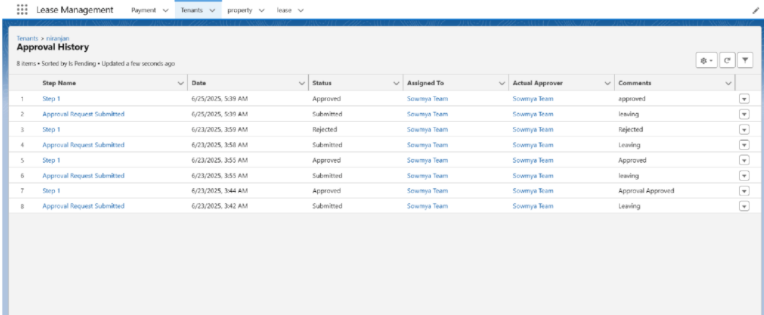
RESULTS

Output Screenshots

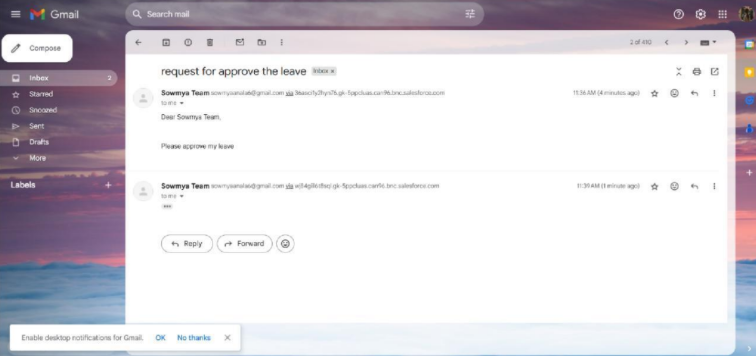
● Tabs for Property, Tenant, Lease, Payment



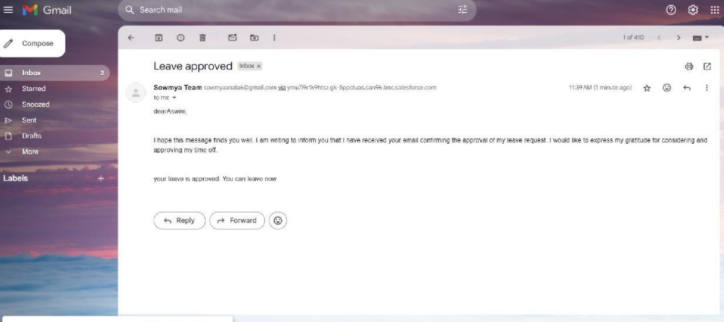
● Email alerts



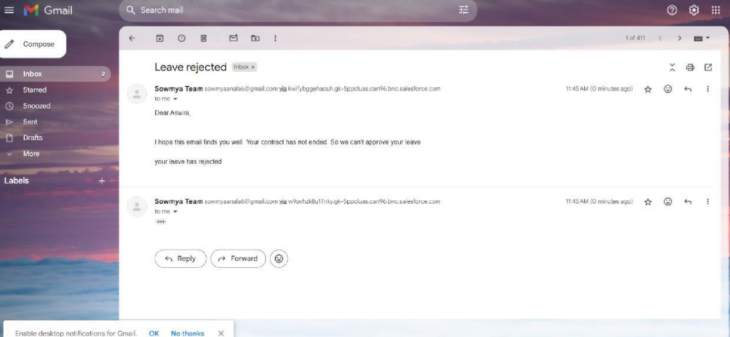
● Request for approve the leave



● Leave approved



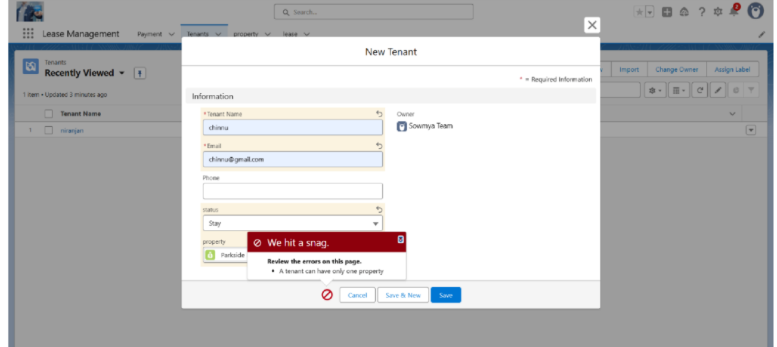
● Leave rejected



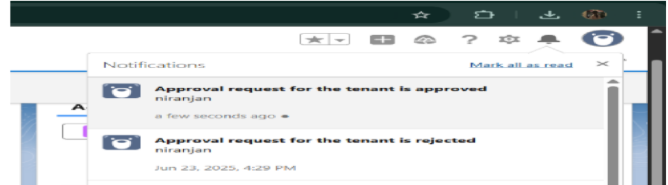
● Flow runs



● Trigger error messages



● Approval process notifications



**ADVANTAGES & DISADVANTAGES**

**Advantages:**

- Automates routine tasks like rent reminders, approvals, and payments.

- Reduces manual errors by using validation rules and flows.

- Improves efficiency and saves time for administrators.

- Provides real-time tracking of properties, tenants, and payments.

- Enhances communication through automated email alerts and notifications.

- Centralized data storage ensures easy access and better decision-making.

**Disadvantages:**

- Requires initial learning effort to understand salesforce setup and features.

- Internet connectivity is mandatory to use salesforce effectively.

- Advanced customization (like Apex coding for triggers/schedulers) may be difficult for

beginners.

- In real-world business, Salesforce licensing costs can be high (though the developer edition is free for projects).

**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});

}

}

}