LEASE MANAGEMENT

**COLLEGE NAME :**SMS COLLEGE OF ARTS AND SCIENCE

**COLLEGE CODE :**bru4s

## TEAM ID :NM2025TMID28691

**TEAM MEMBERS:**

**Team LeaderName :** Mythili.P

**[Email:](mailto:mythilipalanisamy2203@gmail.com)**[mythilipalanisamy2203@gmail.com](mailto:mythilipalanisamy2203@gmail.com)

**Team Member 1 :**Sriramkumar.S

**Email [:](mailto:sriramkumar14356@gmail.com)**[sriramkumar14356@gmail.com](mailto:sriramkumar14356@gmail.com)

**Team Member 2 :**Poojasri.J

**Email** [:jawaharlalpooja23@gmail.com](mailto:jawaharlalpooja23@gmail.com)

**Team Member 3:**Poovarasan.M

**Email [:](mailto:poovupoovu905@gmail.com)**[poovupoovu905@gmail.com](mailto:poovupoovu905@gmail.com)

## INTRODUCTION

* 1. **Project Overview**

The Lease Management System is a Salesforce-based application that automates and streamlines real estate leasing processes. It manages tenant records, lease contracts, rent payments, and communication within a centralized platform, reducing manual effort and errors. Using Salesforce features like Flows, Approval Processes, and Email Alerts, the system ensures timely reminders, accurate payment tracking, and proper contract approvals. With dashboards and reports, stakeholders gain insights into occupancy, payments, and renewals, enabling better decision-making. Overall, it enhances efficiency, improves tenant satisfaction, and secures data with role-based access



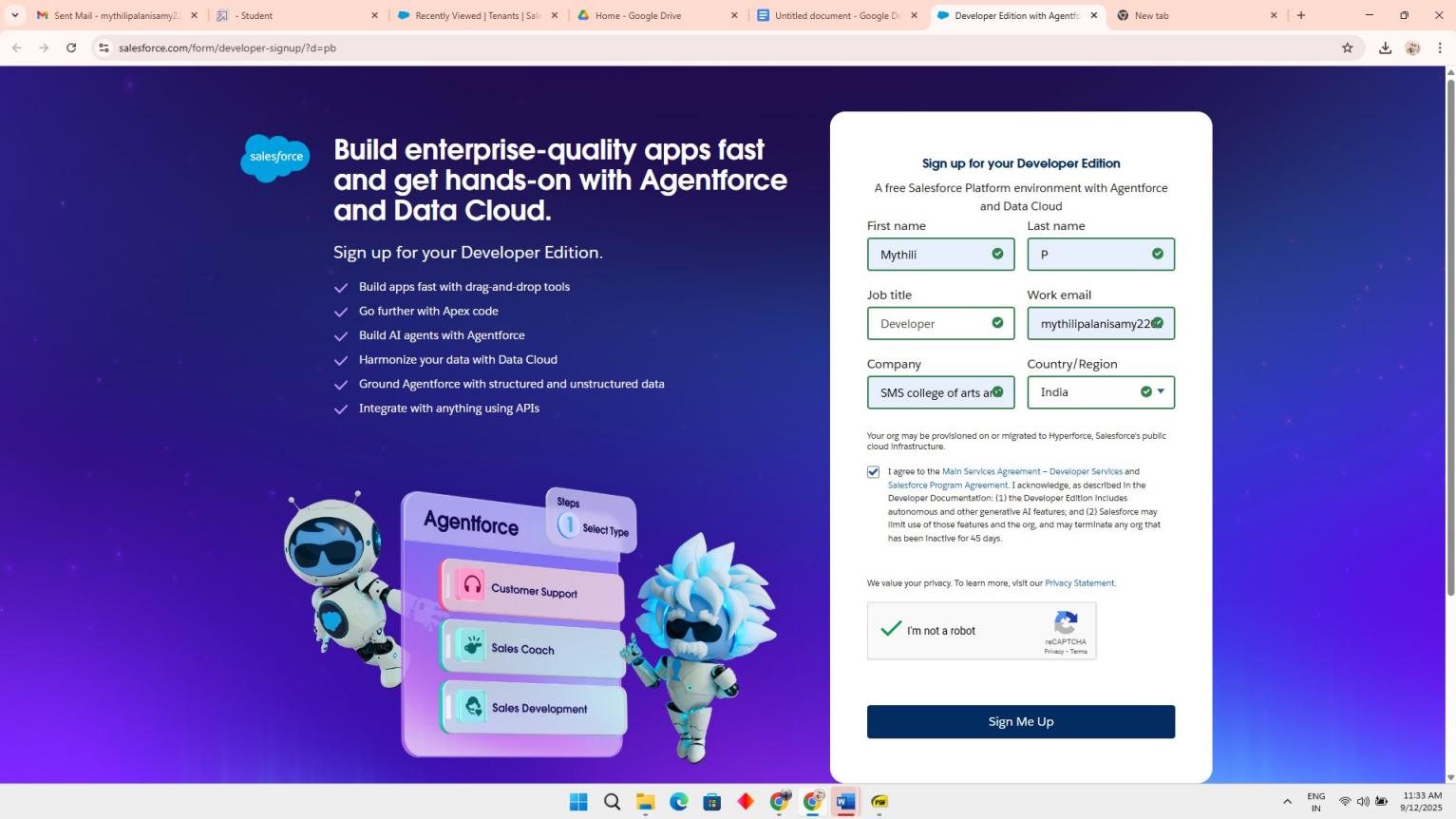
* 1. **Purpose**

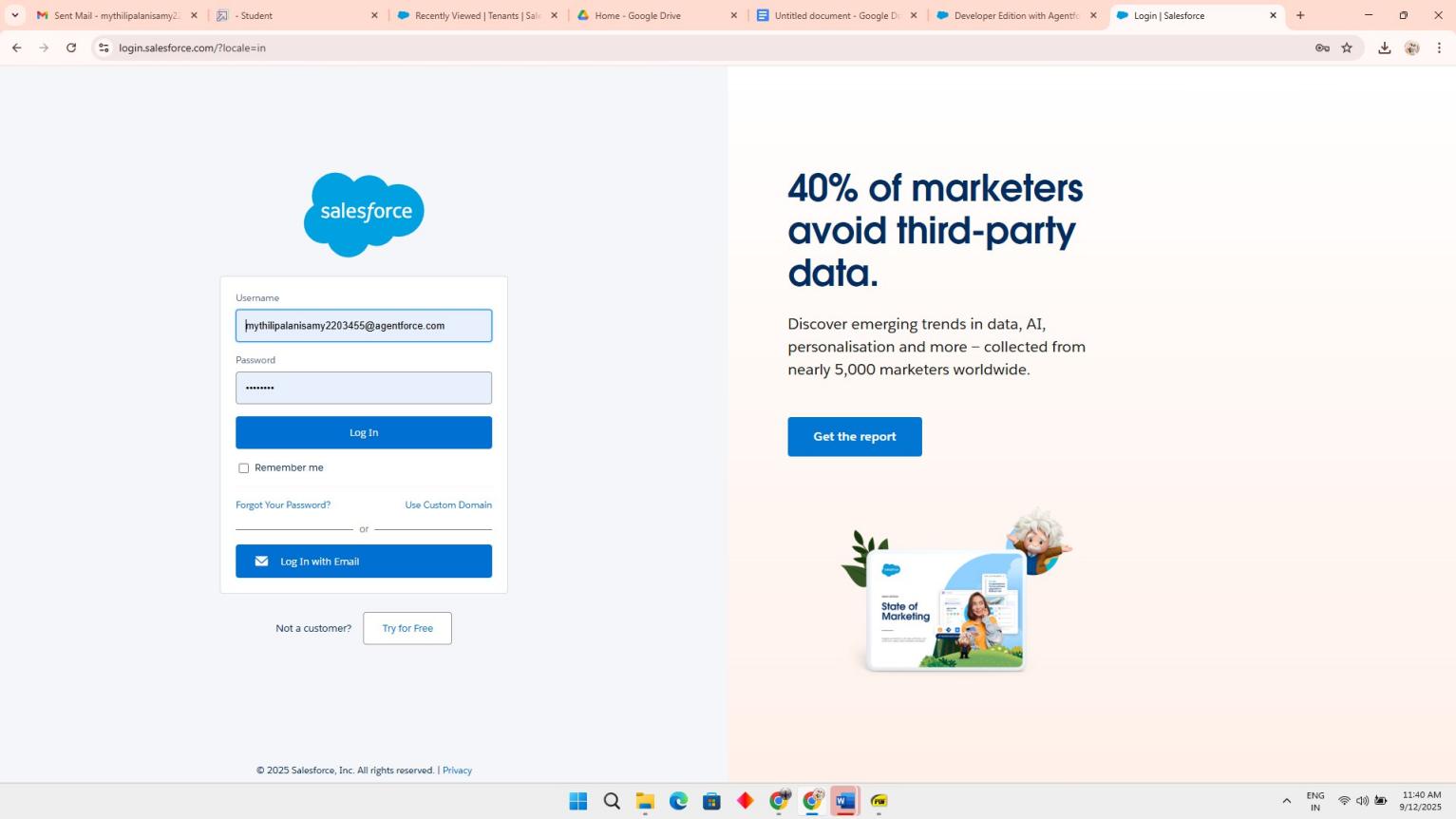
The purpose of the Lease Management System is to provide a centralized, automated solution for managing properties, tenants, and lease activities. By using Salesforce

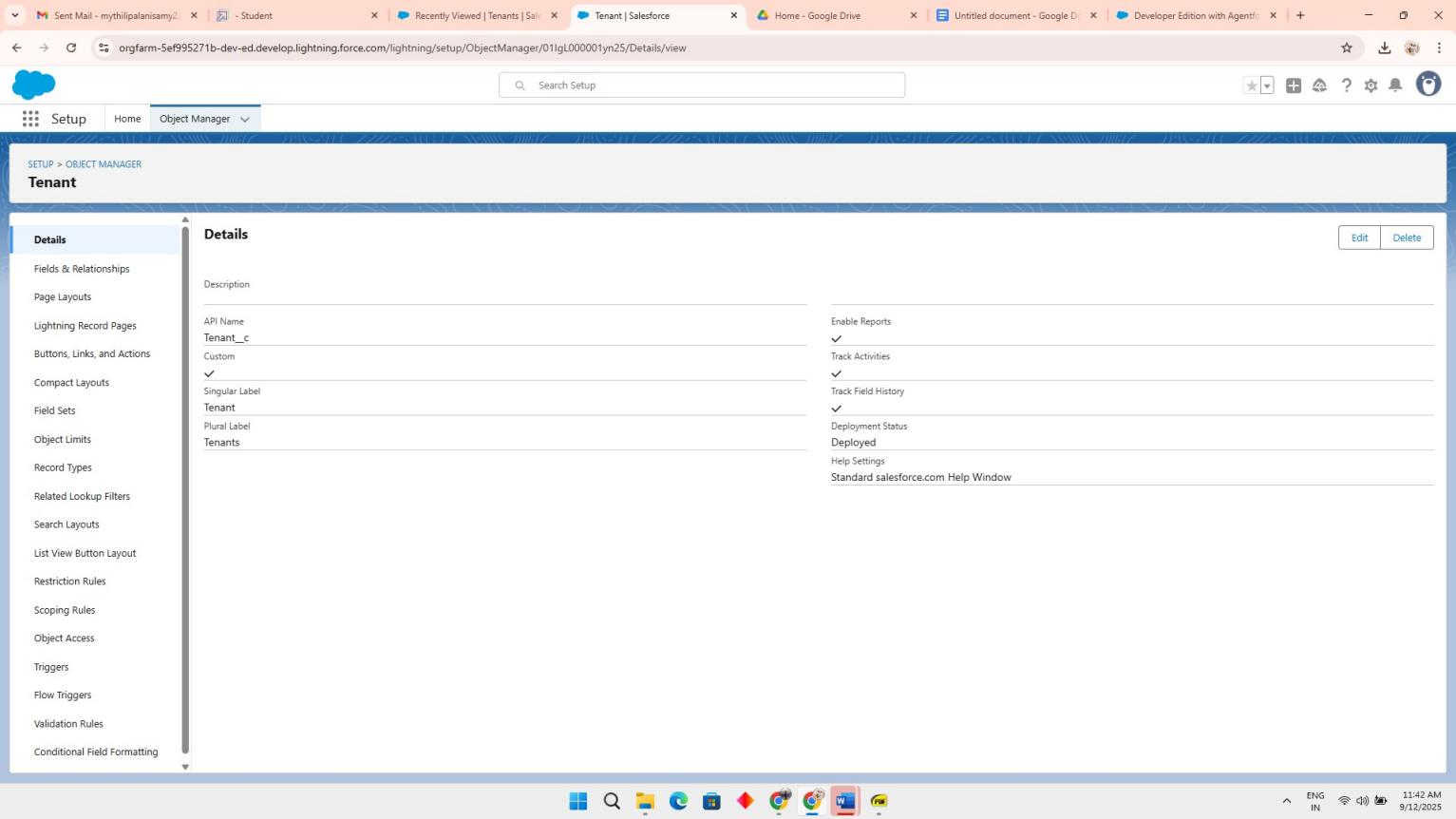
features such as Flows, Approval Processes, and Email Alerts, the system reduces manual effort, improves accuracy, and ensures timely communication. It enhances compliance through secure role-based access and strengthens tenant–landlord relationships with transparent, efficient processes, ultimately increasing productivity and supporting better decision-making.

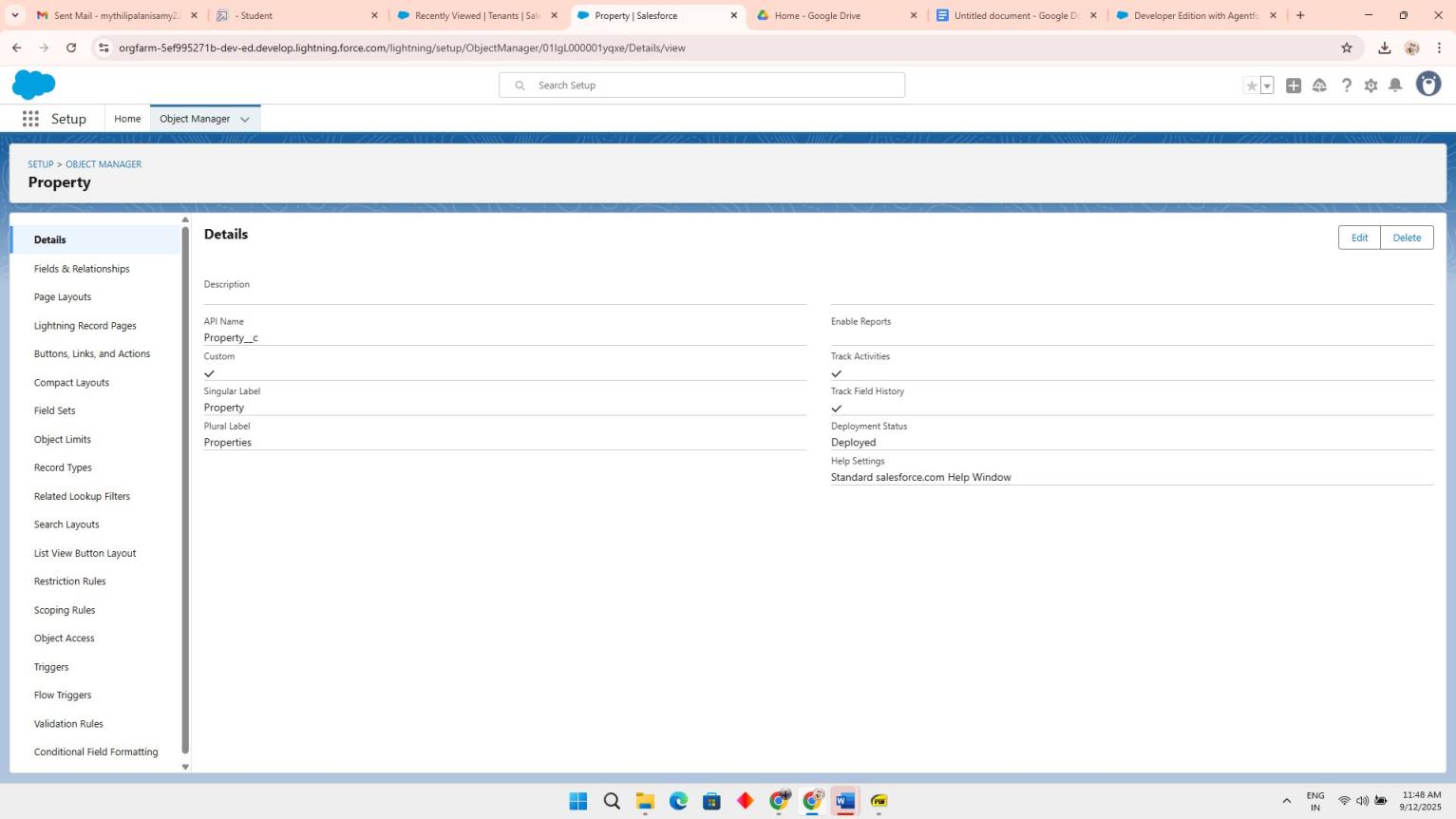
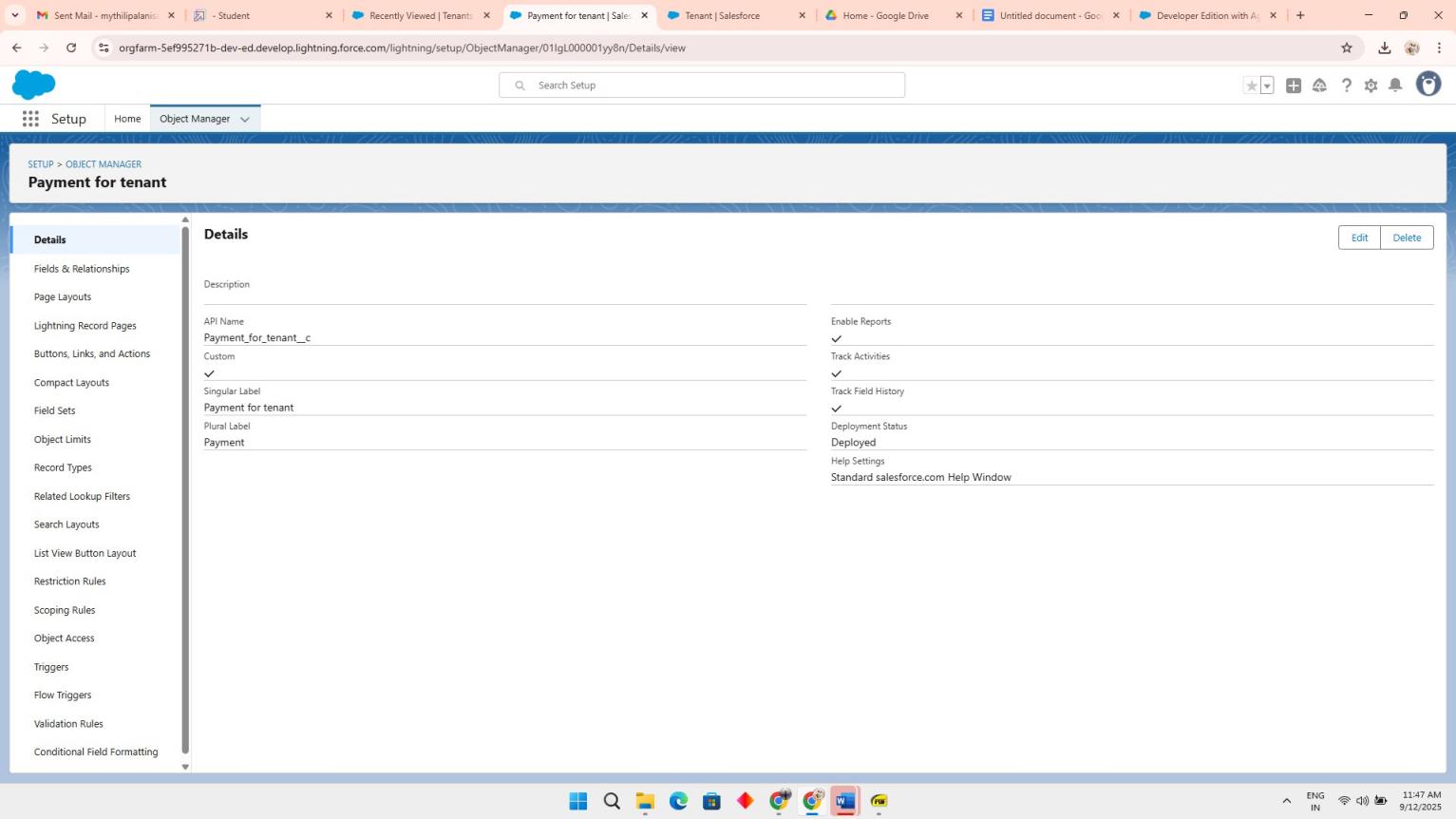
**DEVELOPMENT PHASE**

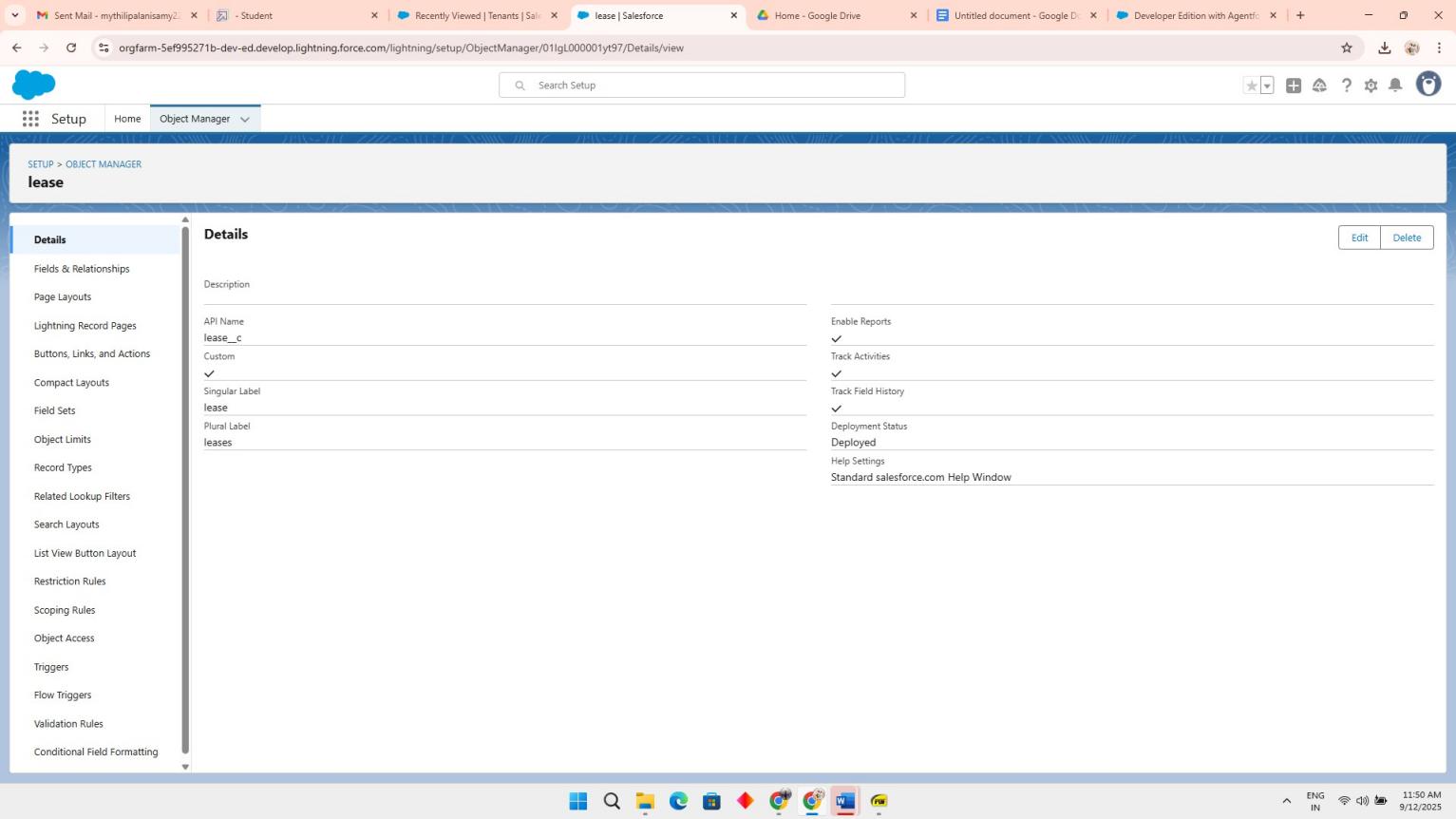
* + - **Creating Developer Account:**

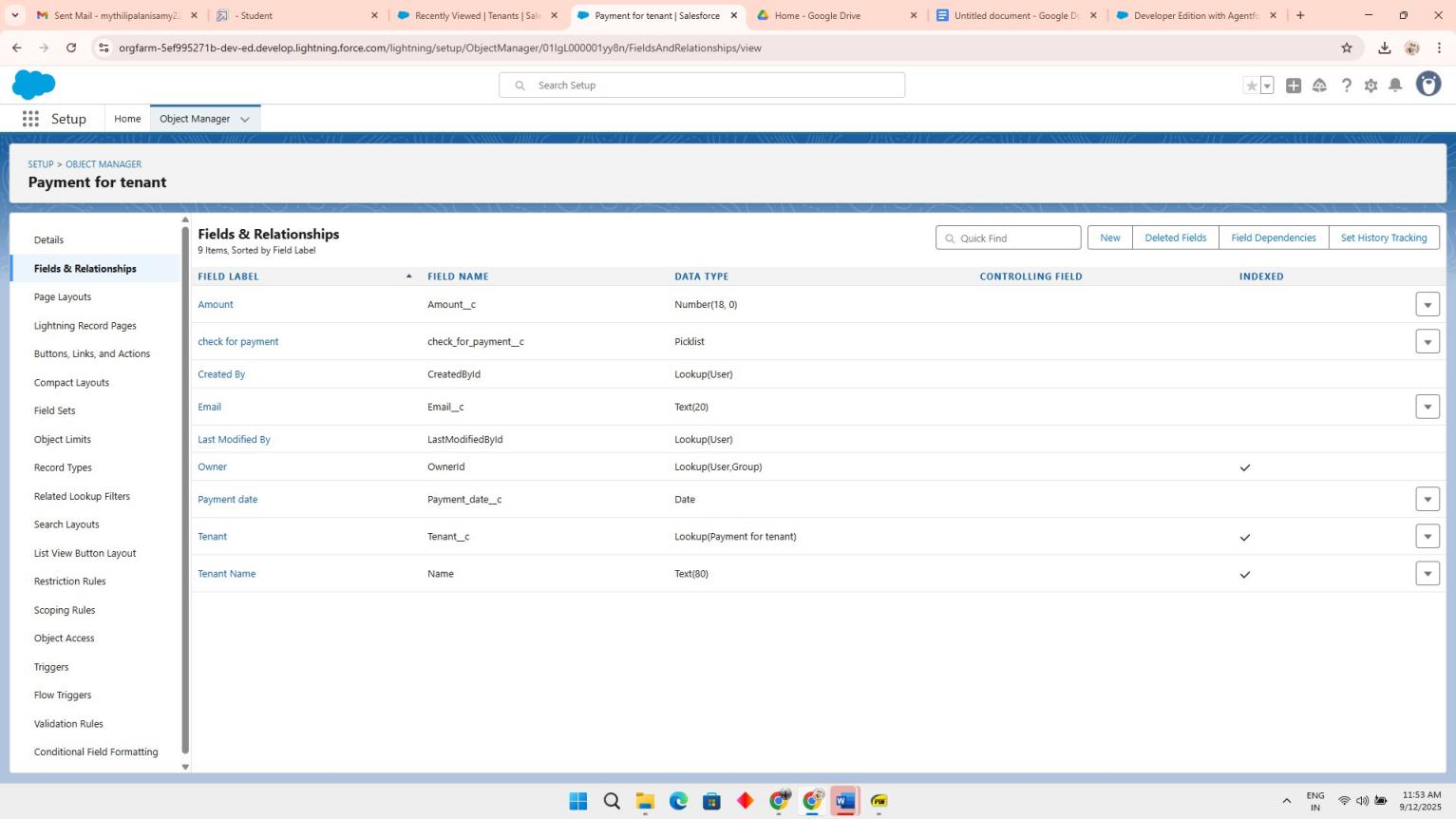
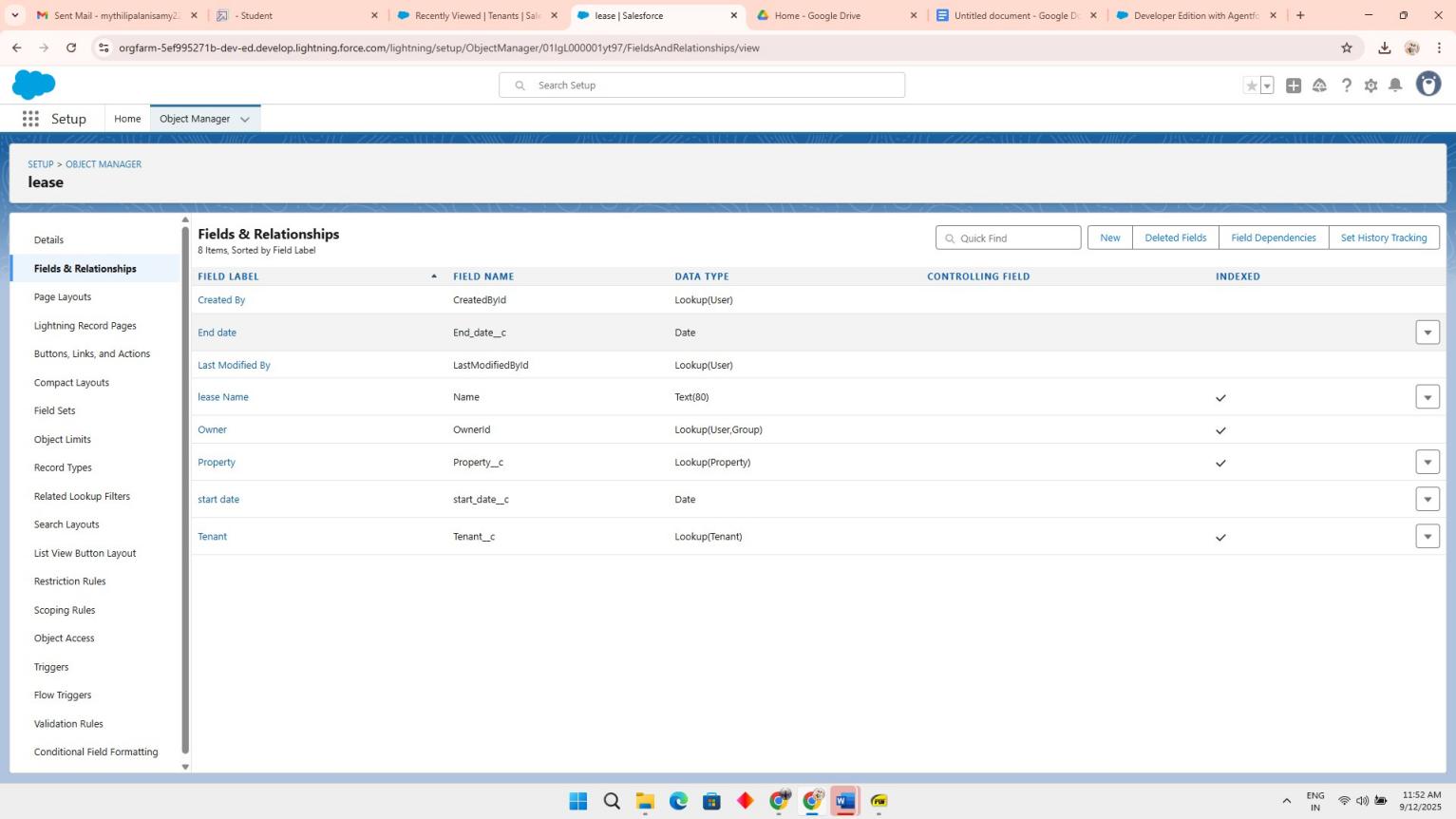
By using this URL: <https://www.salesforce.com/form/developer-signup/?d=pb>

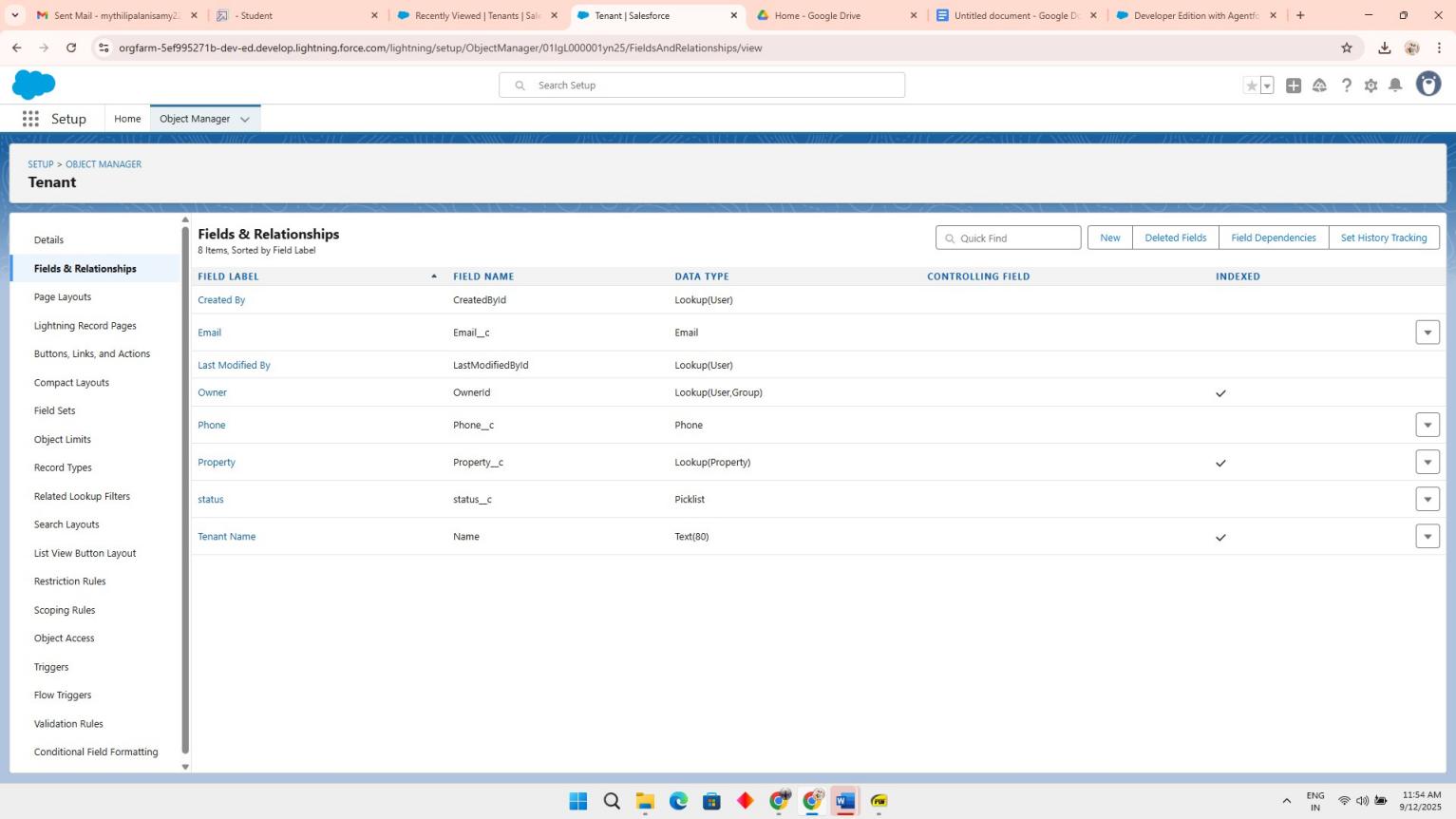
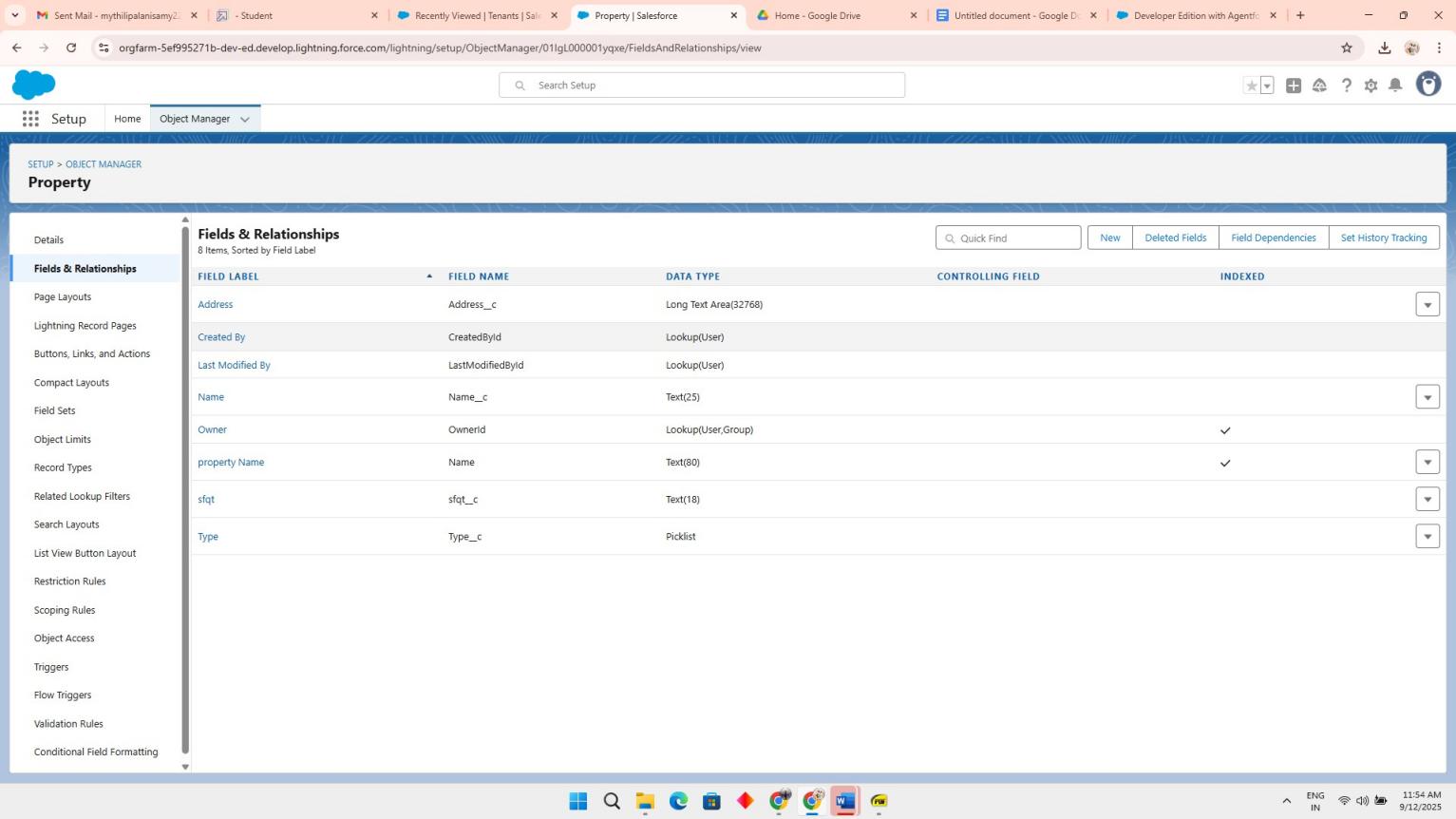
Account activation then next login this link <https://login.salesforce.com/>

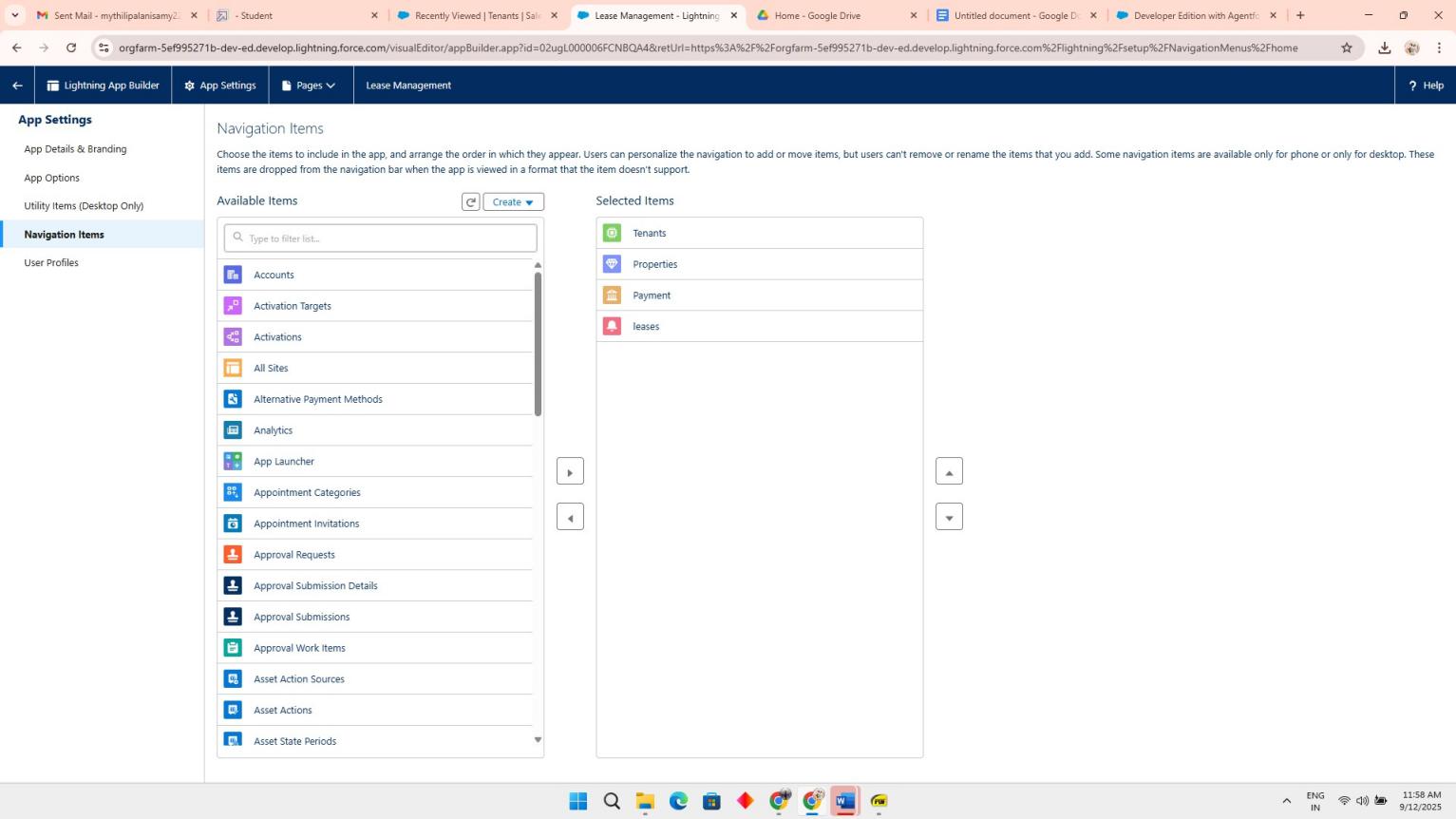
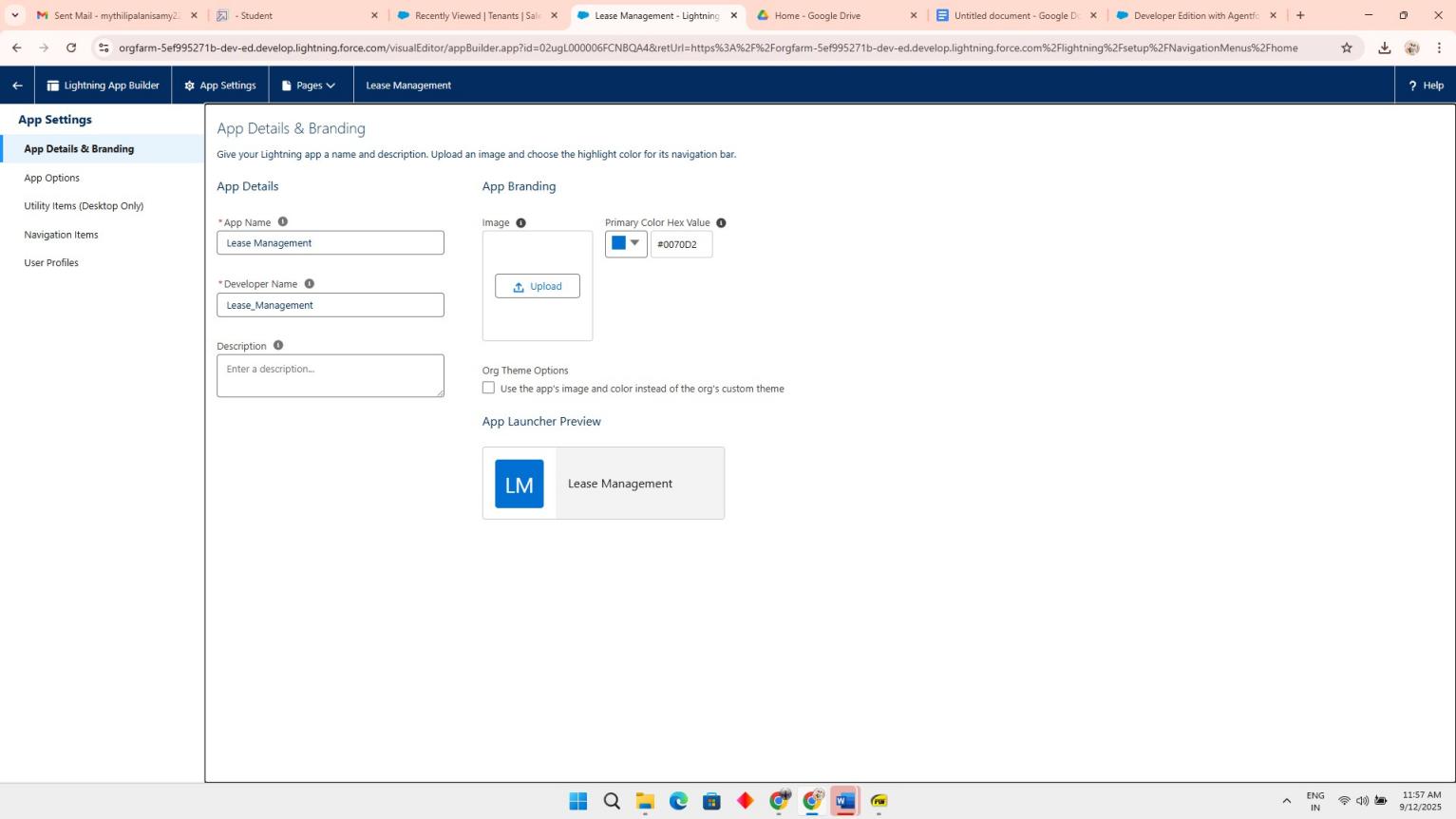
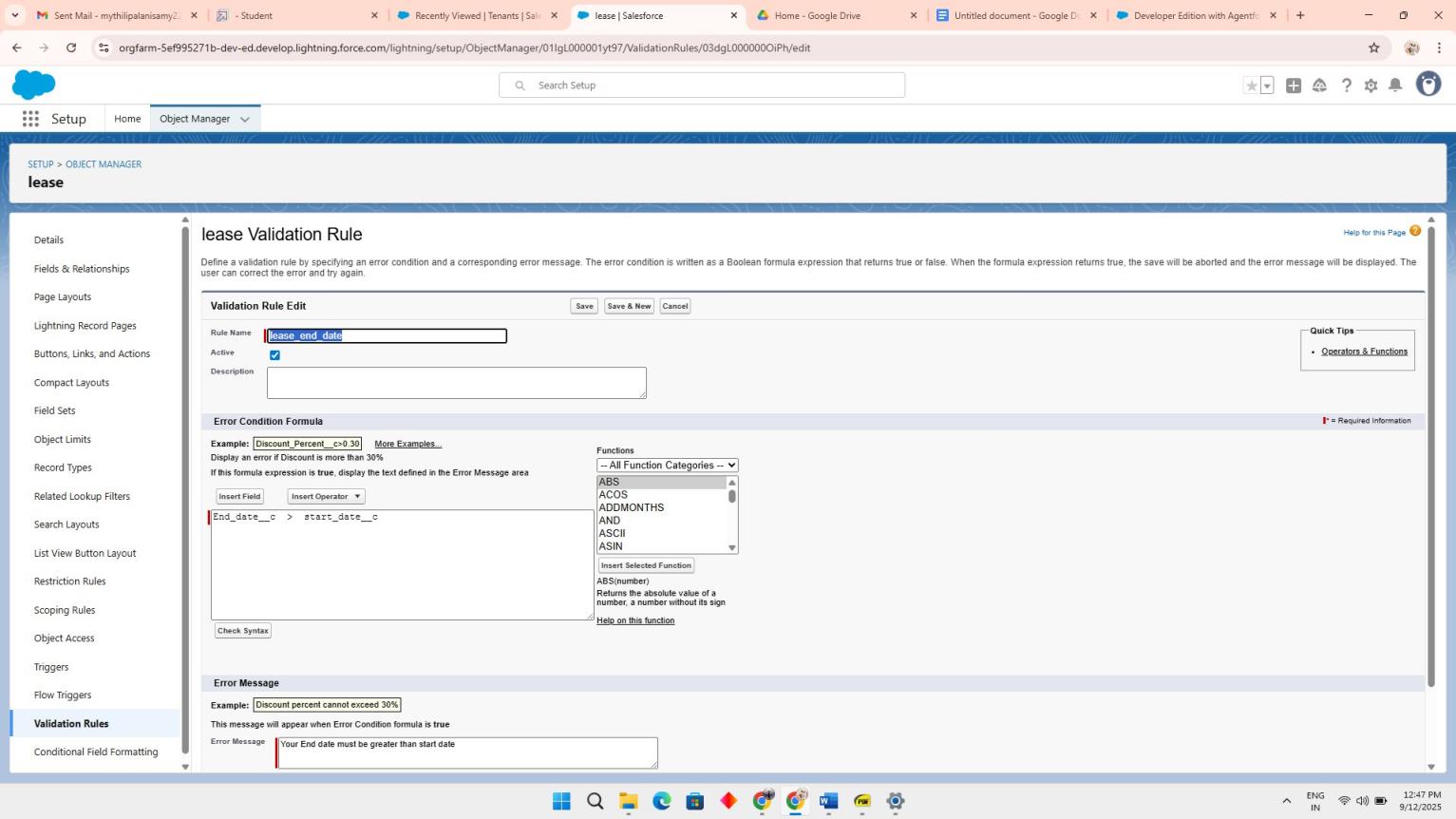
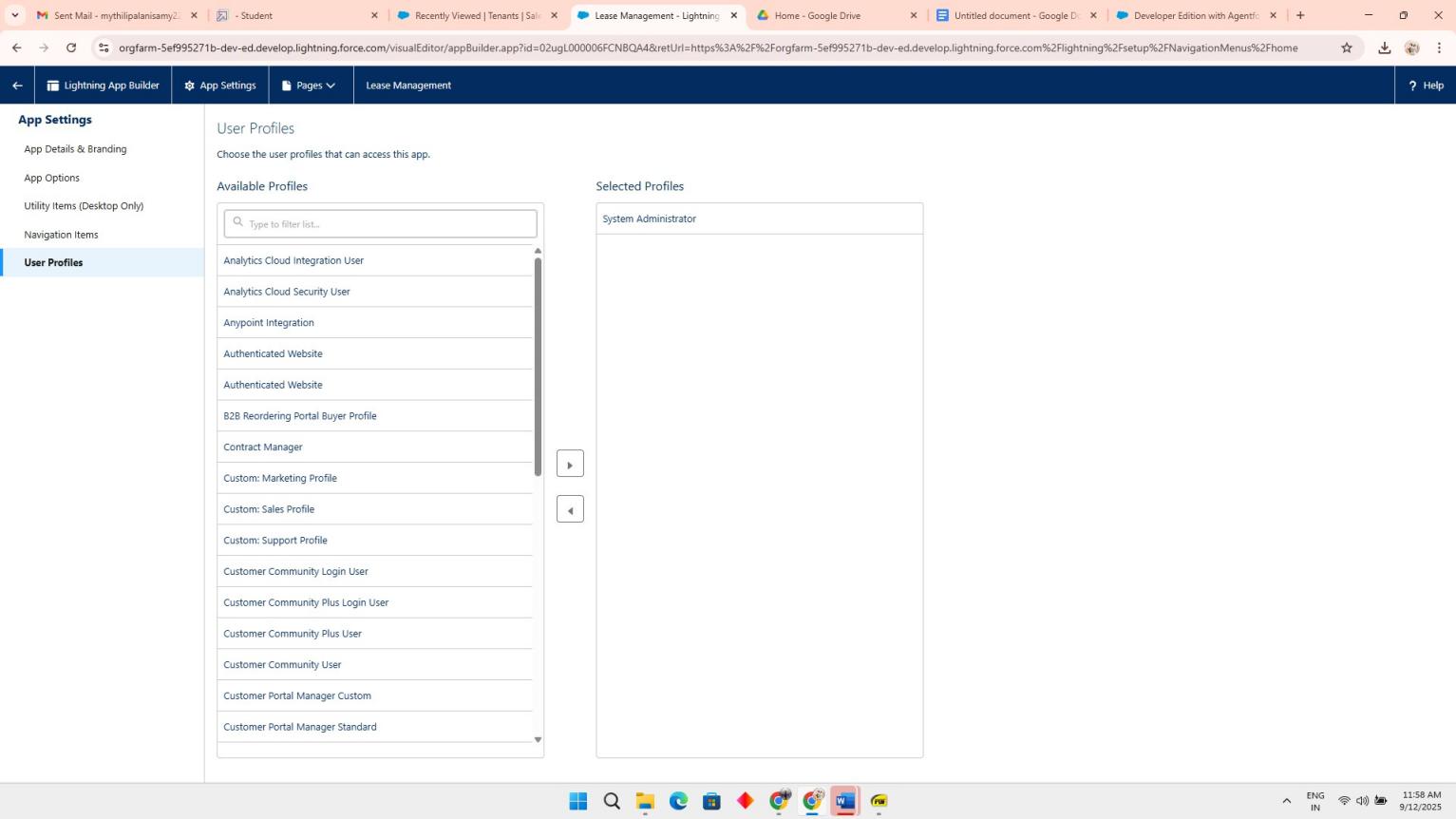
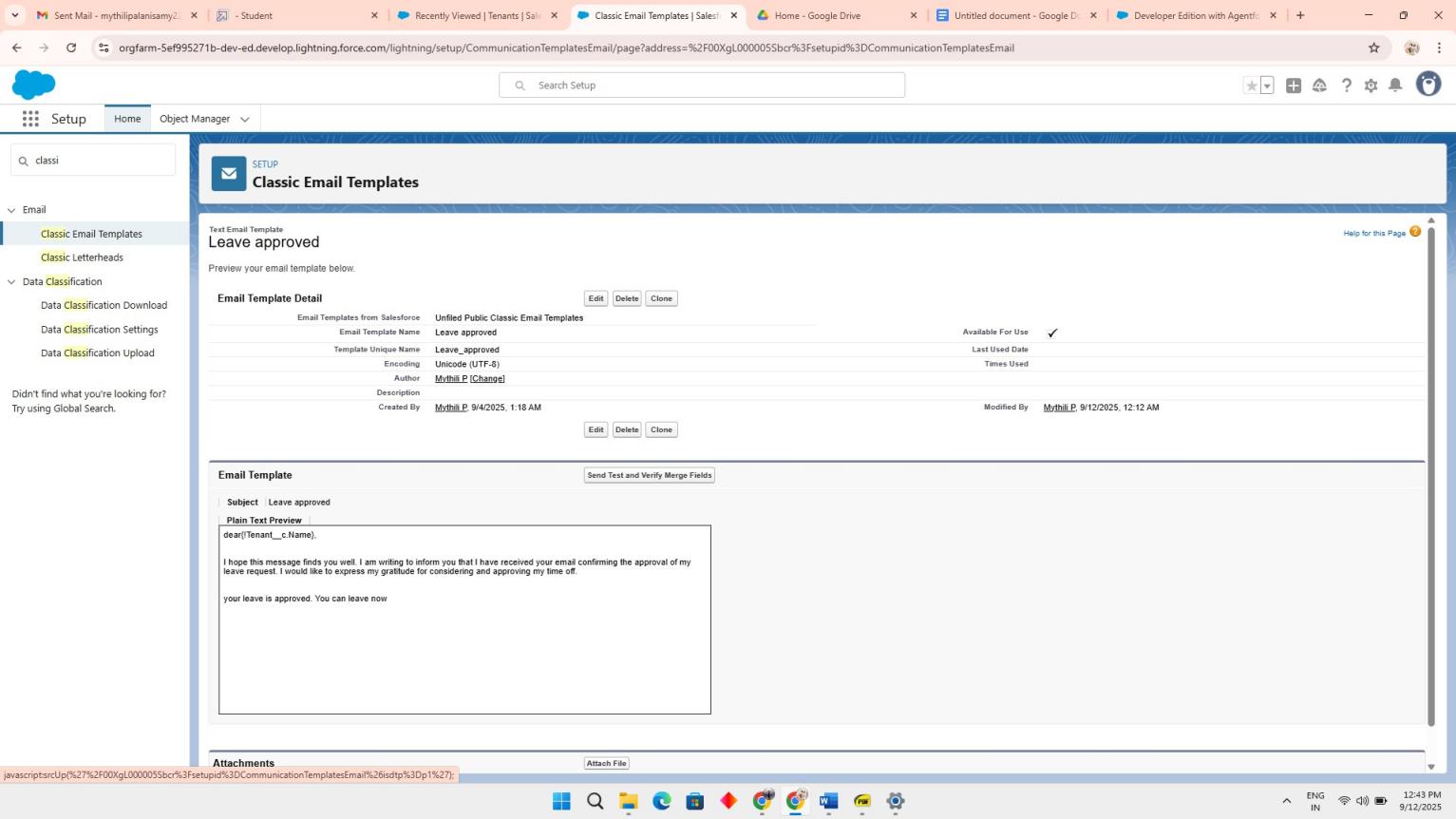
* + - * Created objects: Property, Tenant, Lease, Payment

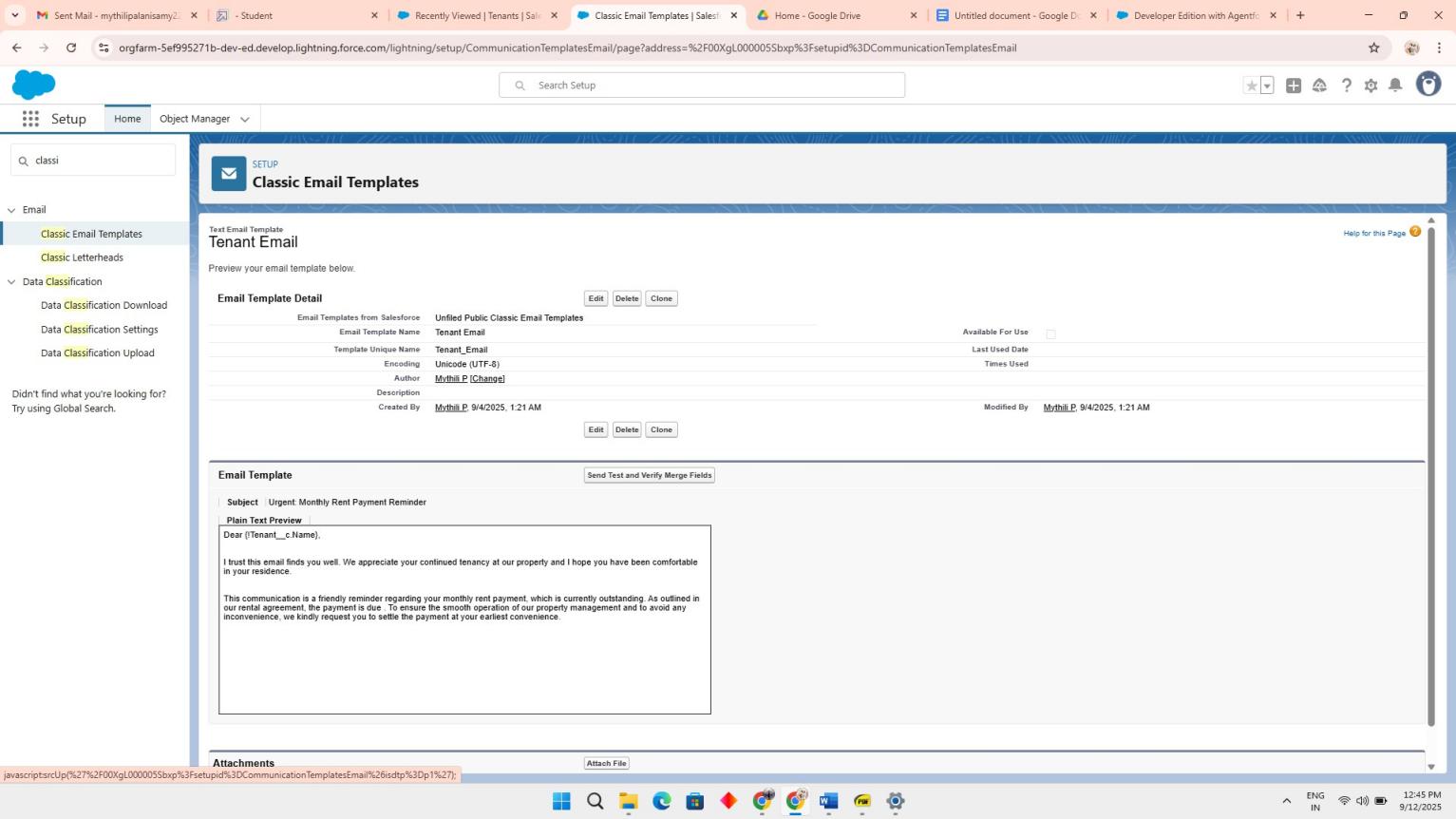
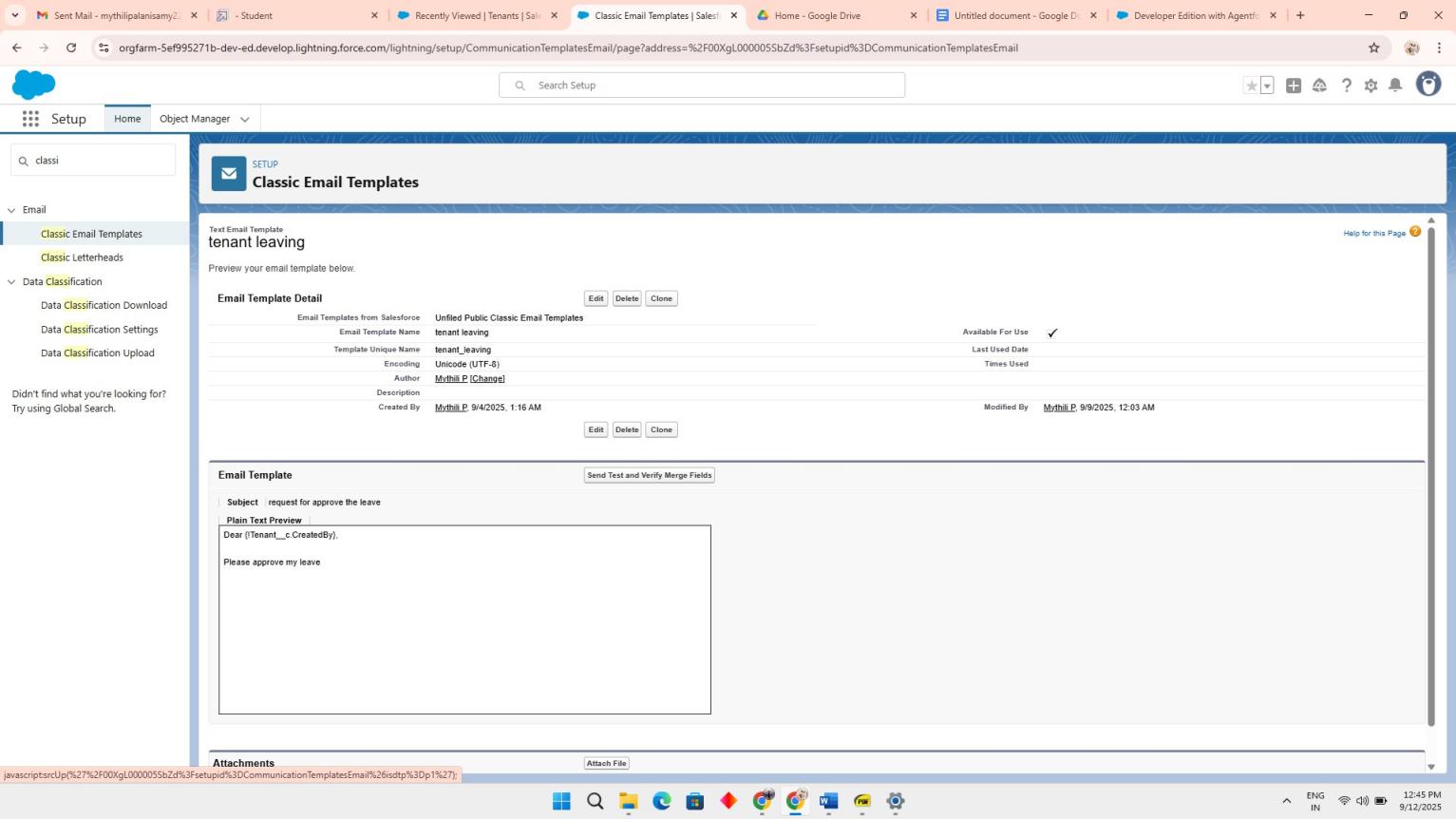


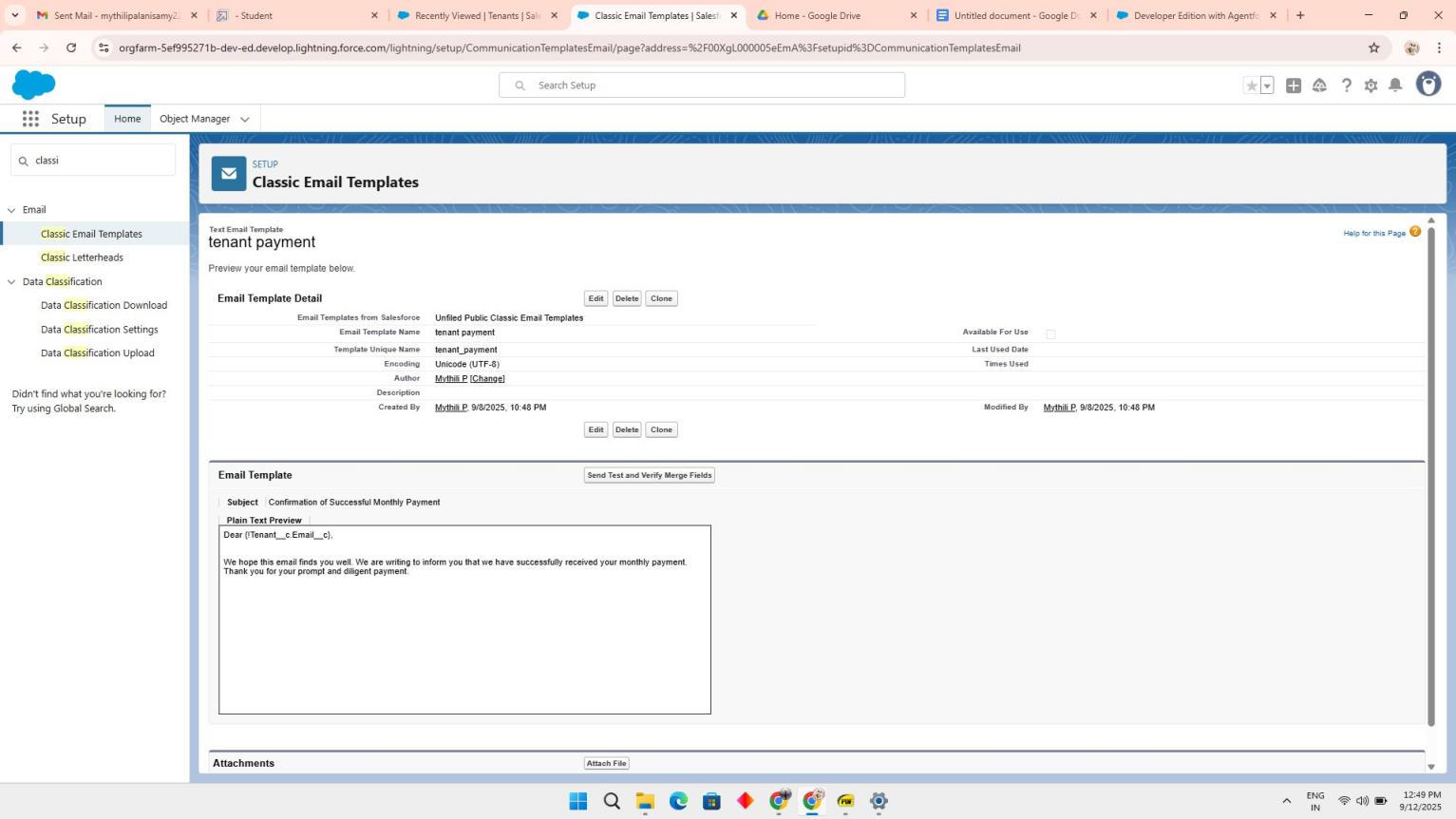


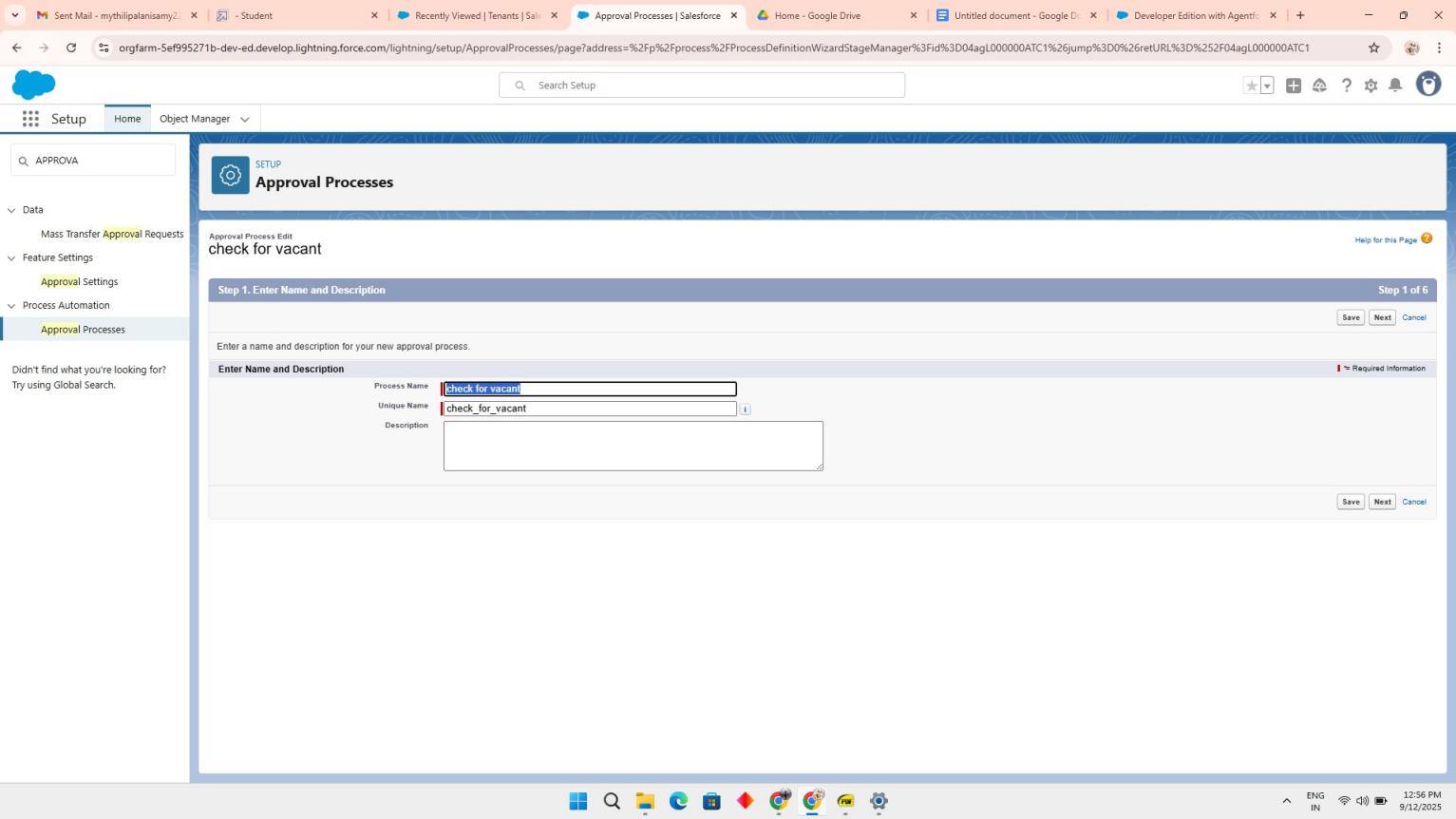
* + - Configured fields and relationships

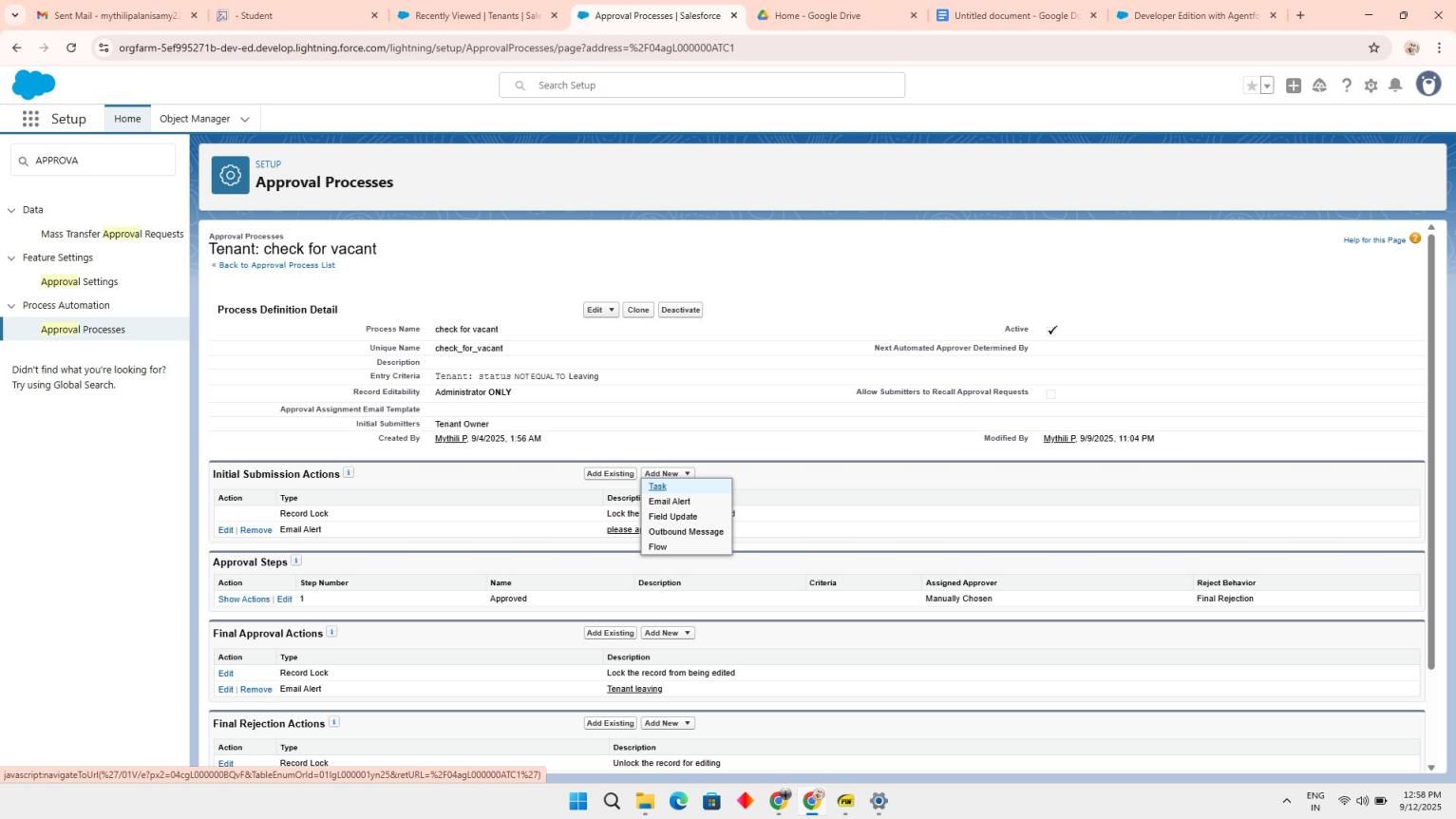
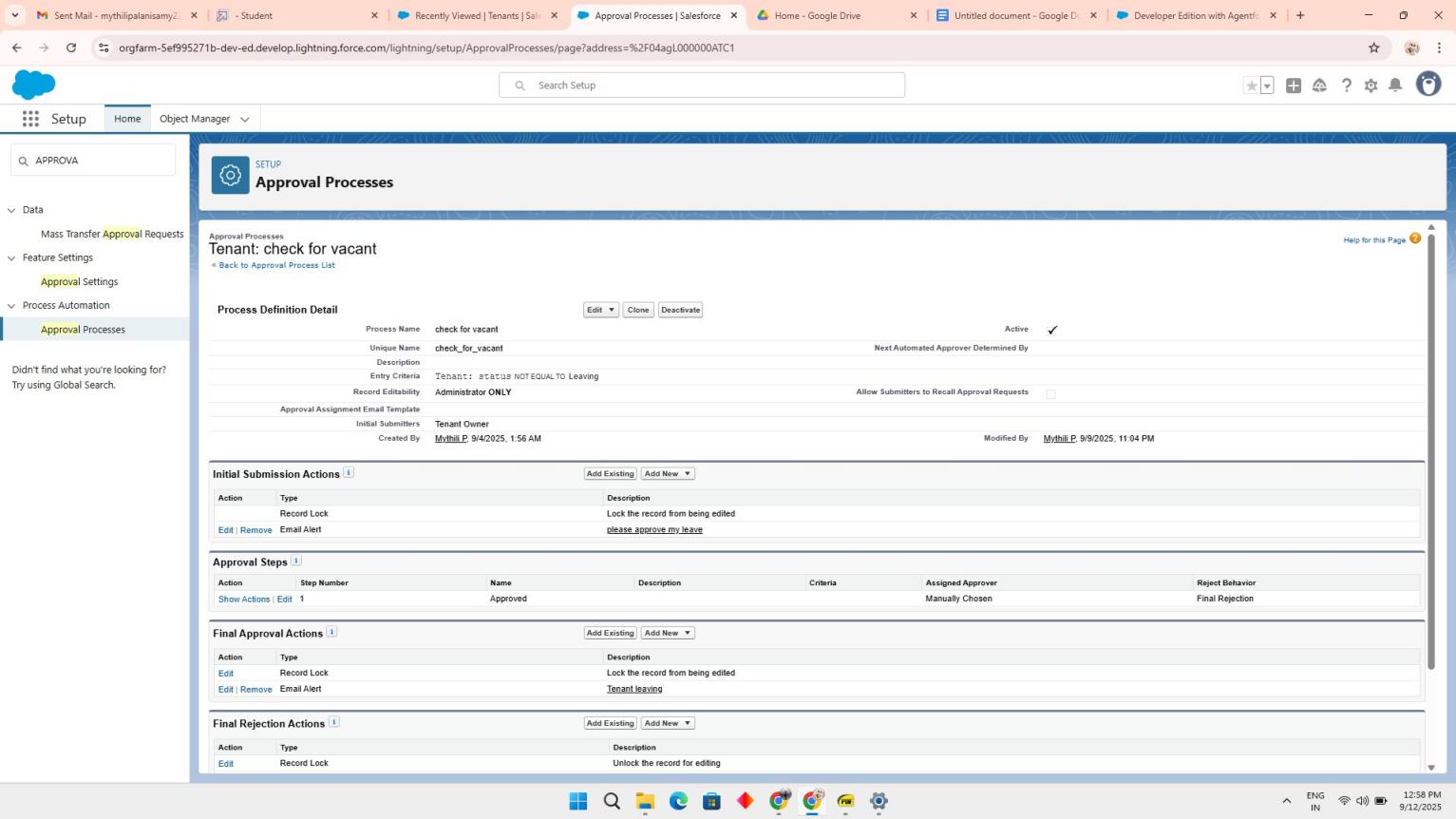


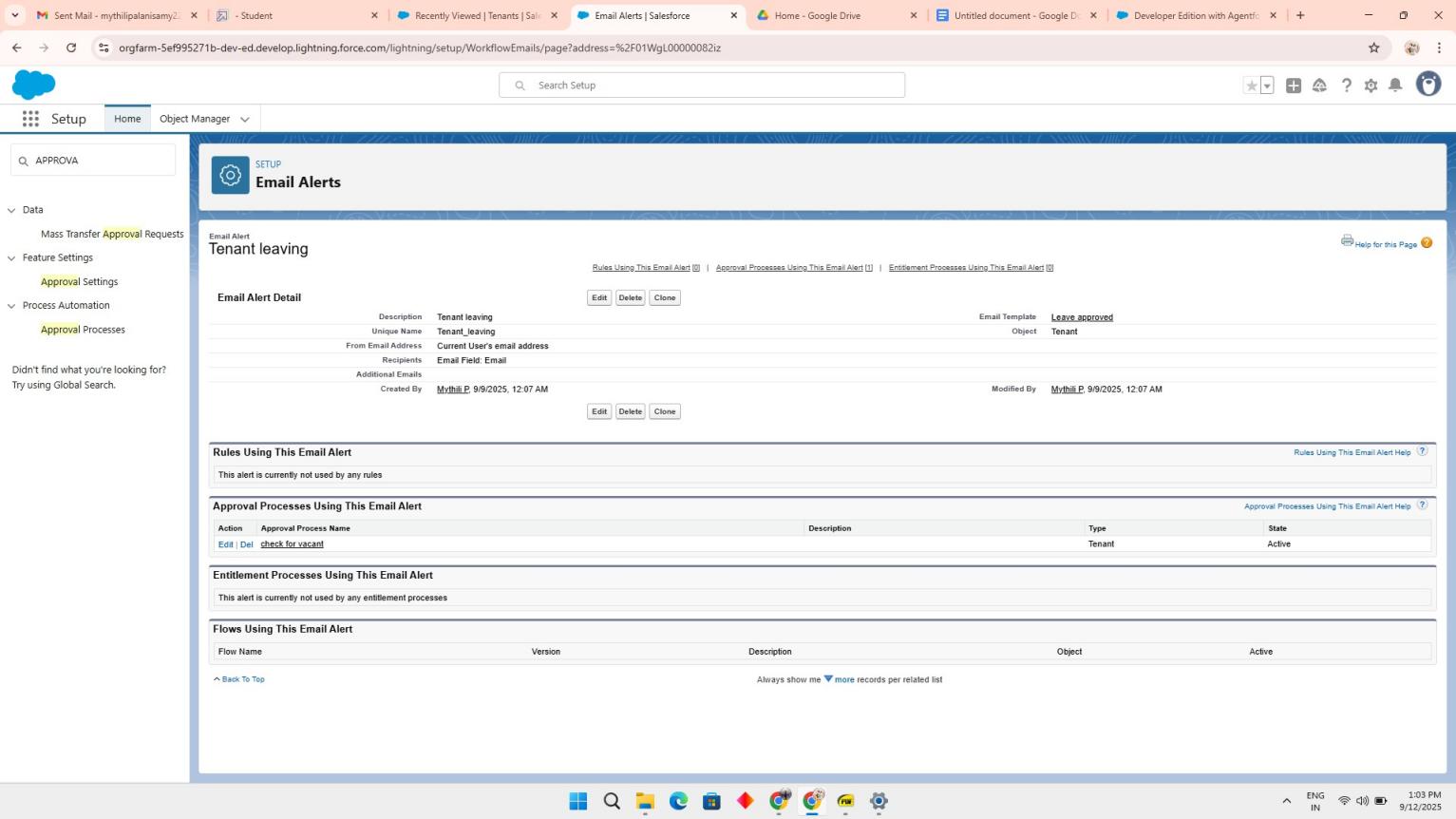
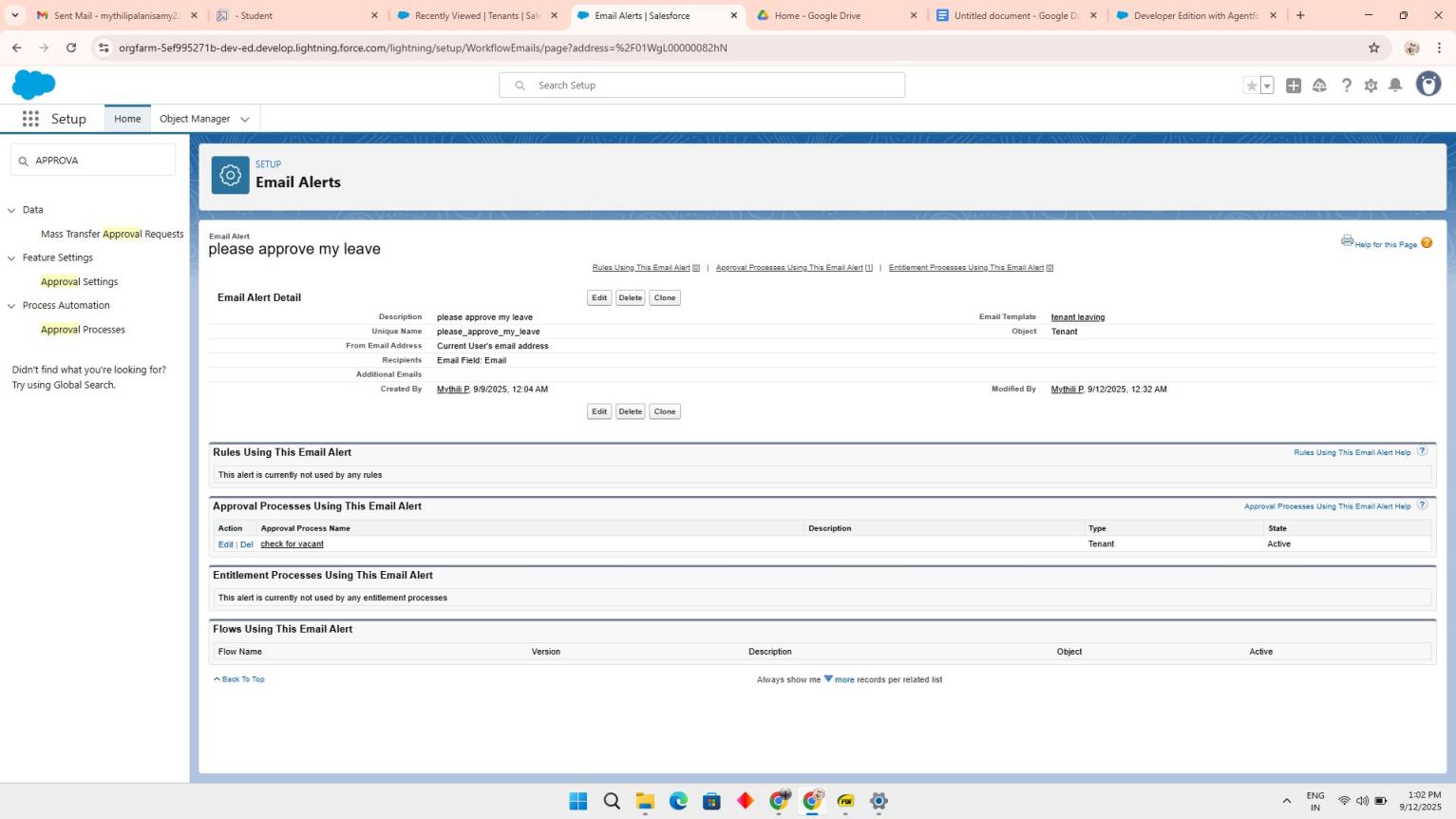
* Developed Lightning App with relevant tabs
* create a validation rule to an Lease Object
  + Built and tested email templates for leave request, approval, rejection, payment, and Reminders

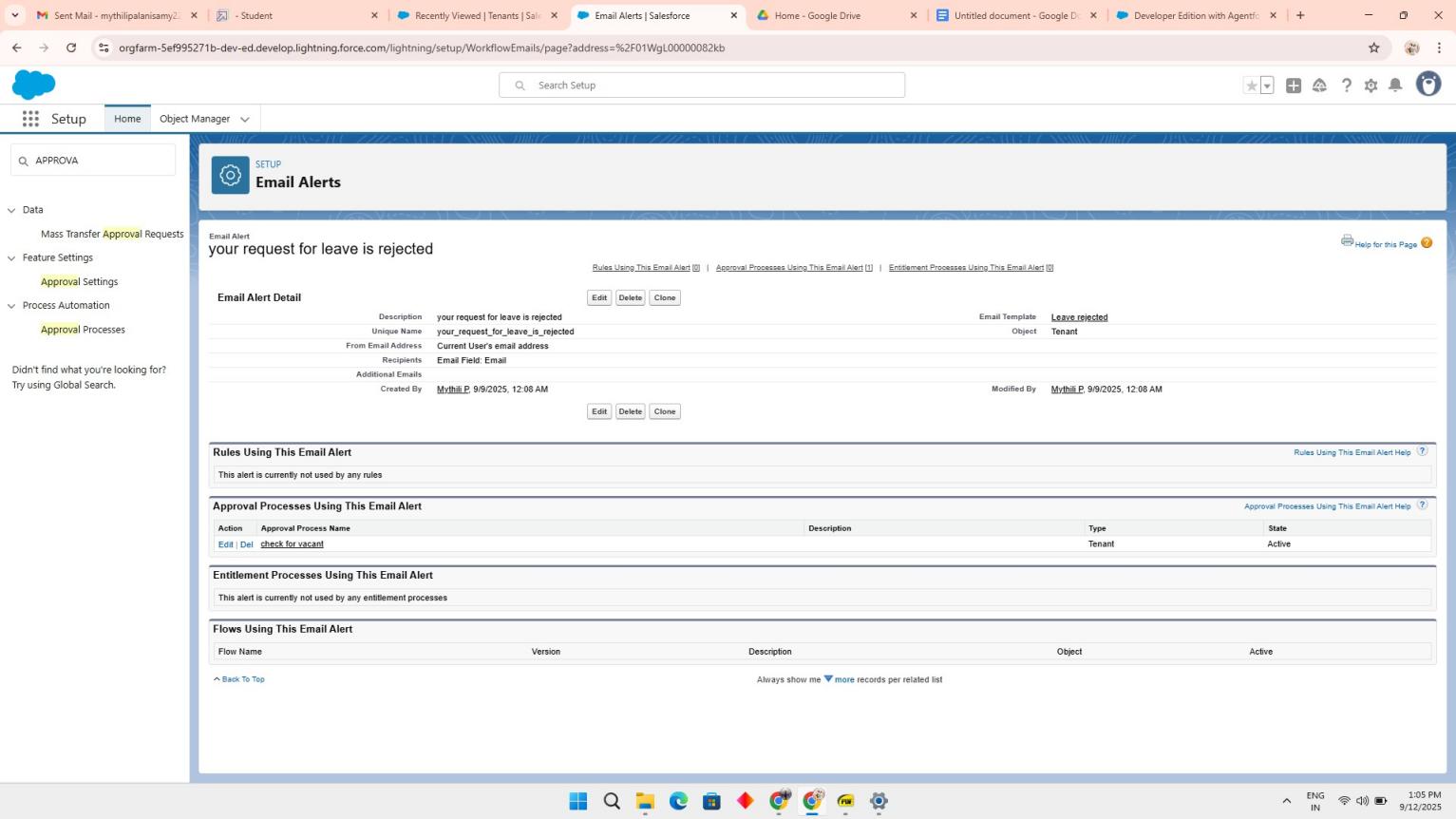


* **Approval Process creation**

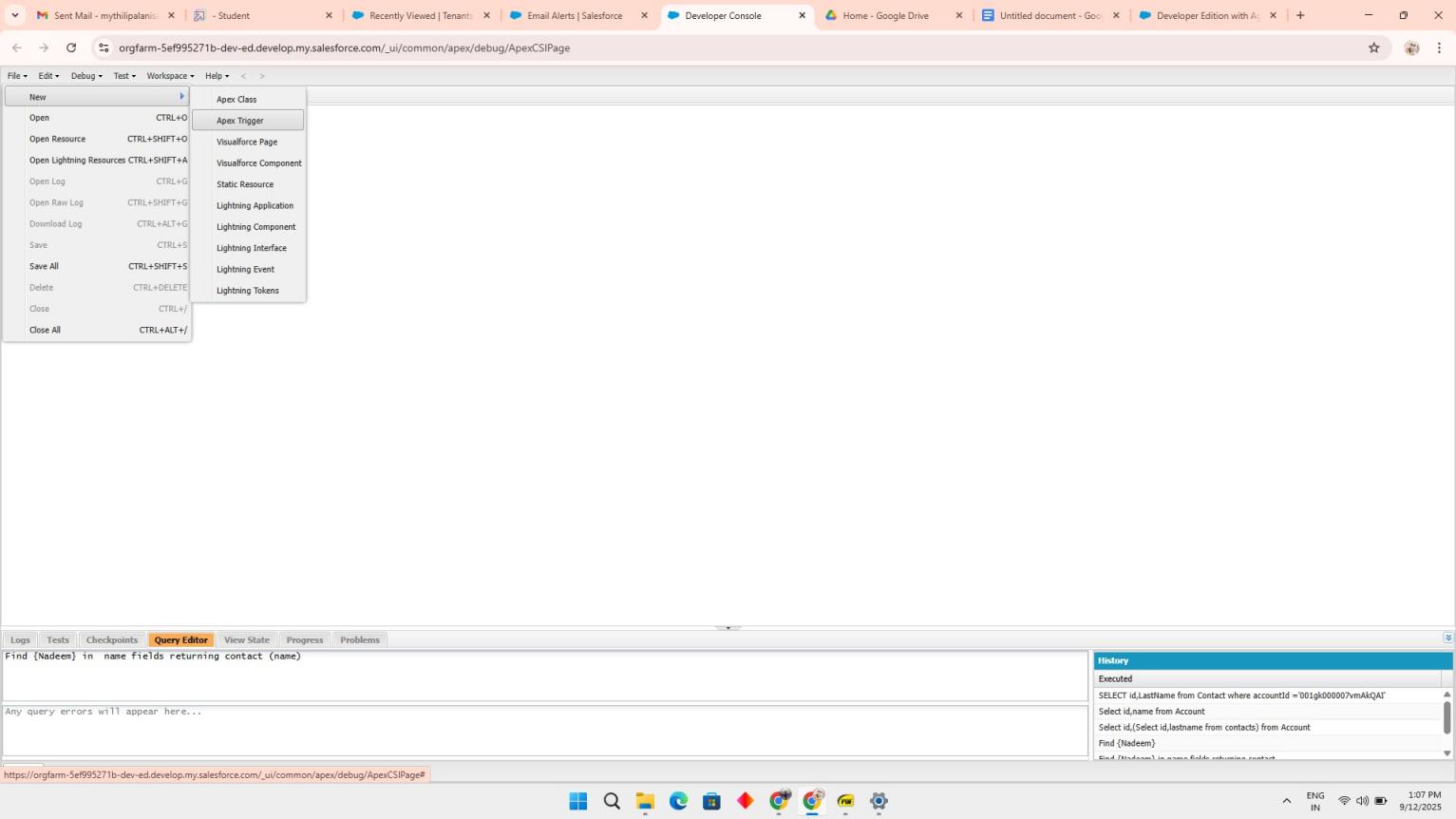
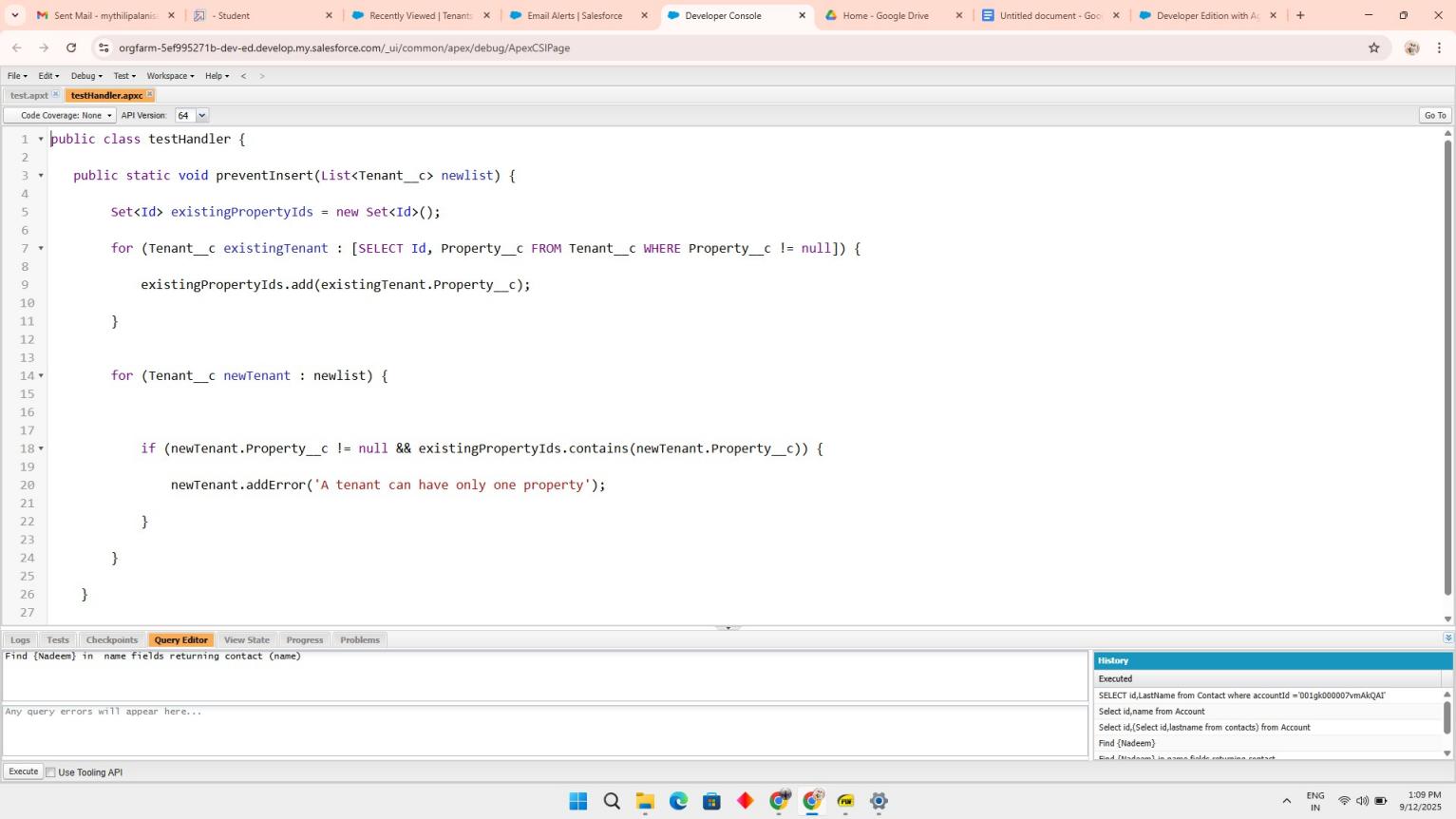
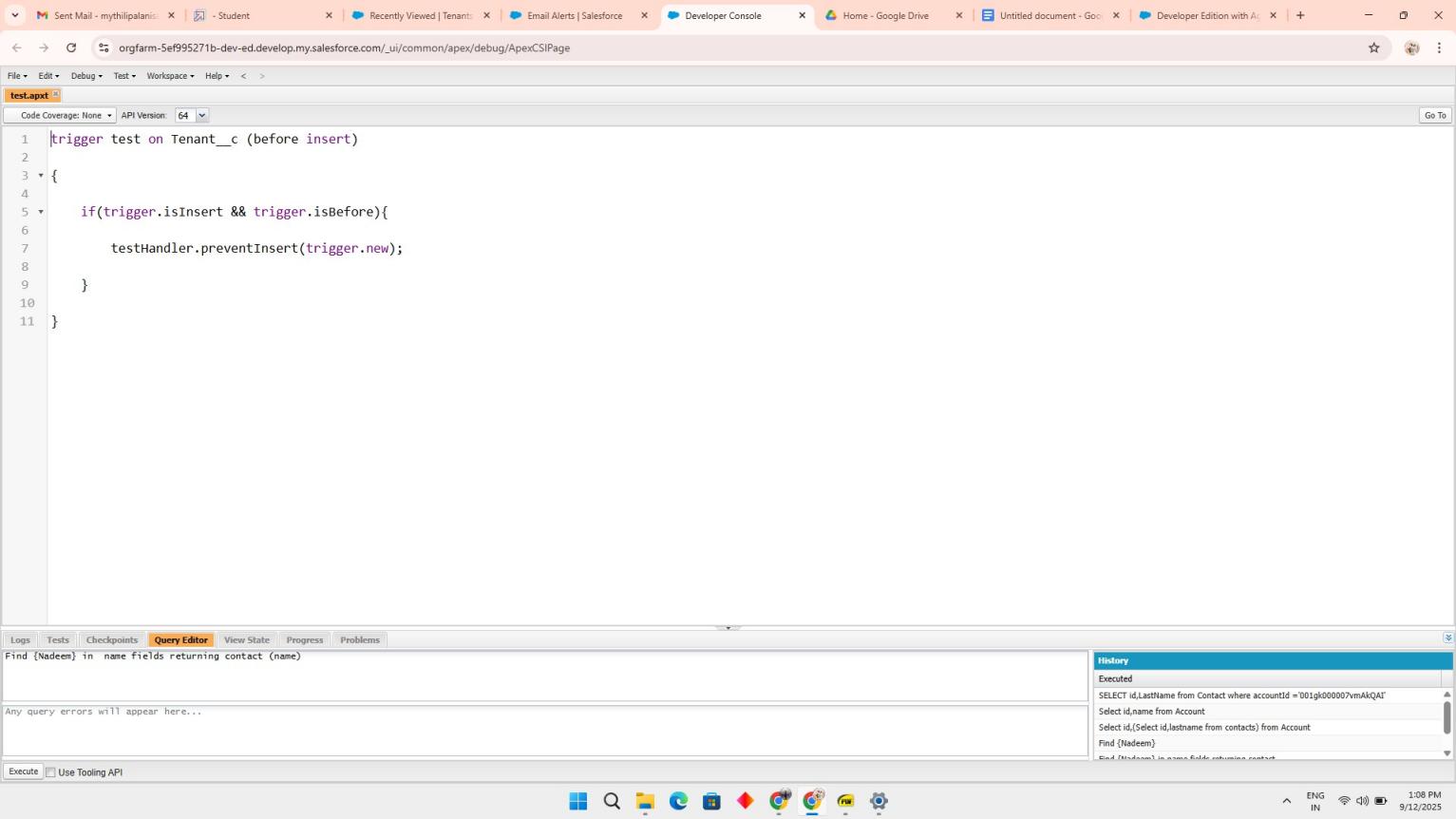
For Check for Vacant :

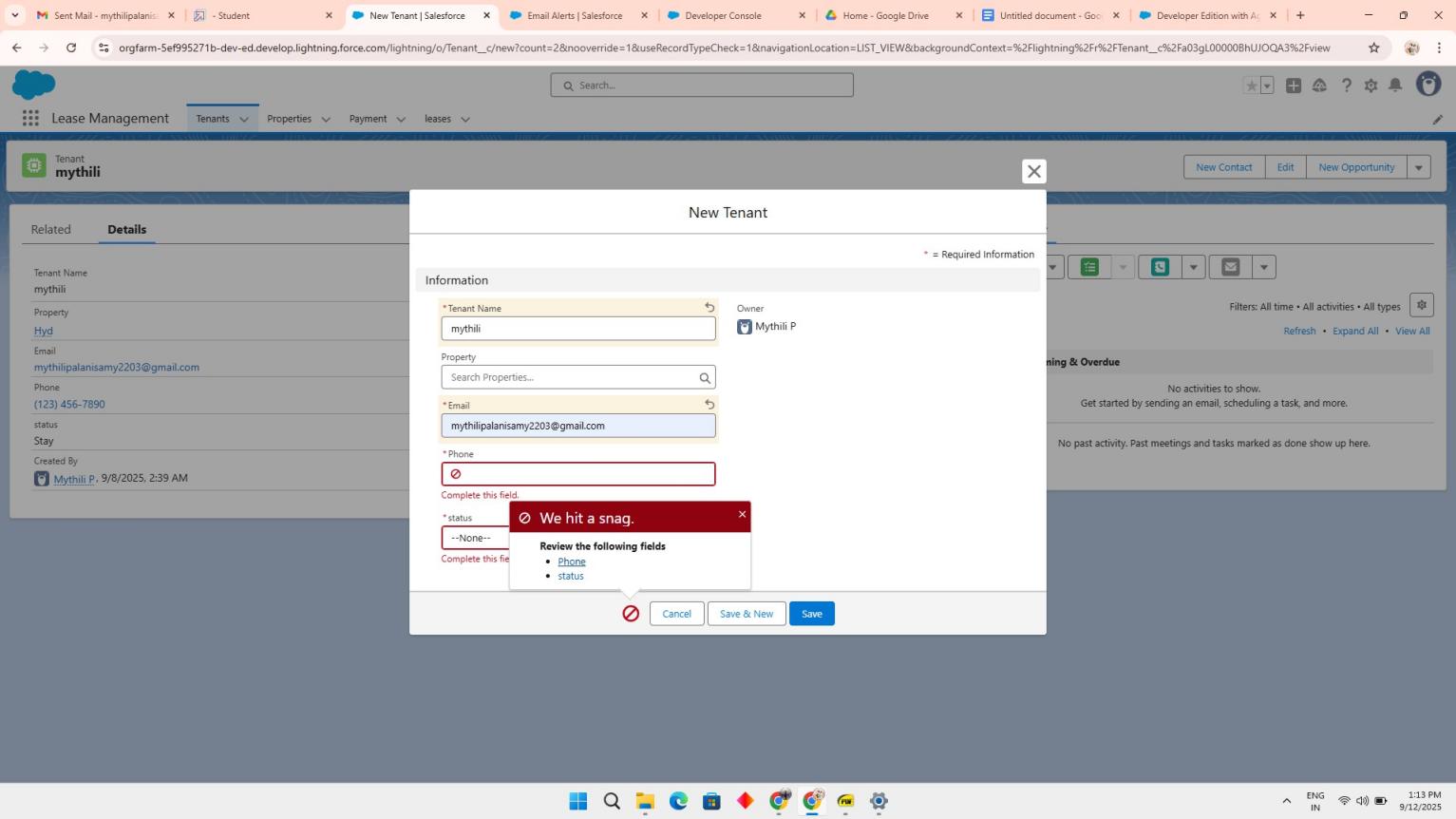


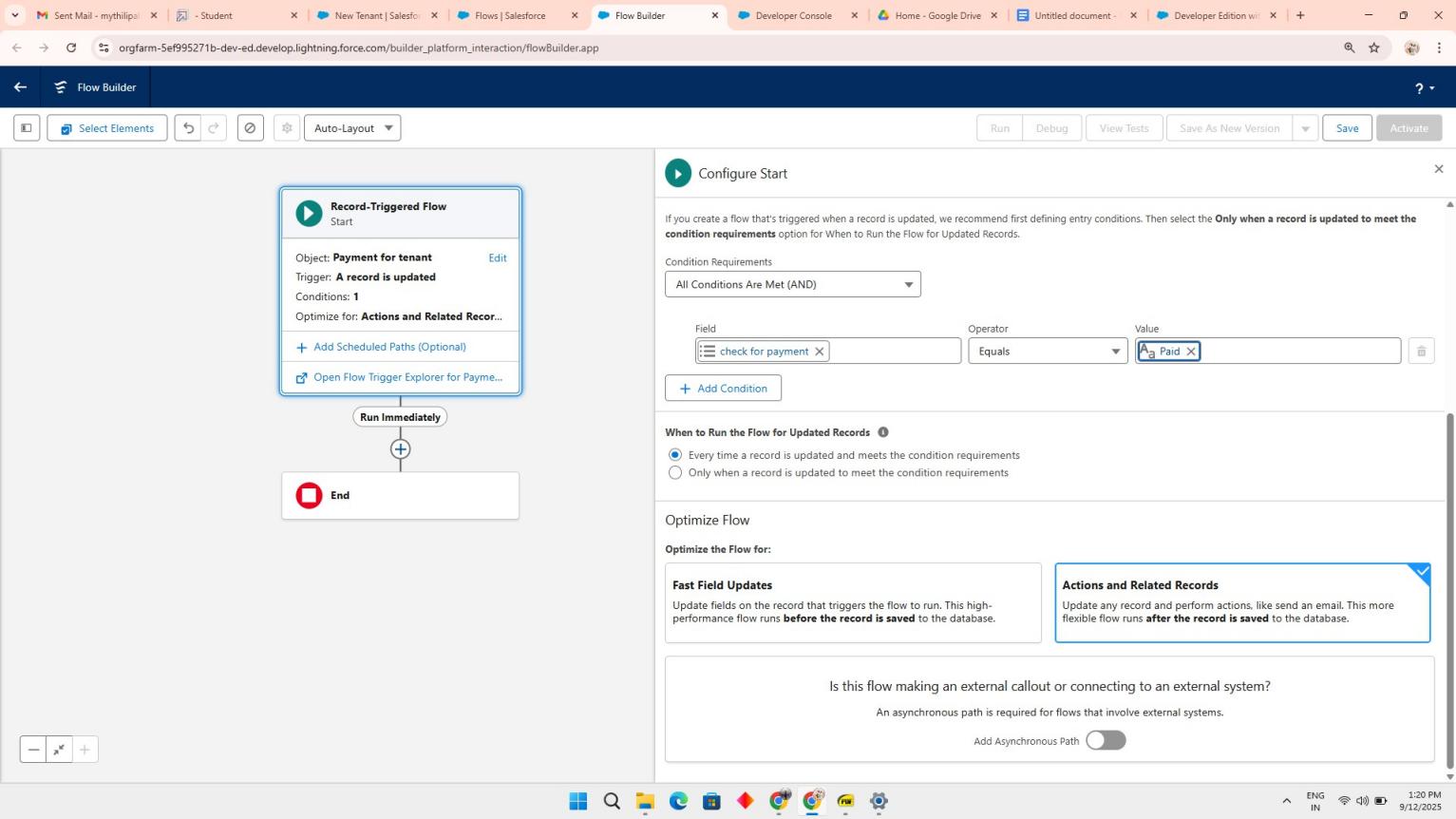
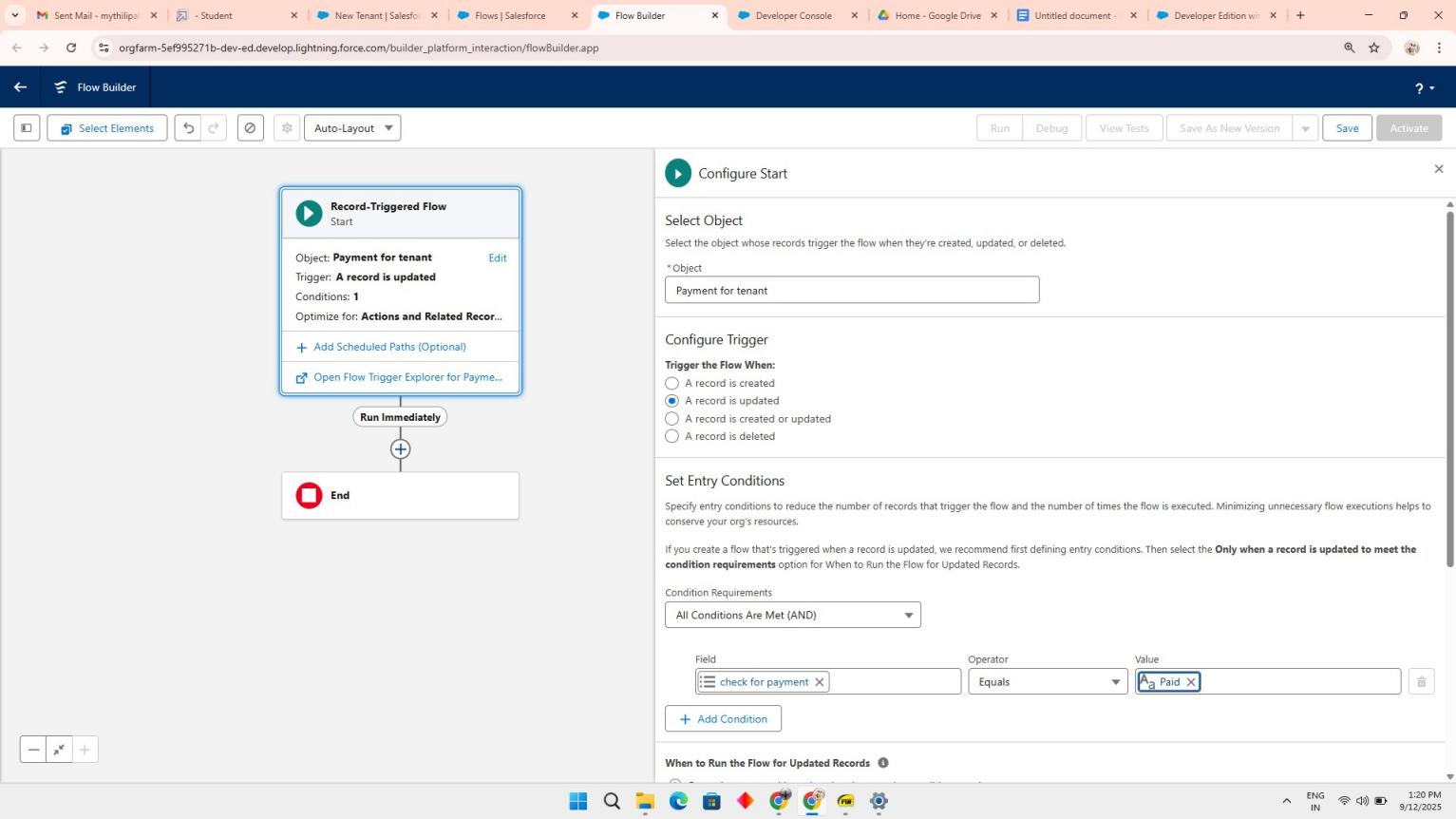
* + Final Approval Action
    - Final Rejection Action

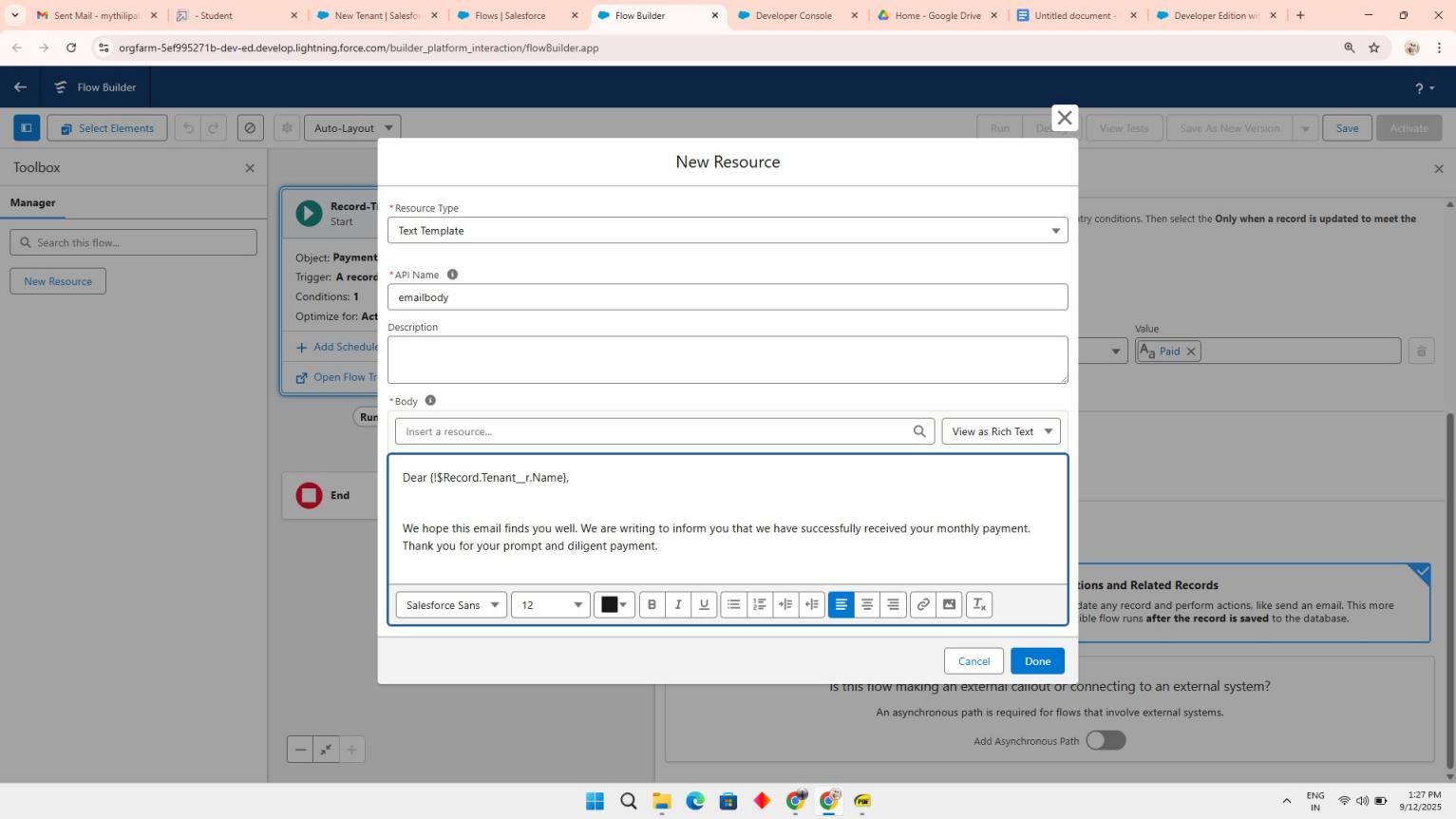
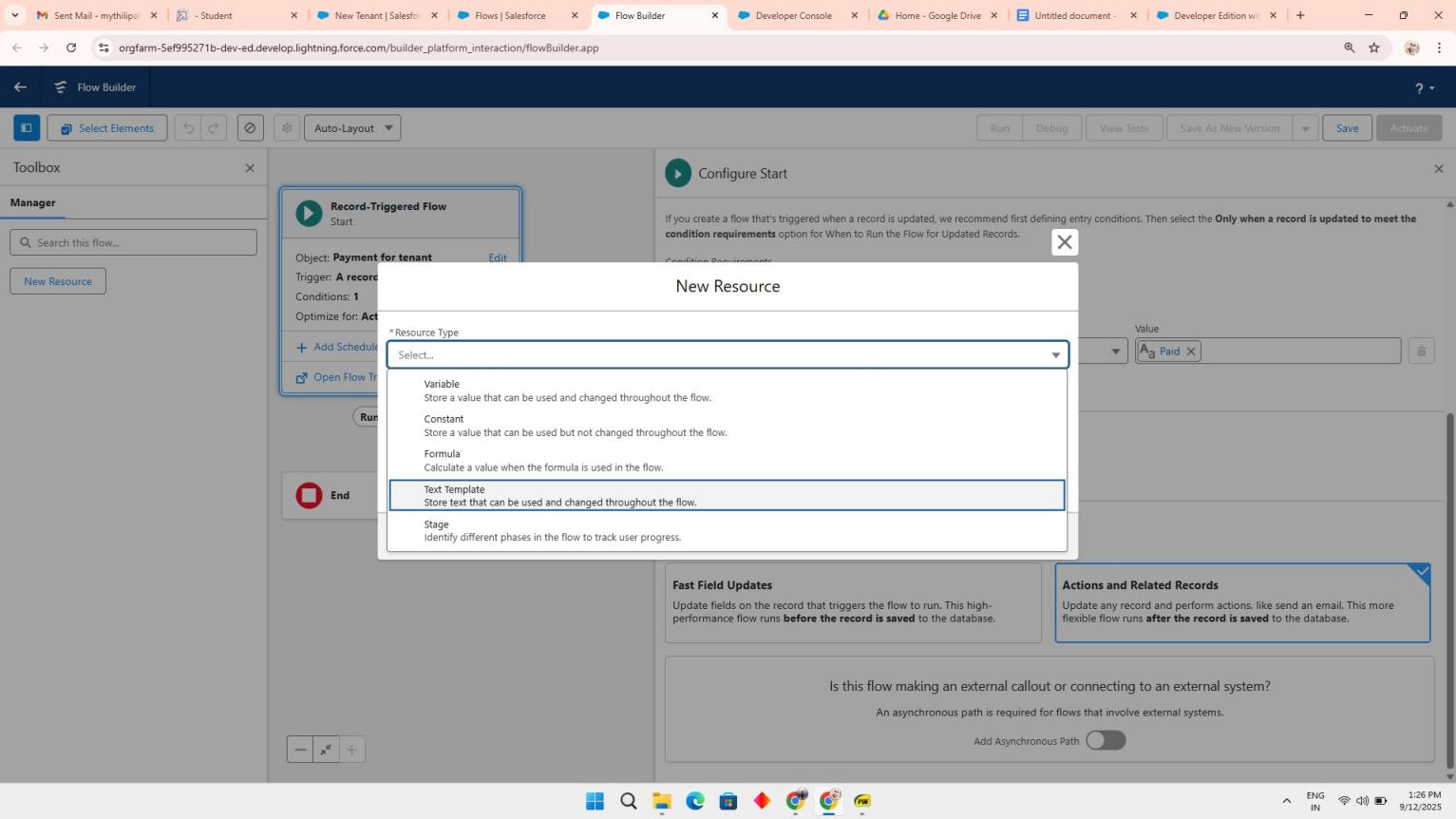


# Apex Trigger

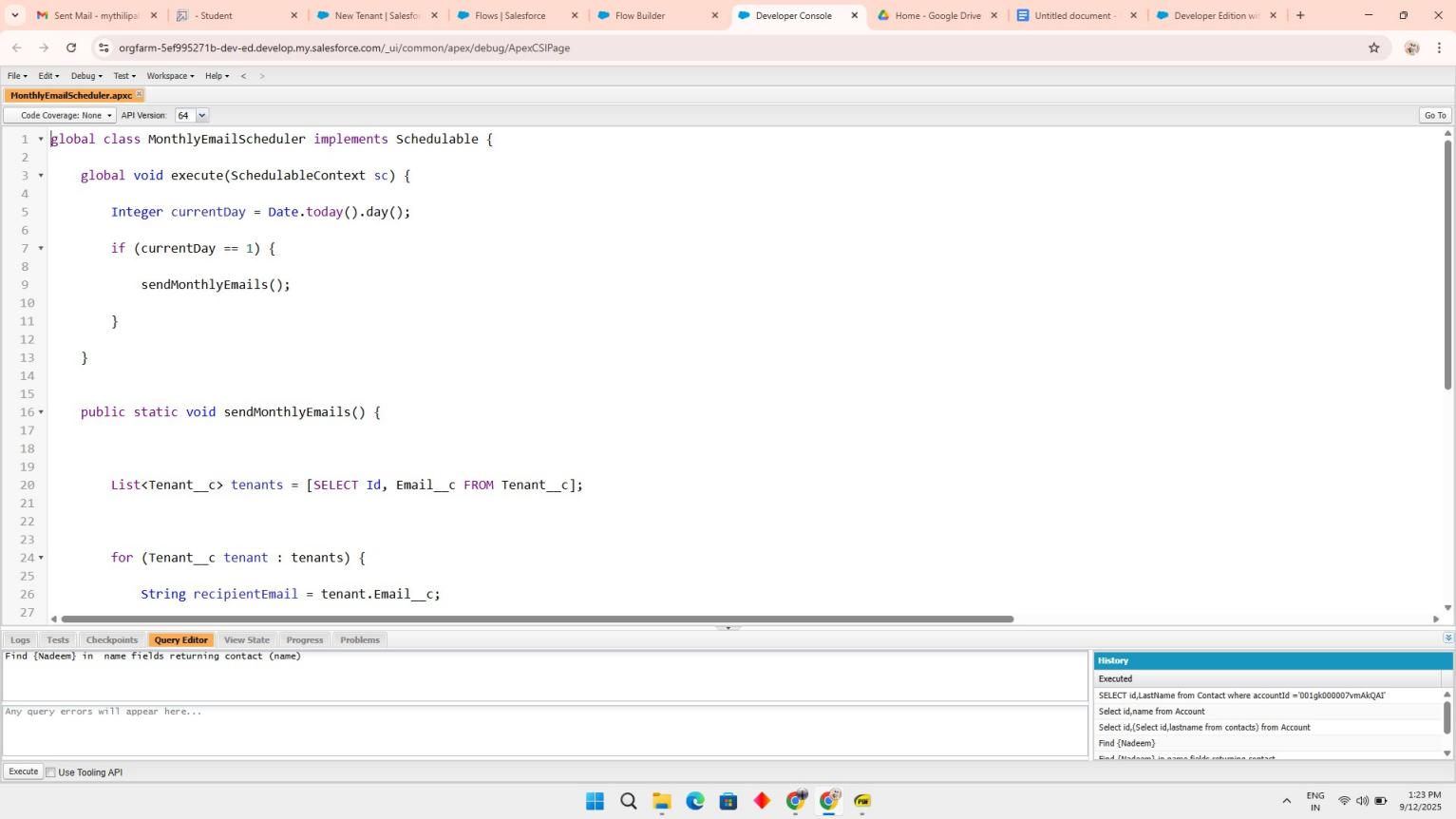
* + Create an Apex Trigger
* Create an Apex Handler class
* Testing the Trigger

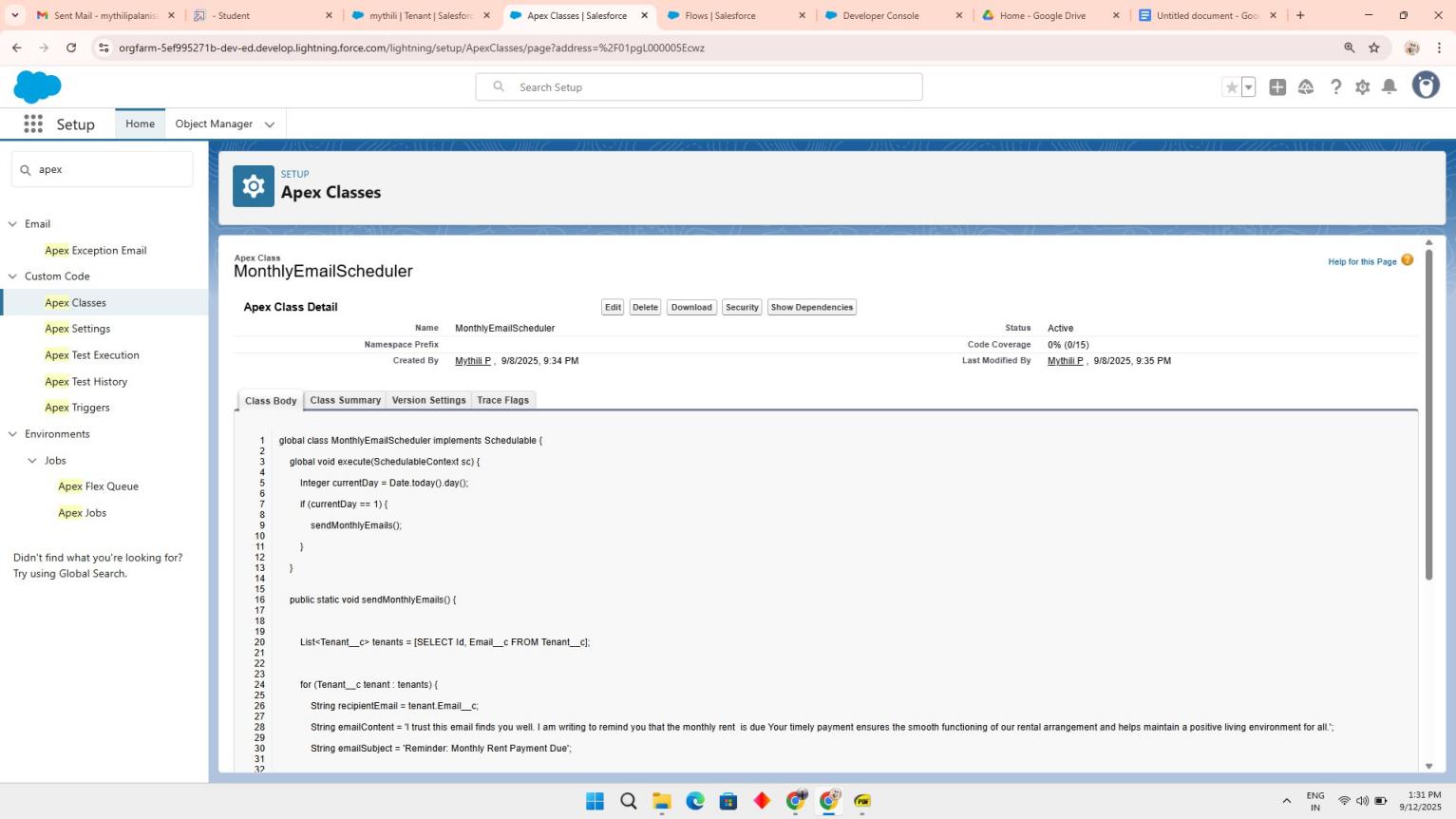
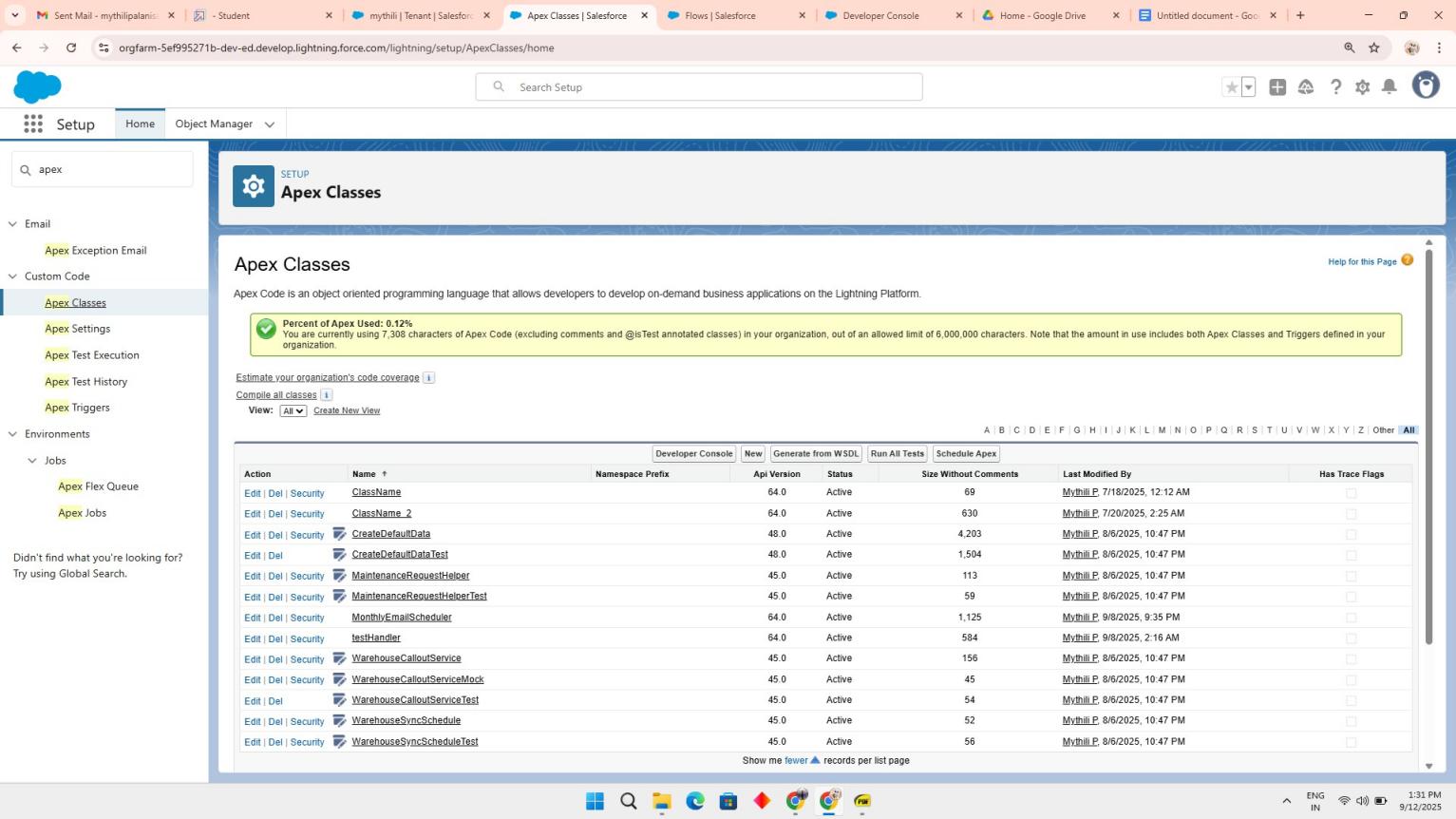


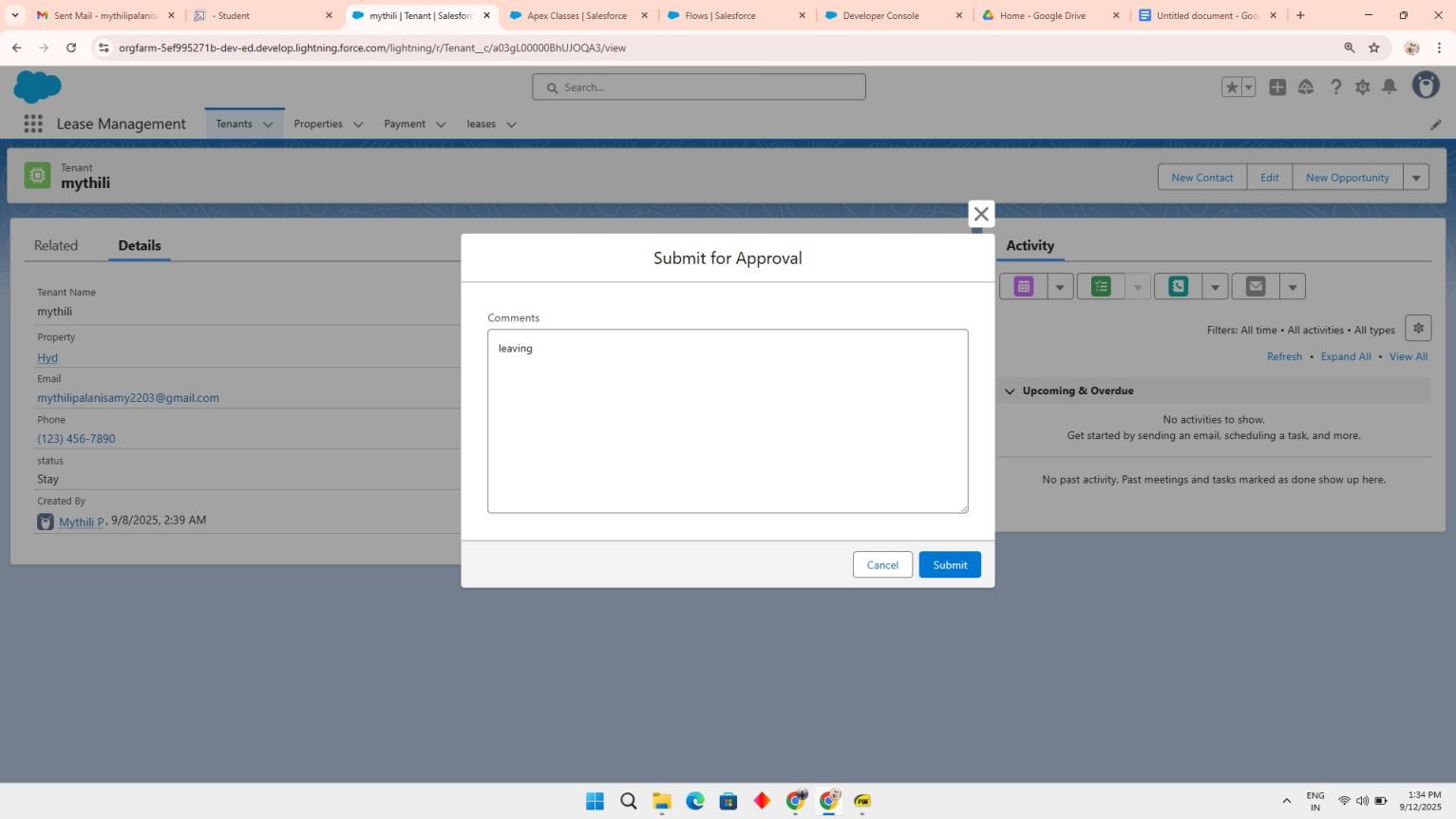
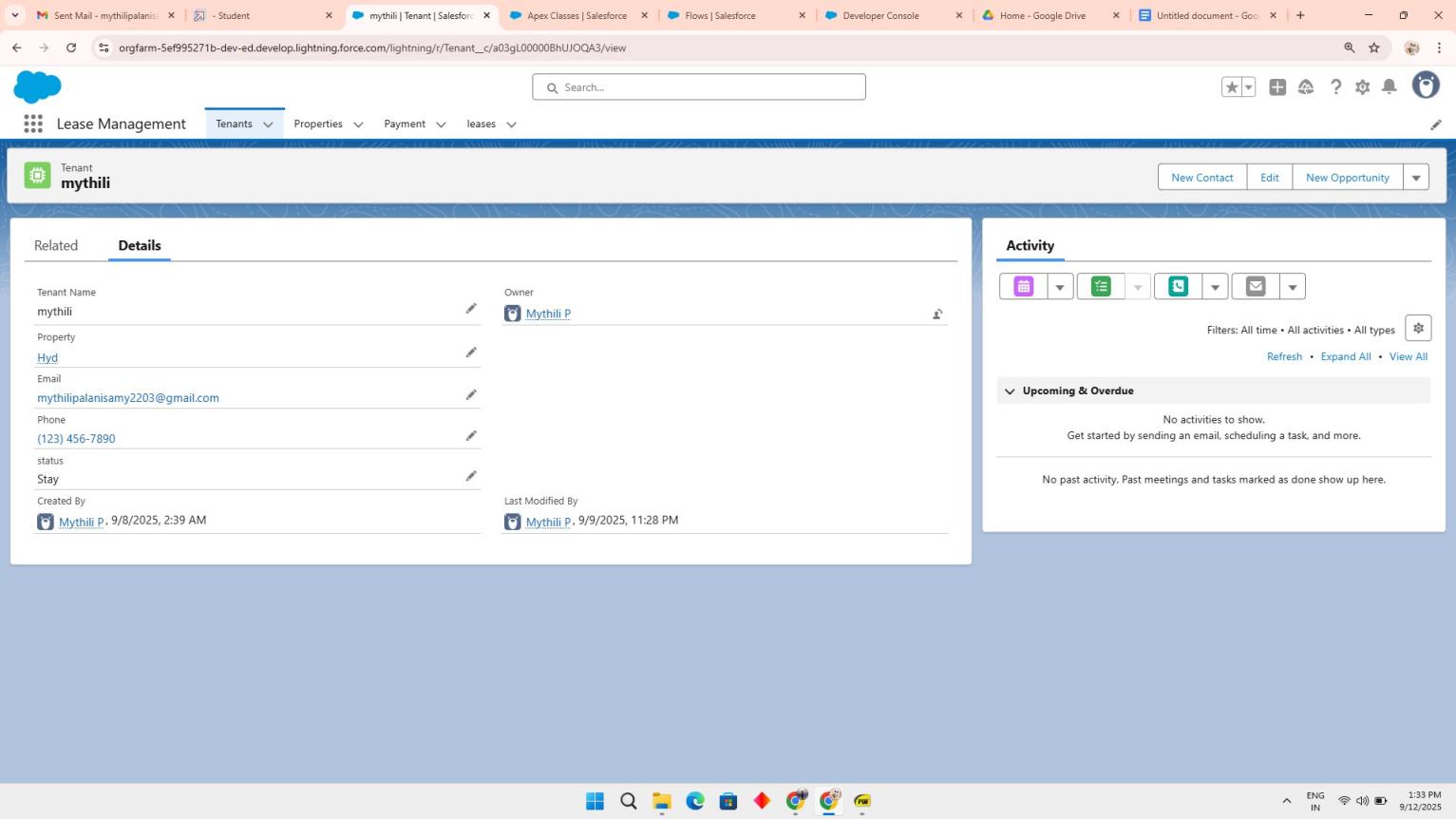
Create Flow for monthly payment

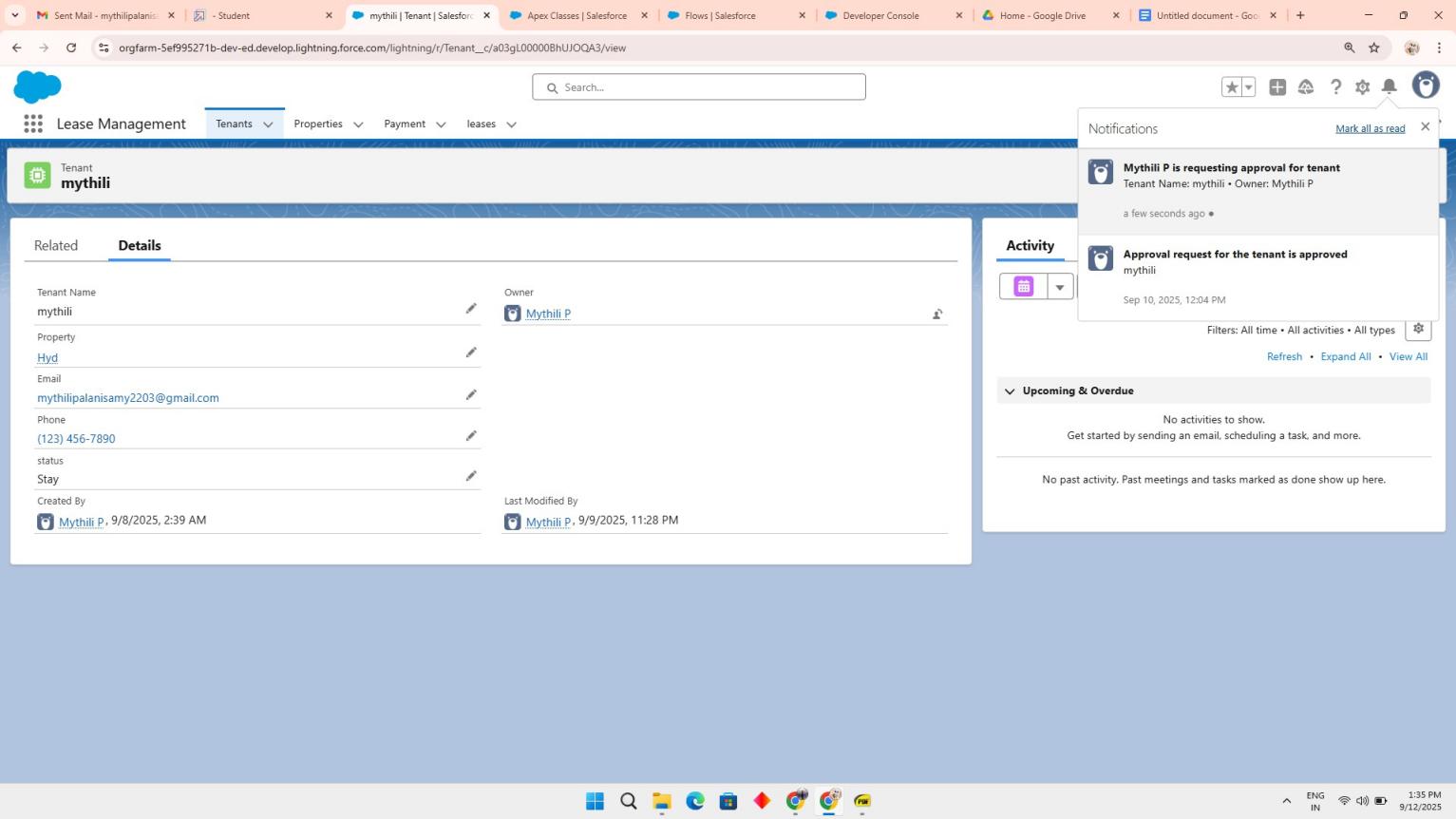
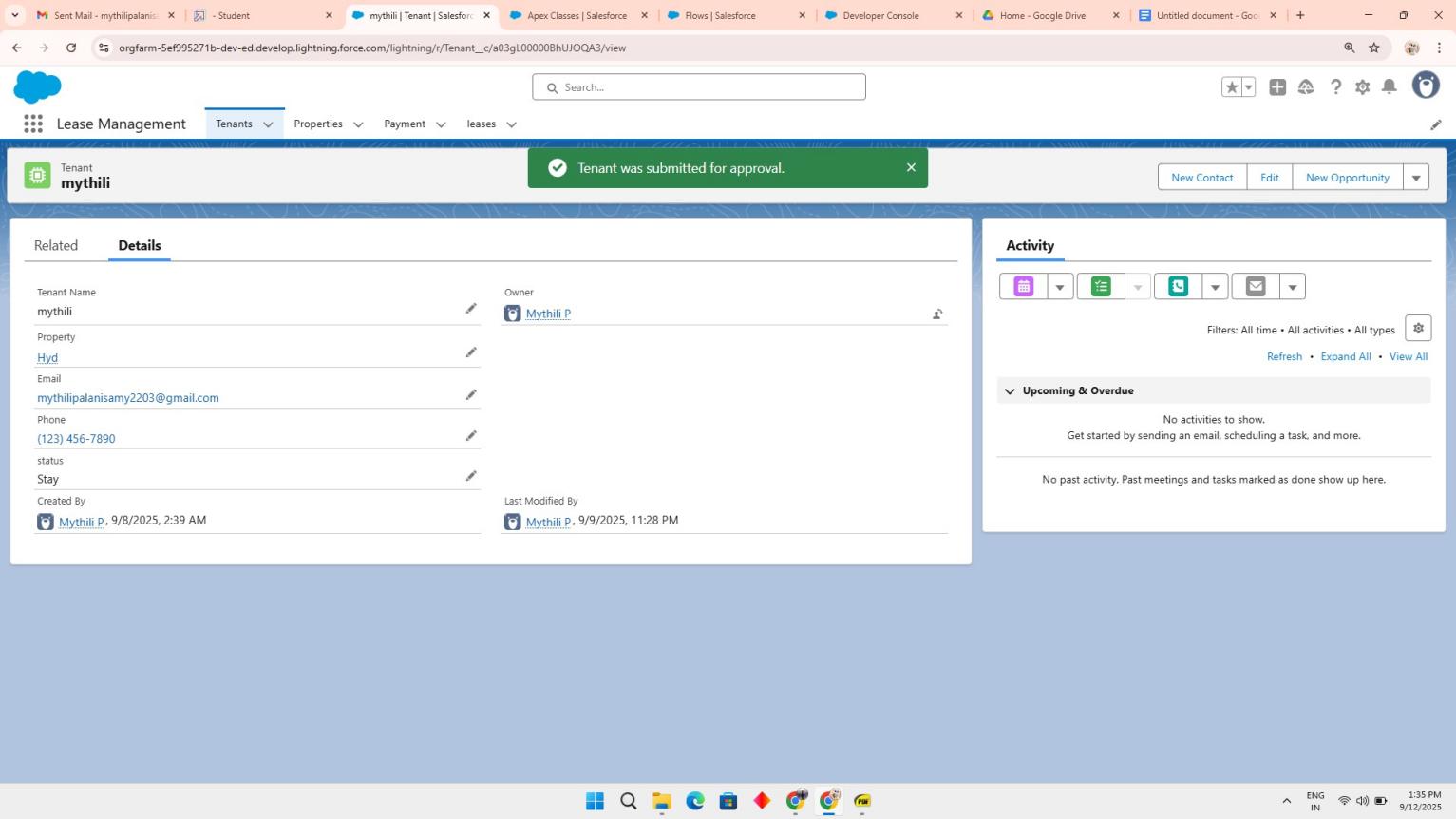


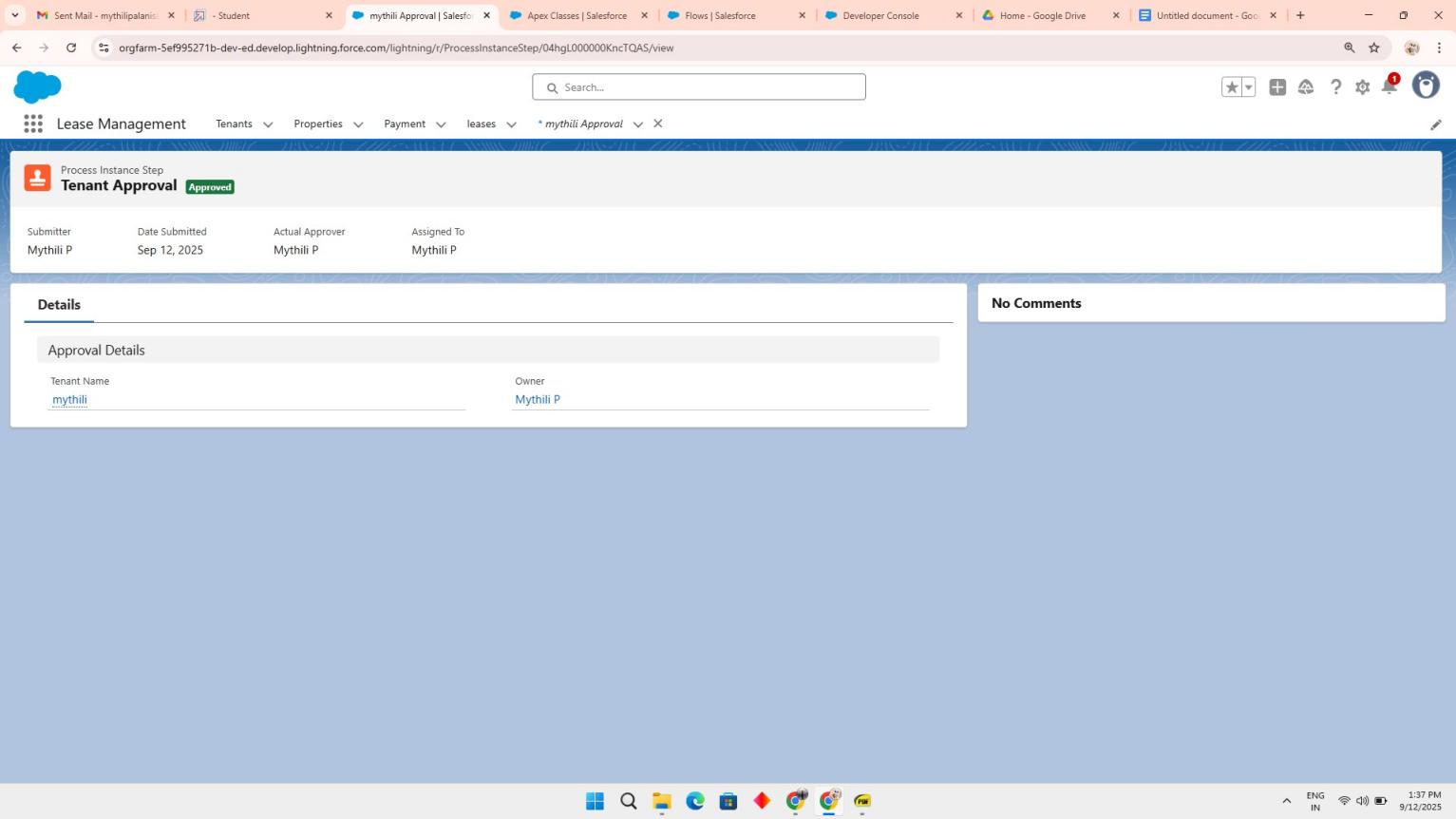
**Schedule class**

* Create an Apex Class



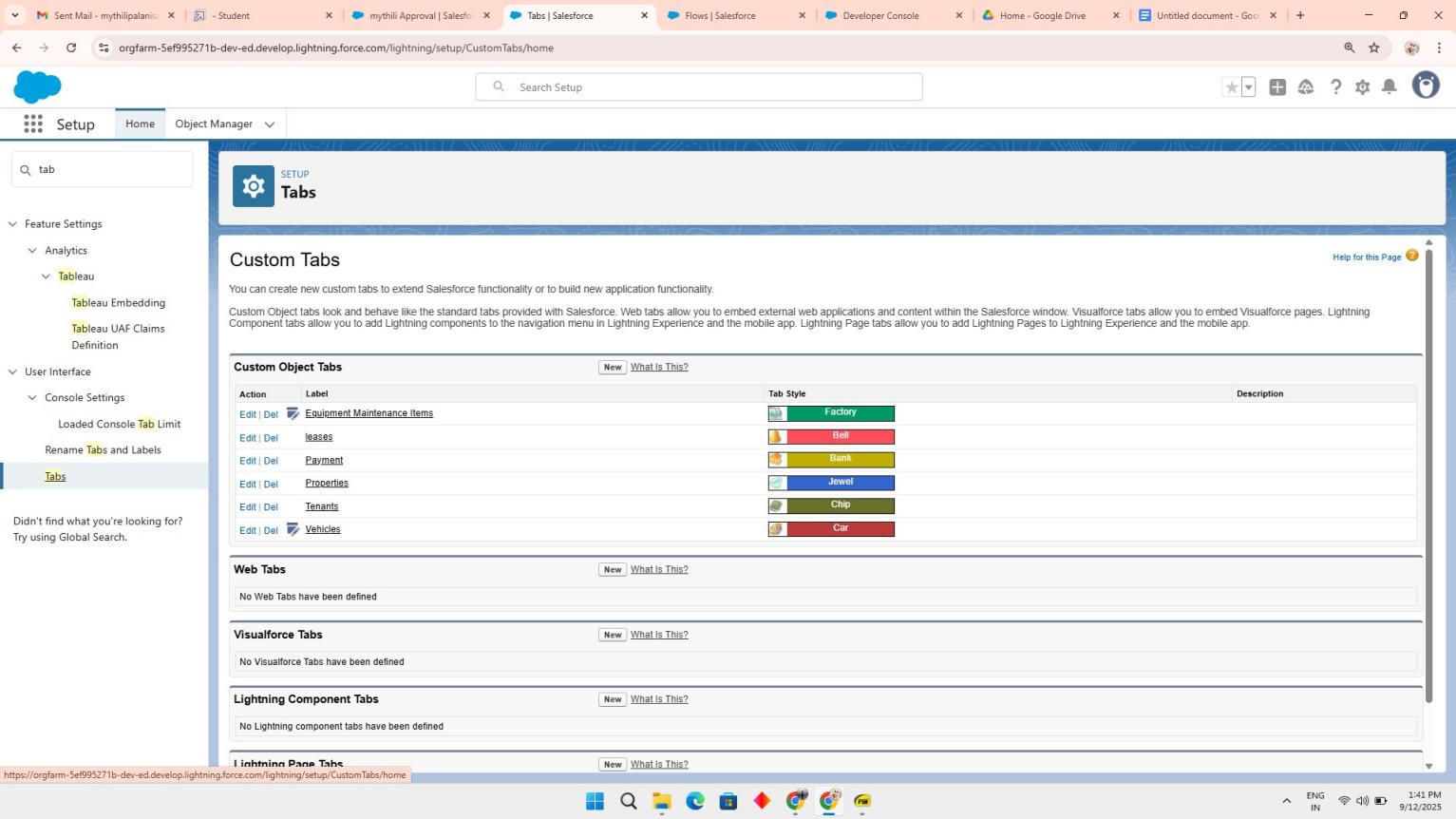
* Testing the approval process





**RESULTS**

Output Screenshots

Tabs for Property, Tenant, Lease, Payment

**ADVANTAGES & DISADVANTAGES**

**Advantages of Lease Management System:**

**Centralized Data Management –**All tenant, property, and lease records are stored in One place, reducing duplication and errors.

**Automation of Processes** – Workflows, approvals, and reminders minimize manual intervention, saving time and effort.

**Improved Accuracy** – Automated tracking of payments and contracts reduces chances of mistakes and missed deadlines.

**Better Communication** – Email alerts and notifications keep tenants and property managers informed in real time.

**Enhanced Compliance** – Role-based access ensures data security and adherence to legal Requirements.

**Data Analytics & Reporting** – Dashboards provide insights into occupancy rates, payment status, and lease renewals.

**Scalability** – Can be extended to manage multiple properties and tenants as the Business grows.

**Tenant Satisfaction** – Timely updates and smoother processes improve the overall tenant experience.

## Disadvantages of Lease Management System

**High Initial Setup Cost** – Implementing Salesforce and customizing it for lease management may require investment.

**Training Requirement** – Staff need to be trained to use Salesforce effectively, which can take time.

**Dependency on Internet** – Being a cloud-based solution, it requires stable internet Connectivity.

**Customization Complexity** – Advanced customizations may need skilled Salesforce developers, adding to costs.

**Data Migration Challenges –** Transferring existing lease and tenant records into the system can be time-consuming.

**Subscription Costs** – Ongoing Salesforce licensing fees may be expensive for smaller organizations

# CONCLUSION

The Lease Management System streamlines leasing operations by centralizing tenant data,contracts, payments, and communication within Salesforce. It enhances accuracy, Reduces manual effort, and ensures timely rent collection and contract renewals through automation.With features like Flows, Approval Processes, and Email Alerts, the system improves efficiency and communication, while dashboards provide

real-time insights for decision-making. Overall, it boosts productivity, tenant satisfaction, and compliance, offering a scalable foundation for future growth.

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});

}

}

}