**LEASE MANAGEMENT**

**College Name : SRI VASAVI COLLEGE (SFW), ERODE.**

# **College code : bru17**

**TEAM ID : NM2025TMID23310**

**TEAM MEMBERS :**

**Team Leader Name:** PAVITHRA M

**Email:** pavithramurugesanopm@gmail.com

**Team Member1:** BOOMIKA K

**Email:** boomika11@gmail.com

**Team Member2:** KAVIYADHARSINI V

**Email:** kdharshini325@gmail.com

**Team Member3:** ABISHEK N

**Email:** abisheknatraj09@gmail.com

**Team Member4**: SANTHOSH V

**Email:** vsanthosh2111@gmail.com

**1.INTRODUCTION**

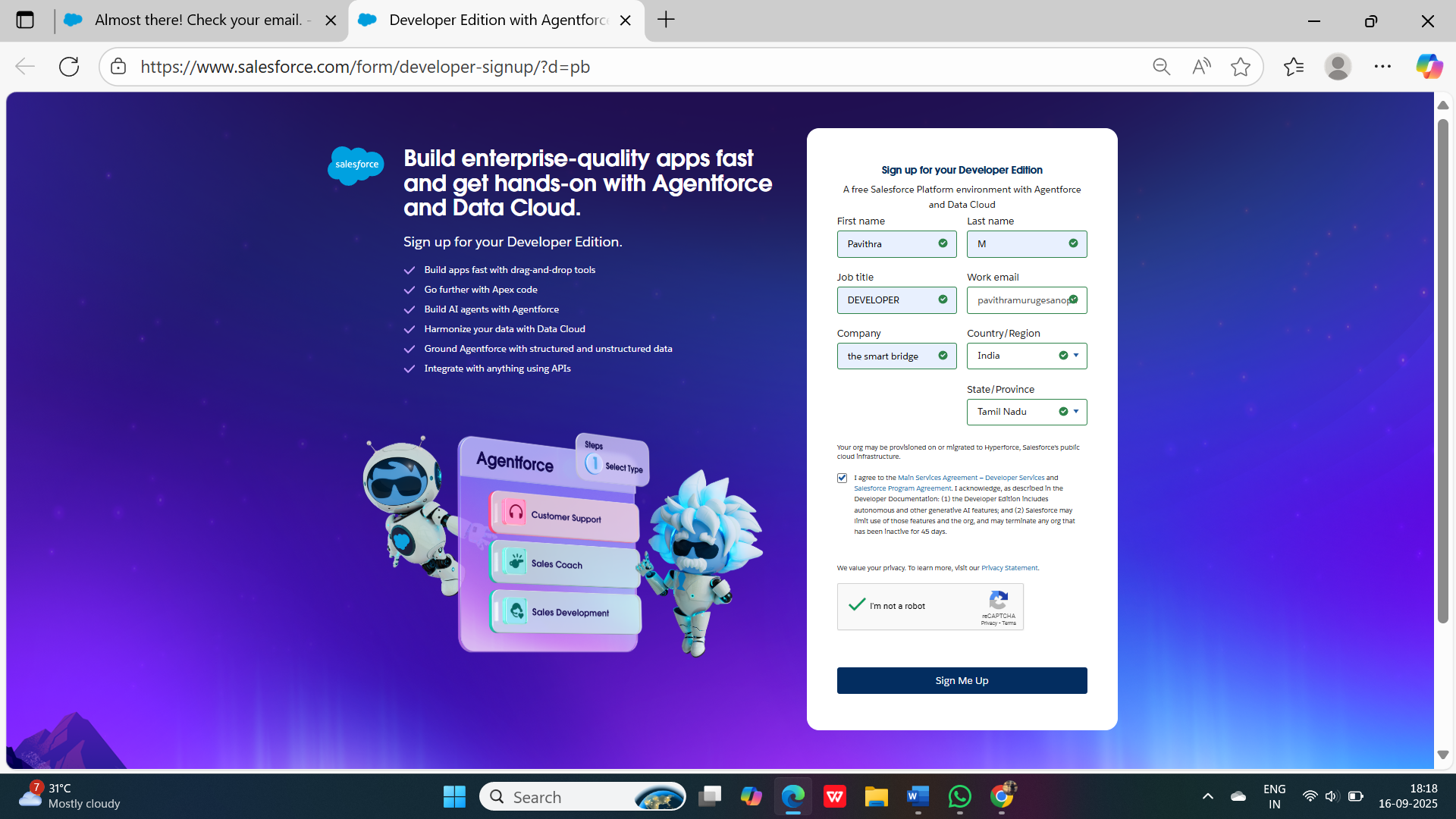
* 1. **Project Overview**

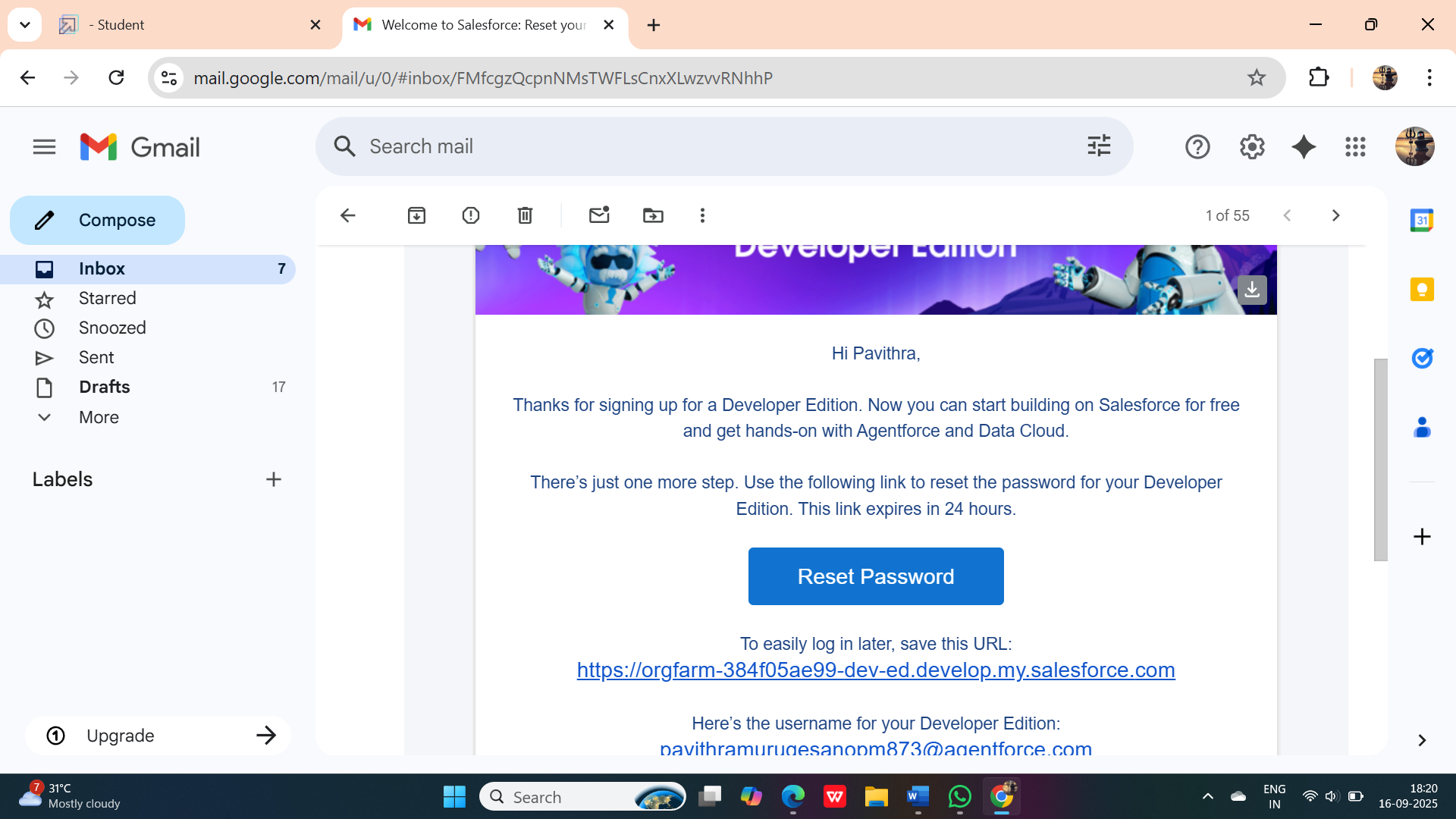
The Lease Management System is a Salesforce-based application designed to streamline the Processes associated with leasing real estate properties. It handles tenant management, lease contracts, payments, and communication with automation features such as flows, approval processes, and email alert

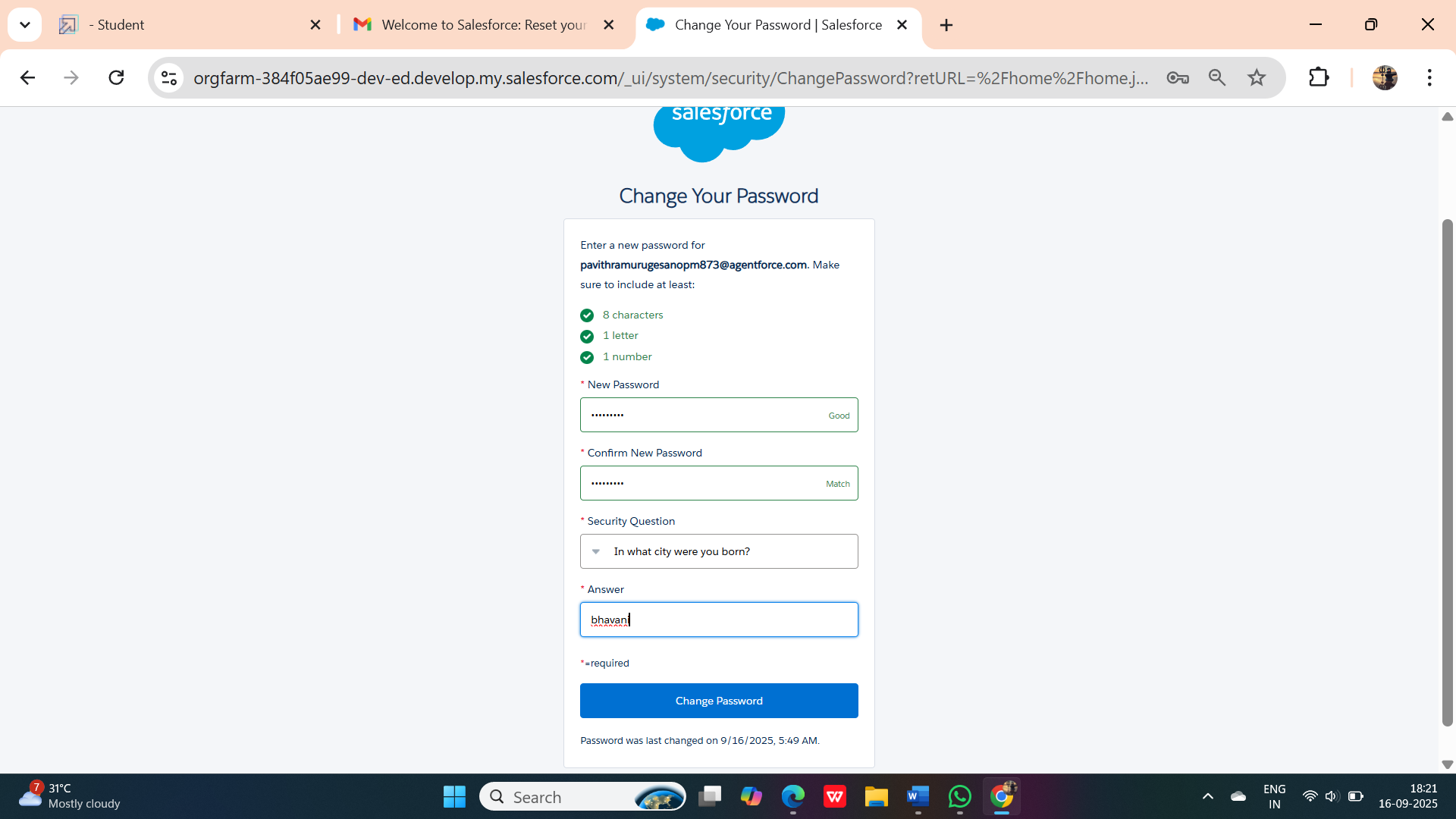
* 1. **Purpose**

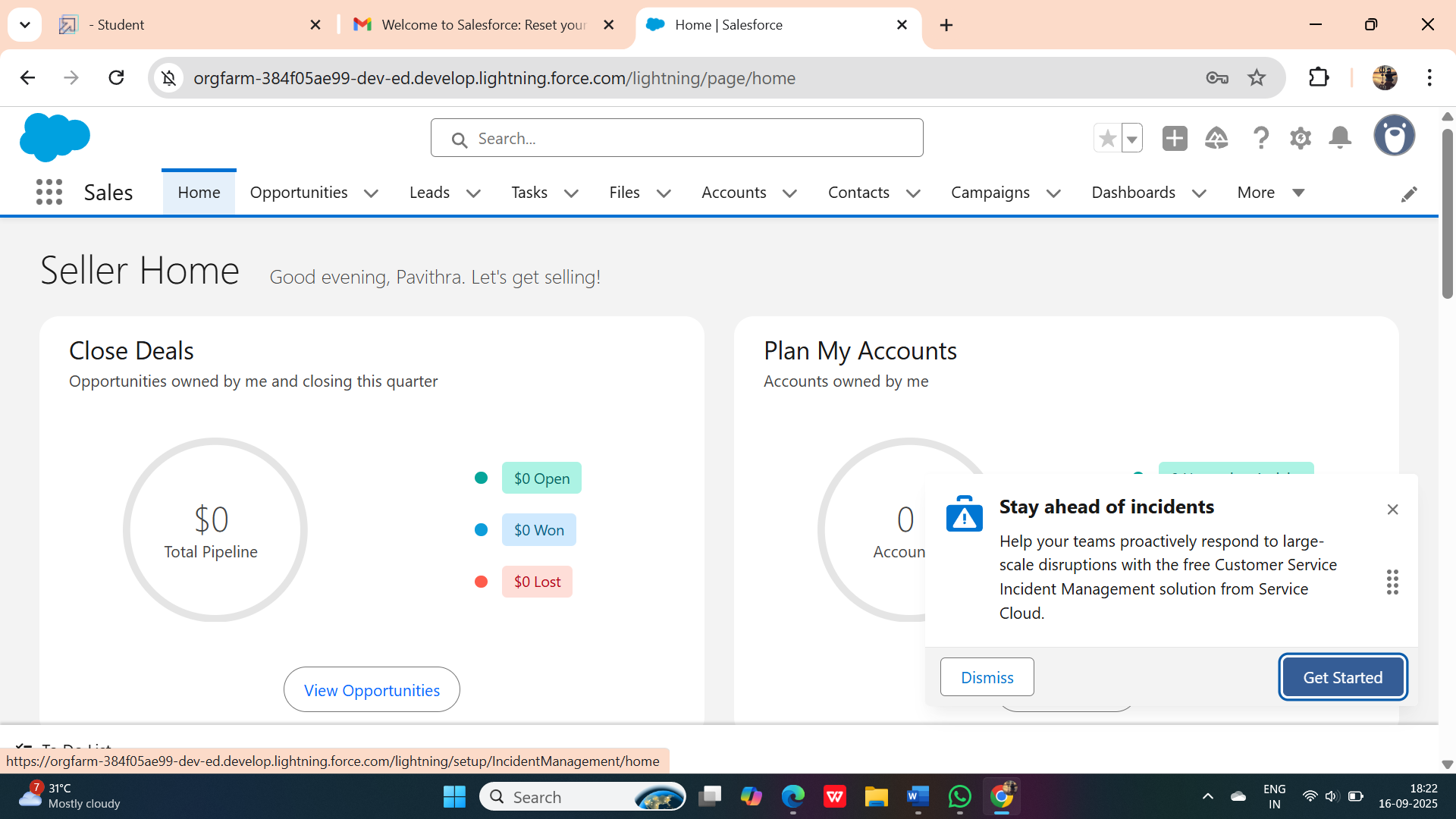
The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

• **DEVELOPMENT PHASE** Creating Developer Account: By using this URL: https://www.salesforce.com

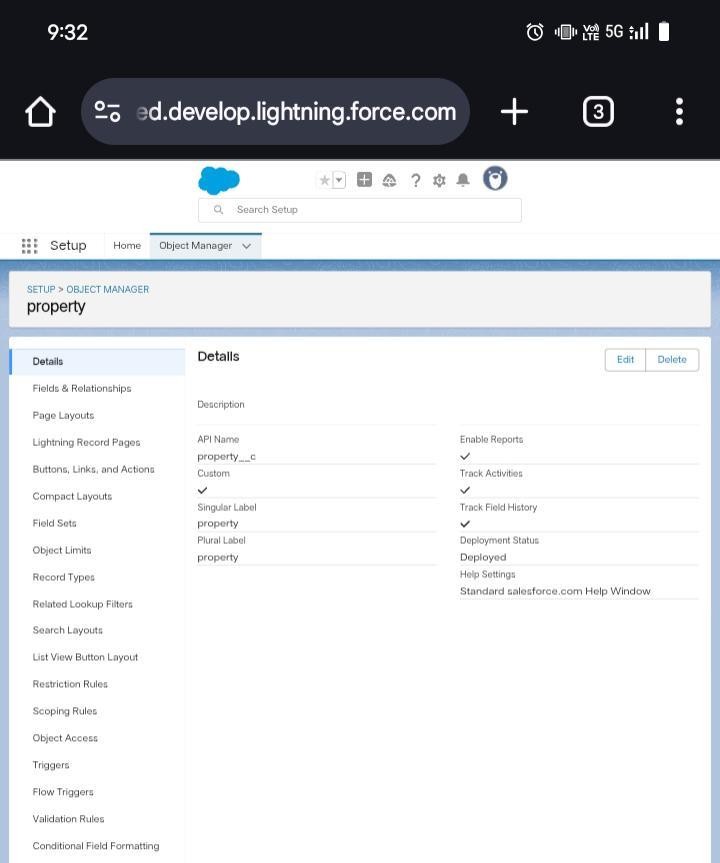


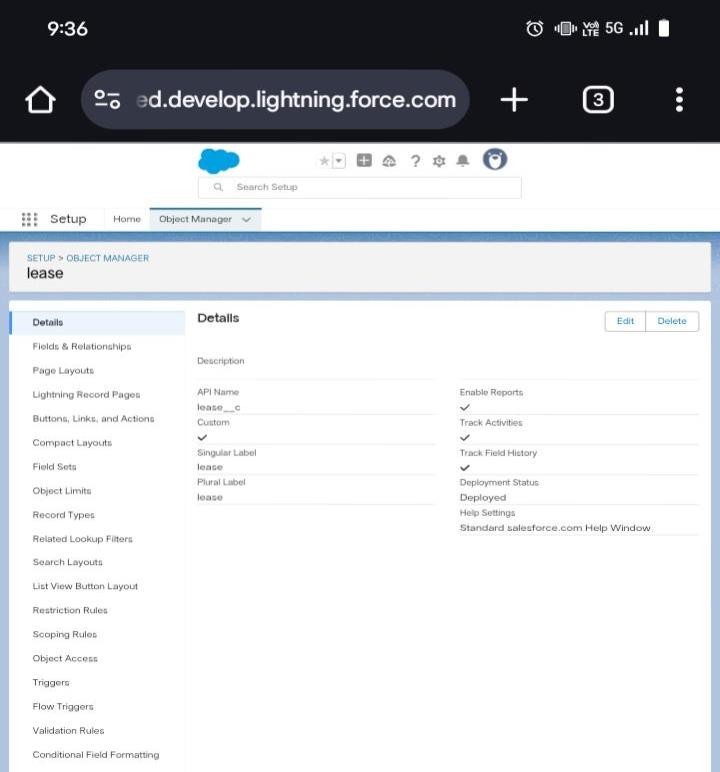


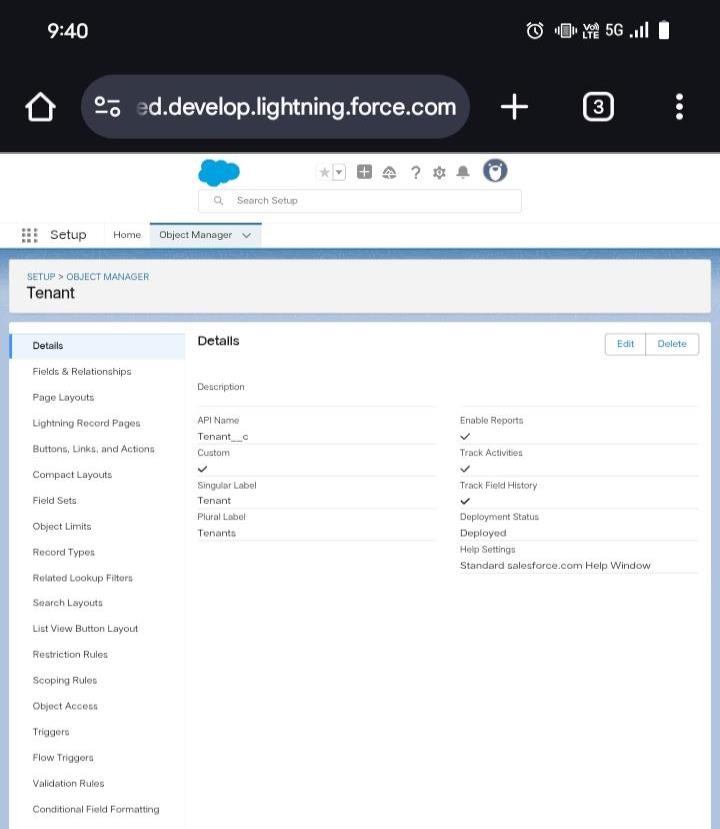


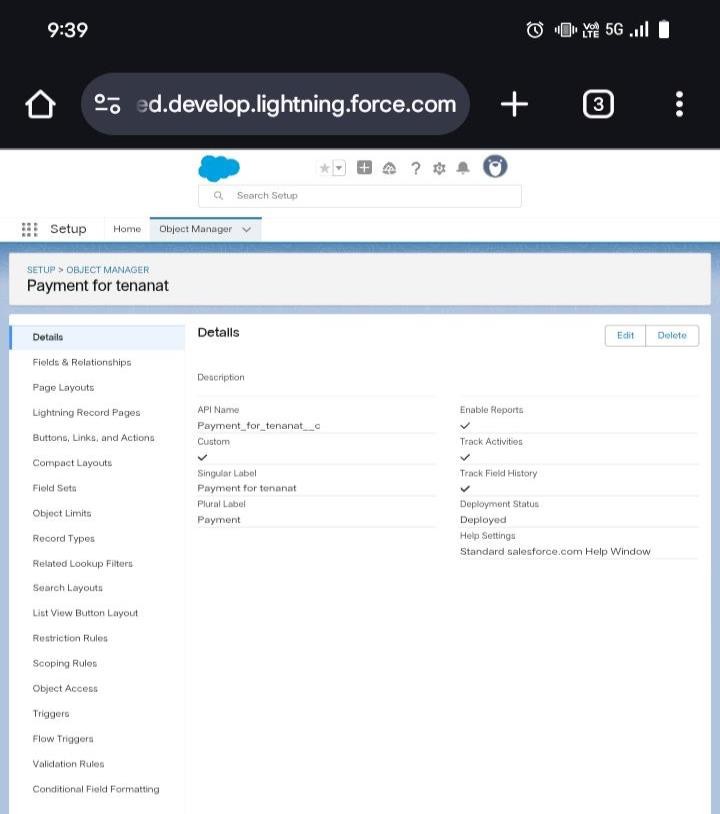


**Created objects:** Property, Tenant, Lease, Payment

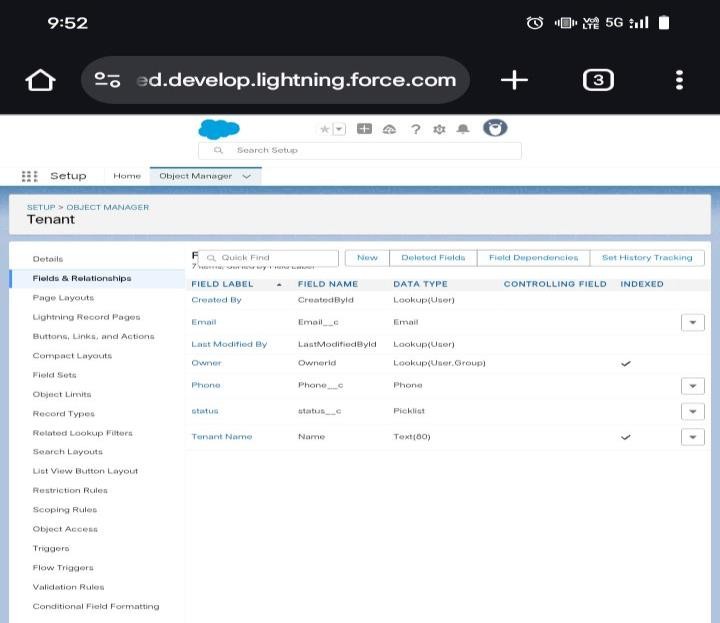


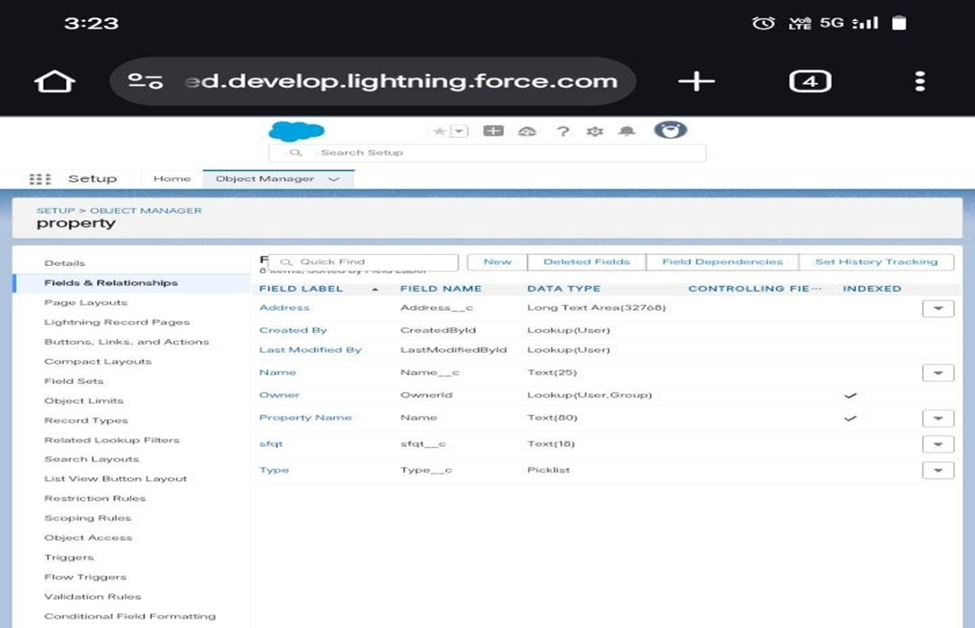


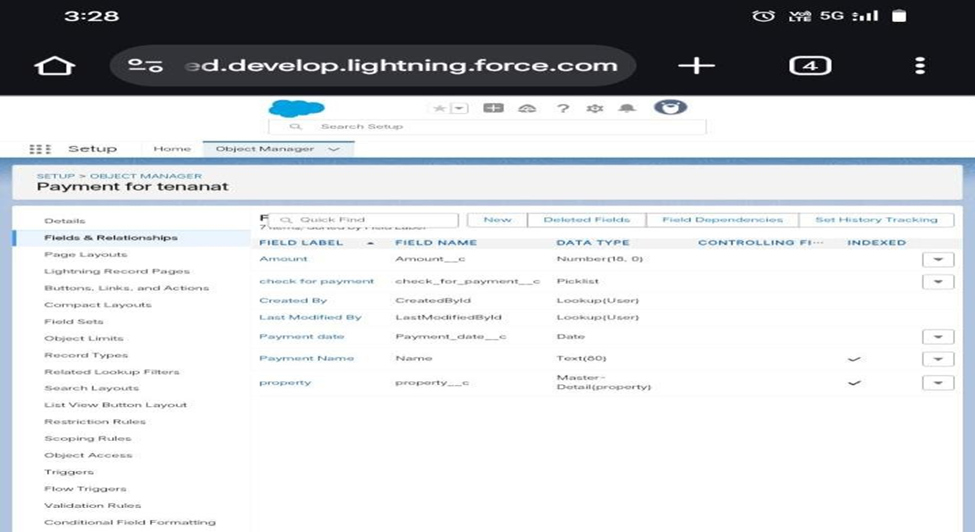


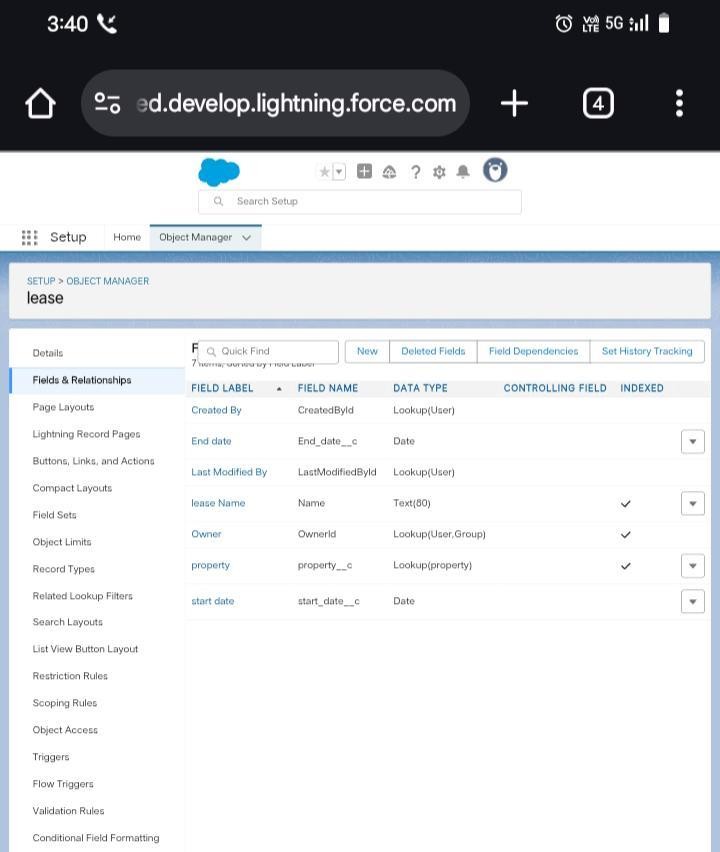


* **Configured fields and relationships**

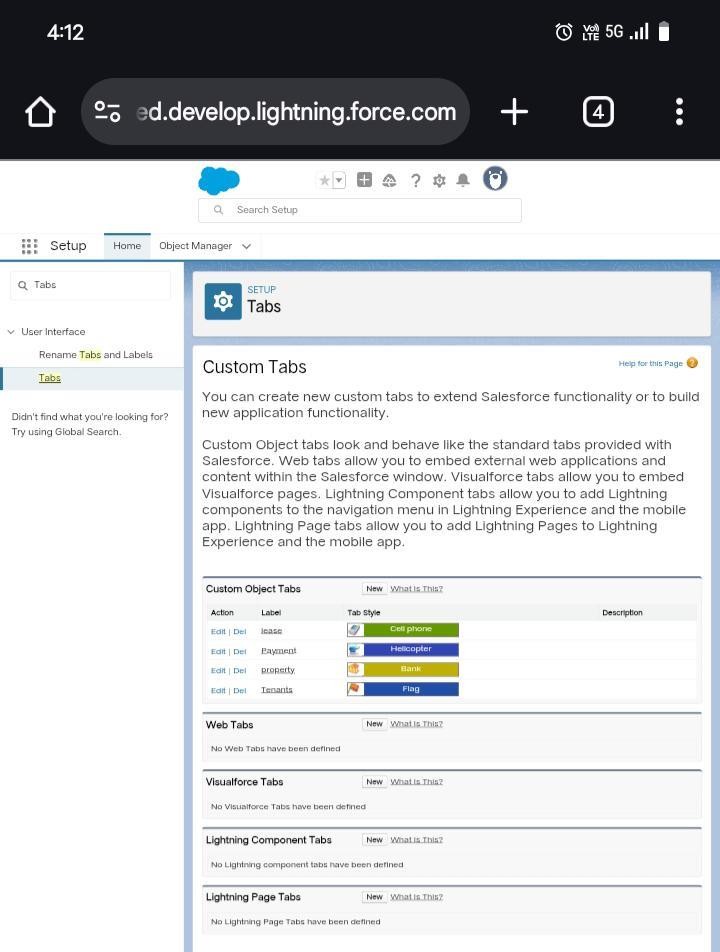




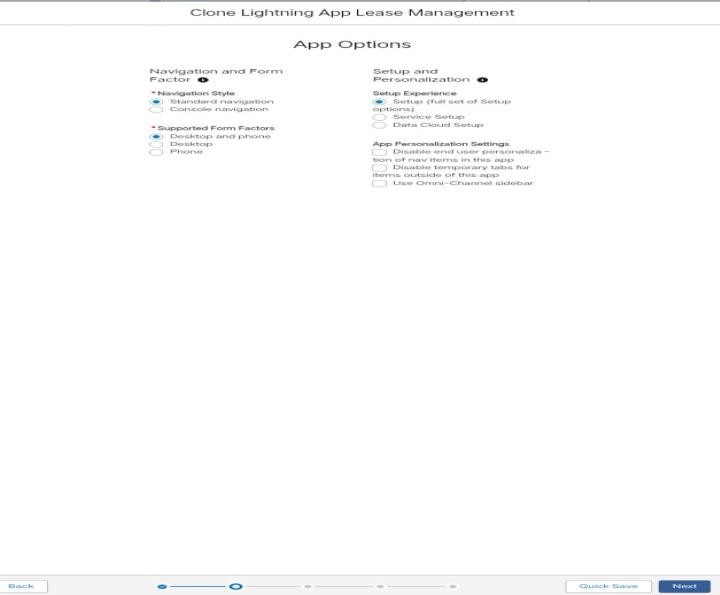
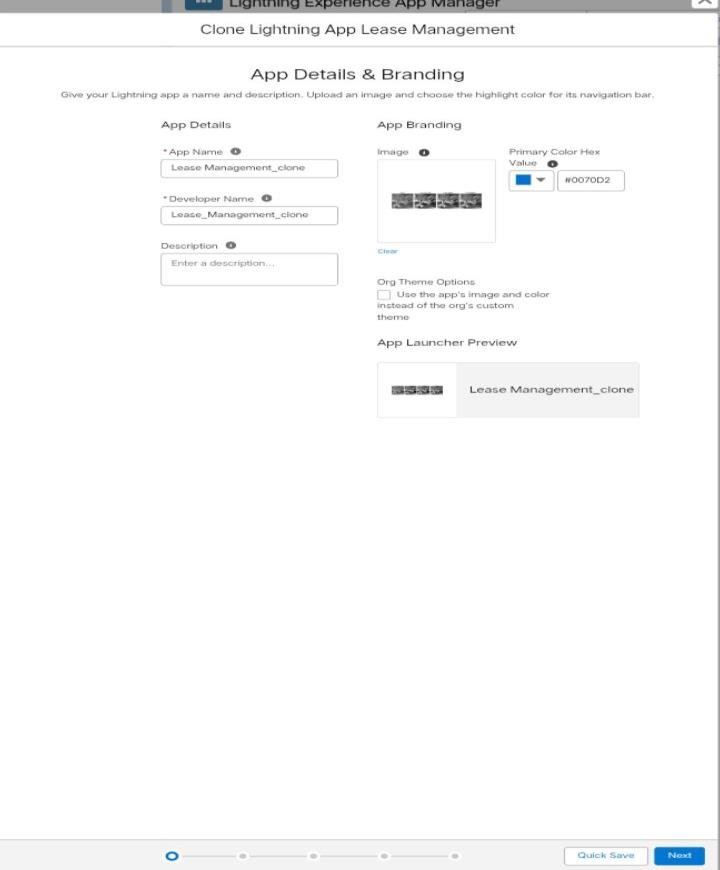


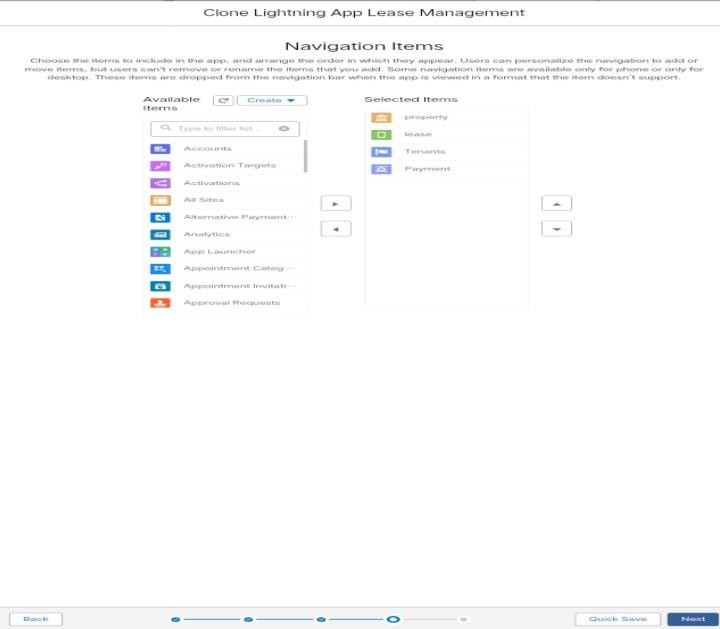


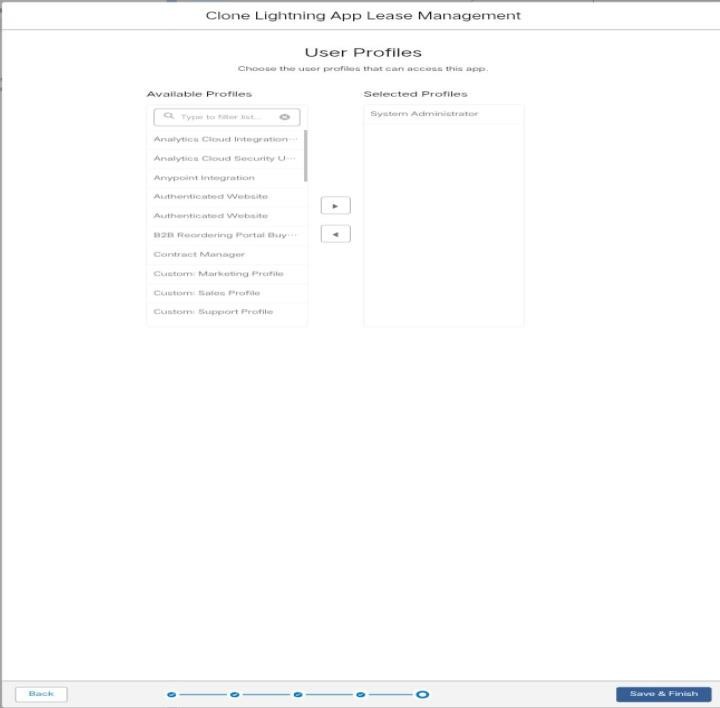
**Configuring the Tabs :**



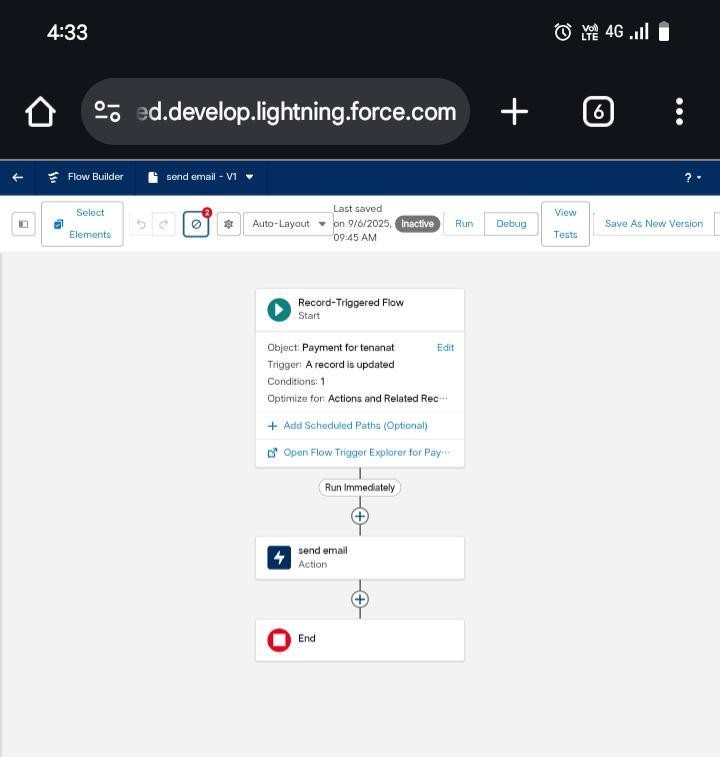
**Developed Lightning App with relevant tabs :**



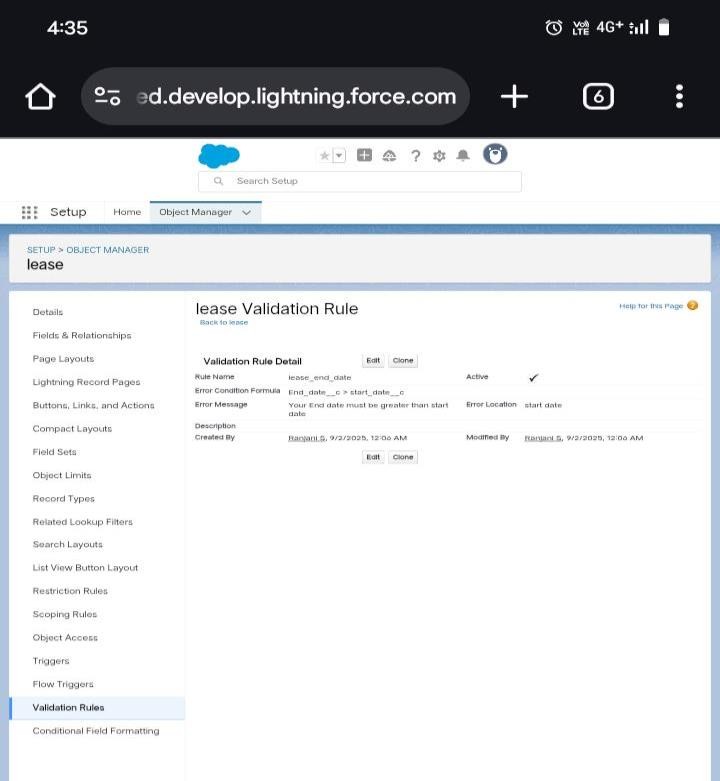


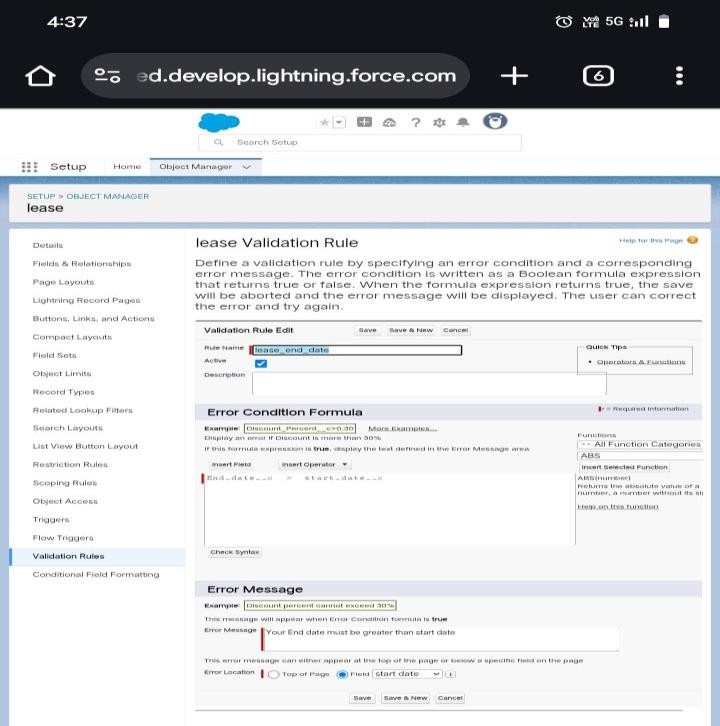


**Implemented Flows for monthly rent and payment success :**

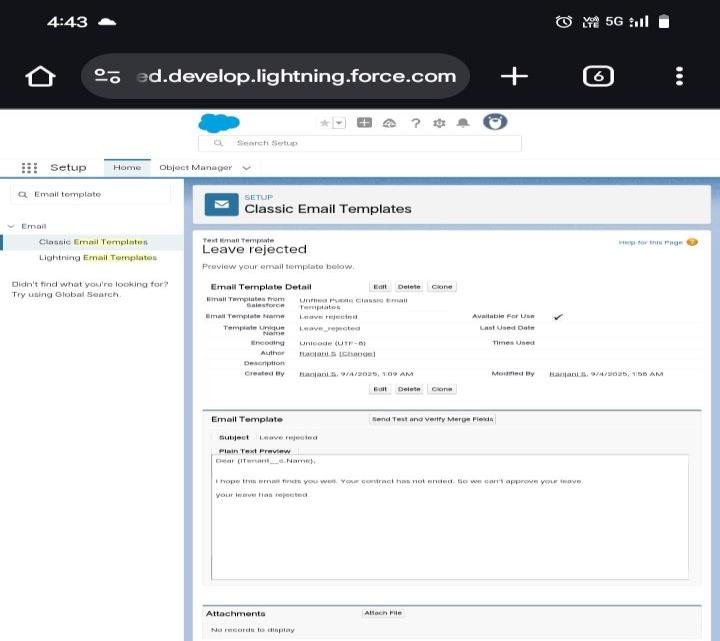


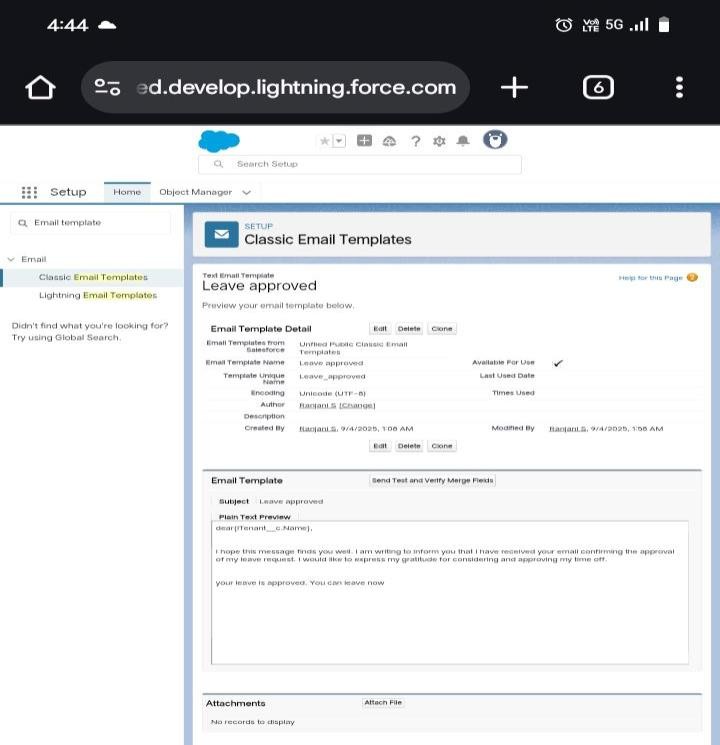
**To create a validation rule to a Lease Object :**

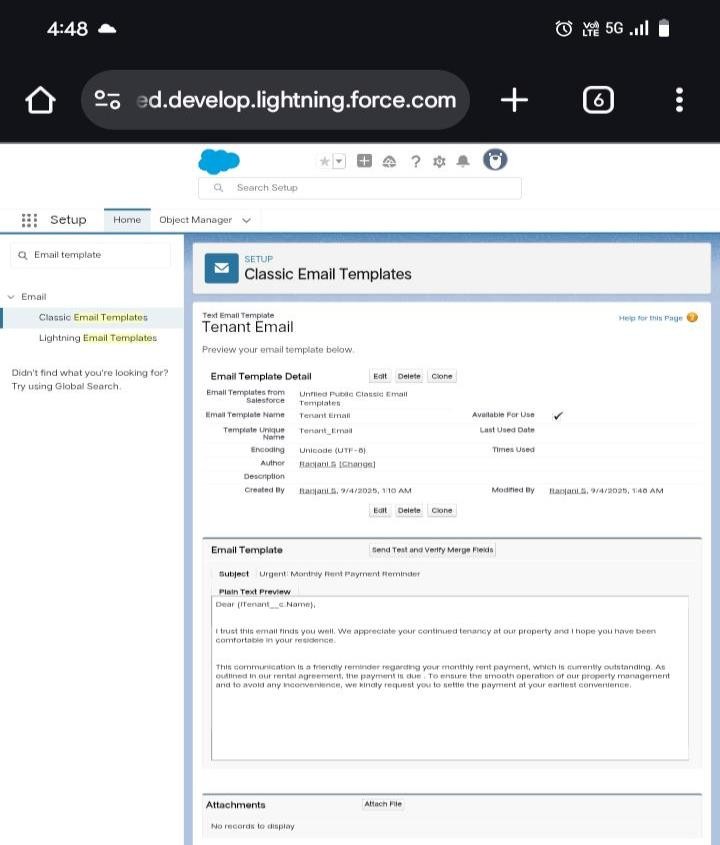


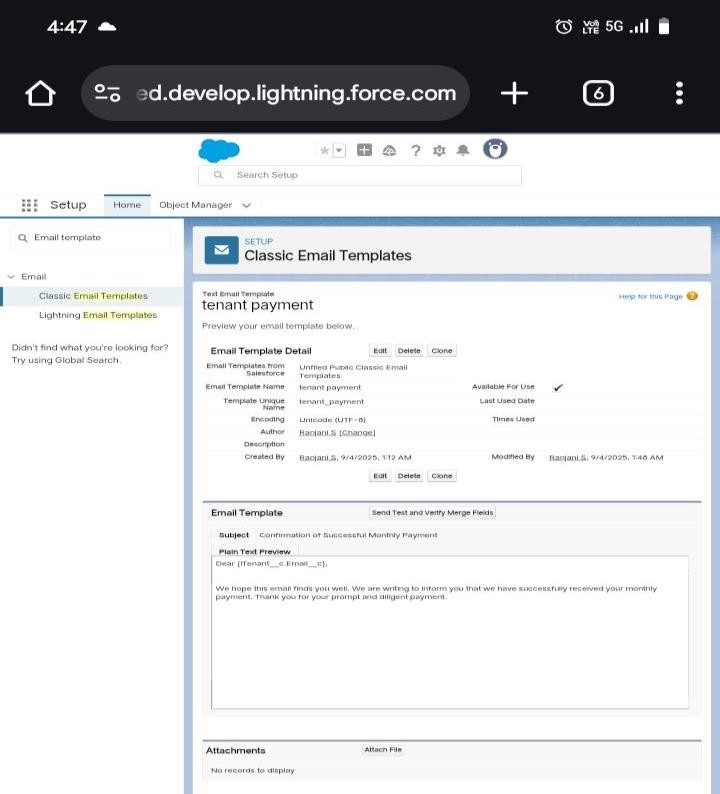


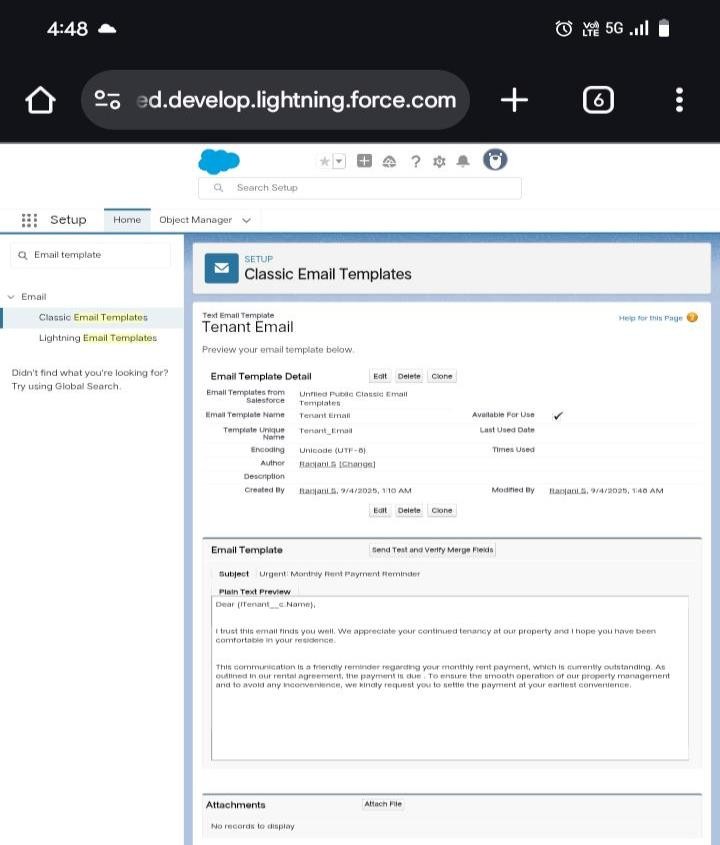
**Built and tested email templates for leave request, approval, rejection, payment, and reminders :**

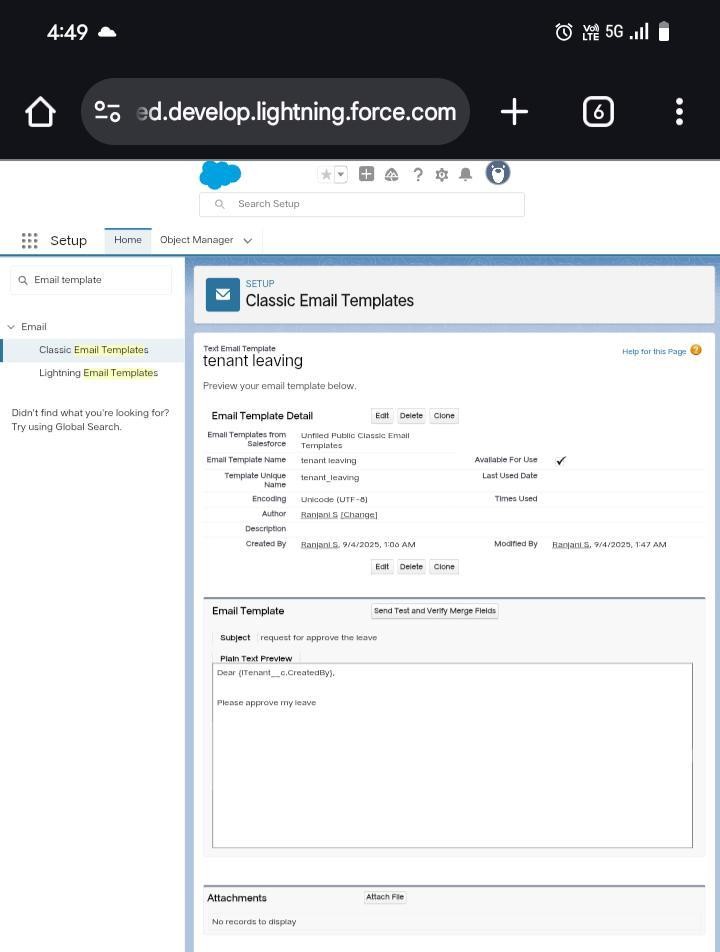






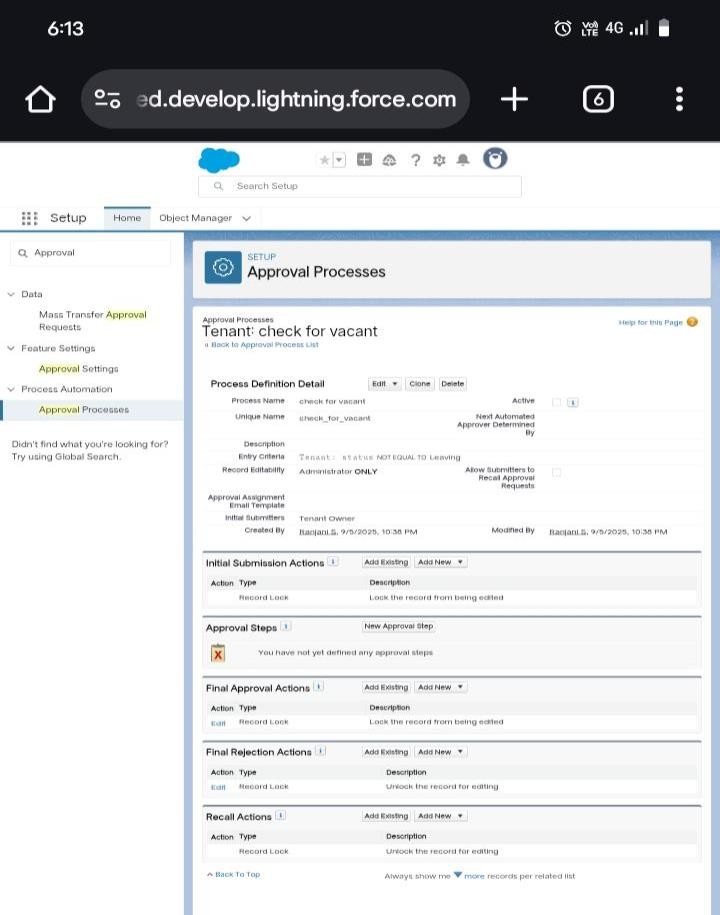




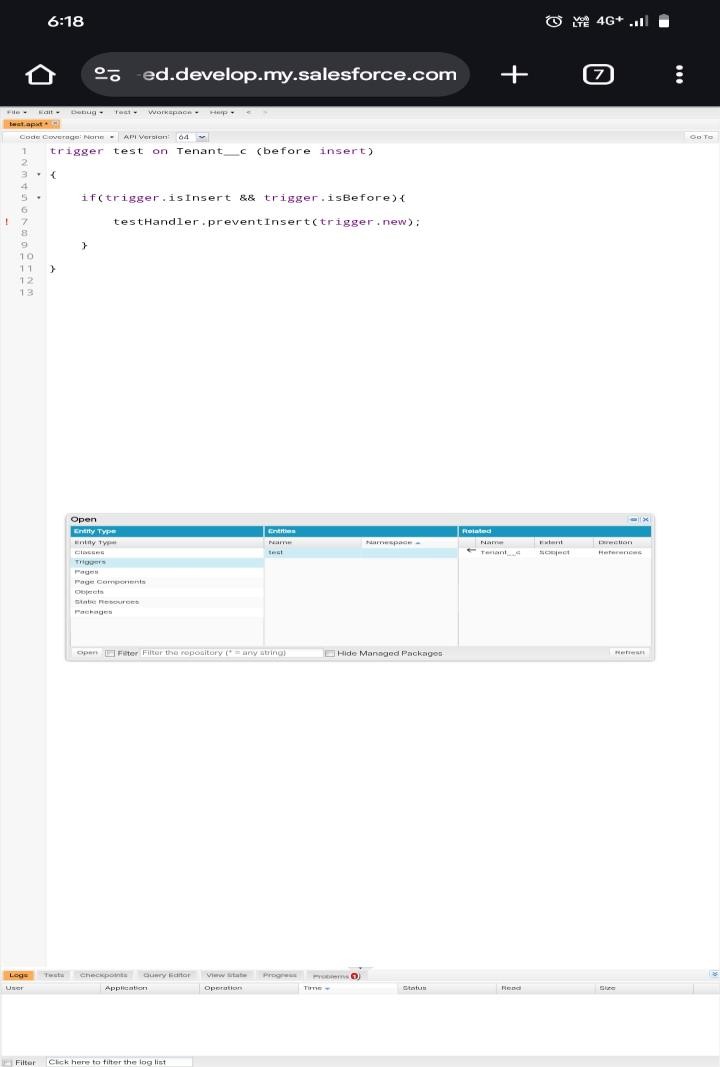


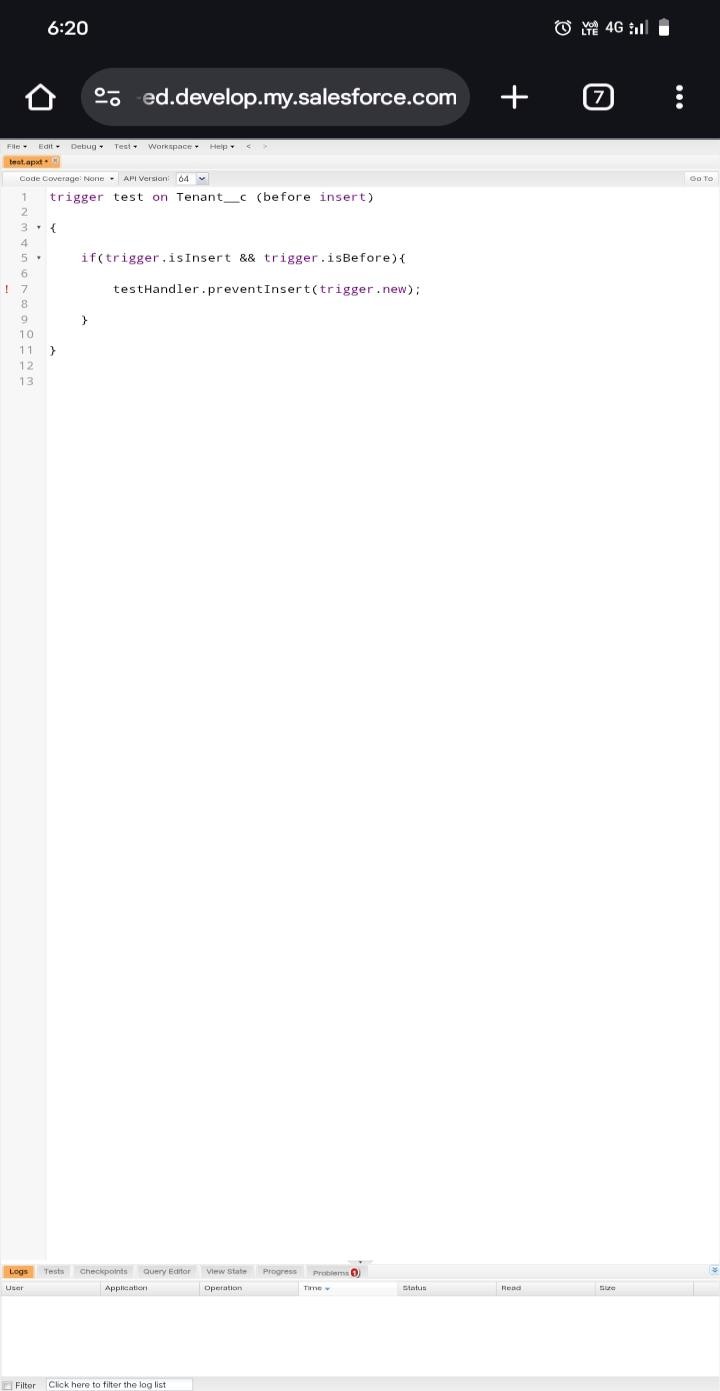
**Approval Process creation:**



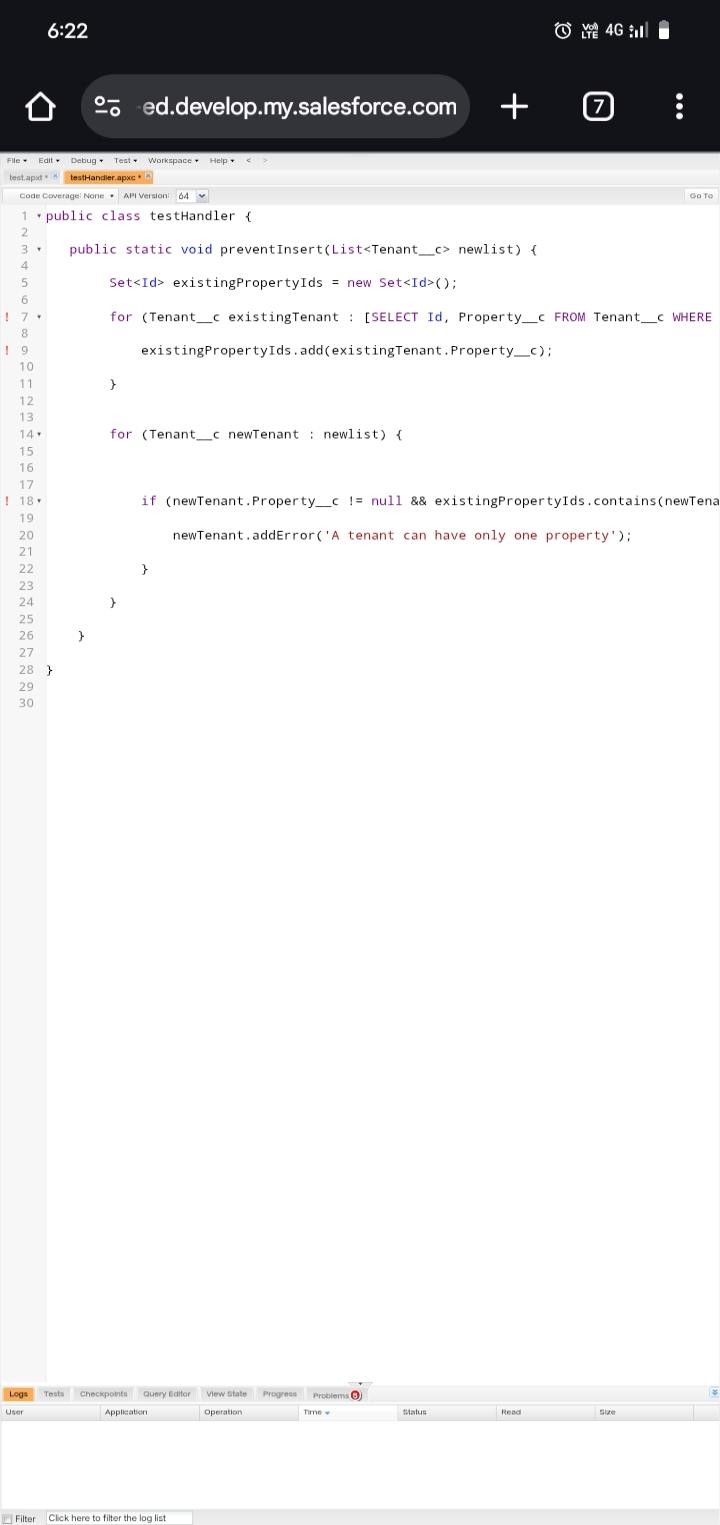


**Apex Trigger :**

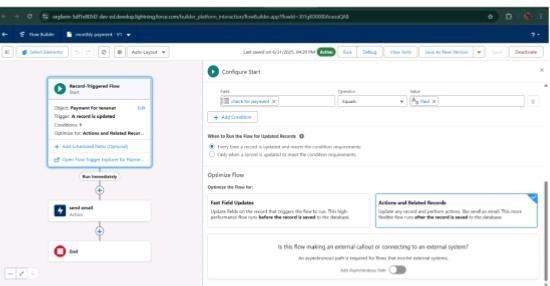
**Create an Apex Trigger:** 

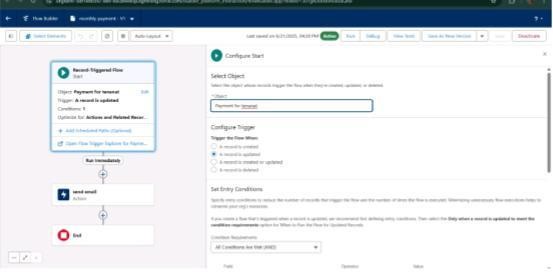


**Create an Apex Handler class :**



**FLOWS :**

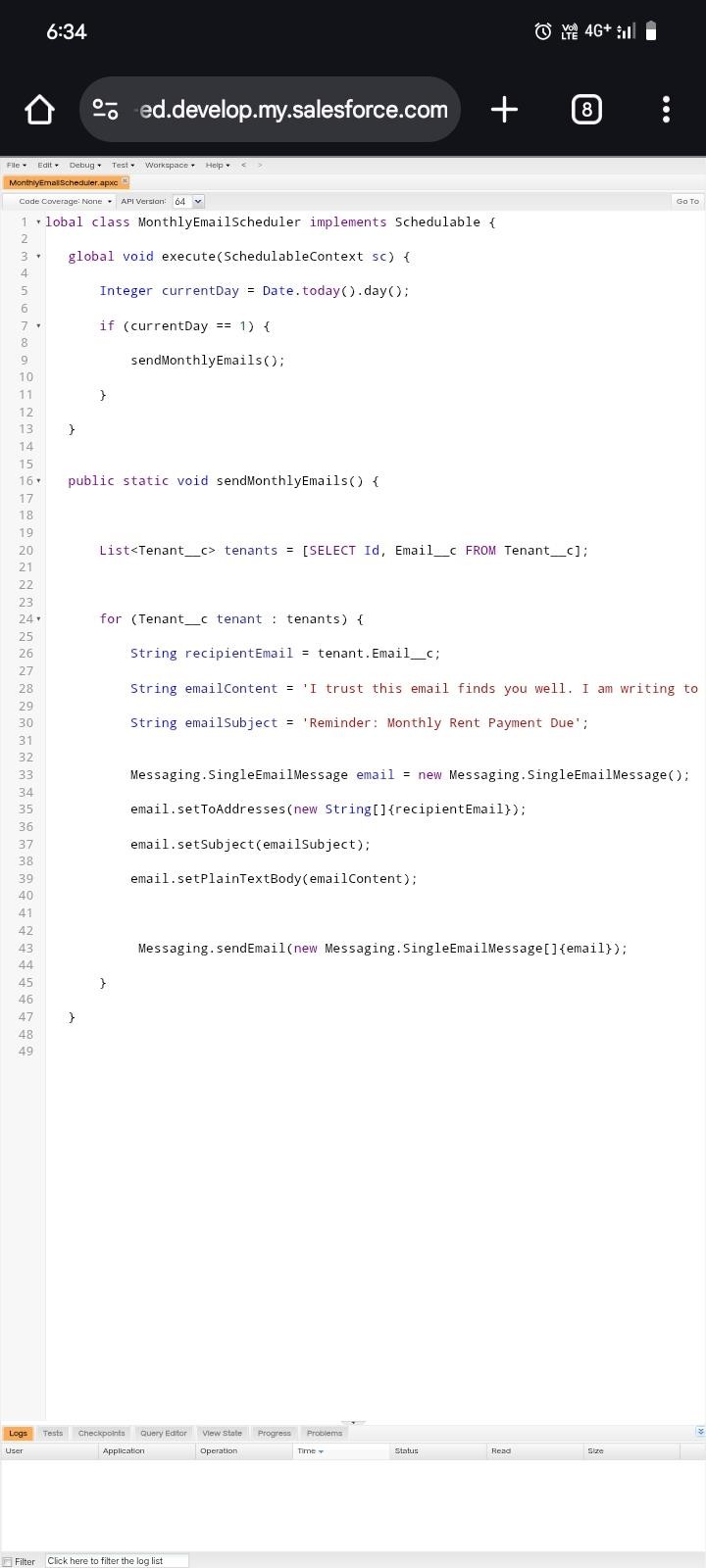




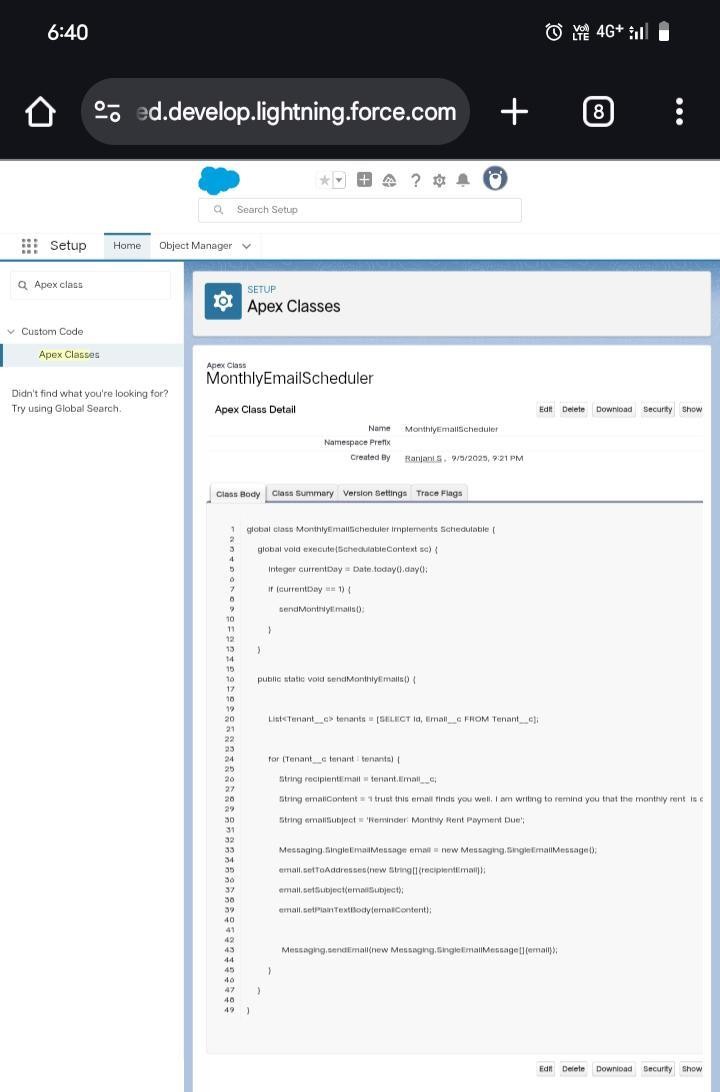
**Schedule class:**

**Create an Apex Class**





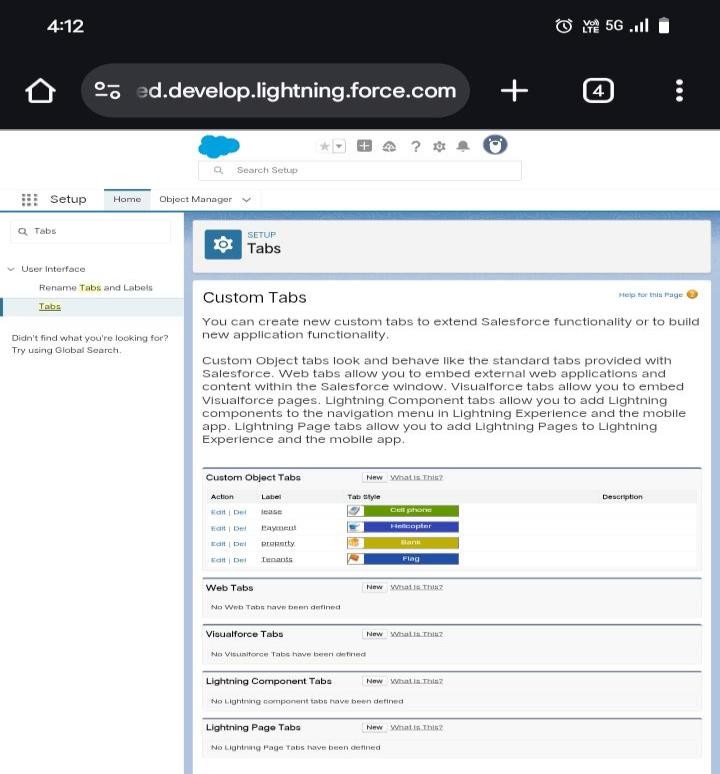
**Schedule Apex class**



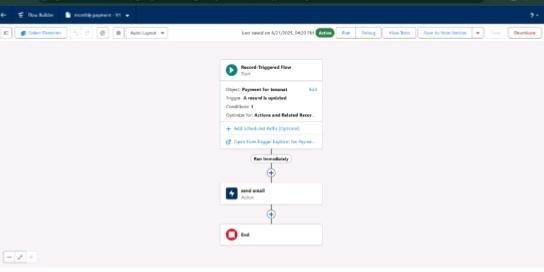
**RESULTS**

**Output Screenshots**

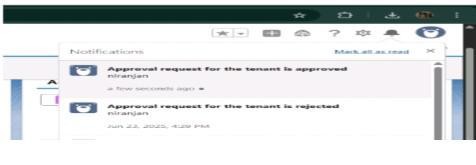
**Tabs for Property, Tenant, Lease, Payment**



**Flow runs:**



**Approval process notifications**



**ADVANTAGES & DISADVANTAGES:**

**Advantages :** Financial Benefits

Operational Efficiency

Legal & Compliance

Flexibility

Transparency & Control

**Disadvantages:** Higher Long-Term Cost

No Ownership

Restrictions & Limitations

Complexity in Management

**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

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});

}

}

}