**LEASE MANAGEMENT**

**College Name: Sree Narayana Guru College**

**College Code:Bru36**

**TEAM ID: NM2025TMID26439**

**TEAM MEMBERS: 4**

Team LeaderName: Akhil M

Email: akhilakhil38044@gmail.com

Team Member: AnithJ

Email: anithani867@gmail.com

Team Member: AL Sabith Z

Email: alsabithhh@gmail.com

Team Member: Anush EK

Email: ekanush78@gmail.com

# 1.INTRODUCTION

**1.1 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 alerts.



**1.2 Purpose**

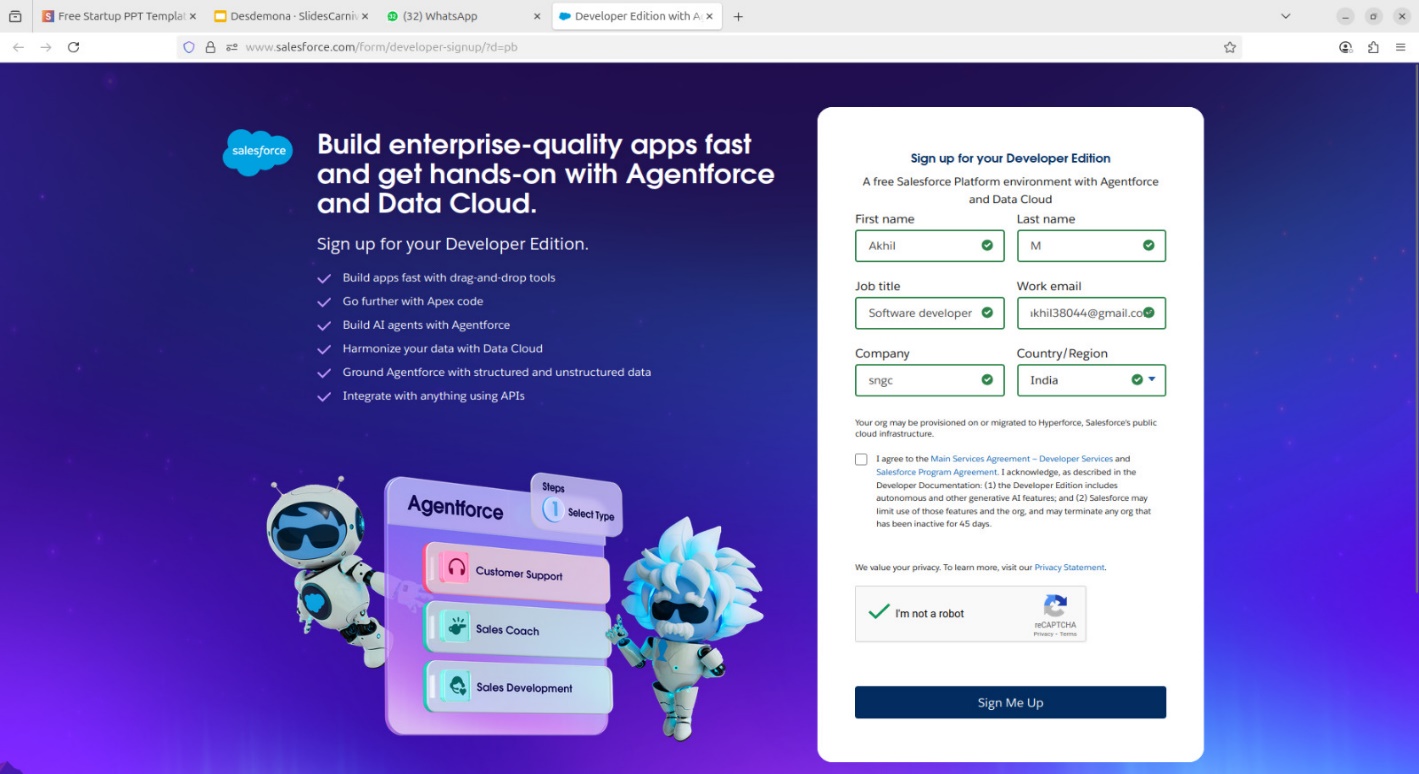
The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

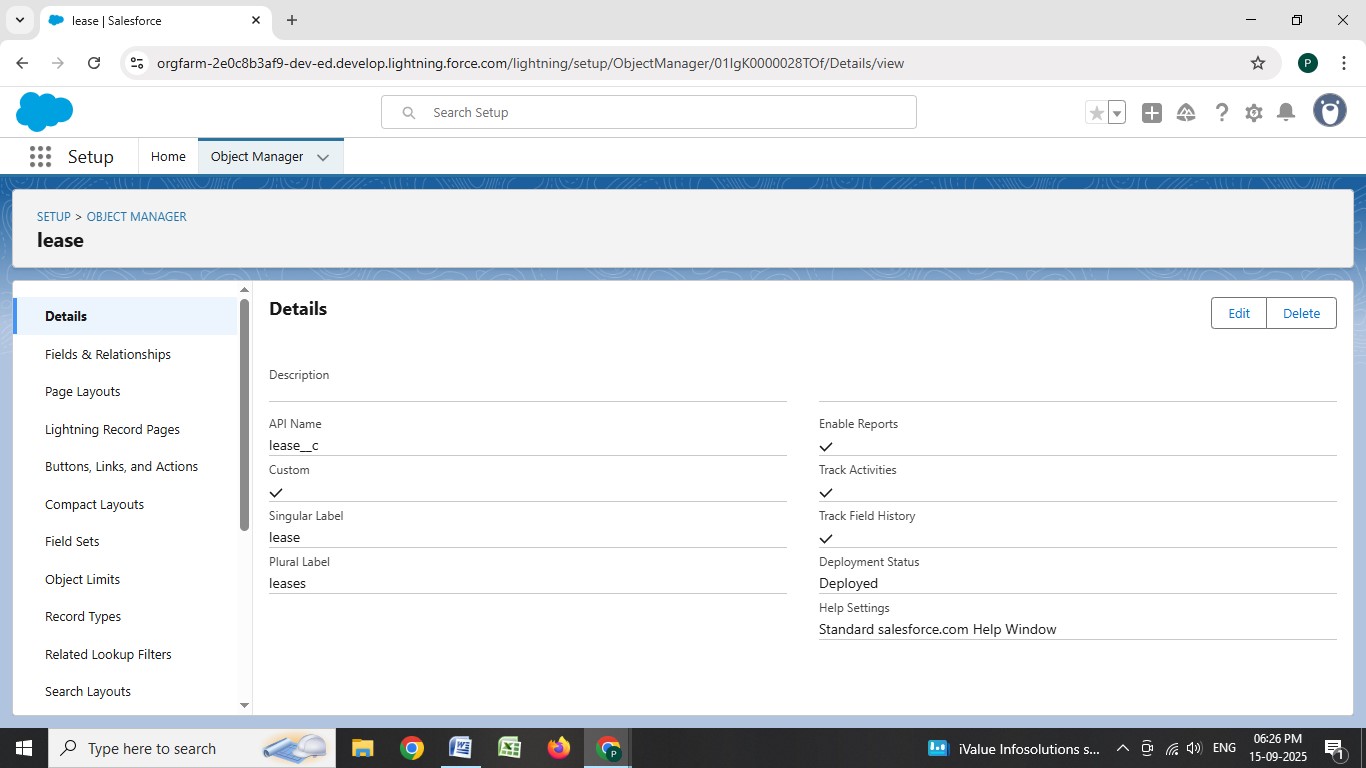
# 2.DEVELOPMENT PHASE

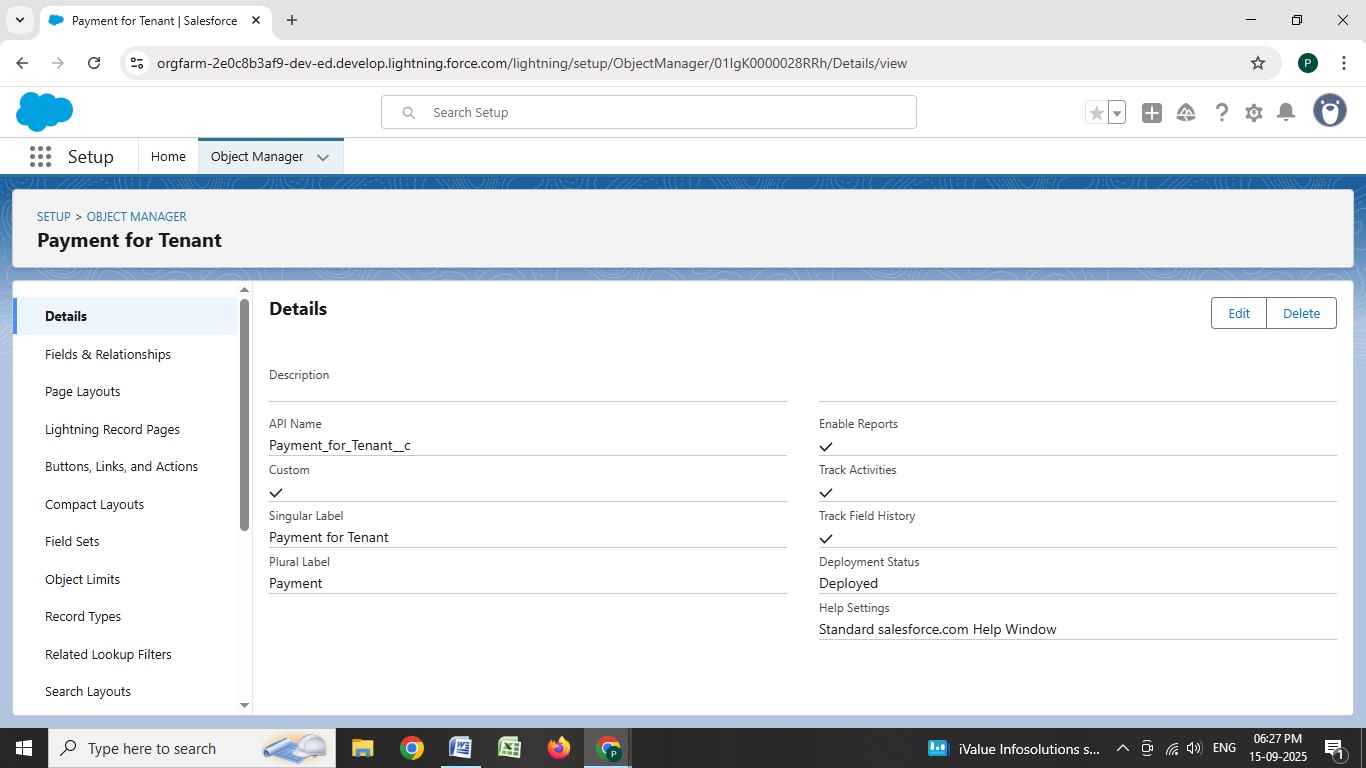
**Creating Developer Account:**

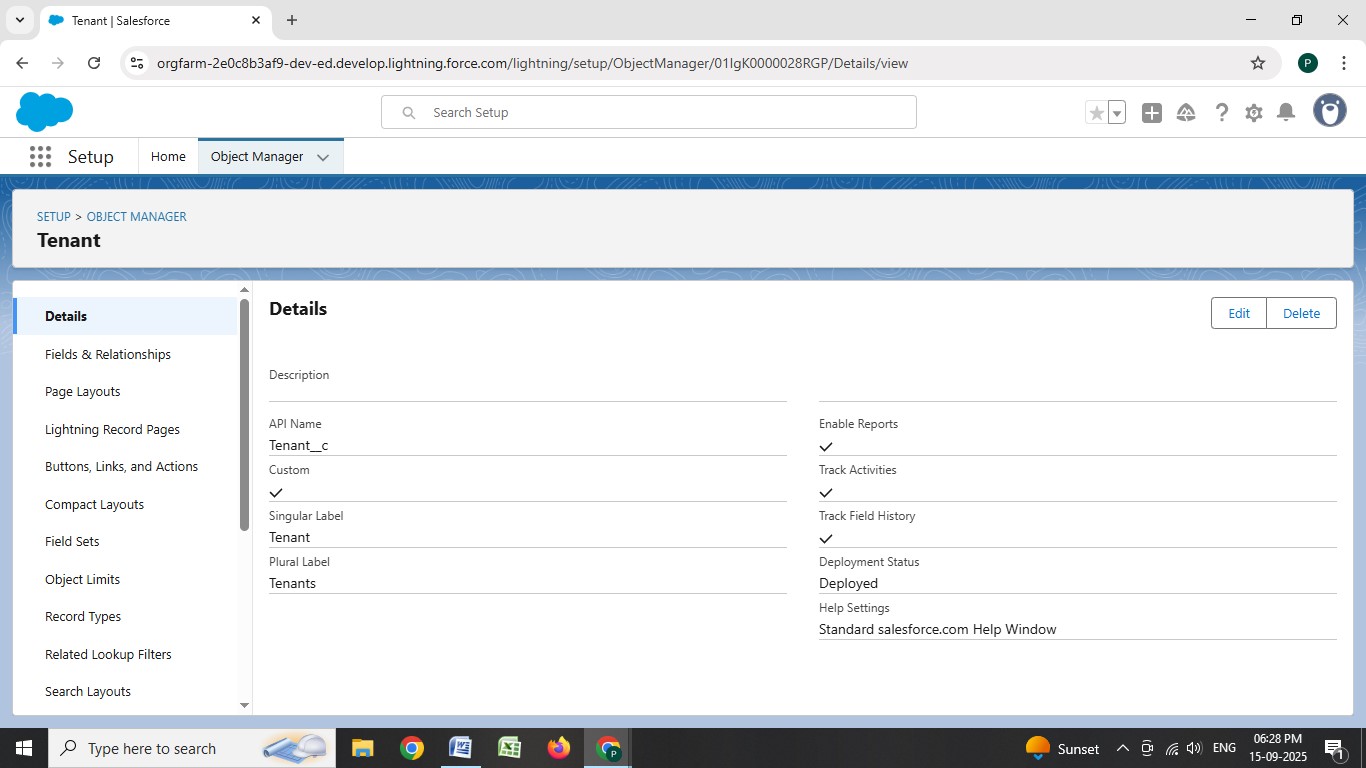
By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>

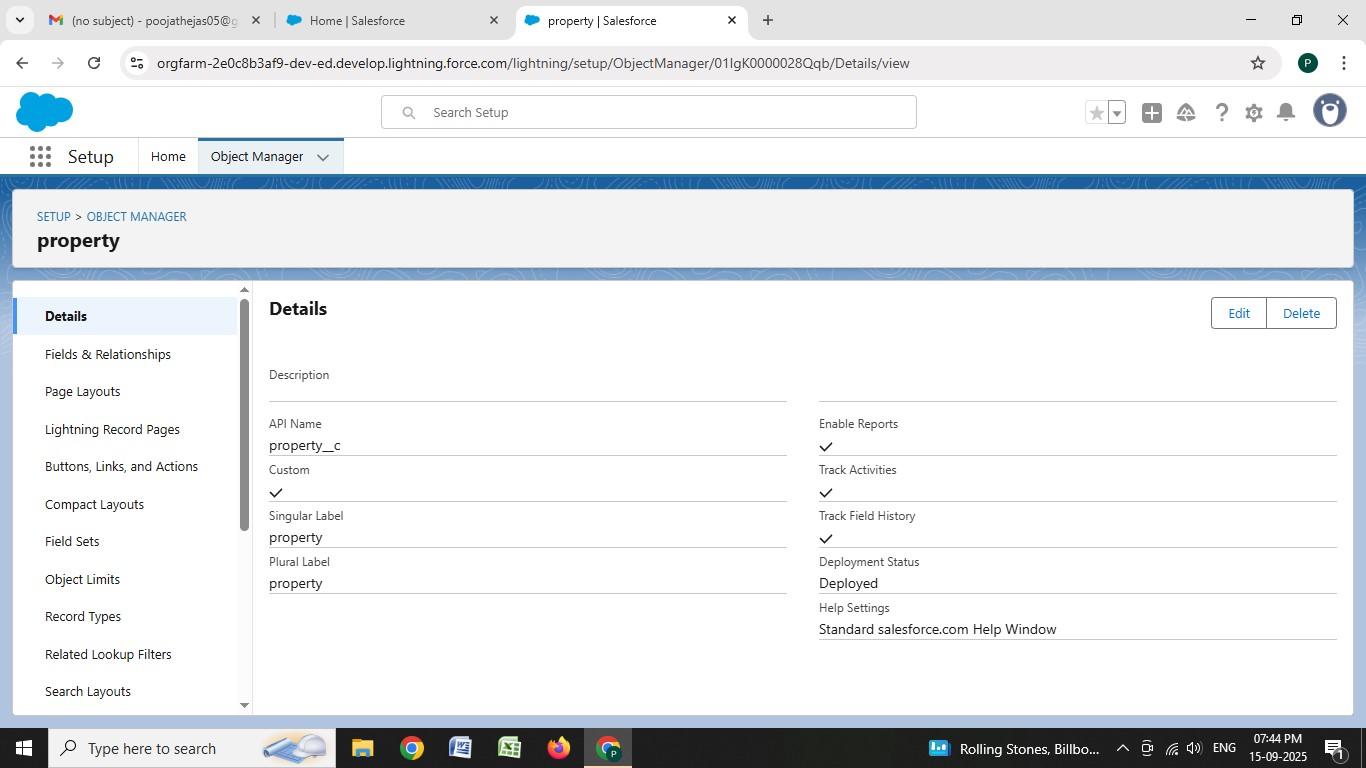
* **Created objects**: Property, Tenant, Lease, Payment



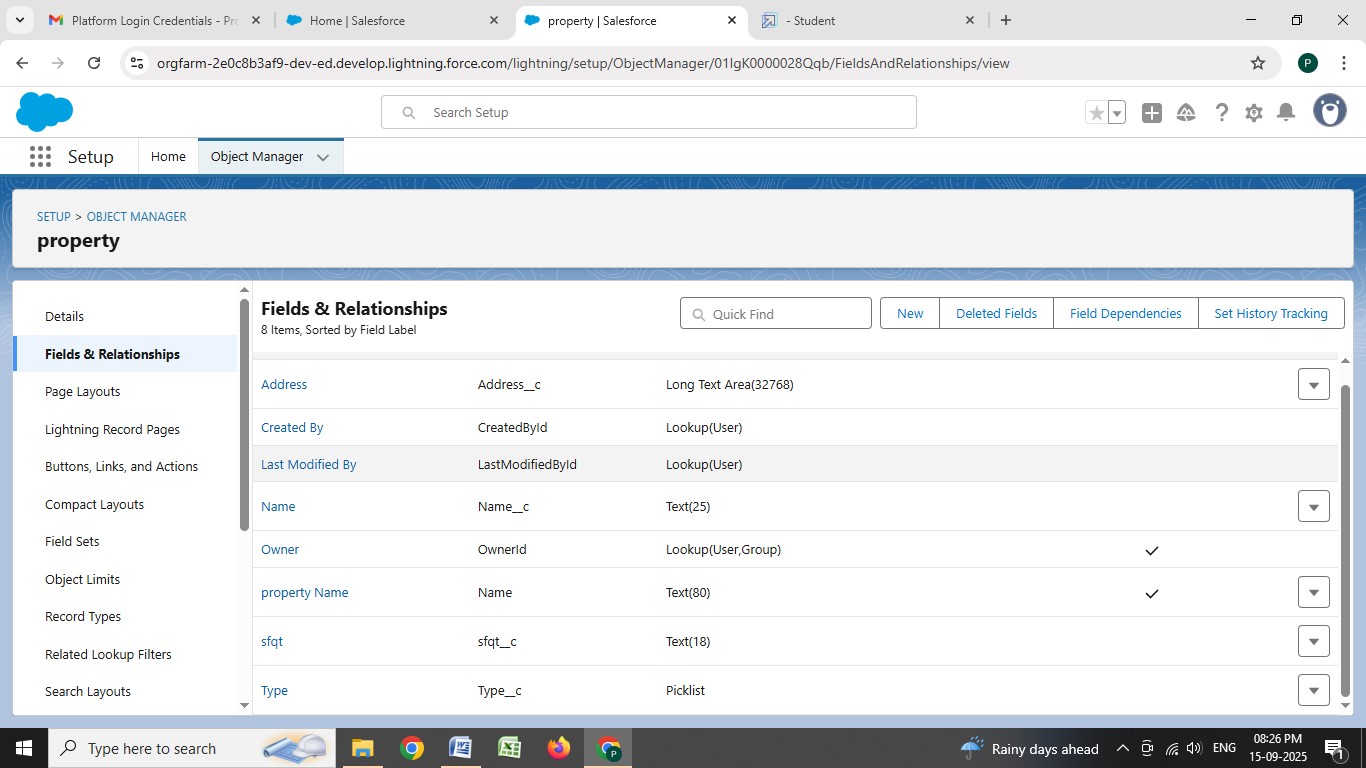


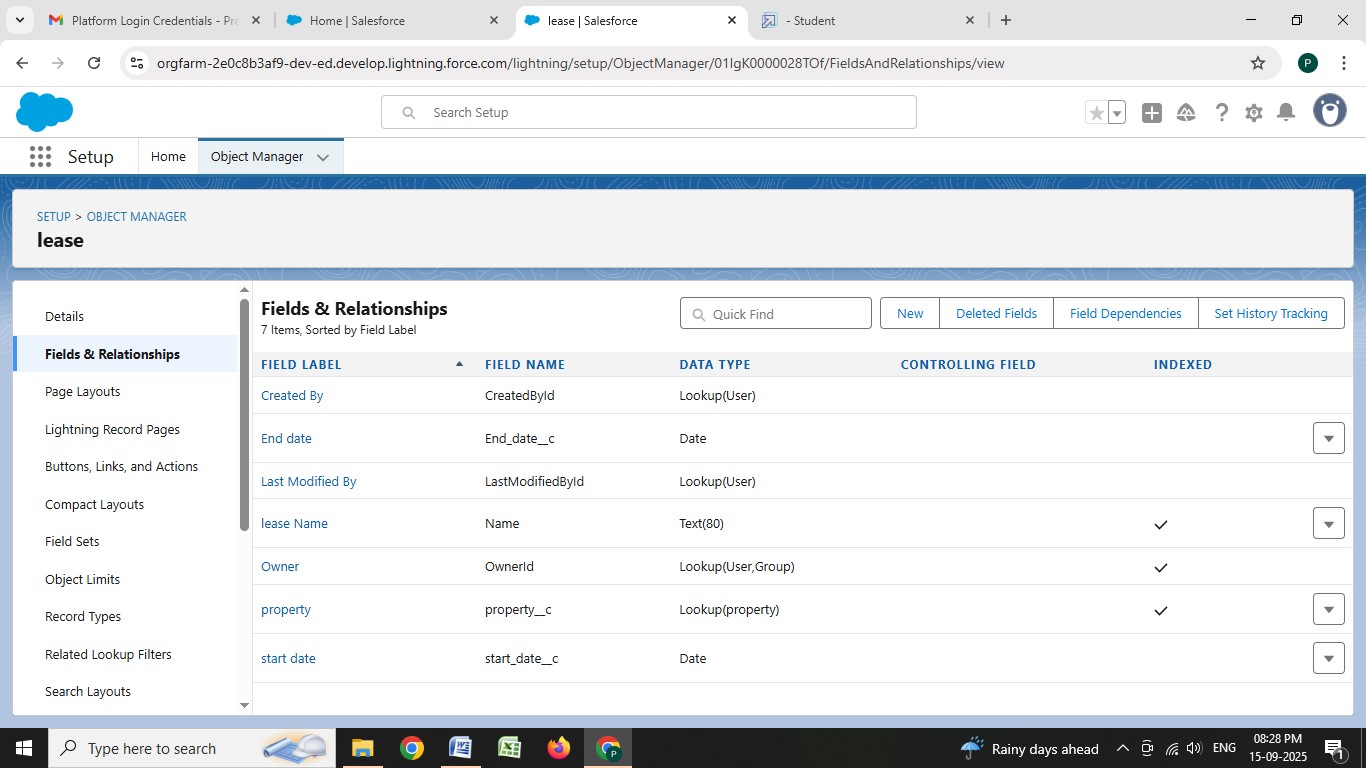


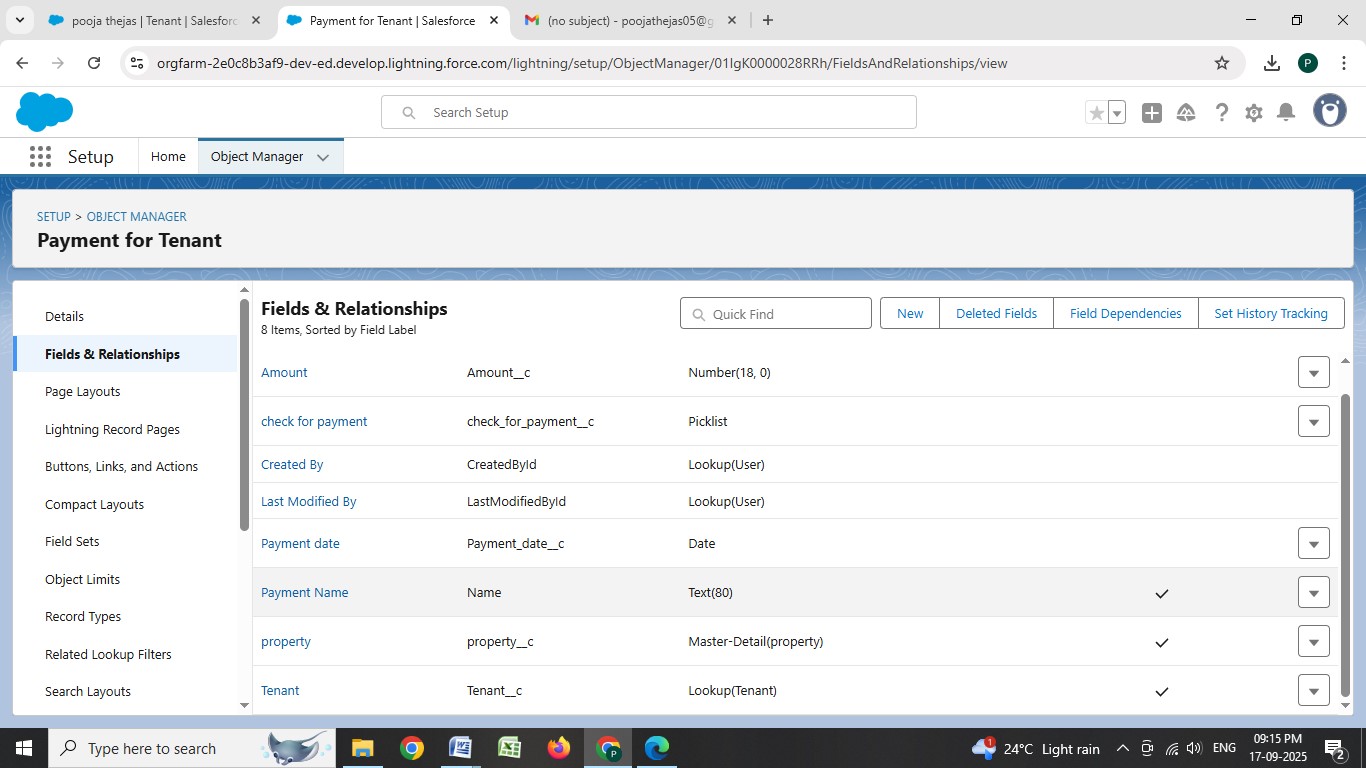




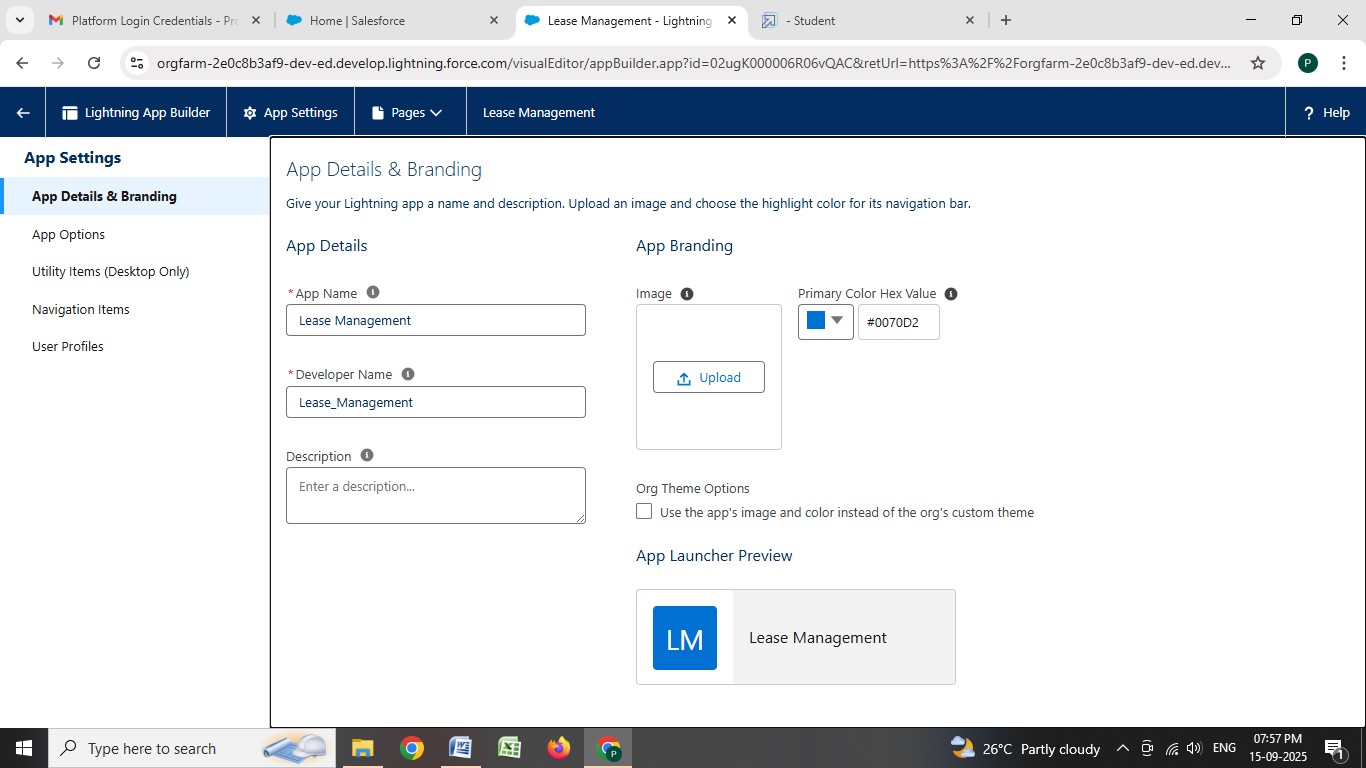
* **Configured fields and relationships**

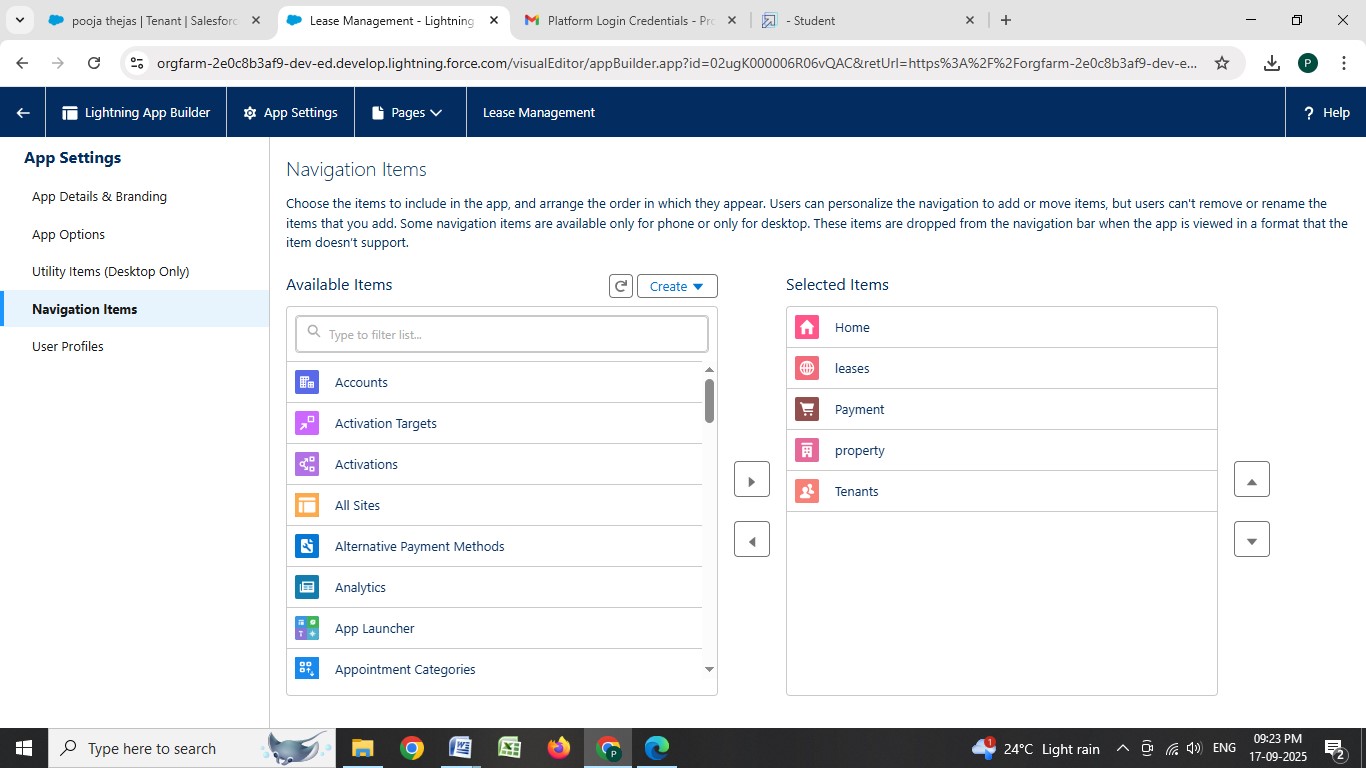


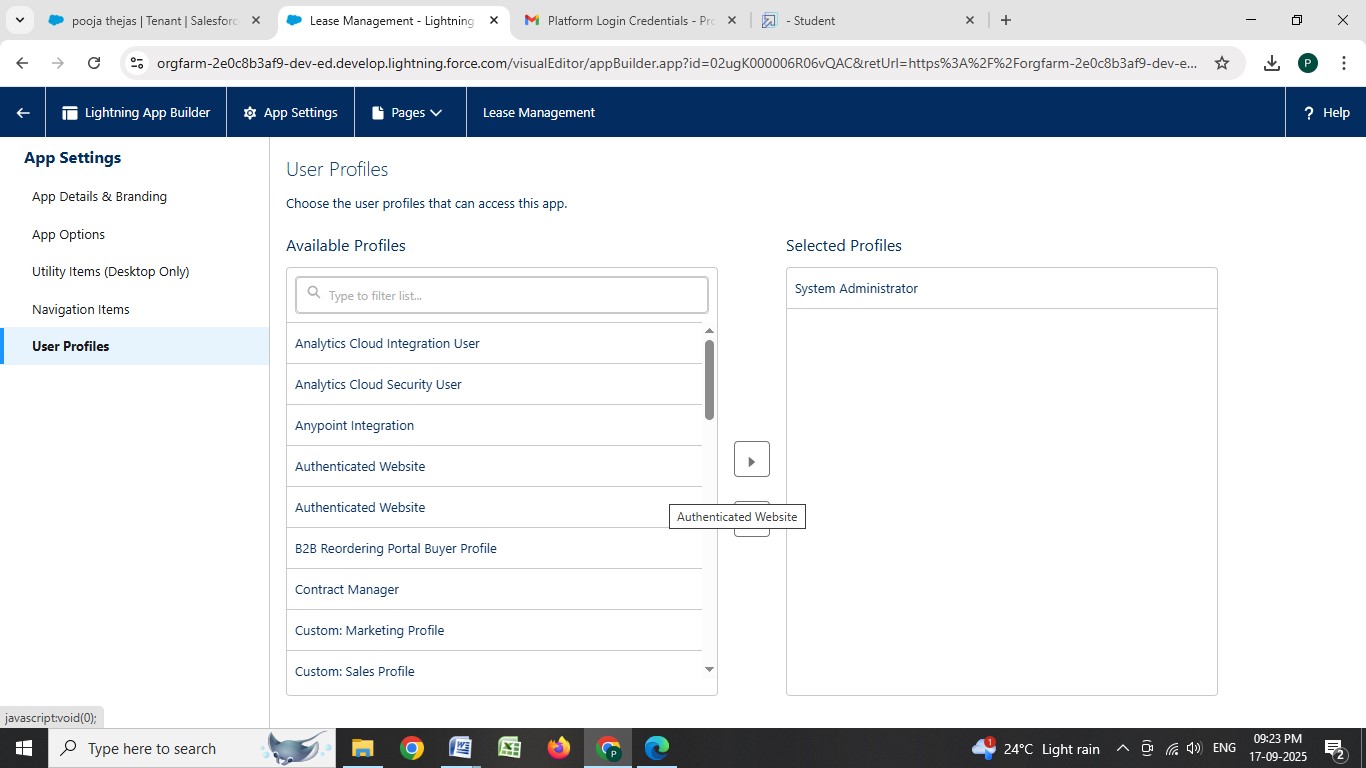




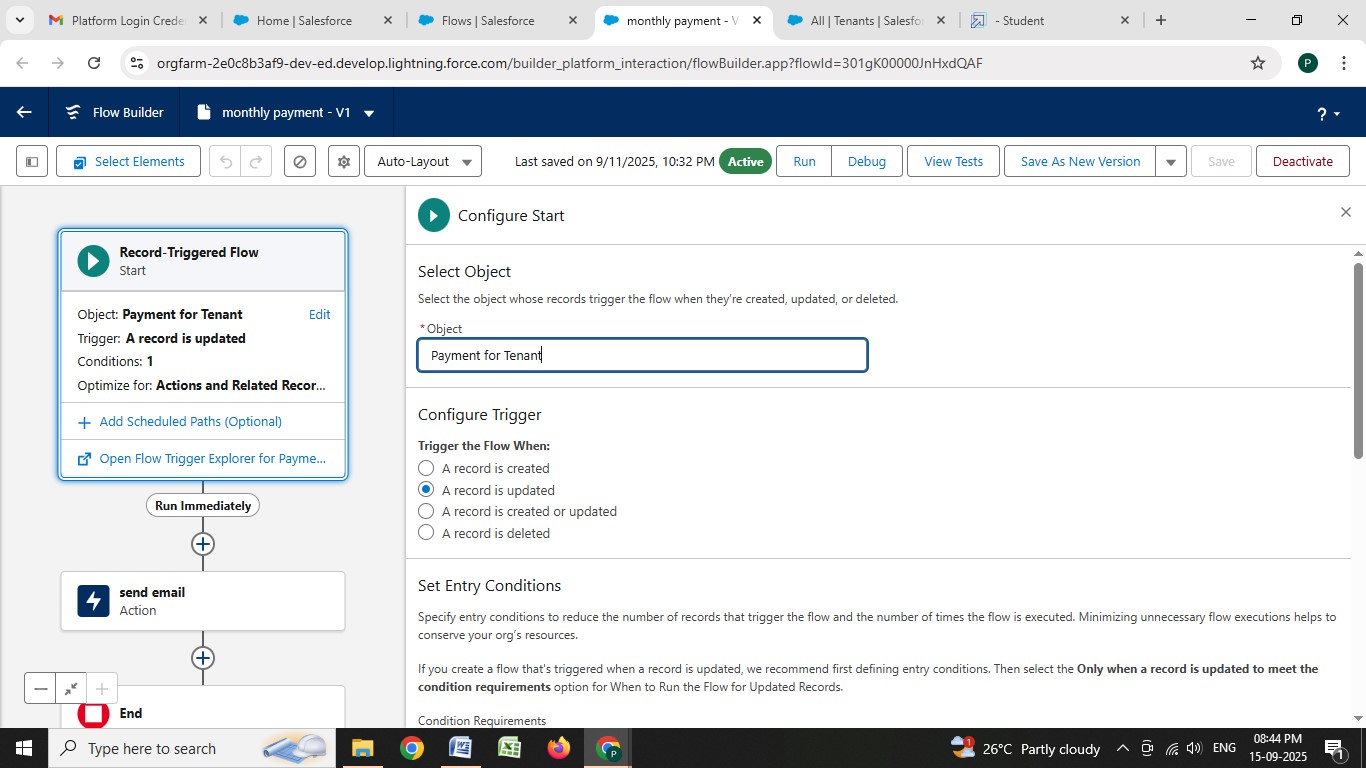
* **Developed Lightning App with relevant tabs**

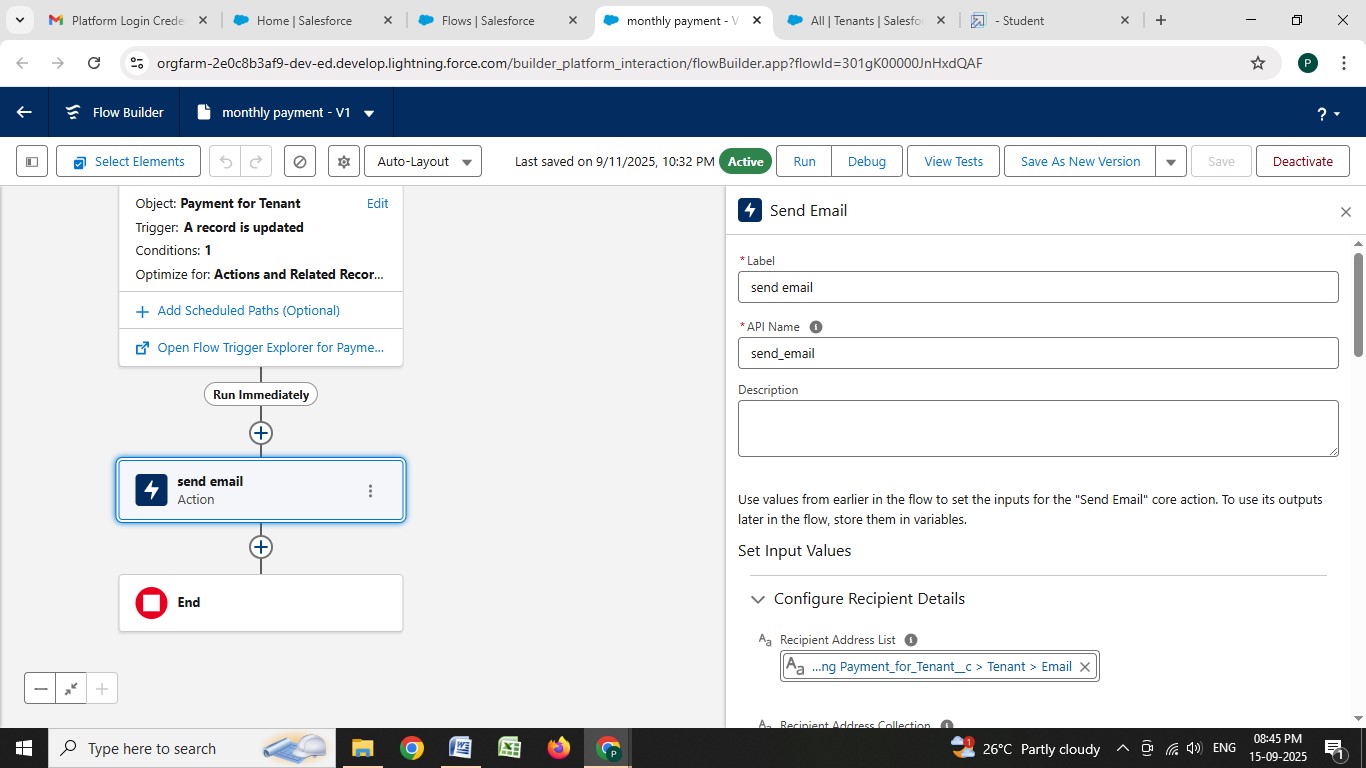




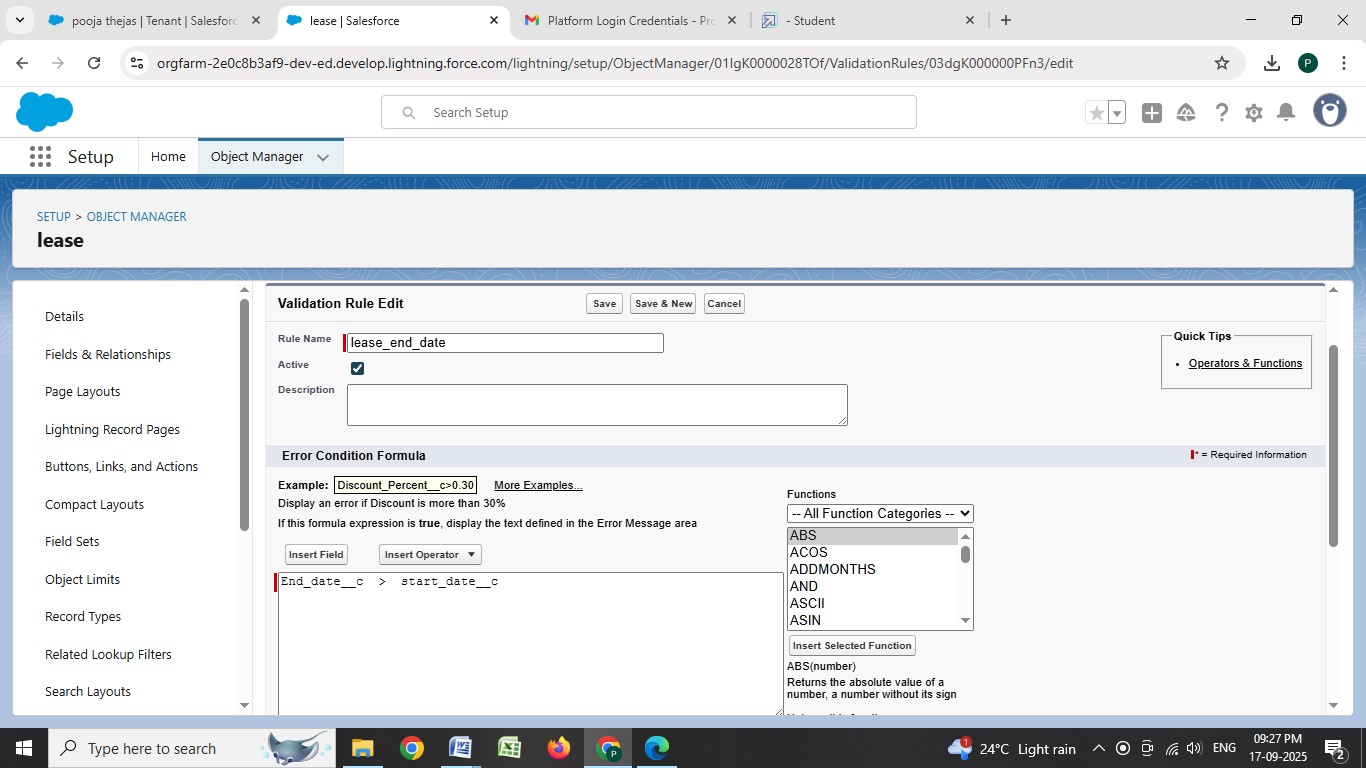


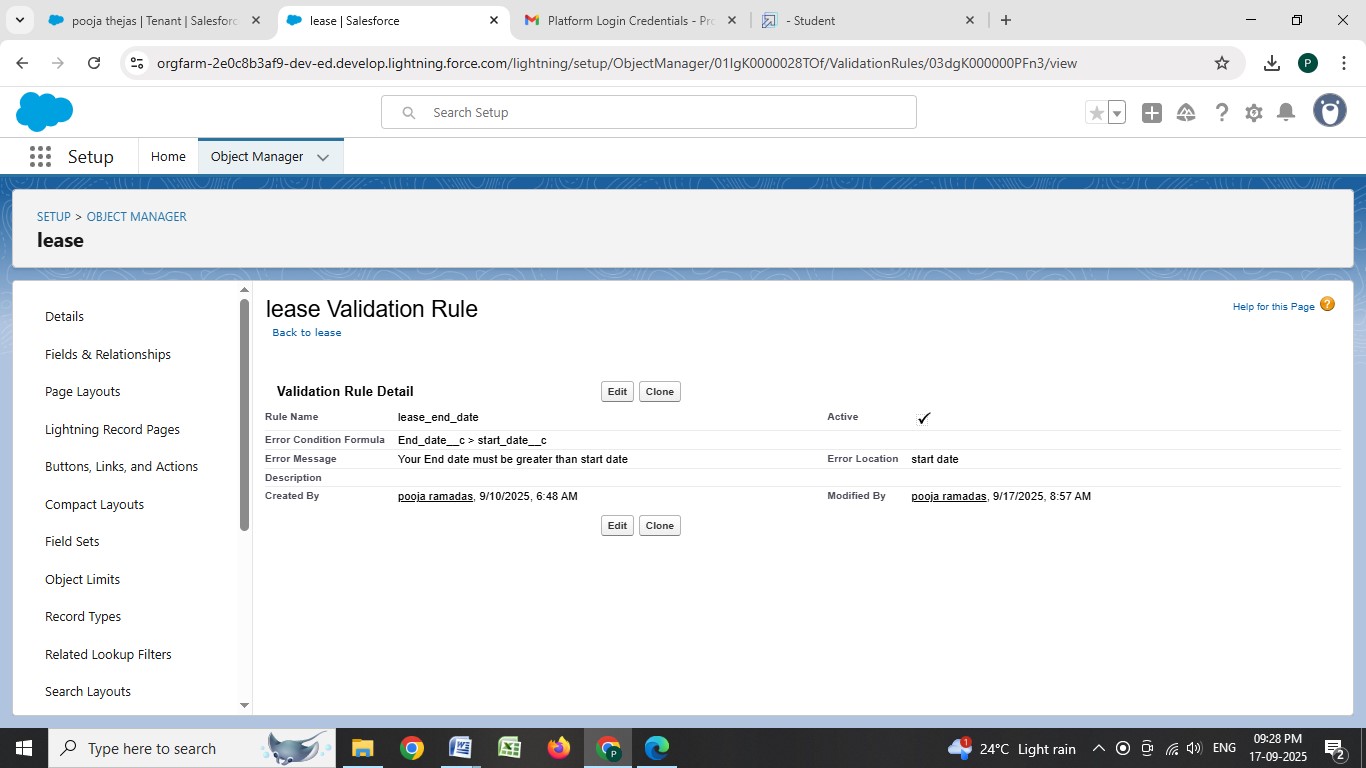
**Implemented Flows for monthly rent and payment success**



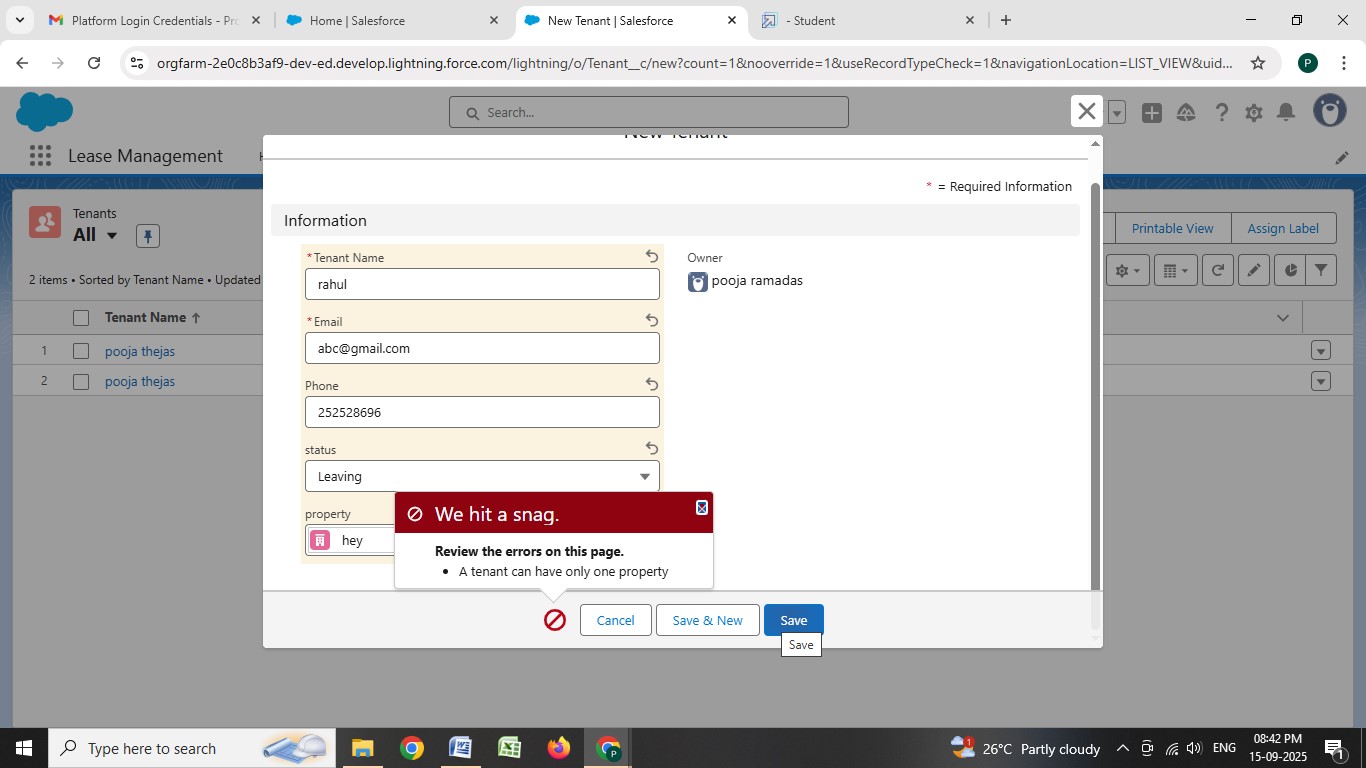


**To create a validation rule to a Lease Object**

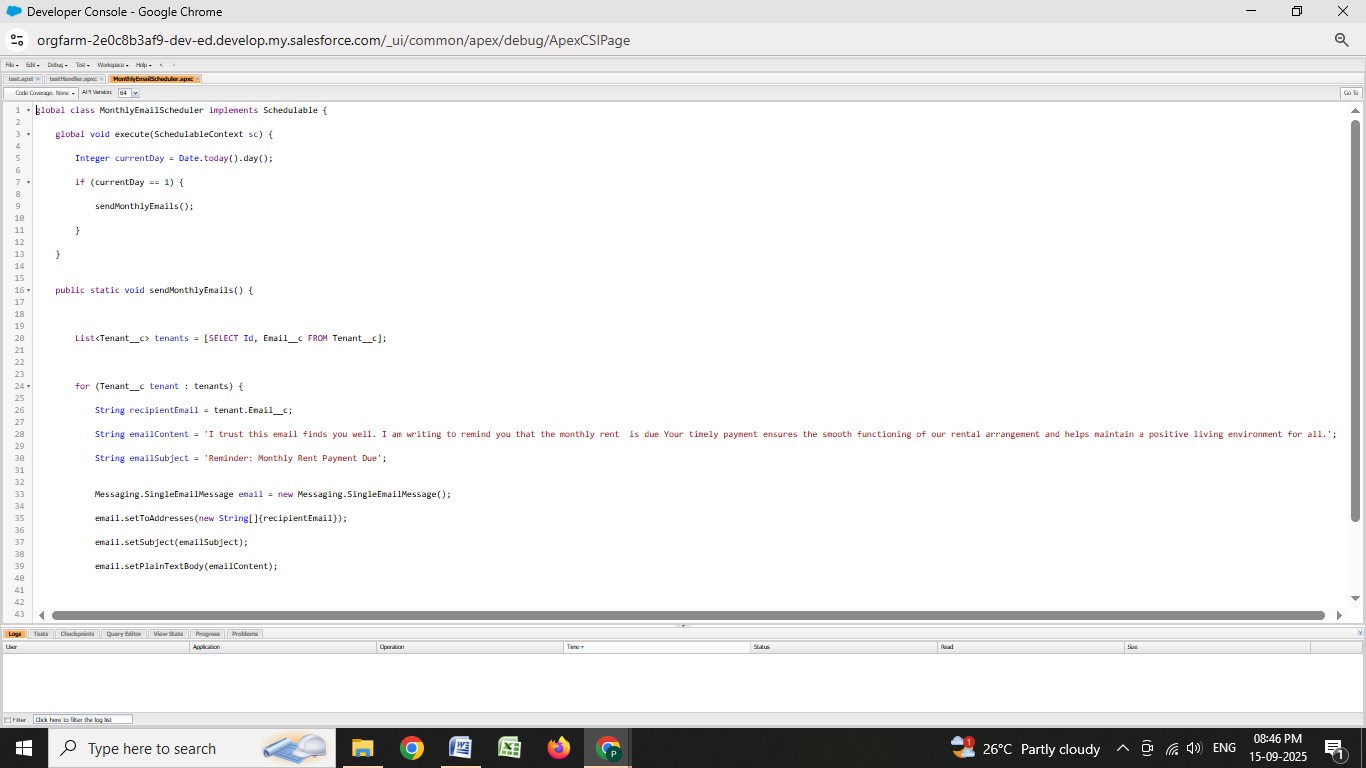




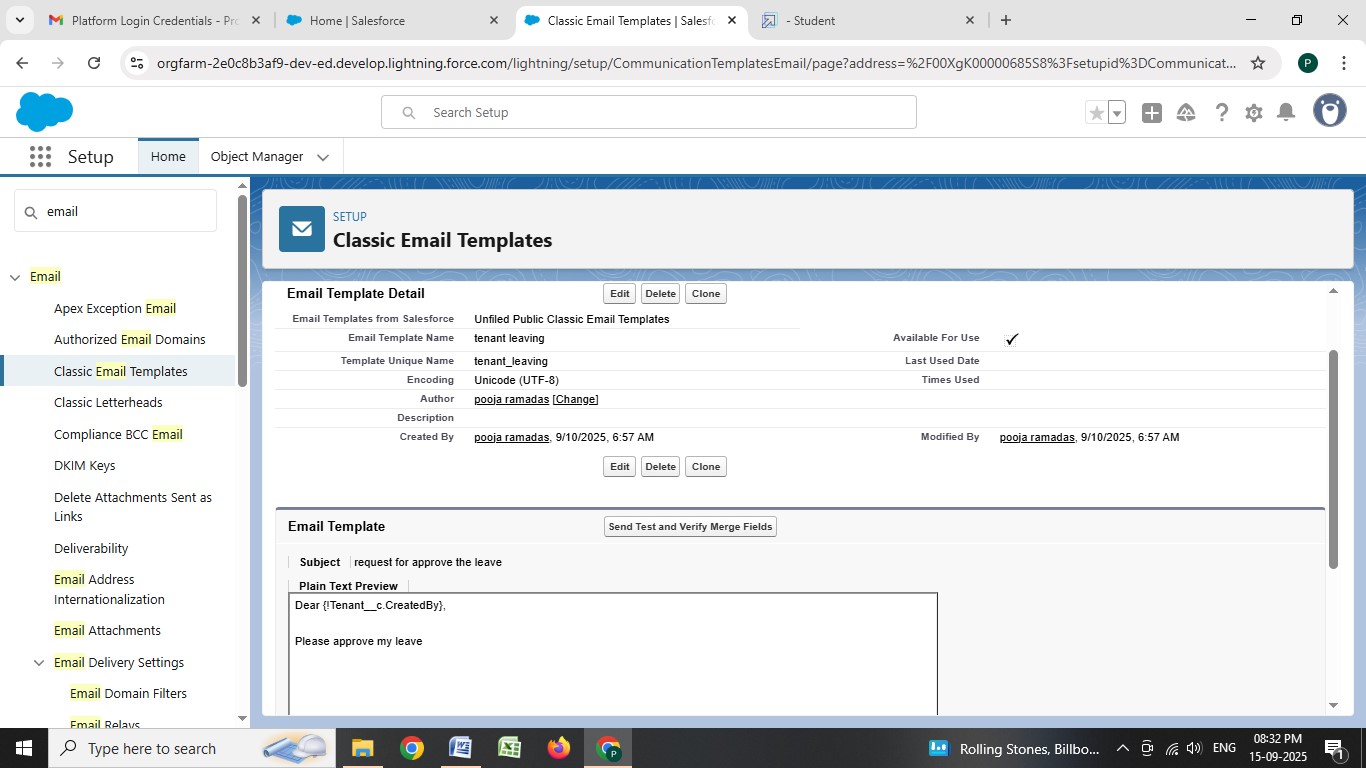
**Added Apex trigger to restrict multiple tenants per property**

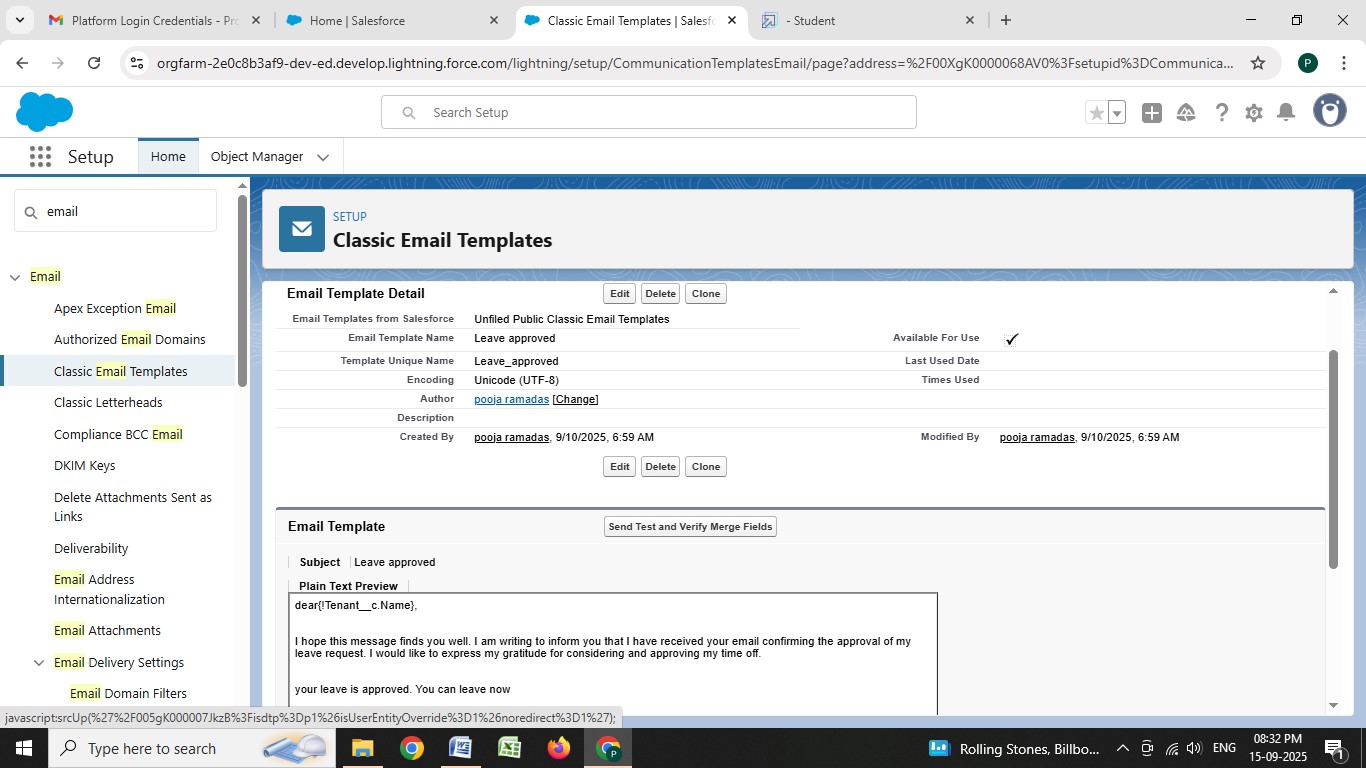


* **Scheduled monthly reminder emails using Apex class**

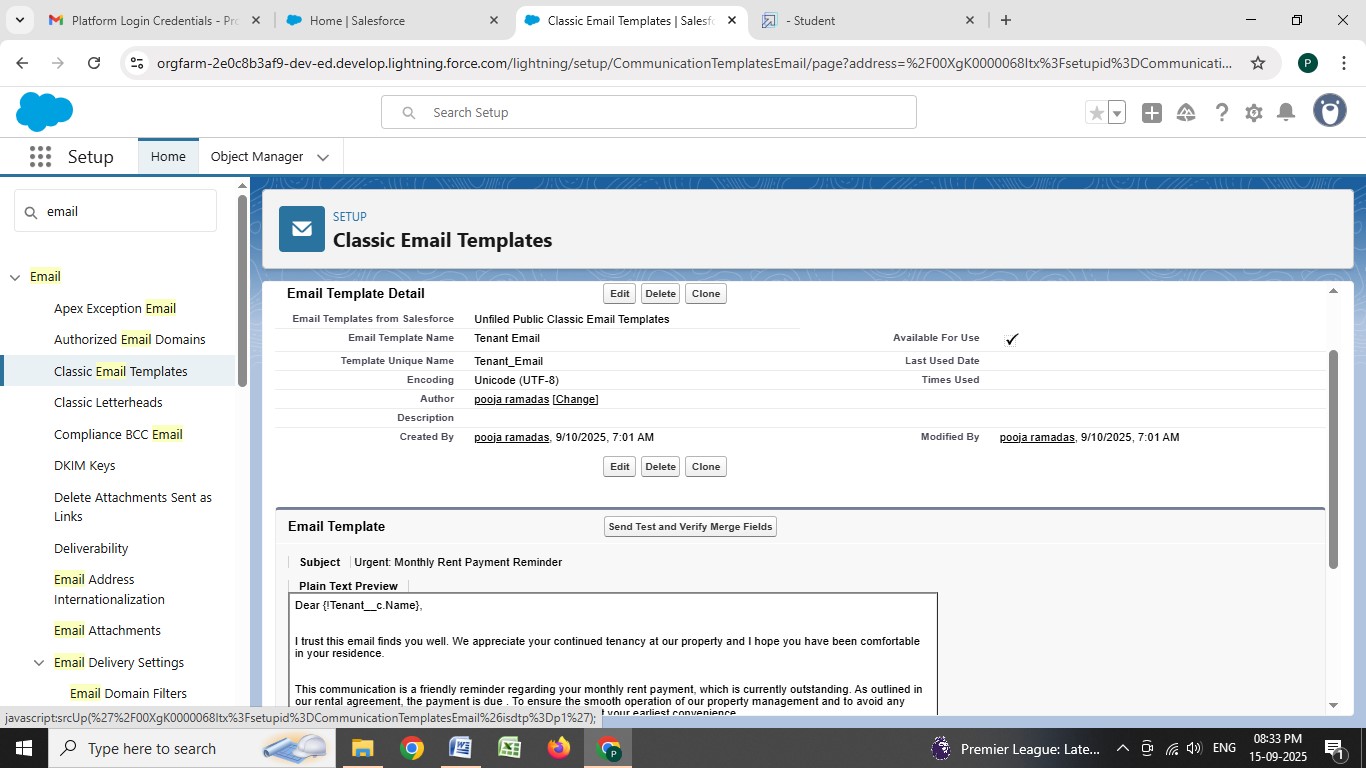


**Built and tested email templates for tenant leaving, leave approval, rejected, monthly payment, success payment**

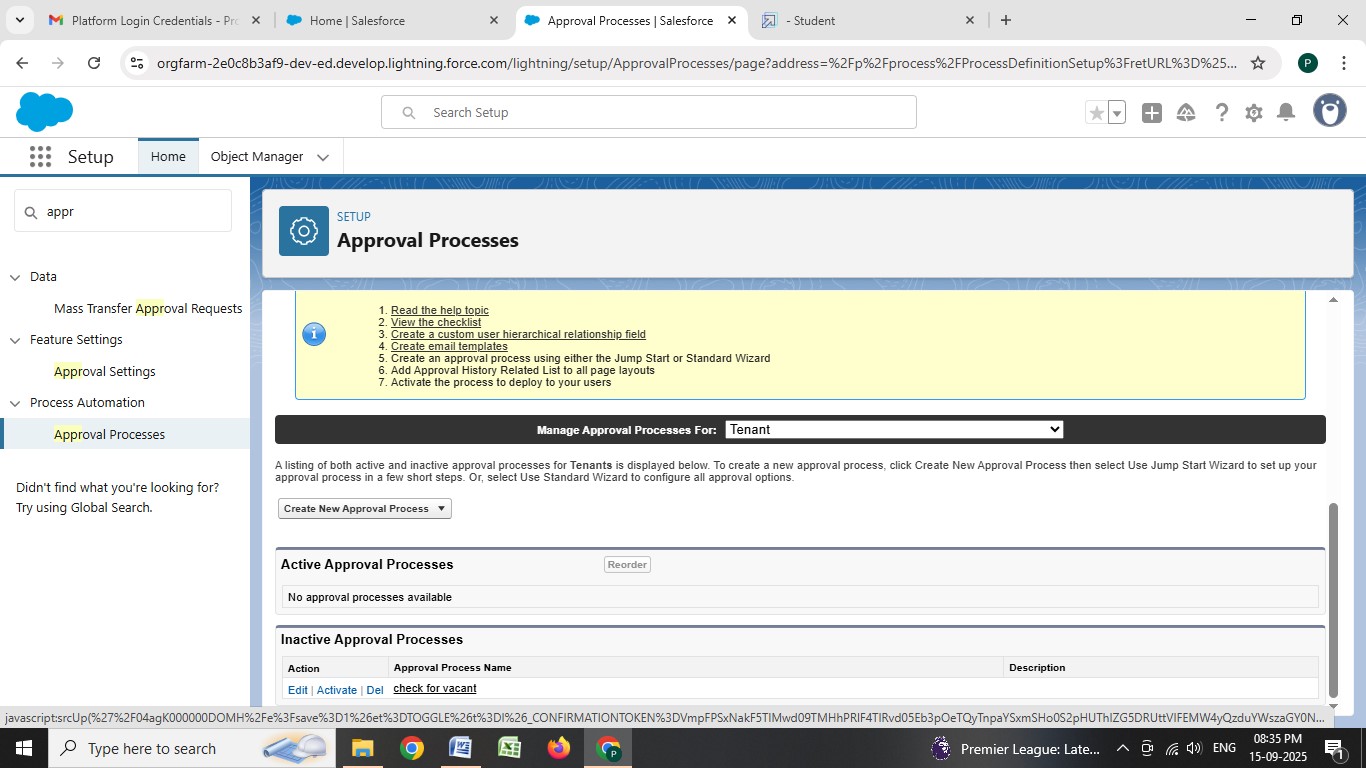




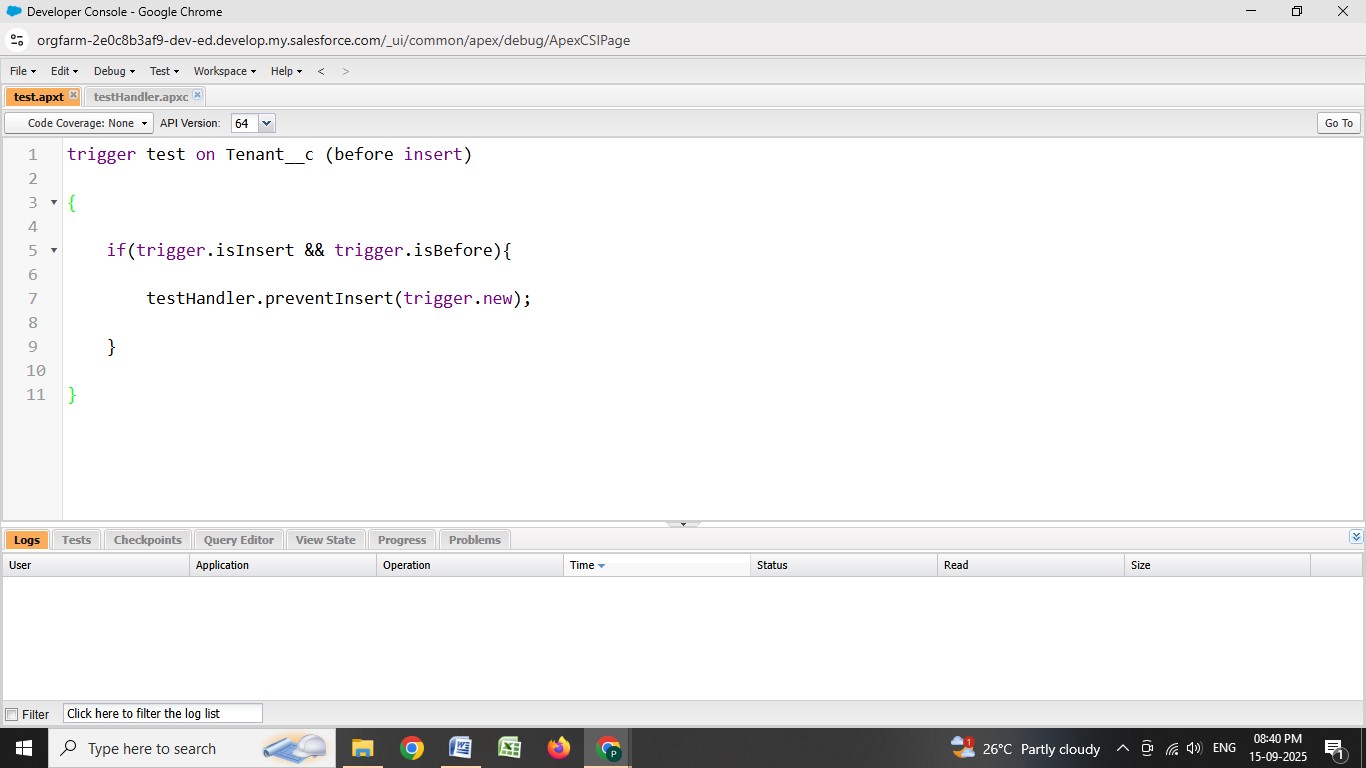




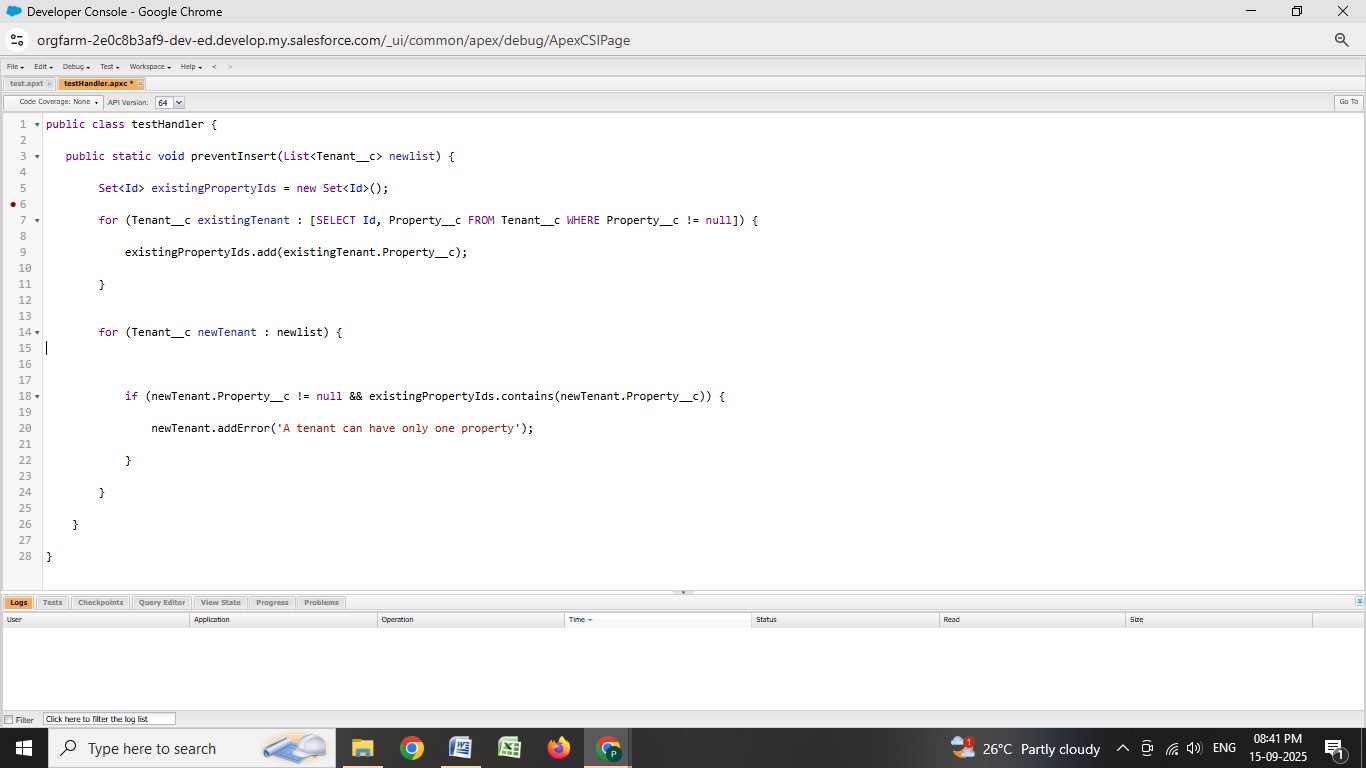
* **For Check for Vacant:**



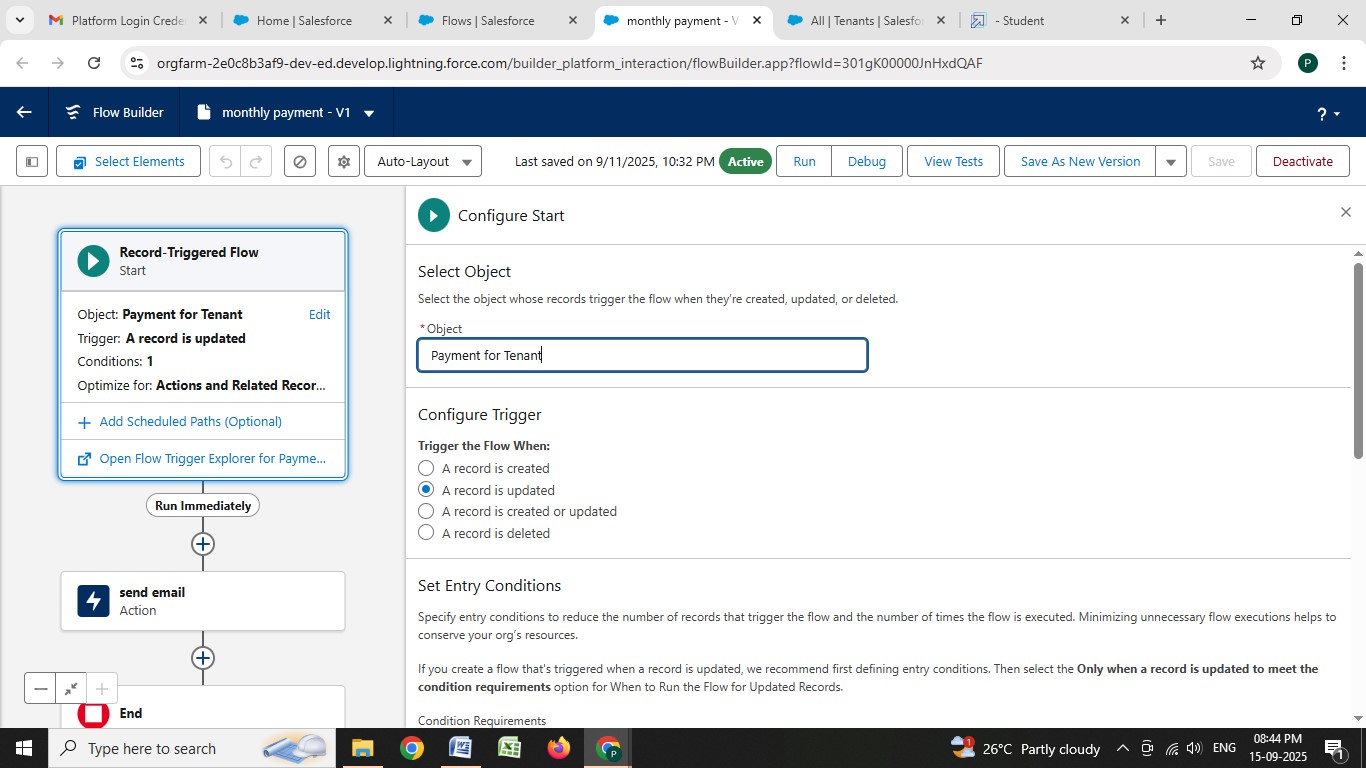
* **Apex Trigger**

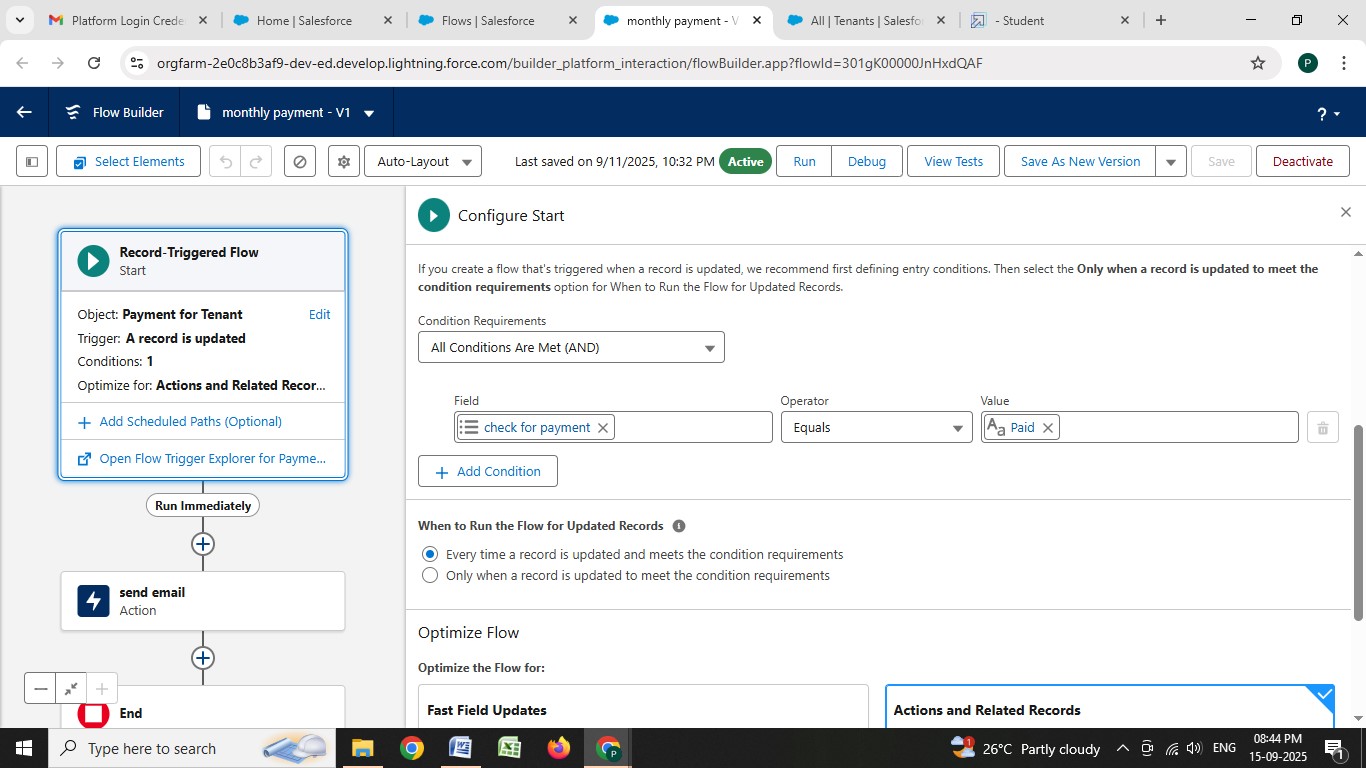


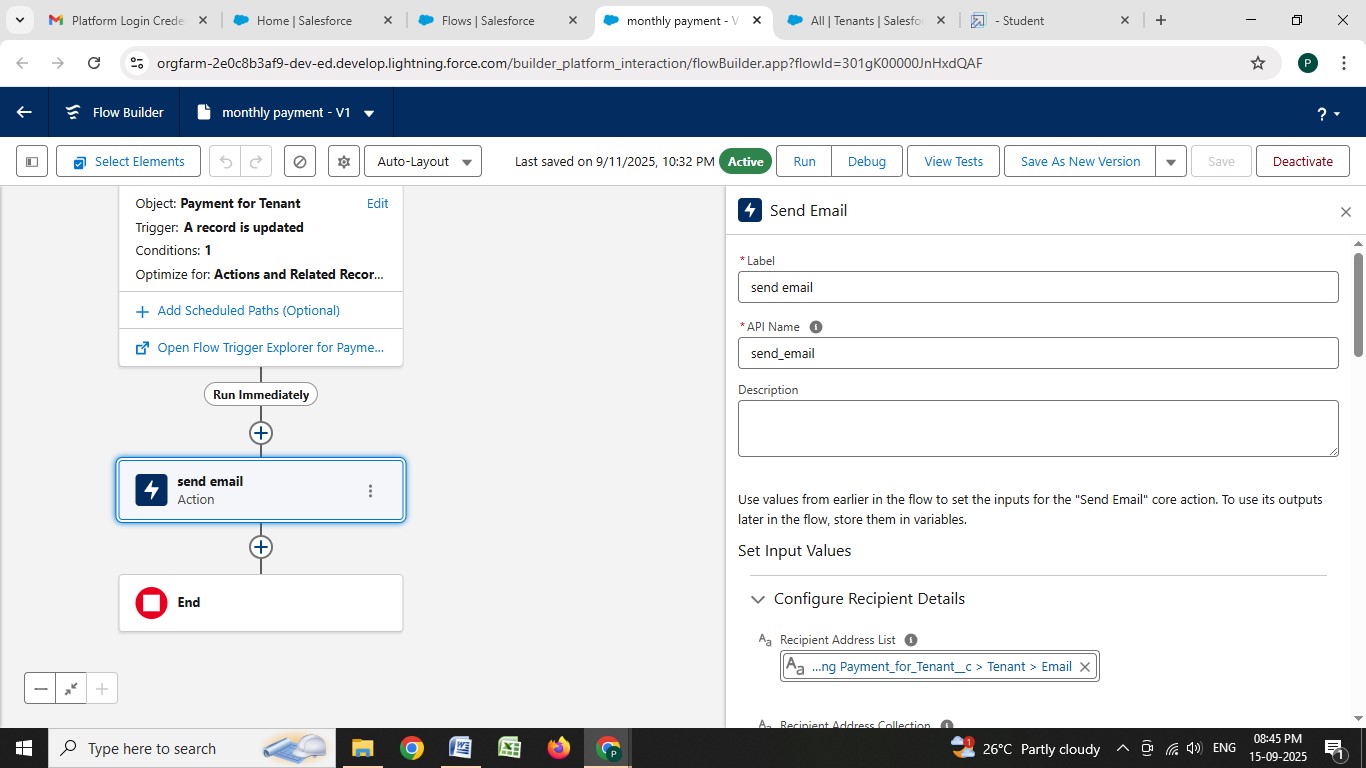
**Create an Apex Handler class**

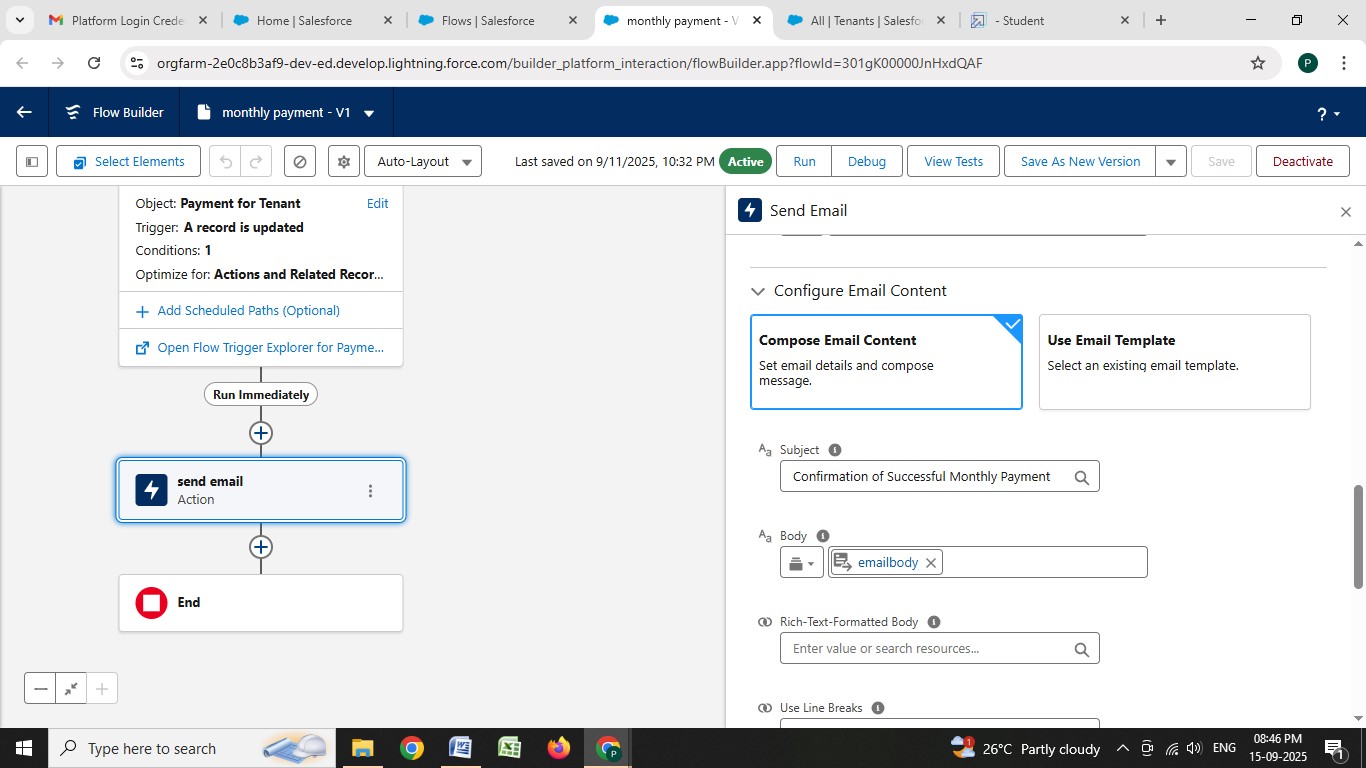


* **FLOWS**



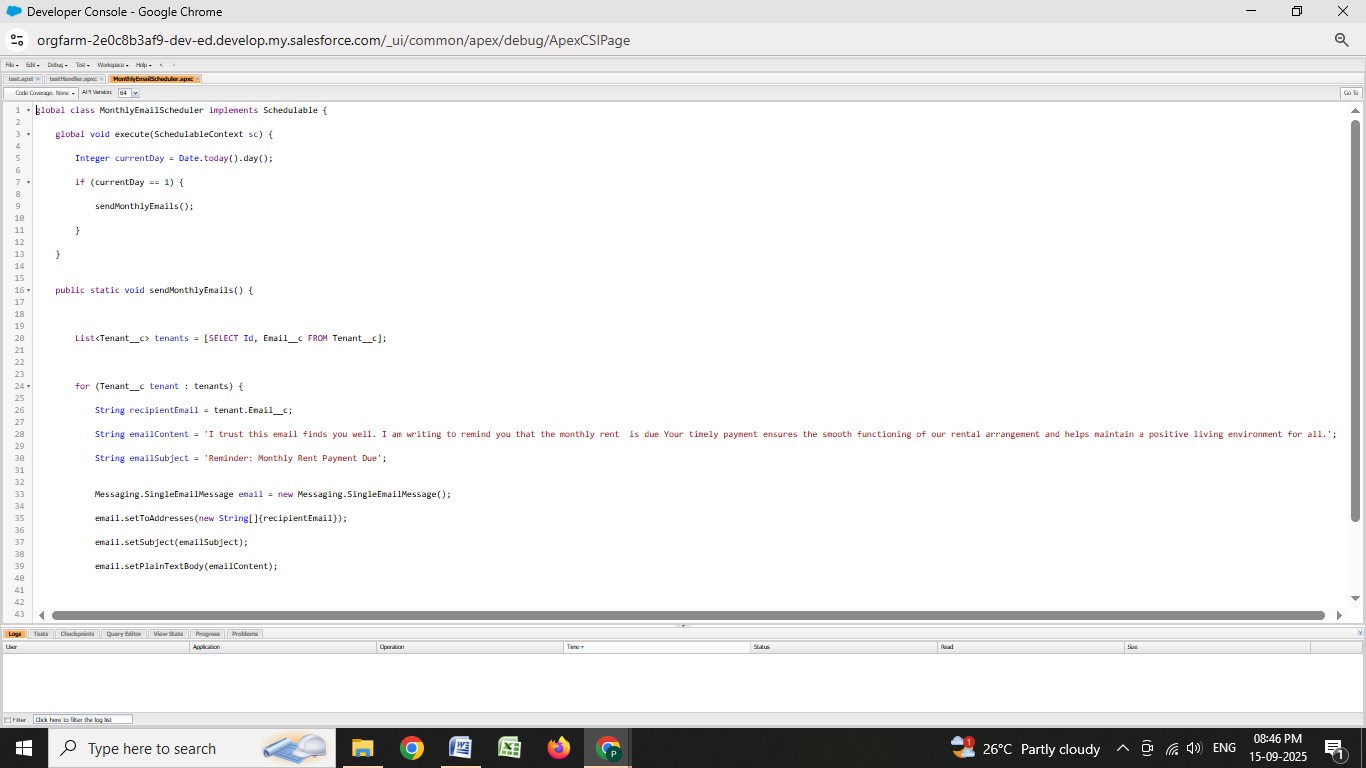






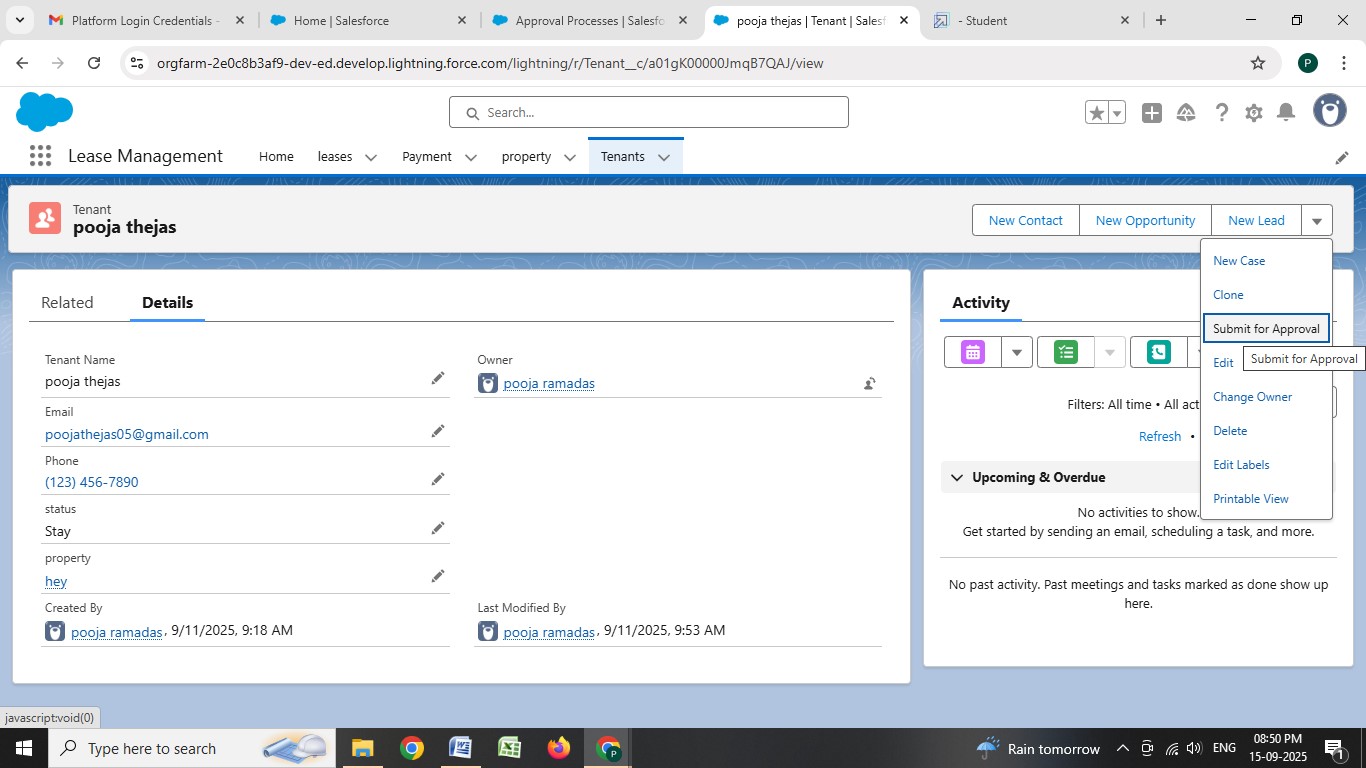
**Schedule class:**

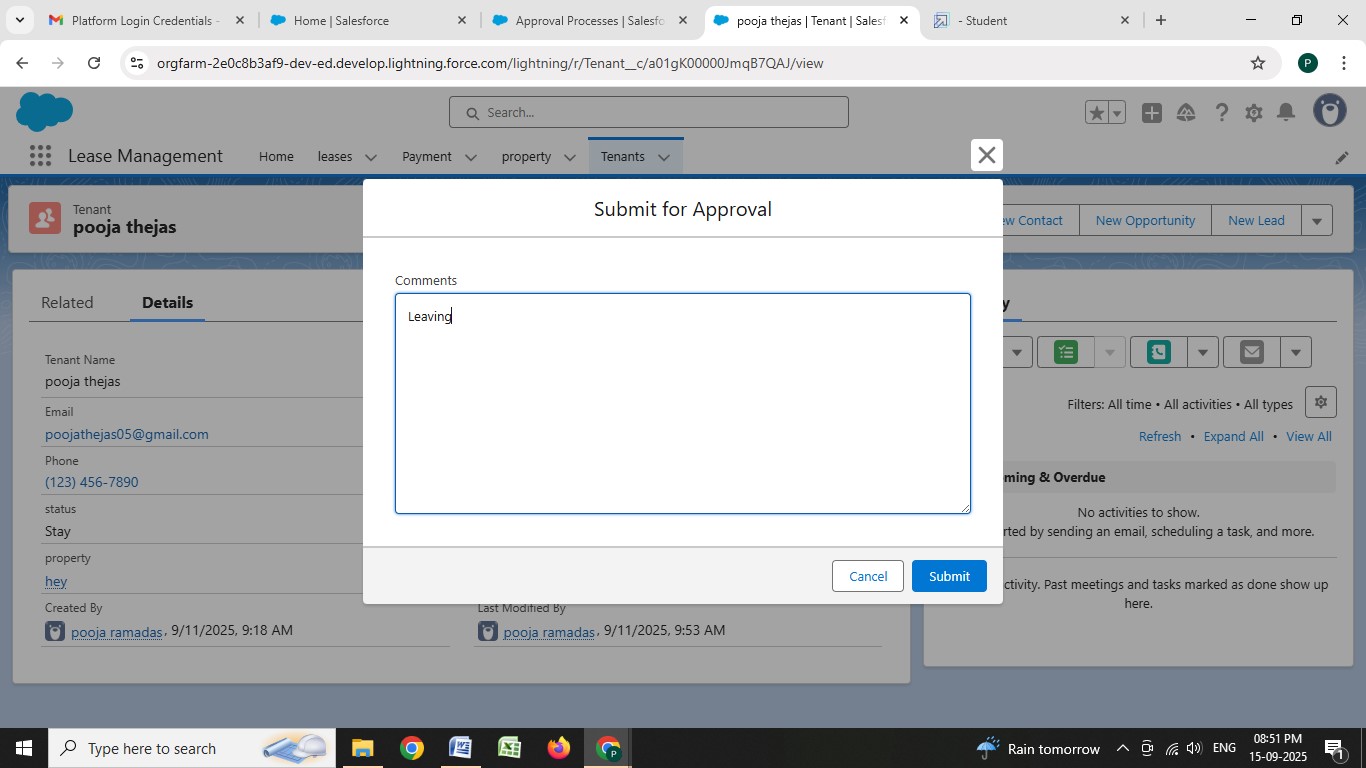
* **Create an Apex Class**

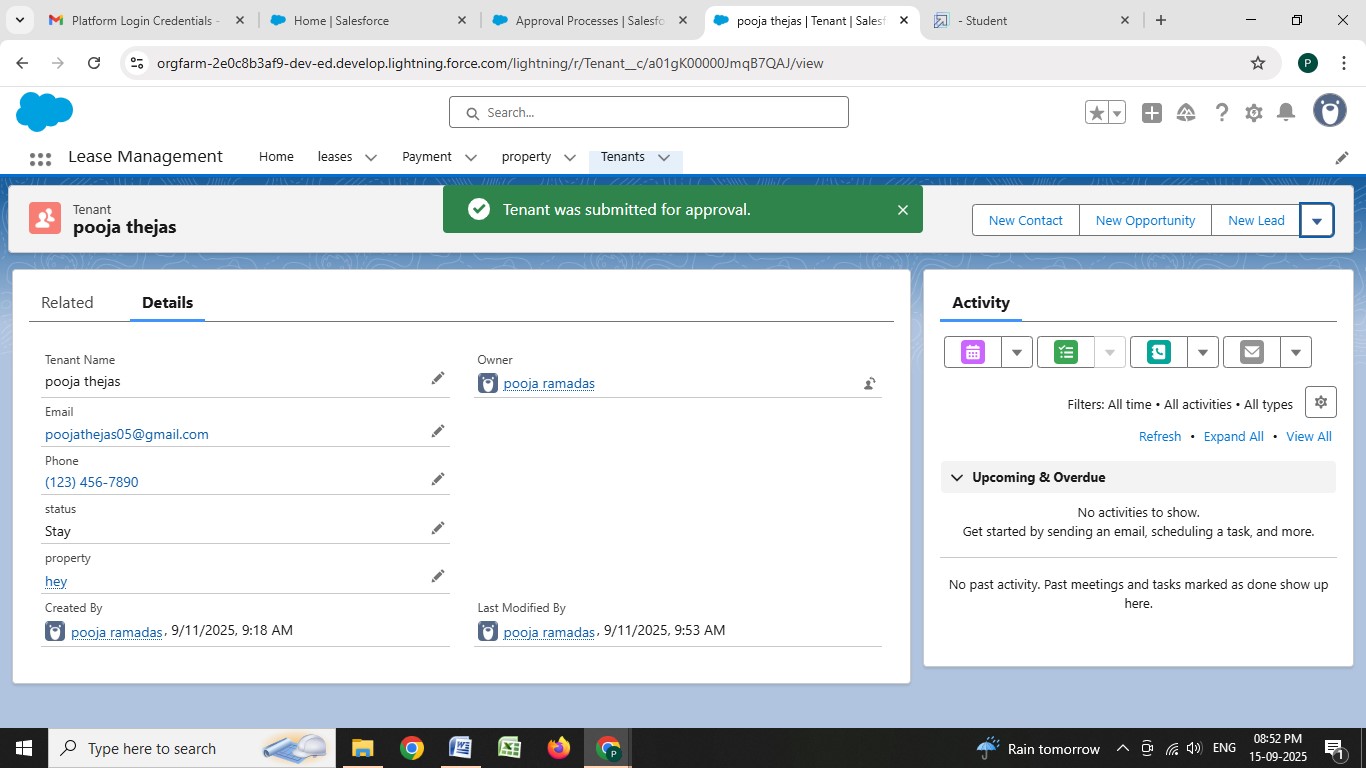


* **Schedule Apex class**





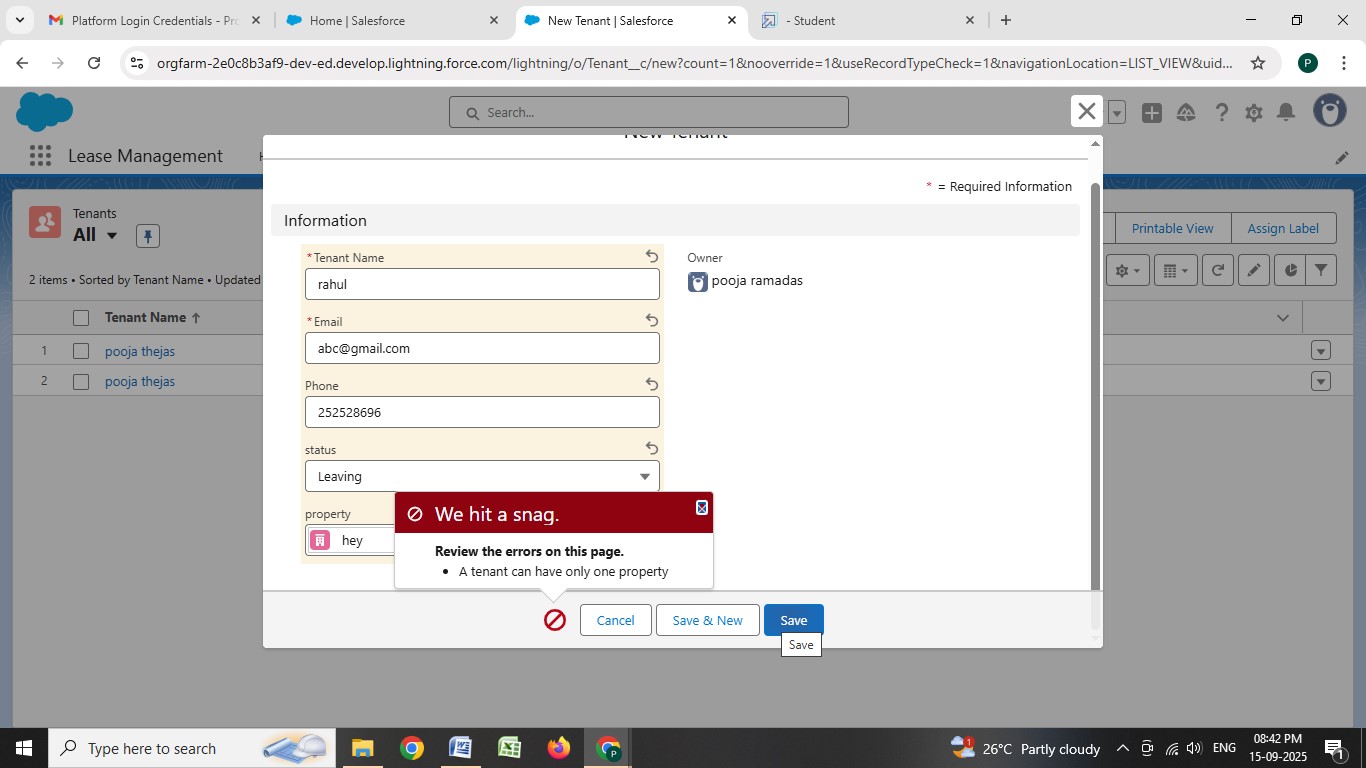




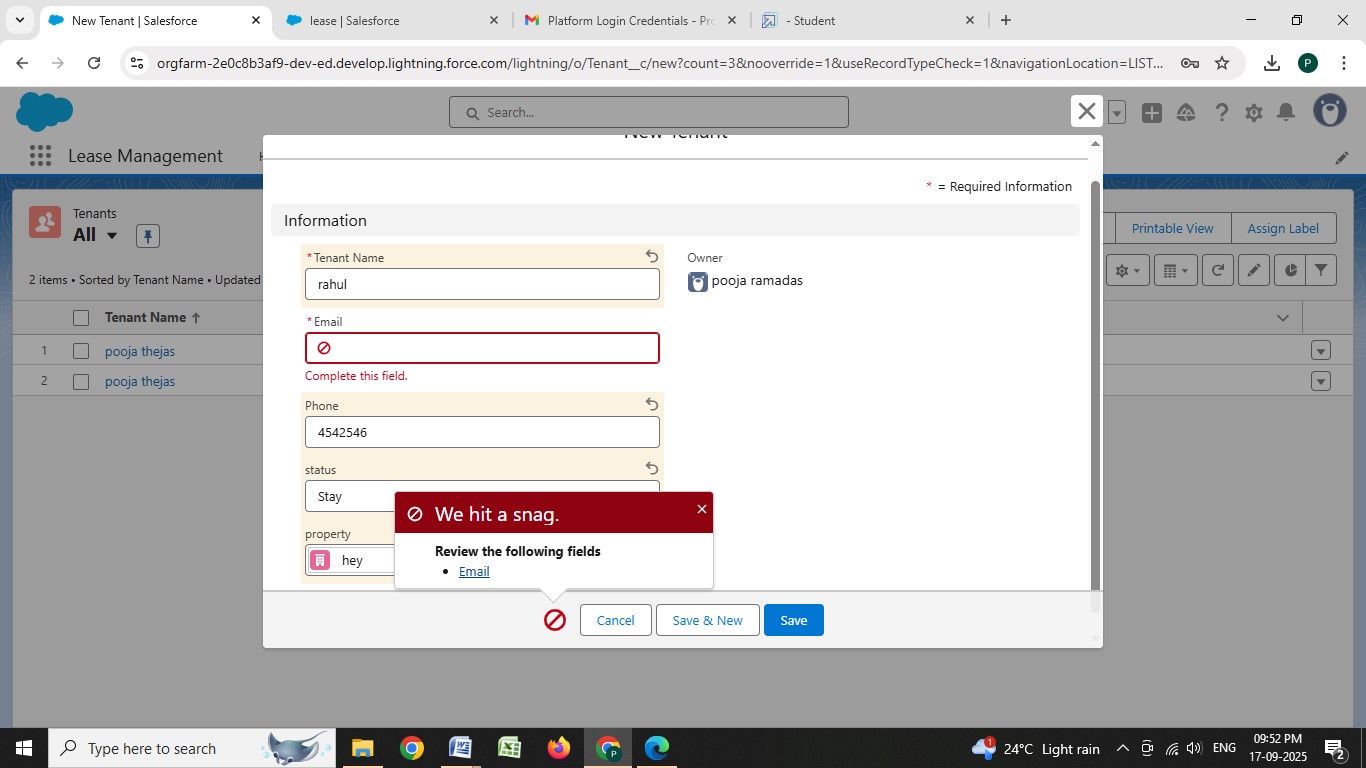
# 3.FUNCTIONAL AND PERFORMANCE TESTING

**Performance Testing**

● Trigger validation by entering duplicate tenant-property records



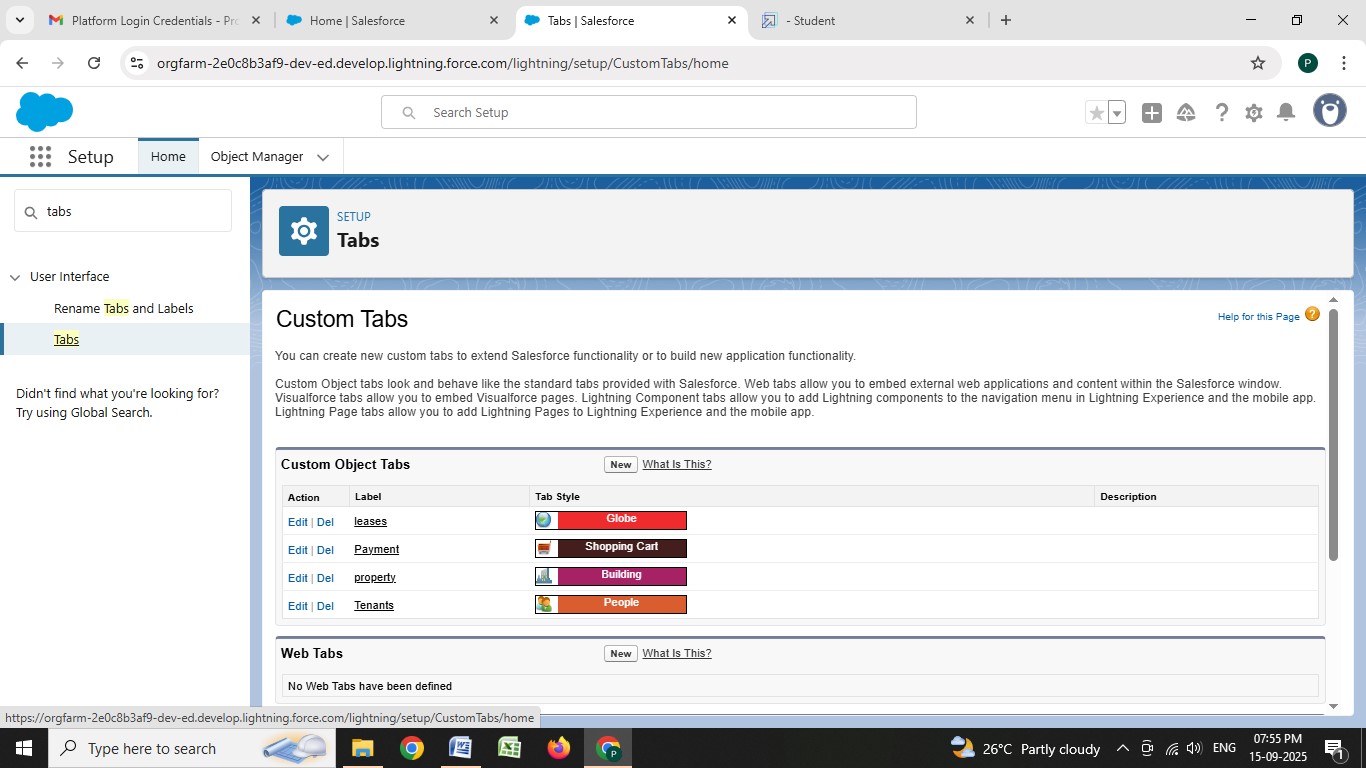
 **Validation Rule checking**



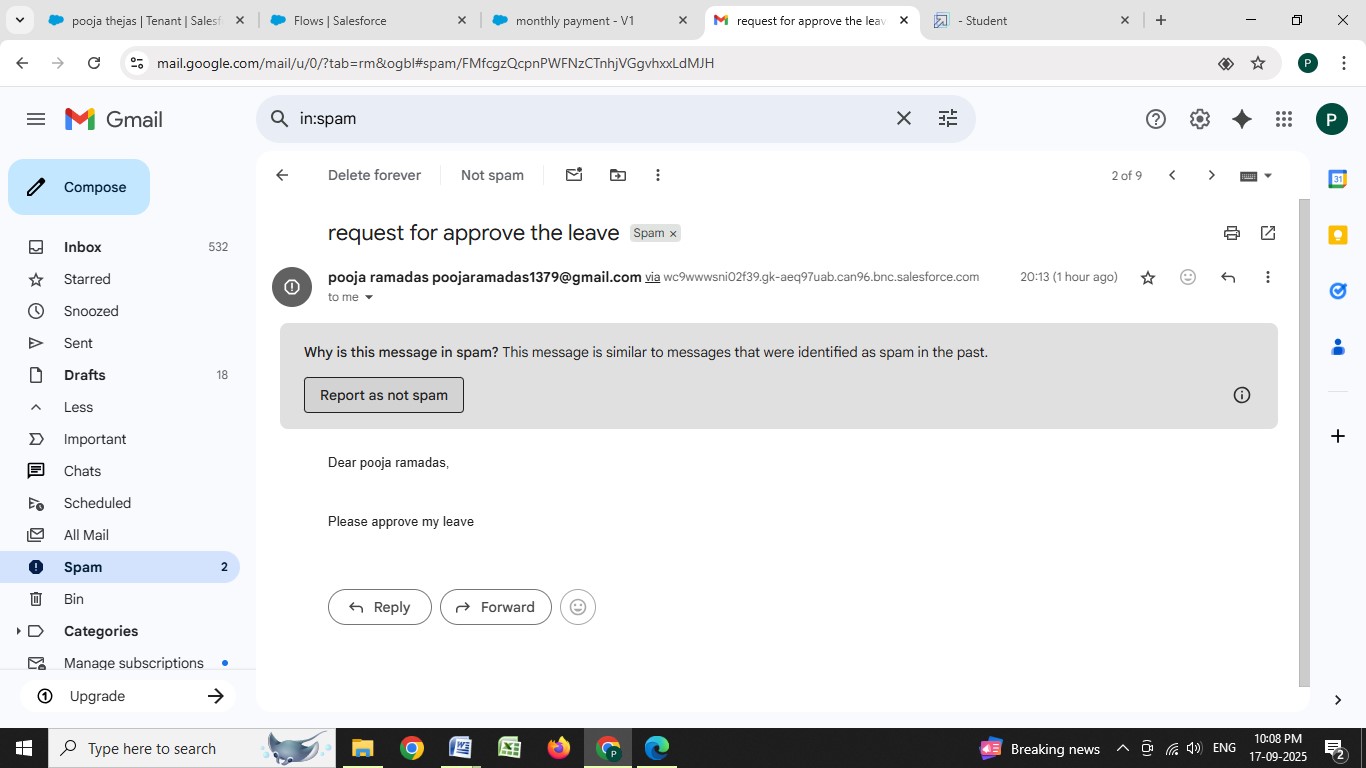
## 4.RESULTS

**Output Screenshots**

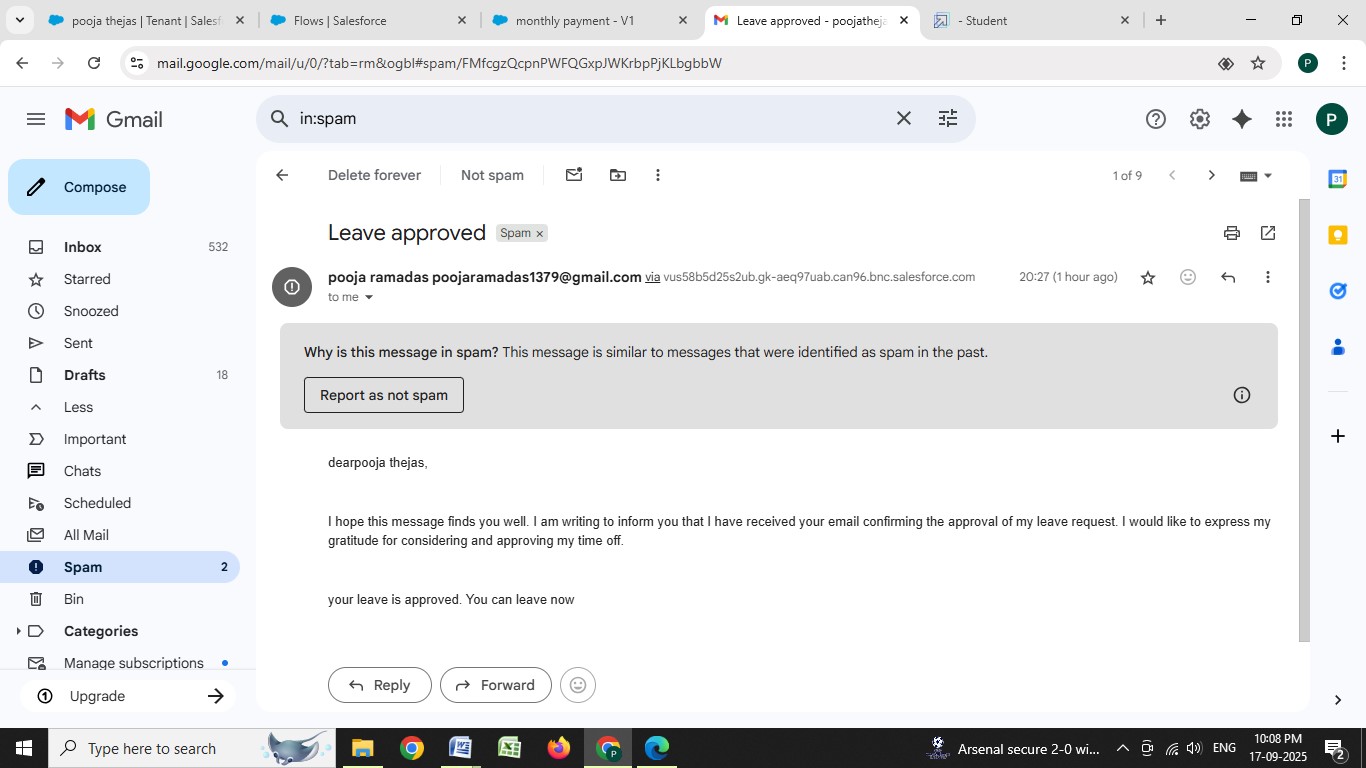
● Tabs for Property, Tenant, Lease, Payment



**Request for approve the leave**



**Leave approved**



# 5. APPENDIX

● Source Code:

Provided in Apex Classes and Triggers

**Test.apxt**

Trigger Code:

trigger test on Tenant\_\_c (before insert)

{

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

}

}

**testHandler.apxc**

Apex logic: 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’);

}

}

}

}

**MonthlyEmailScheduler.apxc**

Apex logic: 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});

}

}

}

## 6. 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.