TITLE: - LEASE MANAGEMENT USING SALESFORCE

College Name: Navarasam Arts and Science College for Women

College Code: bru3a

Team Members:

Team Leader Name: Shanmuga Sree C

E-mail: shanmugasree093@gmail.com

Team Member1: Monisha V

E-mail: monishav25806@gmail.com

Team Member2: Vaishnavi V

E-mail: v43810051@gmail.com

Team Member3: Sandhiya Devi E

E-mail: sarankumar112233sa@gmail.com

INTRODUCTION

Project Overview:

The Lease Management project develops a Lease Management system on the Salesforce platform to handle agreements for properties and assets. Salesforce objects and workflows are used to track lease details, monitor payments, and manage compliance. The system provides alerts, reports, and dashboards to support decision-making. It ensures a structured approach to managing the complete lease lifecycle.

Purpose: This project provides a system to manage lease agreements effectively. It automates key tasks to minimize errors and support compliance.

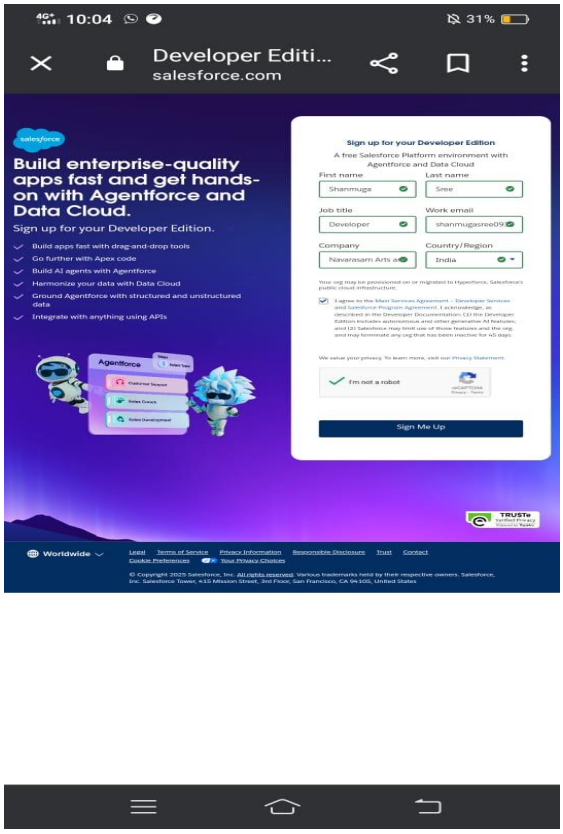
Objectives:

- **Streamline Lease Agreements:** Automate the creation, renewal, and termination of lease contracts.
- **Ensure Accuracy in Records:** Maintain centralized, accurate, and up-to-date information on all leases.
- **Improve Communication**: Facilitate better collaboration between property, owners, tenants, and managers**.**
- **Optimize Payment Tracking:** Automate rent/lease payment schedules, reminders, and receipts.
- **Provide Real-Time Insights:** Generate reports and dashboards for better decision-making and performance monitoring**.**

1.DEVELOPMENT PHASE

1.1 Creating Developer Account in Salesforce:

By using this URL- https://developer.salesforce.com/signup

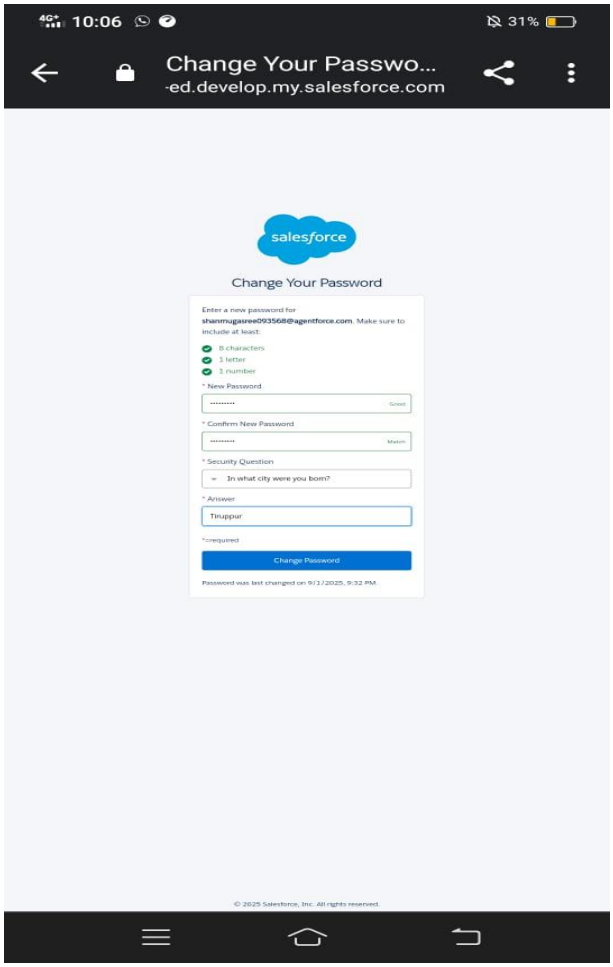


 Enter essential details to sign up in developer edition as shown in image,

1. Name – First Name & Last Name.
2. Job title – Developer.
3. E-mail.
4. Company Name or College Name.

1.2 Account Activation

Verifying the e-mail account to reset the password for activating salesforce account.



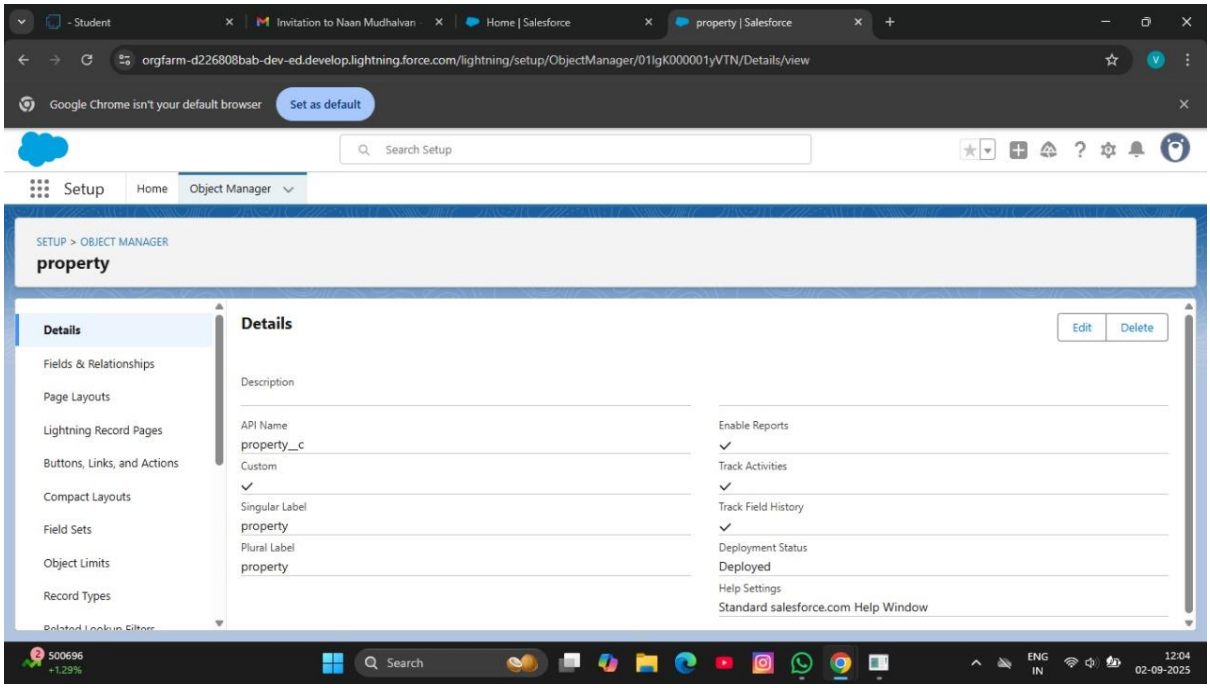
- Set the new password.
- Confirm the new password.
- Set answer for security question.
- Now you will redirect it to the salesforce setup page.

2. OBJECT CREATION PHASE

To create an object:

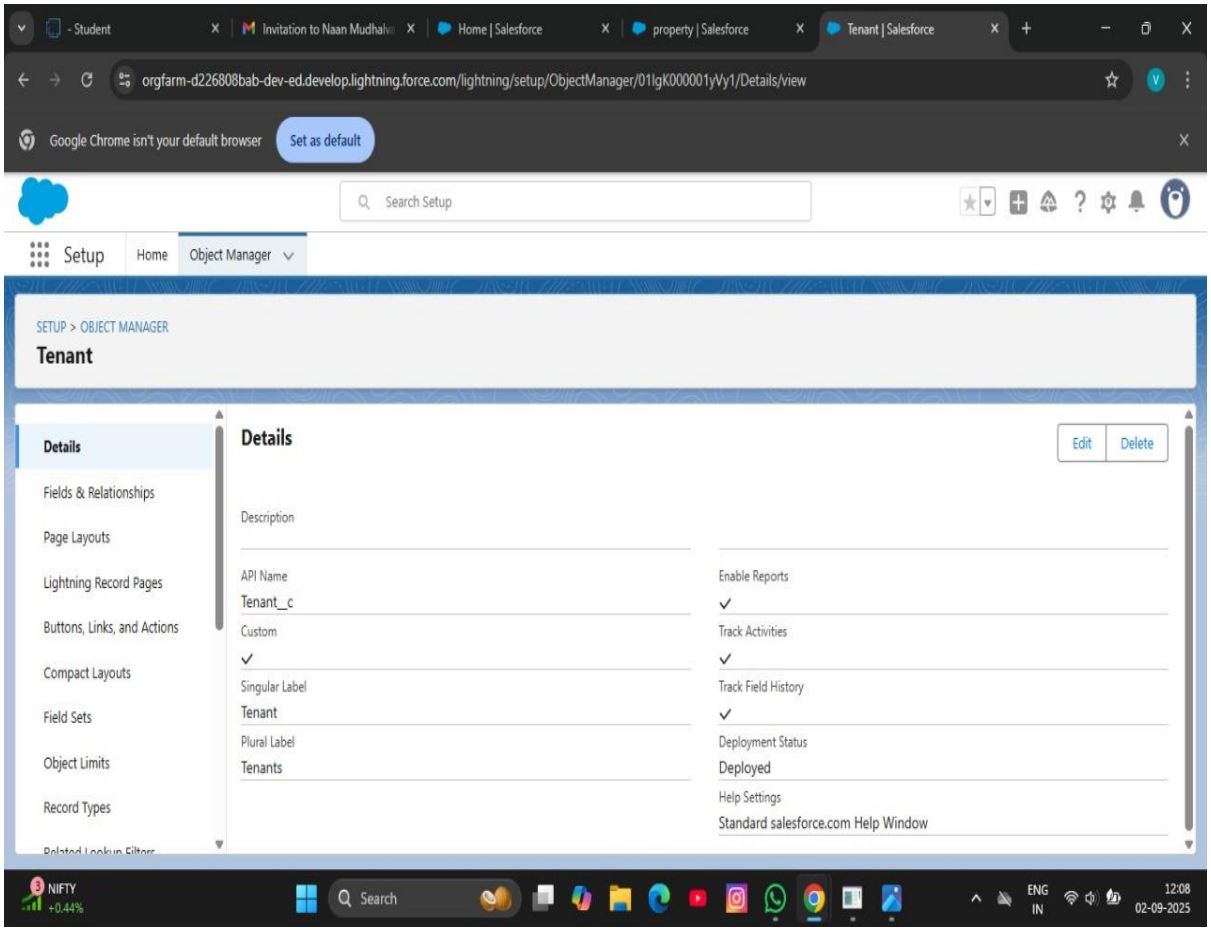
From the setup page > Click on Object Manager > Click on Create > Click on Custom Object.

2.1 Creating Property Object



- **Created object property.**
- It stores details of properties available for lease, such as location and type. It helps identify assets in the system. Properties are linked to lease records for tracking.

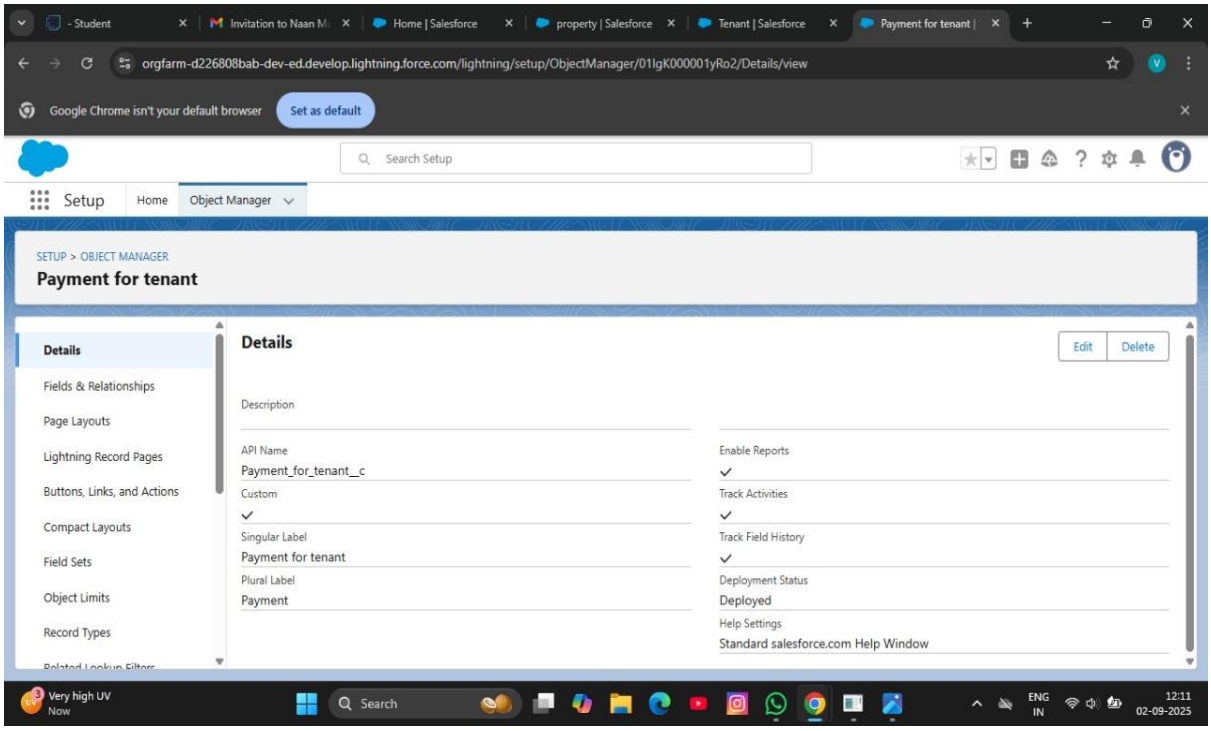
2.2 Creating Tenant Object



- **Created object Tenant.**

- It holds tenant information including name, contact, and lease history. It allows easy management of client details. Tenants are connected to properties and leases for clarity.

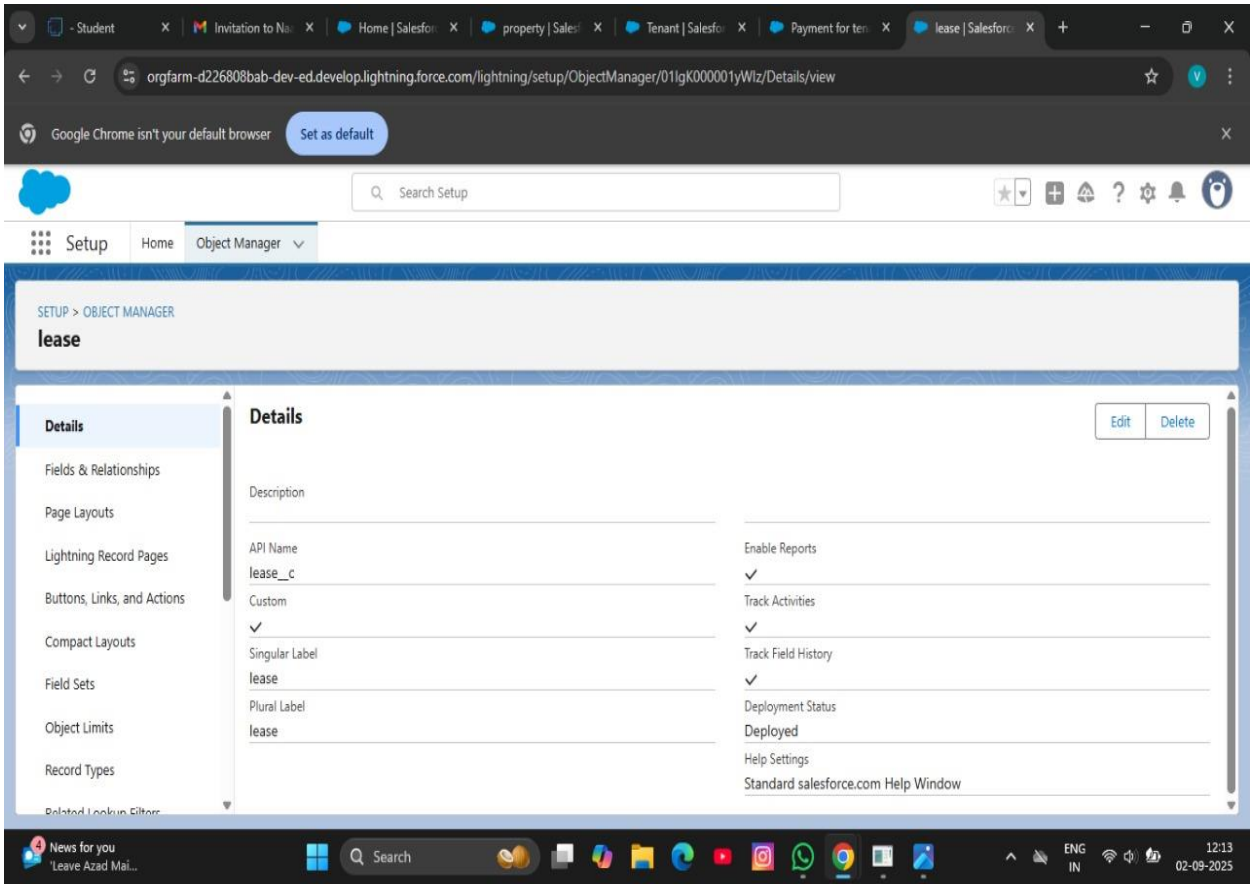
2.3 Creating Payment Object



- **Created object Payment for tenant.**

- It records all payment transactions related to leases. It keeps track of due amounts, status, and history. Payments are tied to leases for accurate financial management.

2.4 Creating Lease Object



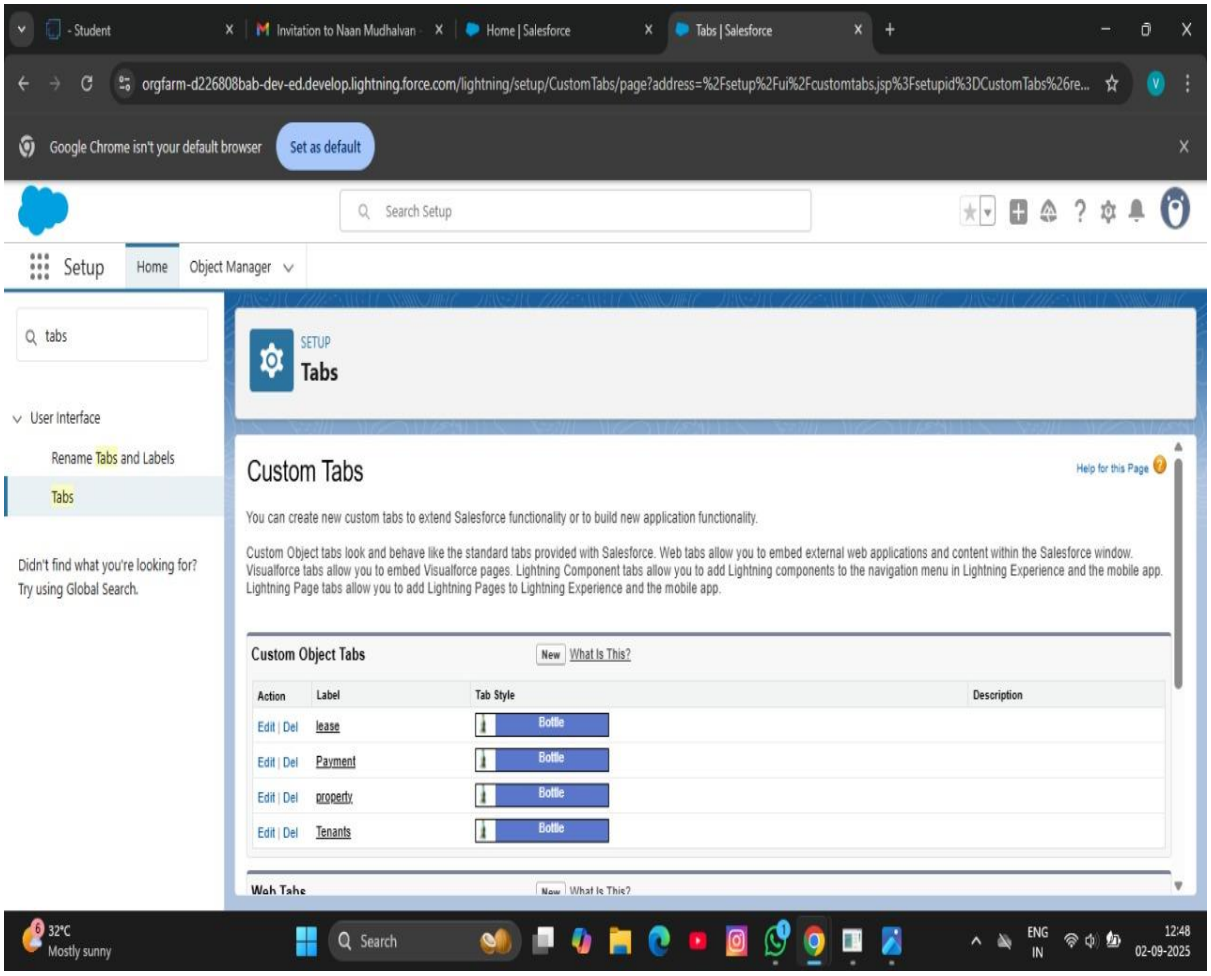
- **Created object lease.**

- It manages agreements between tenants and properties with start/end dates and terms. It acts as the central link between tenants, properties, and payments. This ensures proper lifecycle tracking.

3.TAB CUSTOMIZATION PHASE

To create customized tab:

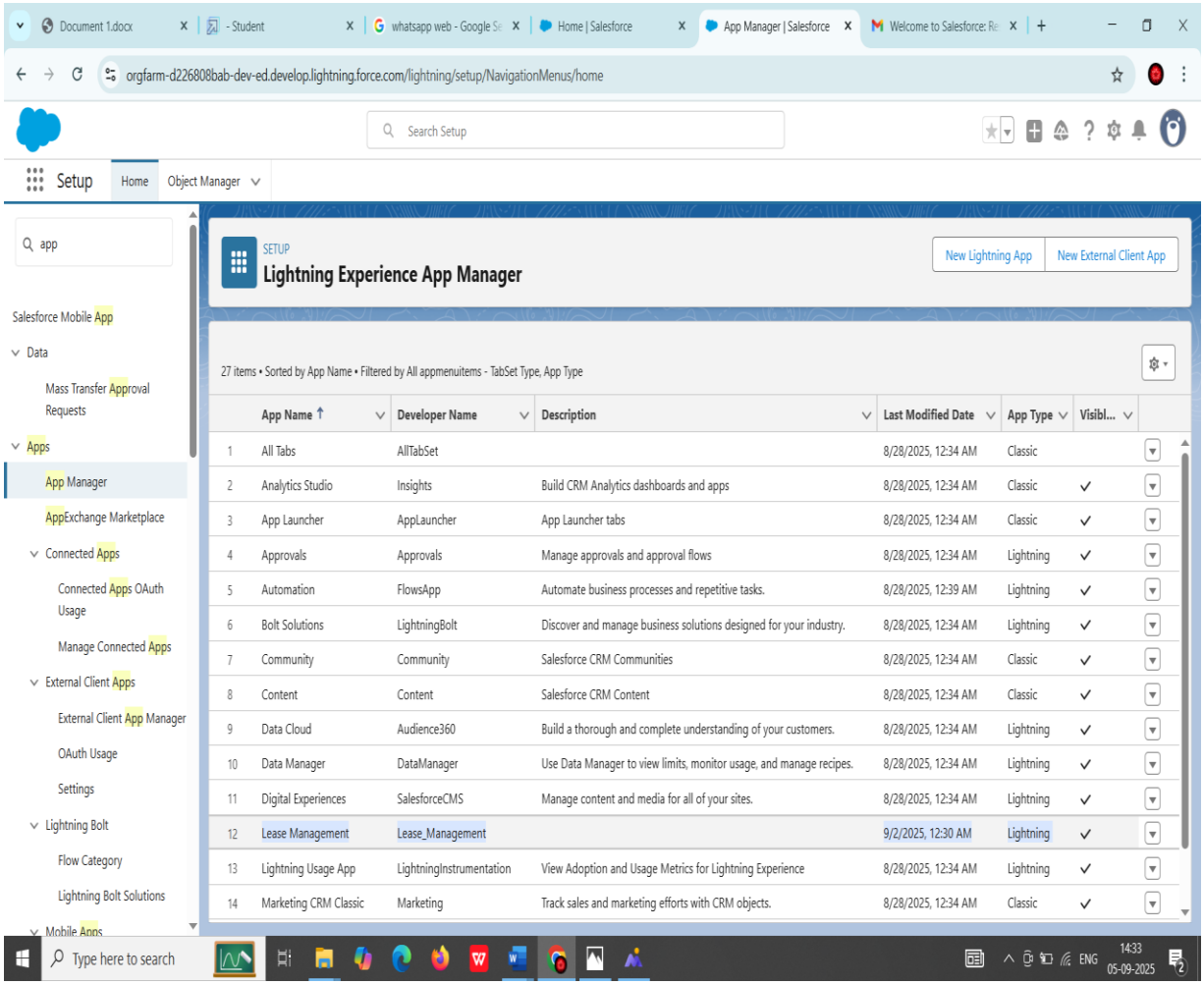
- From the setup page > **type Tabs in Quick Find bar >** click on tabs **>** New.
- Select object > **Tab Style.**



- Select the desired tab styles for the objects lease, Payment, property, Tenants.

4. APPLICATION DEVELOPMENT PHASE

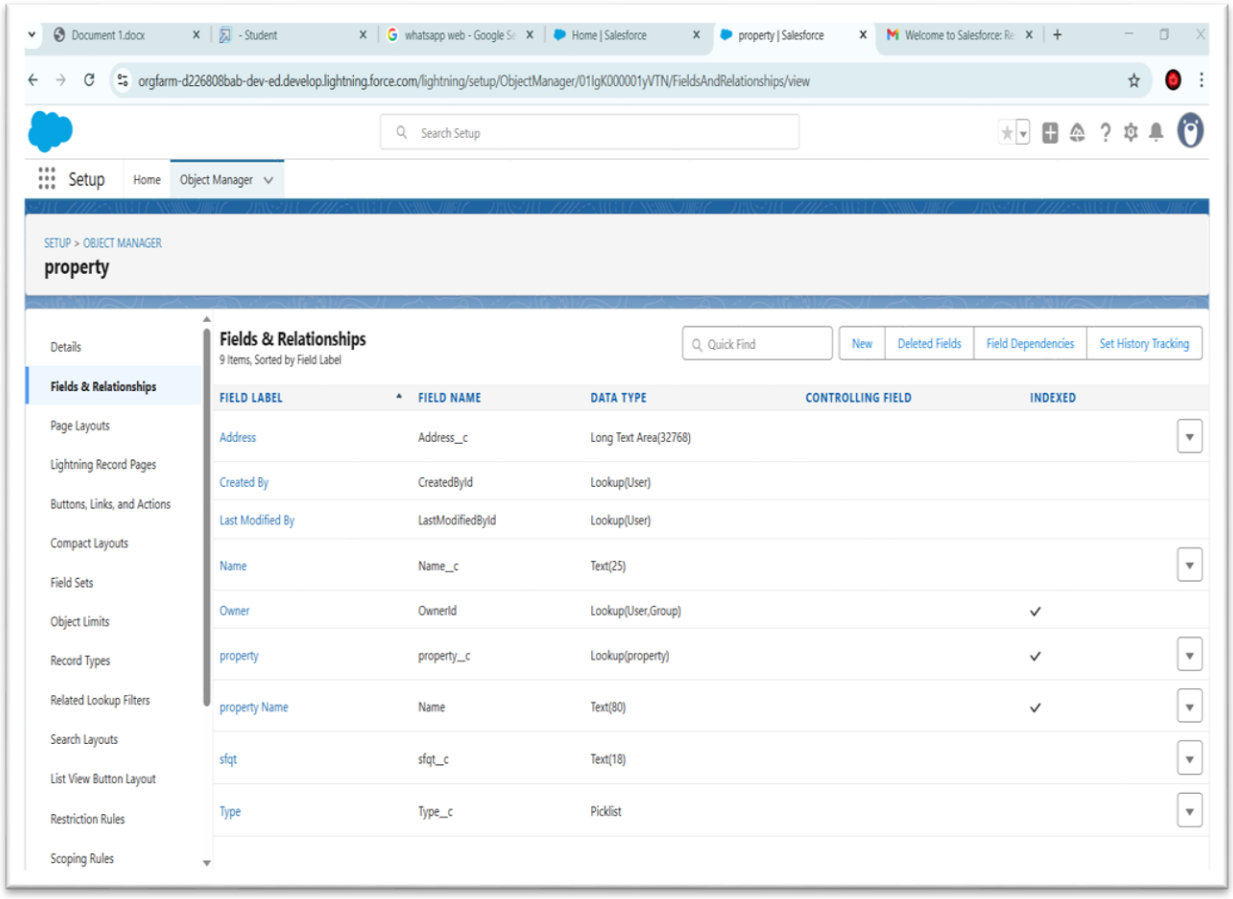
- **Creating Lightning app for Lease Management**



- A Lightning App was created to bring all lease-related objects like Property, Tenant, Lease, and Payment under one workspace.
- It provides users with an organized view and easy navigation between records.
- This improves efficiency by managing all lease processes within a single app.

5.FIELDS AND RELATIONSHIPS PHASE

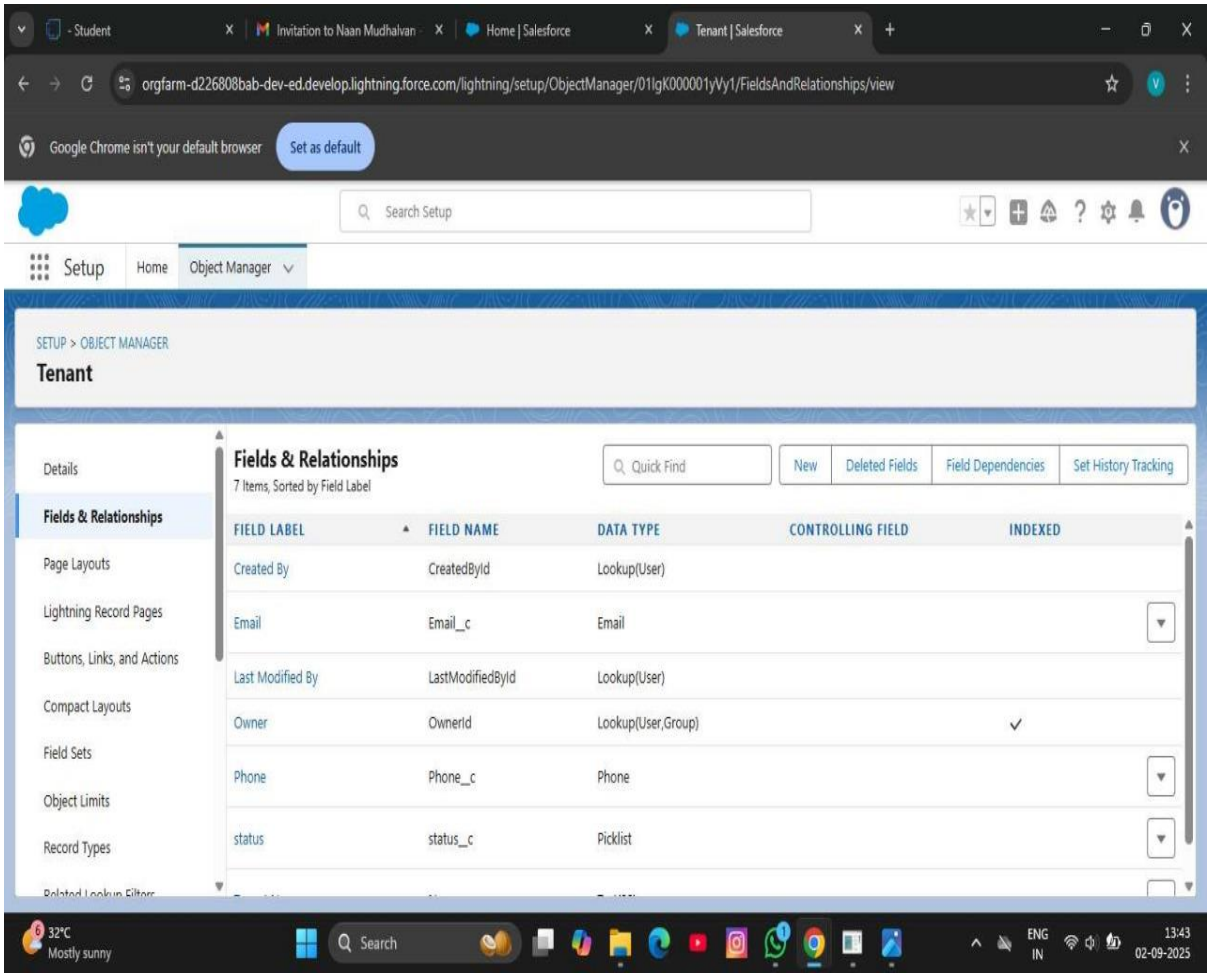
5.1 Creating Fields for Property Object



- Created field labels of **name, address, type, sfqt** for the property object.

- Custom fields were added to capture details like property size, type, and availability. These fields help maintain accurate property records. They also support linking with lease agreements.

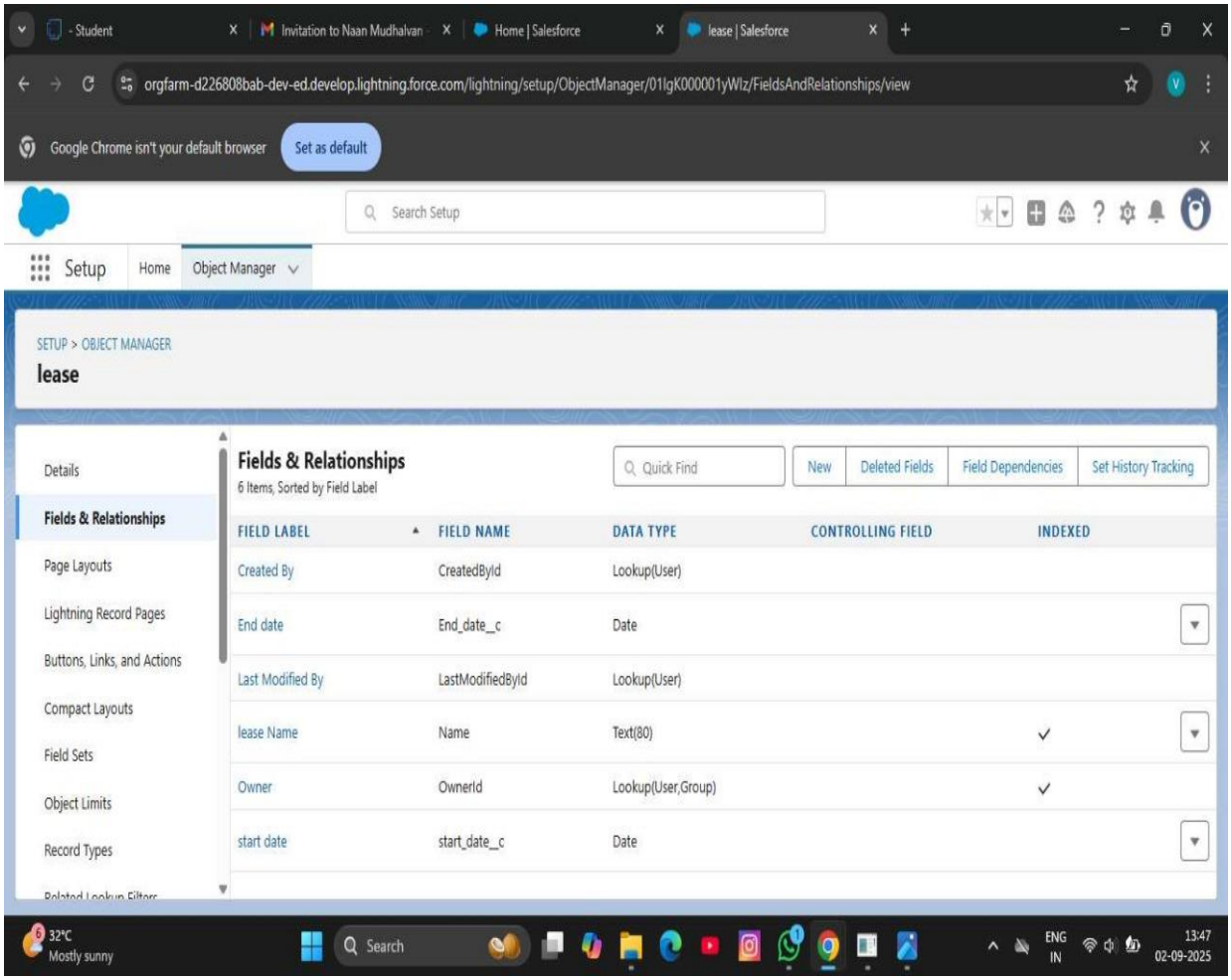
5.2 Creating Fields for Tenant Object



- Created field labels of **Email, Phone, status** for the Tenant object.

- Fields were created to store tenant information such as contact details, ID proof, and lease preferences. This ensures complete tenant profiles. The fields also help in tracking tenant history.

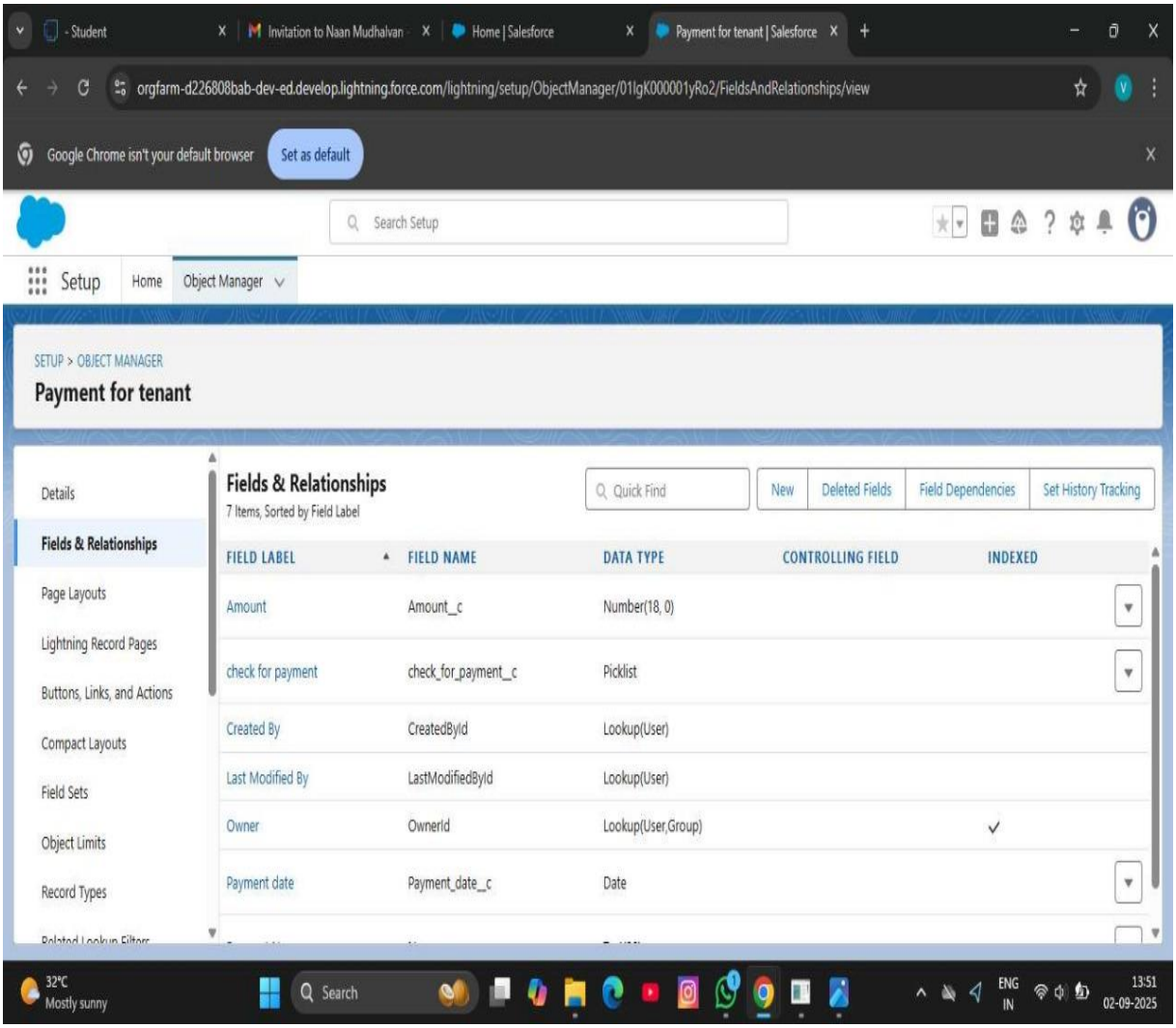
5.3 Creating Fields for Lease Object



- Created field labels of **start date, End date** for the lease object.

- Custom fields were designed to record lease terms, duration, and agreement status. These fields act as the backbone of lease tracking. They also connect with payment schedules and property details.

5.4 Creating Fields for Payment for Tenant Object



- Created field labels of **Payment date, Amount, check for payment** for the payment for tenant object.

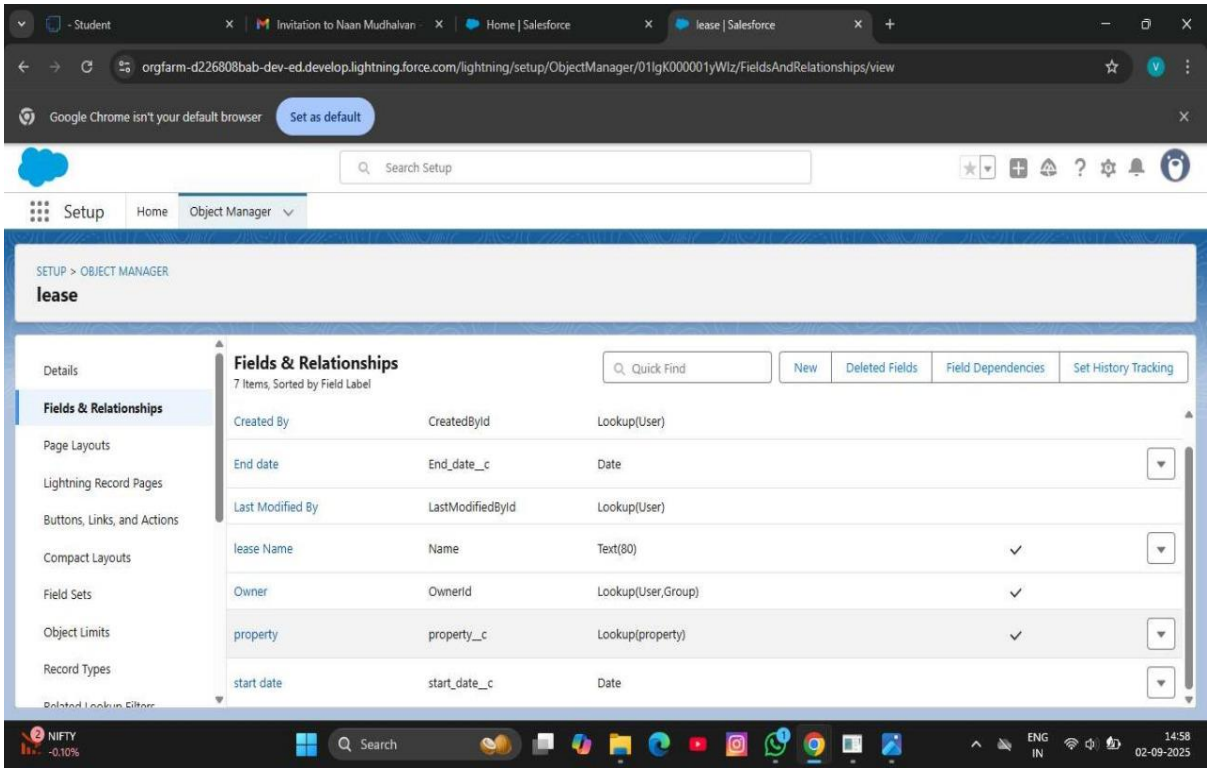
- Custom fields were designed to record lease terms, duration, and agreement status. These fields act as the backbone of lease tracking. They also connect with payment schedules and property details.

5.5 Creation of Lookup Fields

- **Creation of Lookup Field on Lease Object**

1. Click on Object Manager >> **(Lease)** >> click on object.

2. Click on “**Fields & Relationships**” >> New
3. Select lookup relationship
4. Select the related object “**property**” and click next.
5. Field Name: property
6. Field label: Auto generated
7. Next >> Next >> Save.

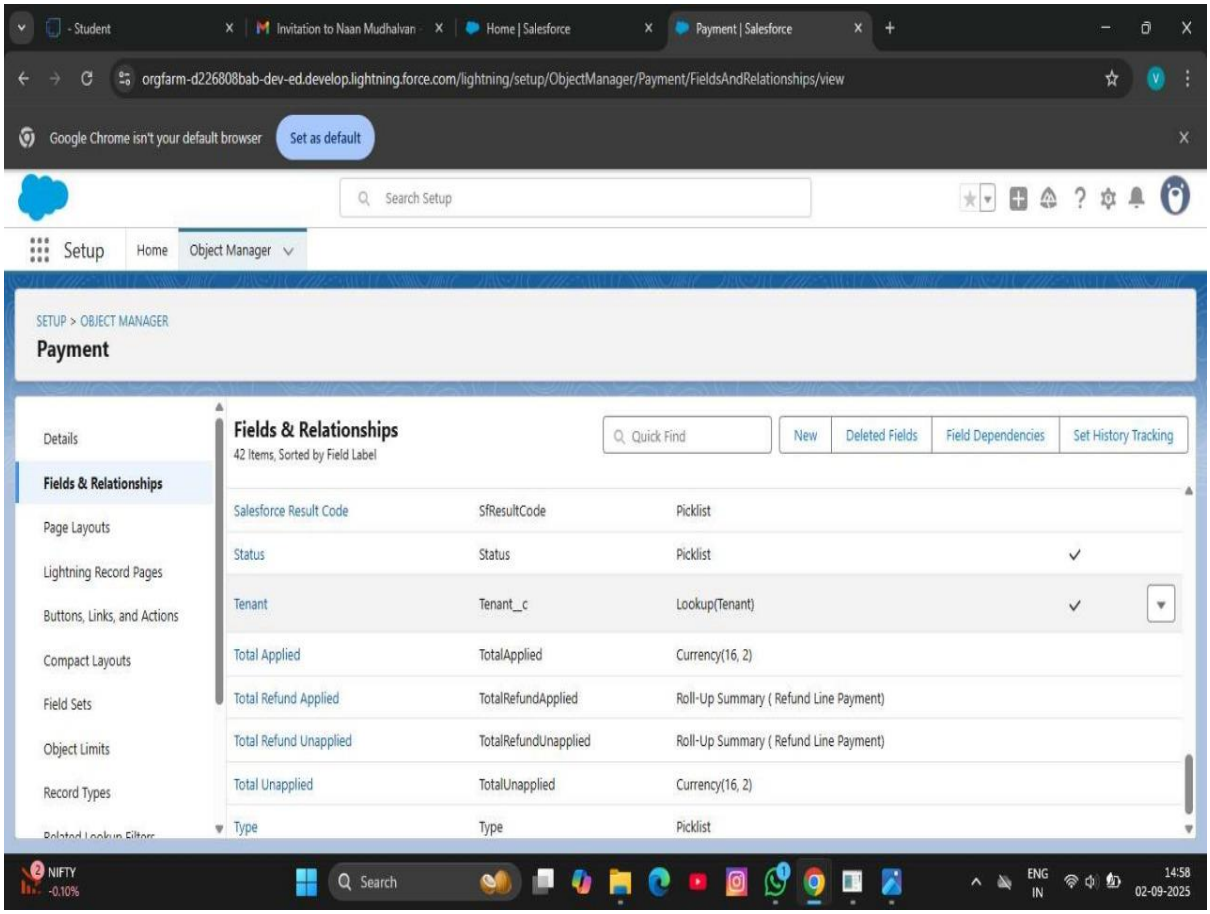


- The lookup field connects Lease with Property or Tenant, ensuring each lease record is linked to the right details.

- **Creation of Lookup Field on Payment Object**

1. Click on Object Manager >> **(Payment)** >> click on object.
2. Now click on “**Fields & Relationships”** >> New
3. Select lookup relationship
4. Select the related object **“Tenant”** and click next.

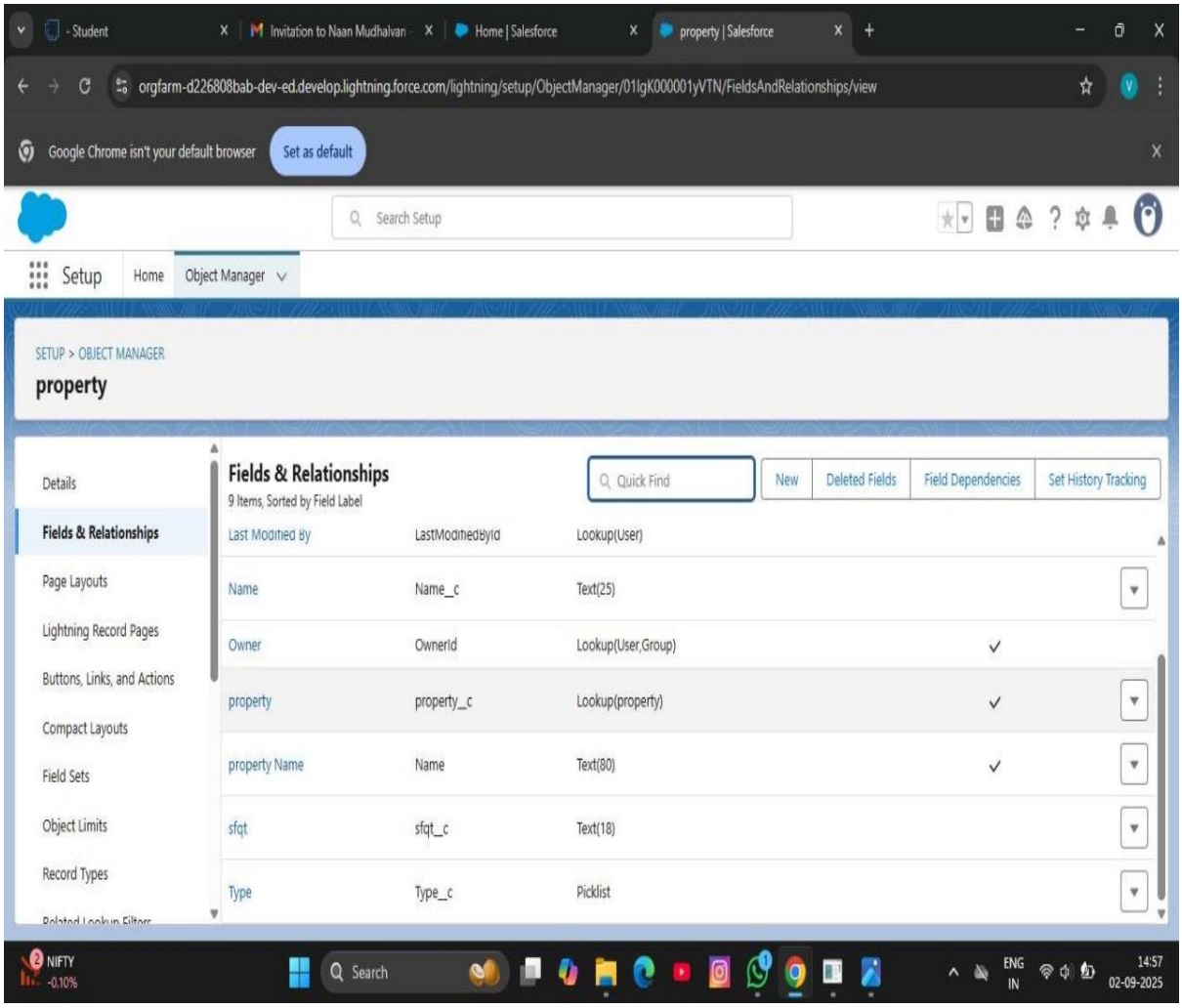
5. Field Name: Tenant
6. Field label: Auto generated
7. Next >> Next >> Save.



- The lookup field ties Payments to a Lease, so all monthly or one-time payments can be tracked under the correct agreement.
    - **Creation of Lookup Field on property Object**

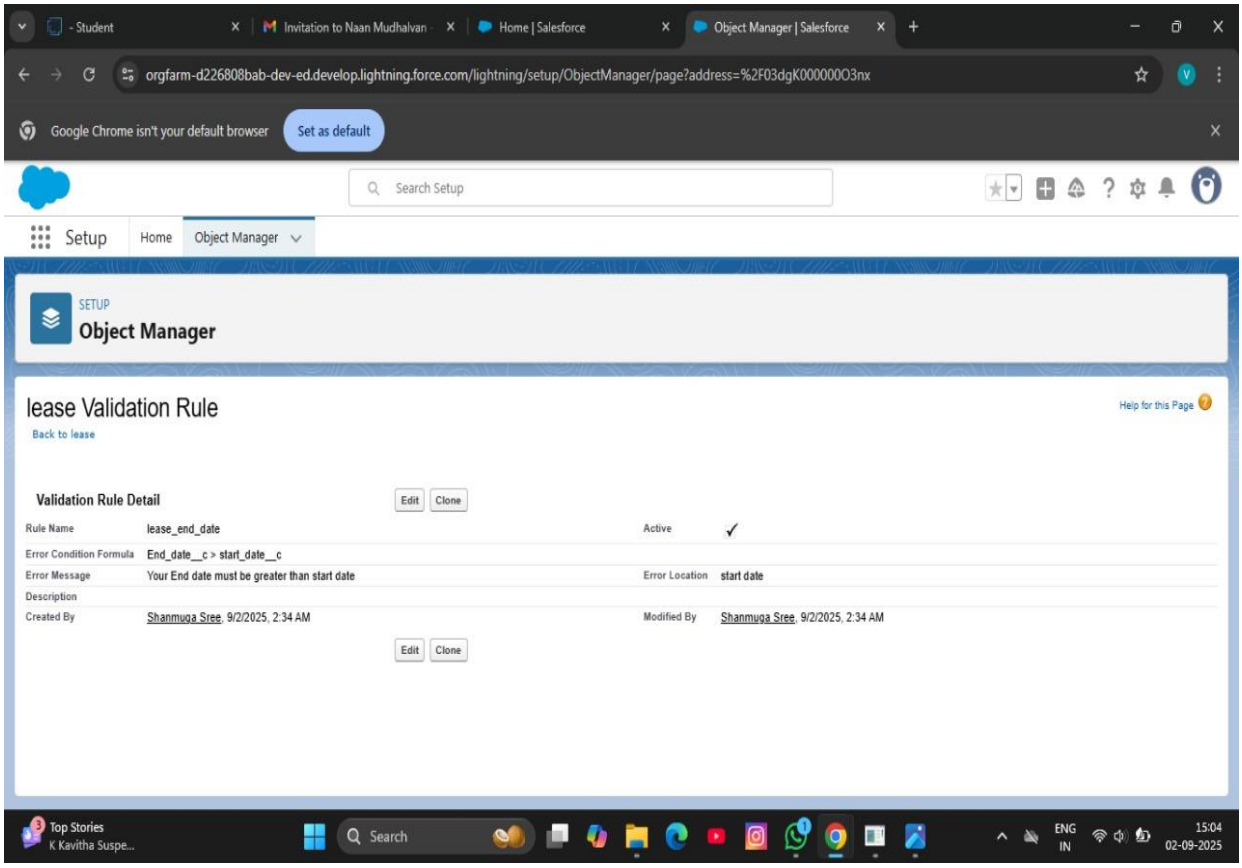
1.Click on Object Manager >>**(property)**>>click on object.
2. Now click on **“Fields & Relationships”** >> New
3. Select lookup relationship
4. Select the related object **“property”** and click next.
5. Field Name: property
6. Field label: Auto generated

7. Next >> Next >> Save.



- The lookup field associates Property with Lease, helping to identify which property is currently leased.

6.VALIDATION RULE PHASE



☐ **Created Validation Rule to a Lease Object.**

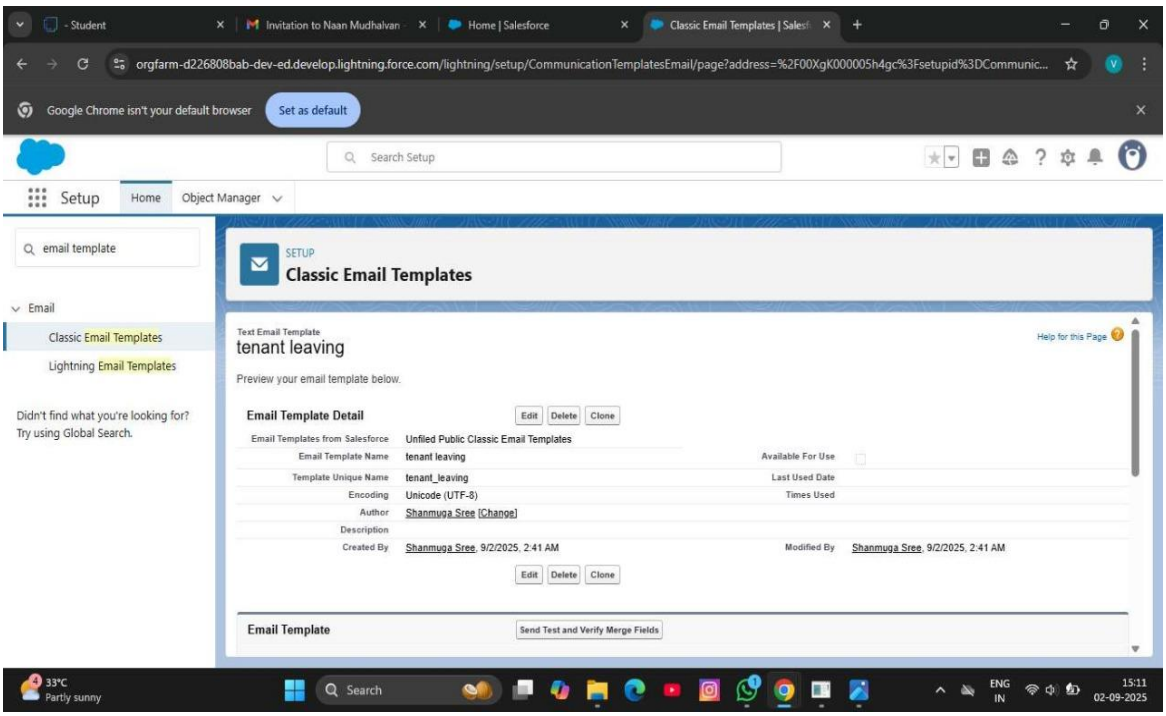
Validation rules ensure data accuracy by restricting invalid entries. In this project, they are applied to lease records, such as checking correct start and end dates, or preventing duplicate property assignments. This helps maintain reliable and error-free lease information.

7. EMAIL TEMPLATES PHASE

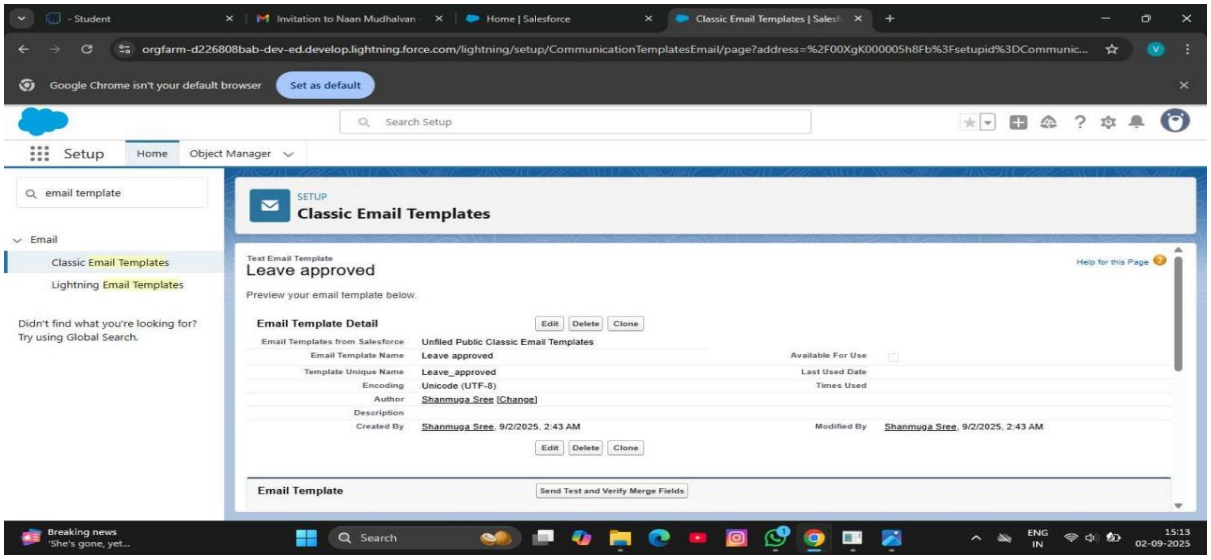
- ☐ Creating Email Template for Tenant Leaving
- ☐ Creating Email Template for Leave Approved
- ☐ Creating Email Template for rejection for leave

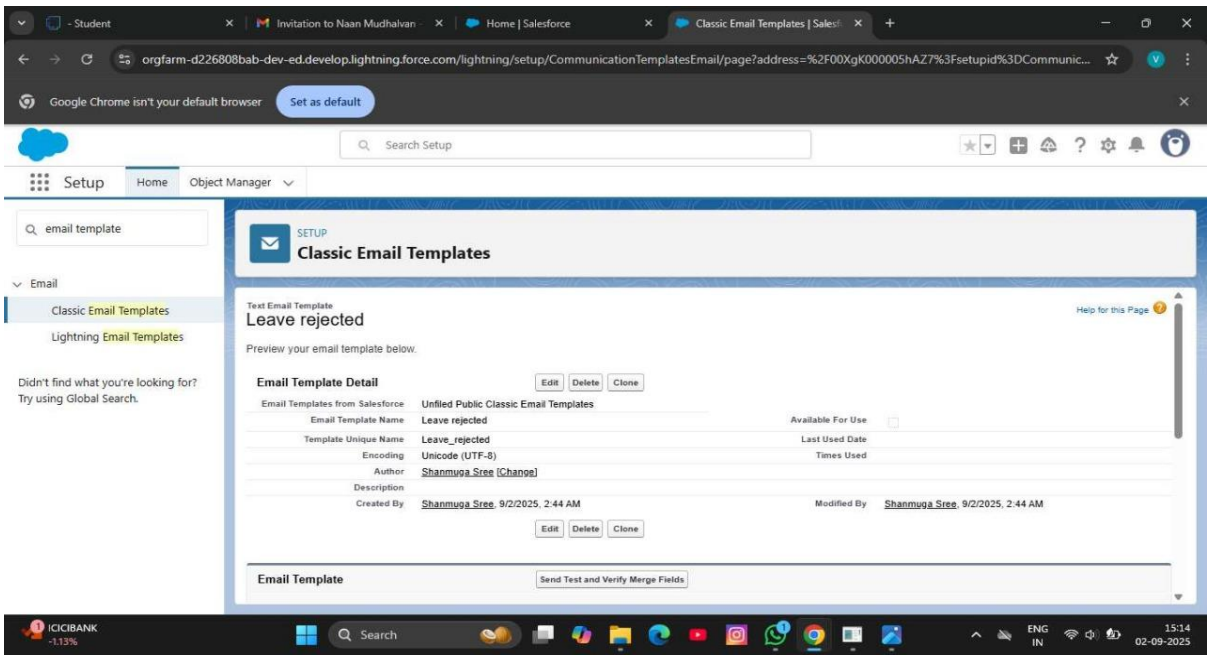
☐ Creating Email Template for Monthly payment

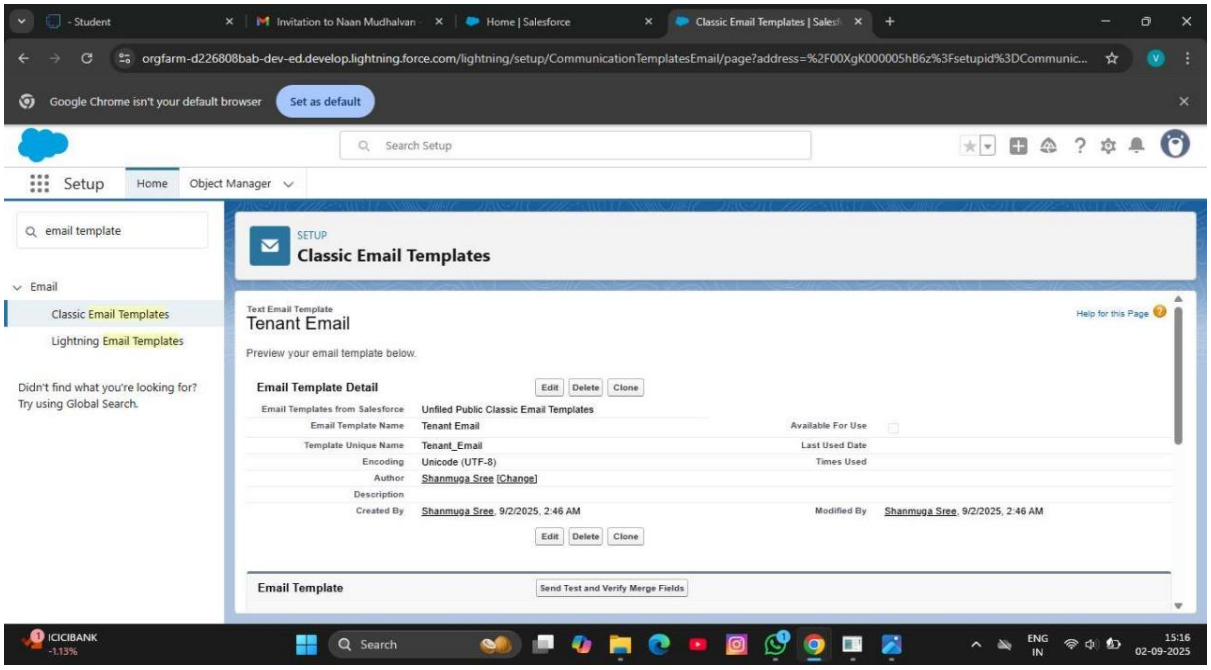
☐ Creating Email Template for successful payment

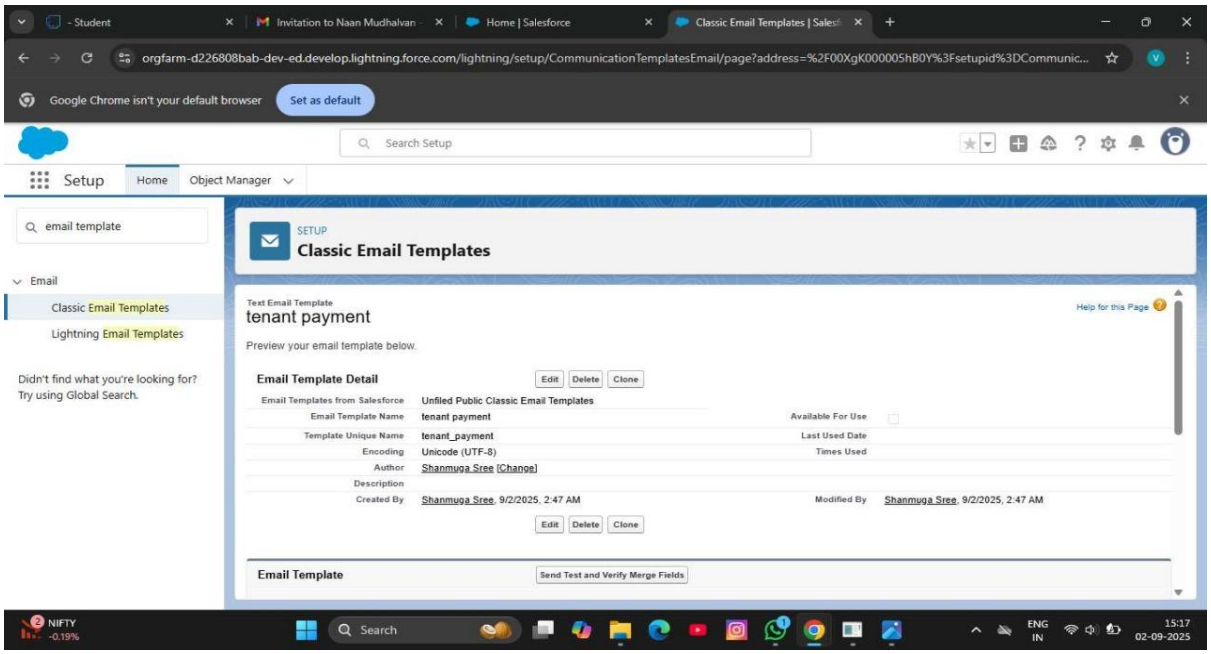


- Email templates were created to send automated notifications for **leave approvals and rejections**.
- This ensures clear communication and quick updates to users without manual effort.


- Student
Invitation to Naan Mudhalvan
Home | Salesforce
Classic Email Templates | Salesf
orgfarm-d226808bab-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK000005h8Fb%3Fsetupid%3DCommunic...
Google Chrome isn't your default browser
Set as default
Search Setup
Setup
Home
Object Manager
email template
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
Leave approved
Help for this Page
Preview your email template below.
Email Template Detail
Edit
Delete
Clone
Email Templates from Salesforce
Unfiled Public Classic Email Templates
Email Template Name
Leave approved
Available For Use
Template Unique Name
Leave_approved
Last Used Date
Encoding
Unicode (UTF-8)
Times Used
Author
Shanmuga Sree [Change]
Description
Created By
Shanmuga Sree, 9/2/2025, 2:43 AM
Modified By
Shanmuga Sree, 9/2/2025, 2:43 AM
Email Template
Send Test and Verify Merge Fields
Breaking news
'She's gone, yet...
Search
ENG
IN
15:13
02-09-2025


- Student
Invitation to Naan Mudhalvan
Home | Salesforce
Classic Email Templates | Salesf
orgfarm-d226808bab-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK000005hAZ7%3Fsetupid%3DCommunic...
Google Chrome isn't your default browser
Set as default
Search Setup
Setup
Home
Object Manager
email template
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
Text Email Template
Leave rejected
Help for this Page
Preview your email template below.
Email Template Detail
Edit
Delete
Clone
Email Templates from Salesforce
Unfiled Public Classic Email Templates
Email Template Name
Leave rejected
Available For Use
Template Unique Name
Leave_rejected
Last Used Date
Encoding
Unicode (UTF-8)
Times Used
Author
Shanmuga Sree [Change]
Description
Created By
Shanmuga Sree, 9/2/2025, 2:44 AM
Modified By
Shanmuga Sree, 9/2/2025, 2:44 AM
Email Template
Send Test and Verify Merge Fields
ICICIBANK
-1.13%
Search
ENG
IN
15:14
02-09-2025





8. APPROVAL PROCESSES PHASE

Approval processes streamline authorization for important actions, like approving lease agreements or payment adjustments. They define a sequence of

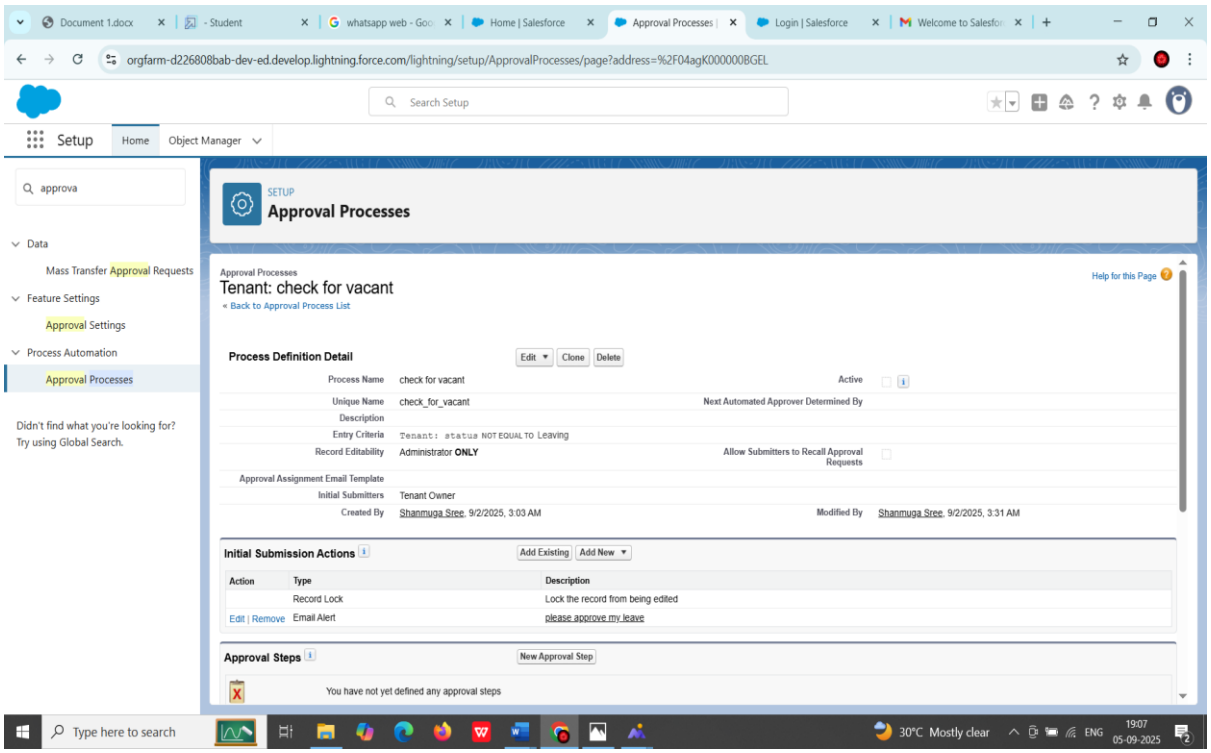
steps where managers or admins must review and approve requests before they are finalized, ensuring compliance and control.

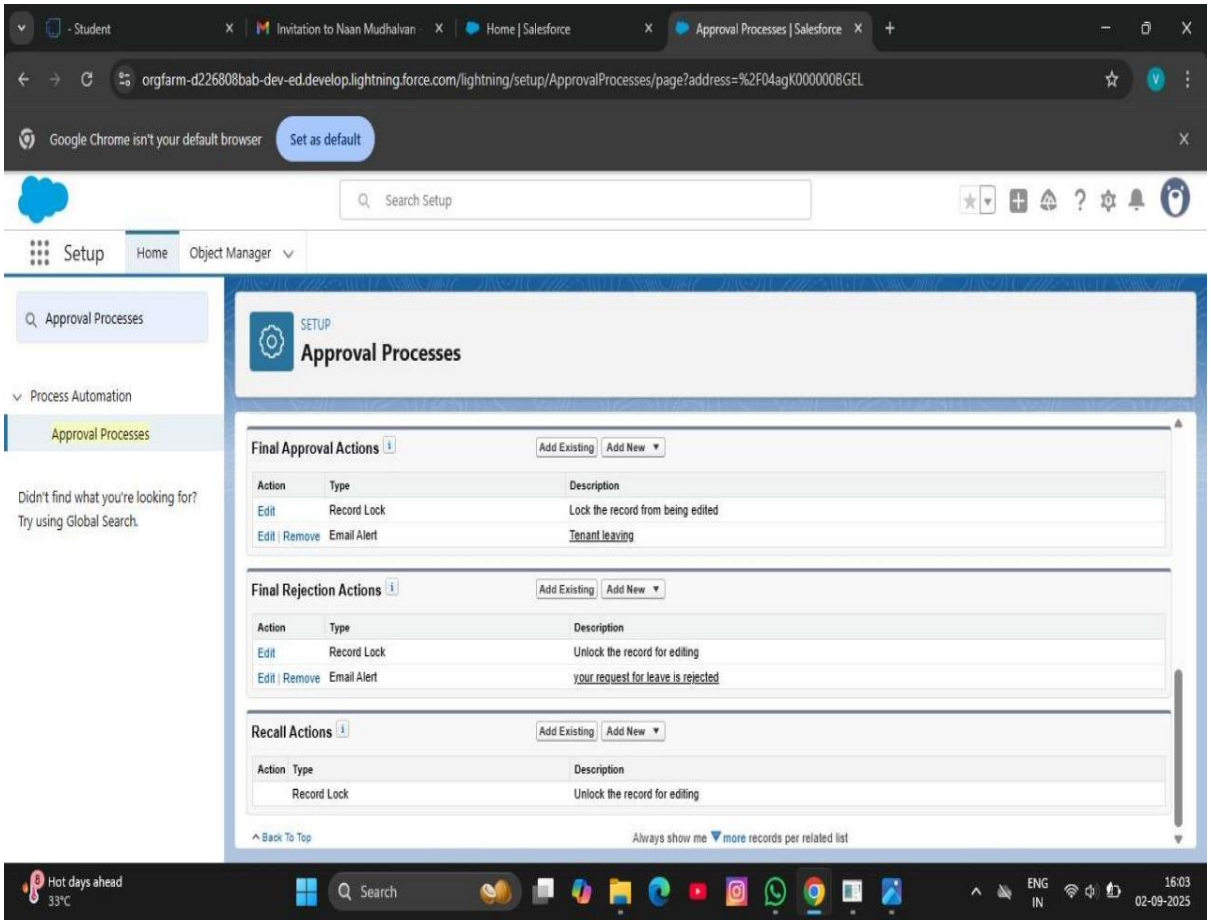
8.1 Create Approval Process For check for vacant

8.2 Initial Submission Action

8.3 Final Approval Action

8.4 Final Rejection Action





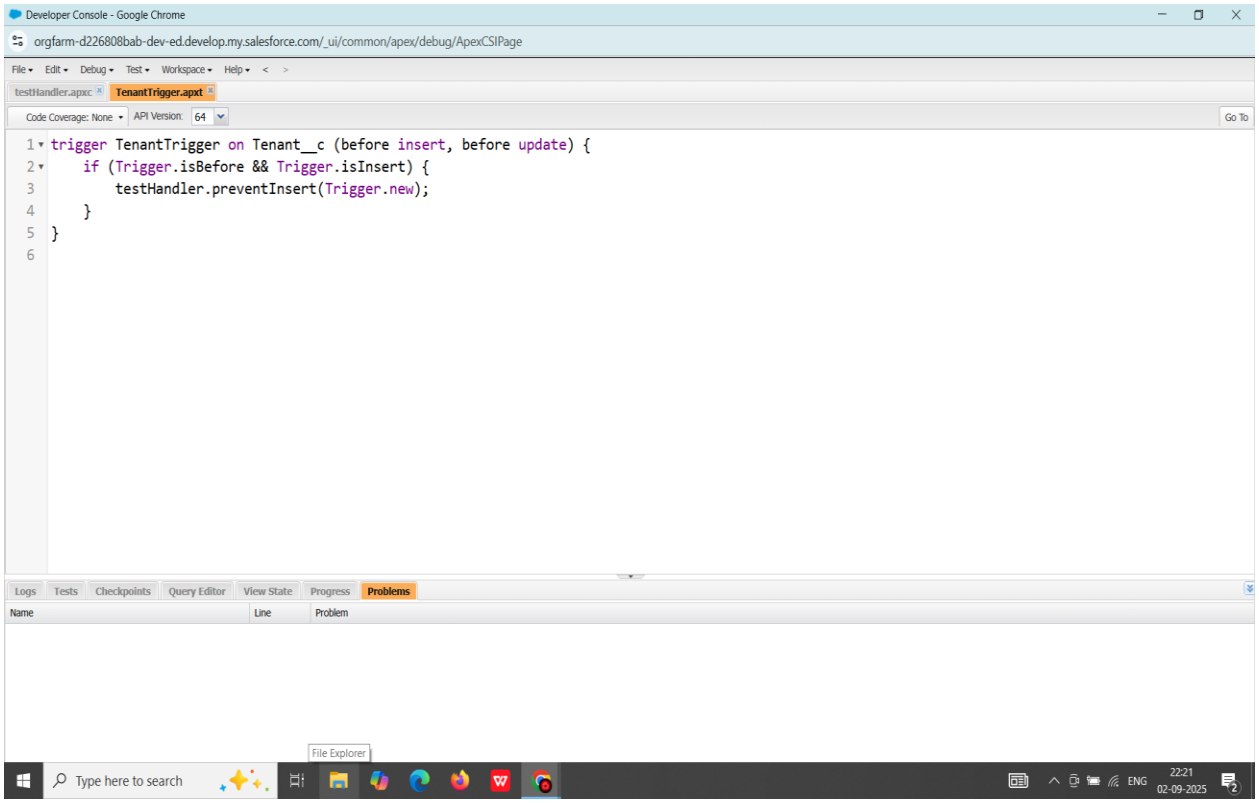
- Initial submission actions are the actions that occur when a user first submits a record for approval.
- Final Approval actions are the actions that occur when a record is approved from all the approval steps.
- Final Rejection actions are the actions that occur when a record is rejected from any of the approval steps.

9. APEX TRIGGER PHASE

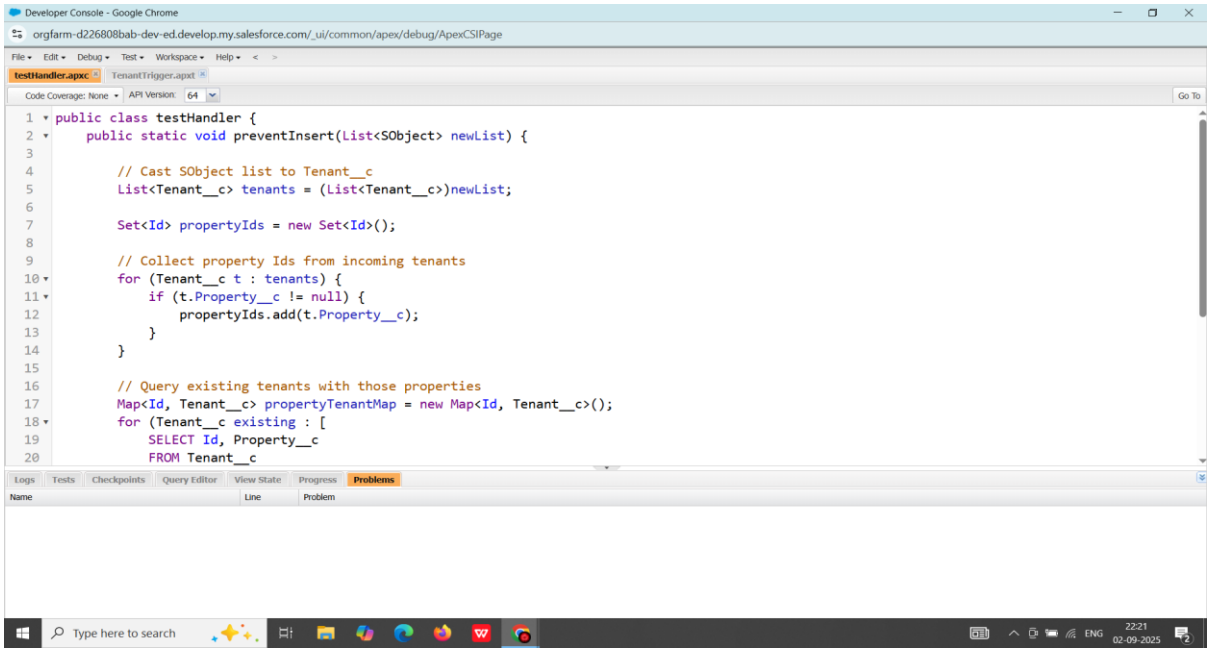
☐ **To create an Apex Trigger:**

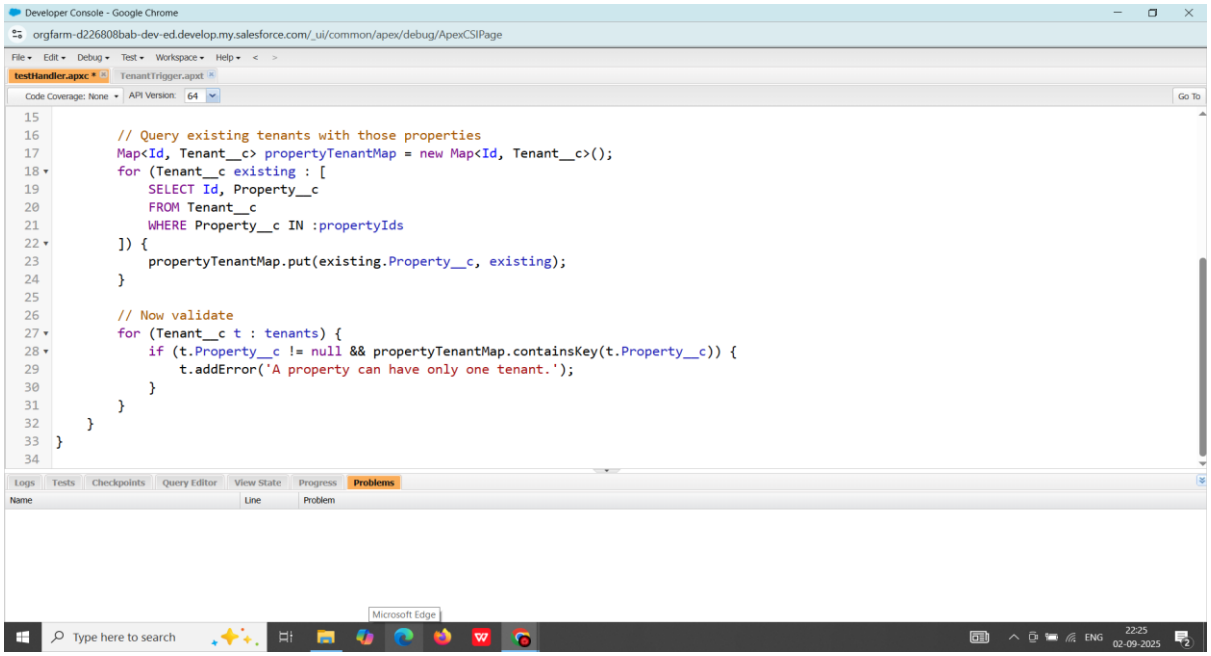
From the developer console >> Click on the file > New > Apex Trigger.

9.1 Creating an Apex Trigger

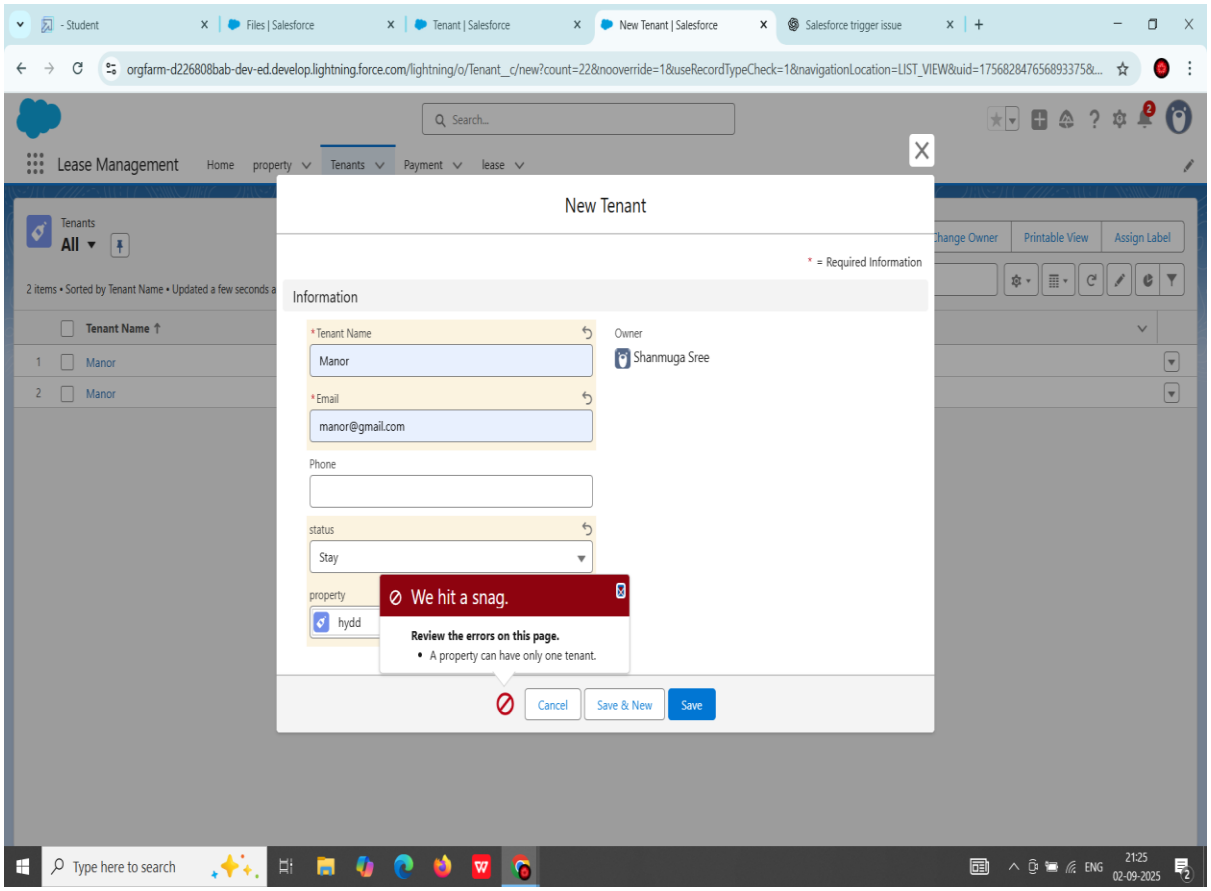


9.2 Creating an Apex Handler class





9.3 Testing the Trigger



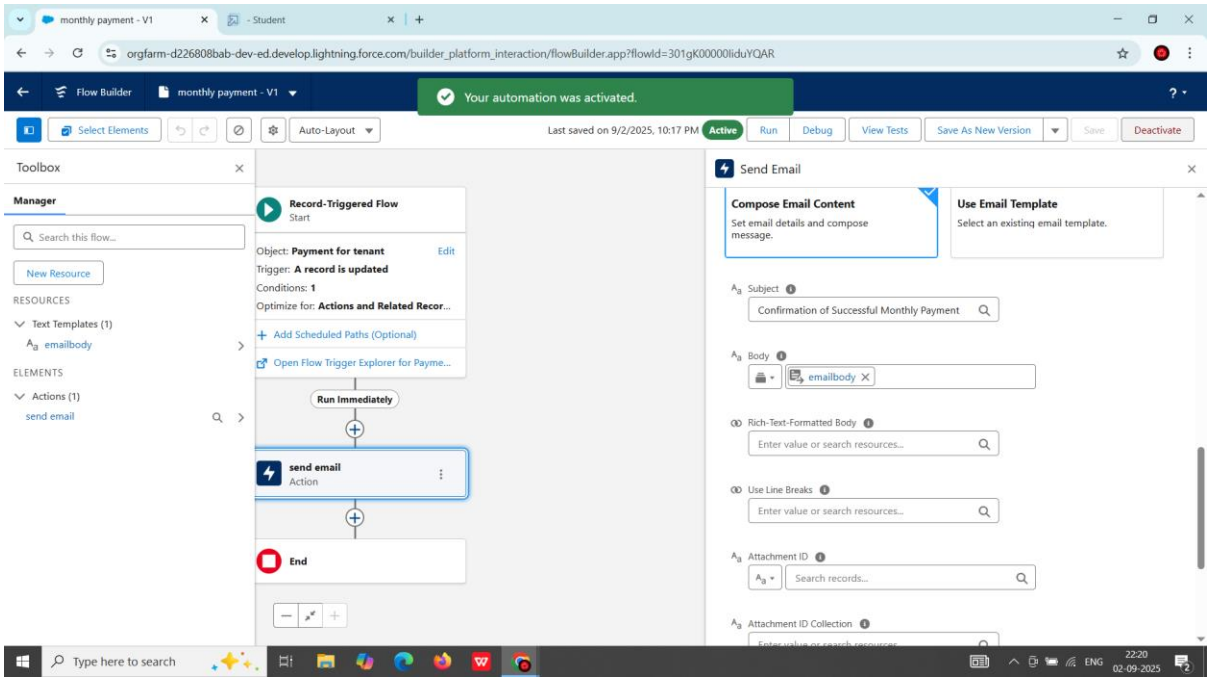
- The trigger ensures that a tenant can be associated with only one property, preventing multiple property allocations for the same tenant.

10. FLOW CREATION PHASE

☐ Creating Flow for Monthly Payment

To Create Flow:

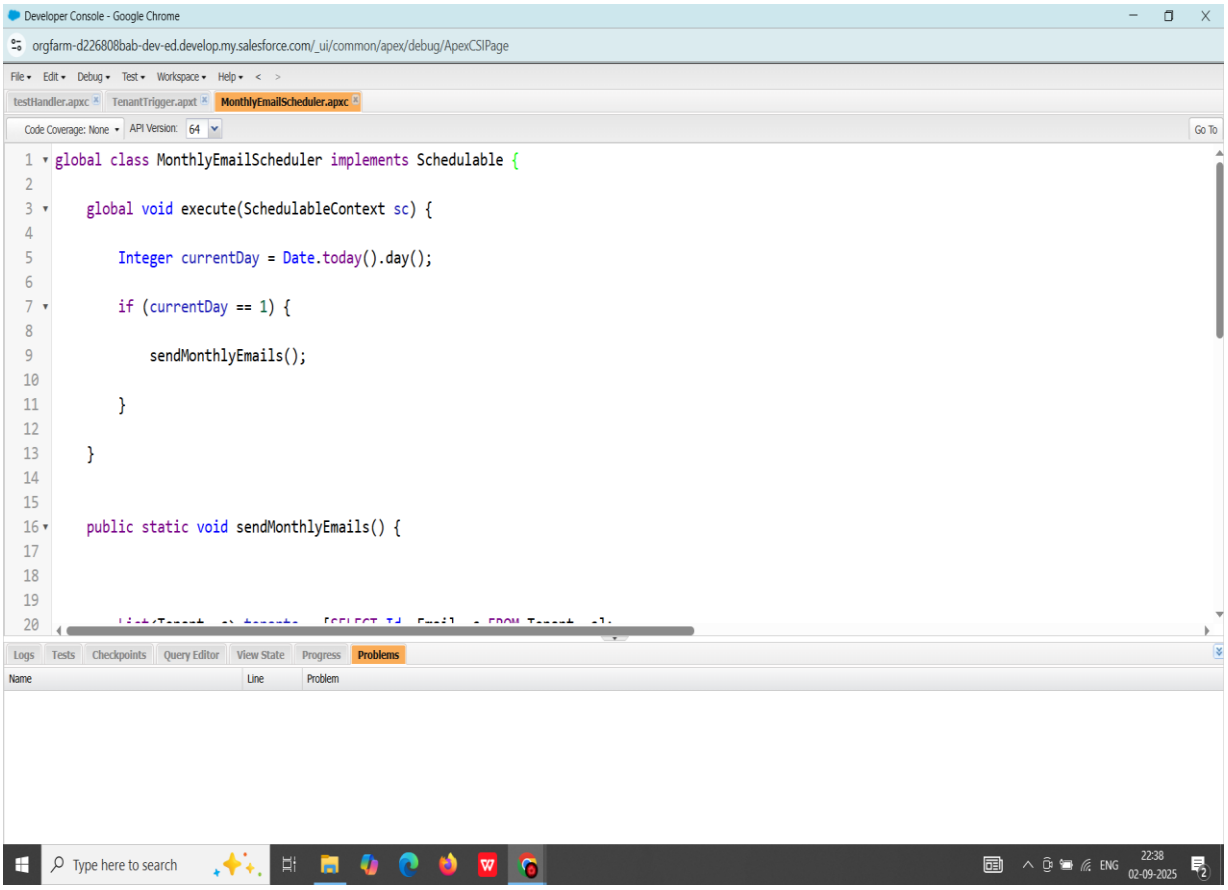
- From the setup page > **type Flow in Quick Find bar >** click on flow **>** New flow.

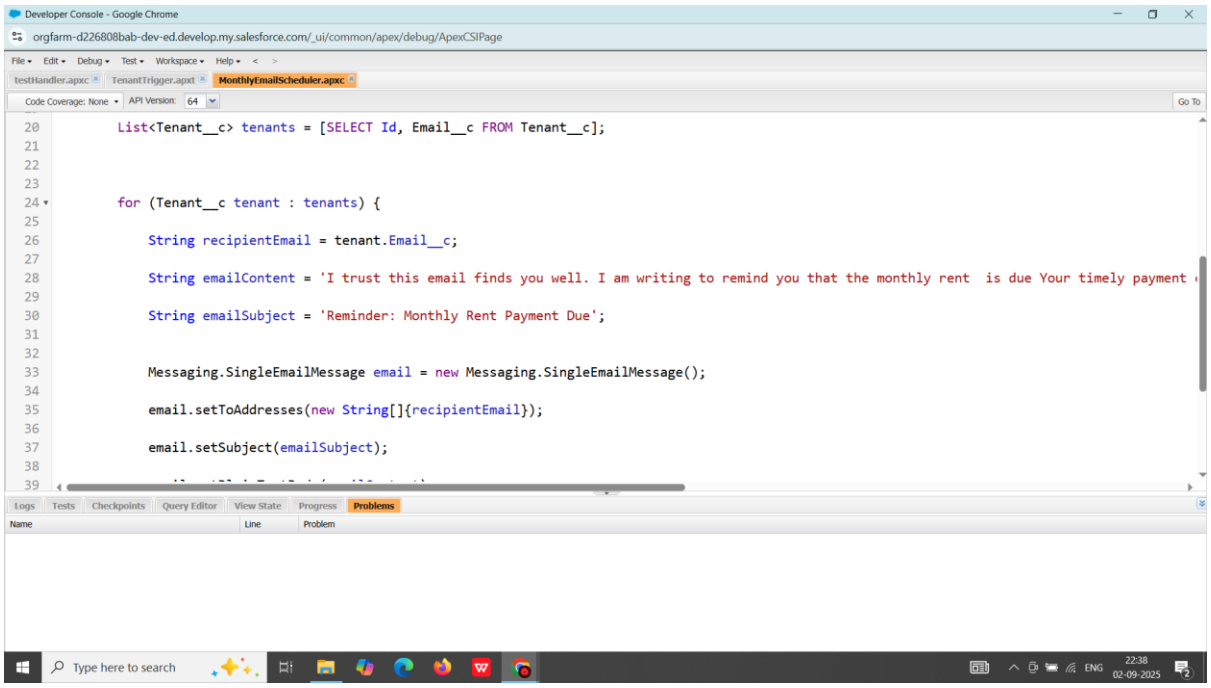


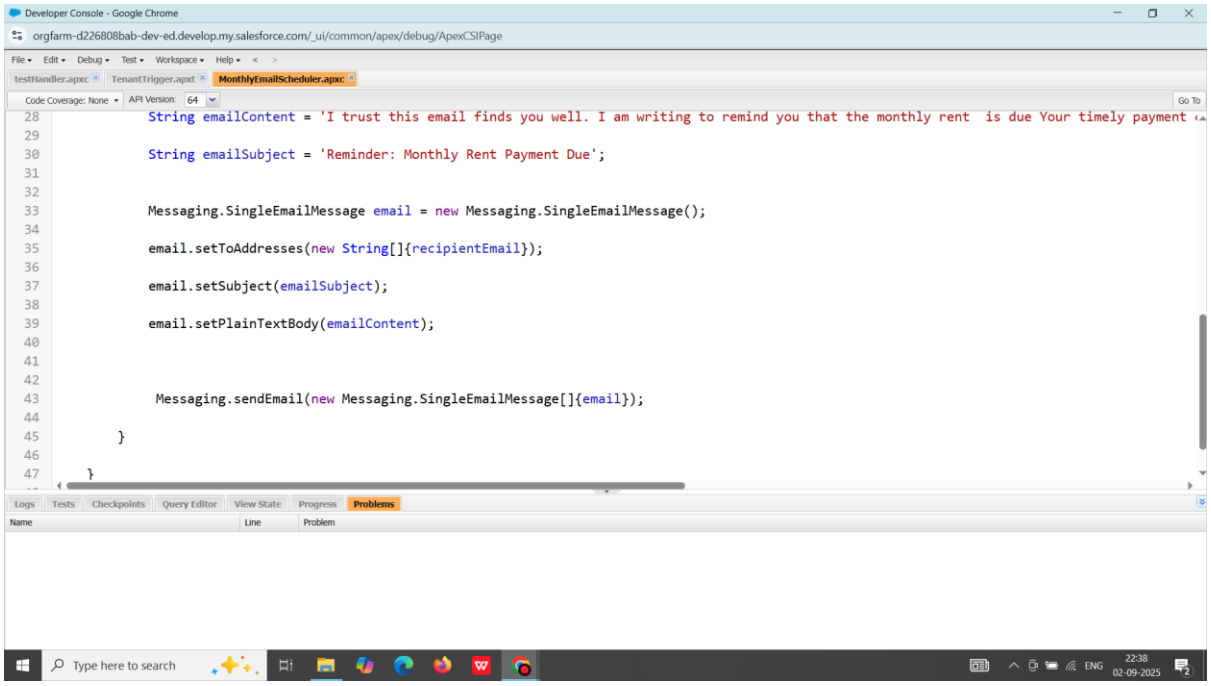
- The flow records monthly payments automatically. It also reminds tenants about due payments on time.
- Flows automate repetitive tasks such as monthly payment reminders, updating records, or sending email notifications.

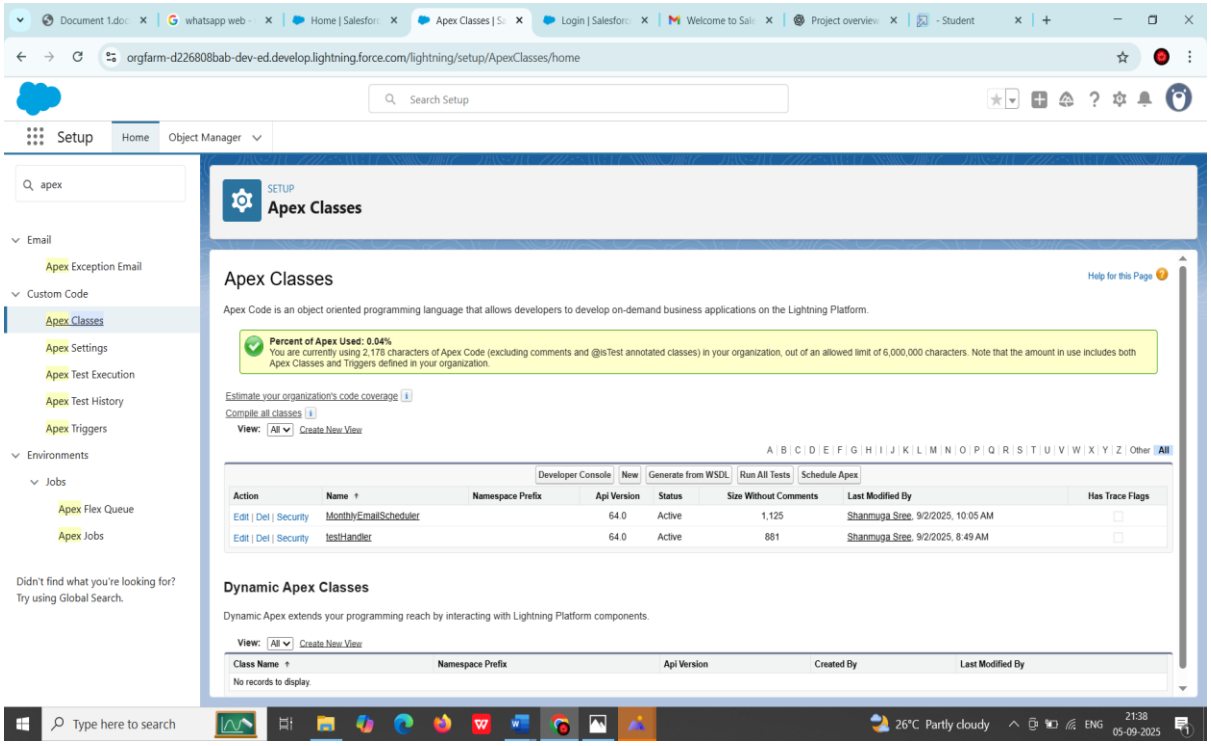
11. SCHEDULE CLASS PHASE

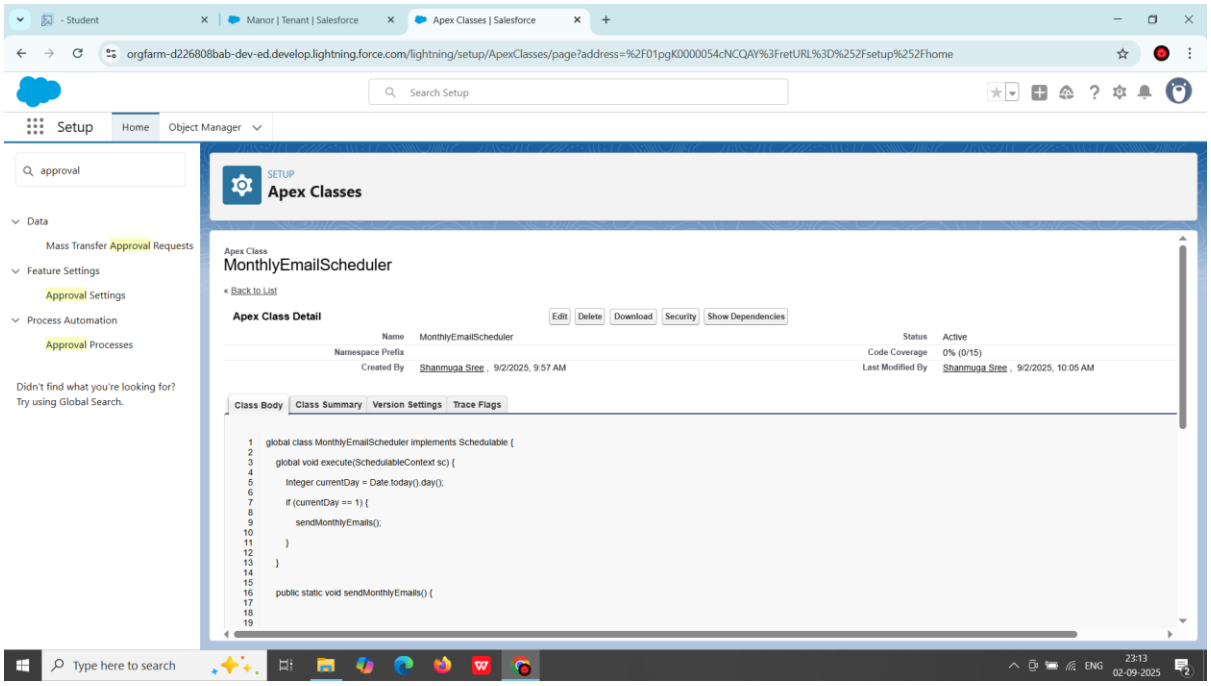
- **Creating Apex Class**
- **Creating Schedule Apex Class**


Developer Console - Google Chrome
orgfarm-d226808bab-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
testHandler.apxc TenantTrigger.apxt MonthlyEmailScheduler.apxc
Code Coverage: None API Version: 64 Go To
1 global class MonthlyEmailScheduler implements Schedulable {
2
3 global void execute(SchedulableContext sc) {
4
5 Integer currentDay = Date.today().day();
6
7 if (currentDay == 1) {
8
9 sendMonthlyEmails();
10
11 }
12
13 }
14
15
16 public static void sendMonthlyEmails() {
17
18
19
20
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
Type here to search
ENG
22:38
02-09-2025


Developer Console - Google Chrome
orgfarm-d226808bab-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
testHandler.apxc TenantTrigger.apxt MonthlyEmailScheduler.apxc
Code Coverage: None API Version: 64 Go To
20 List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
21
22
23
24 for (Tenant__c tenant : tenants) {
25
26 String recipientEmail = tenant.Email__c;
27
28 String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment
29
30 String emailSubject = 'Reminder: Monthly Rent Payment Due';
31
32
33 Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
34
35 email.setToAddresses(new String[]{recipientEmail});
36
37 email.setSubject(emailSubject);
38
39
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
Type here to search
ENG
22:38
02-09-2025


Developer Console - Google Chrome
orgfarm-d226808bab-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
testHandler.apxc
TenantTrigger.apxt
MonthlyEmailScheduler.apxc
String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment
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});
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem
Type here to search
ENG
22:38
02-09-2025

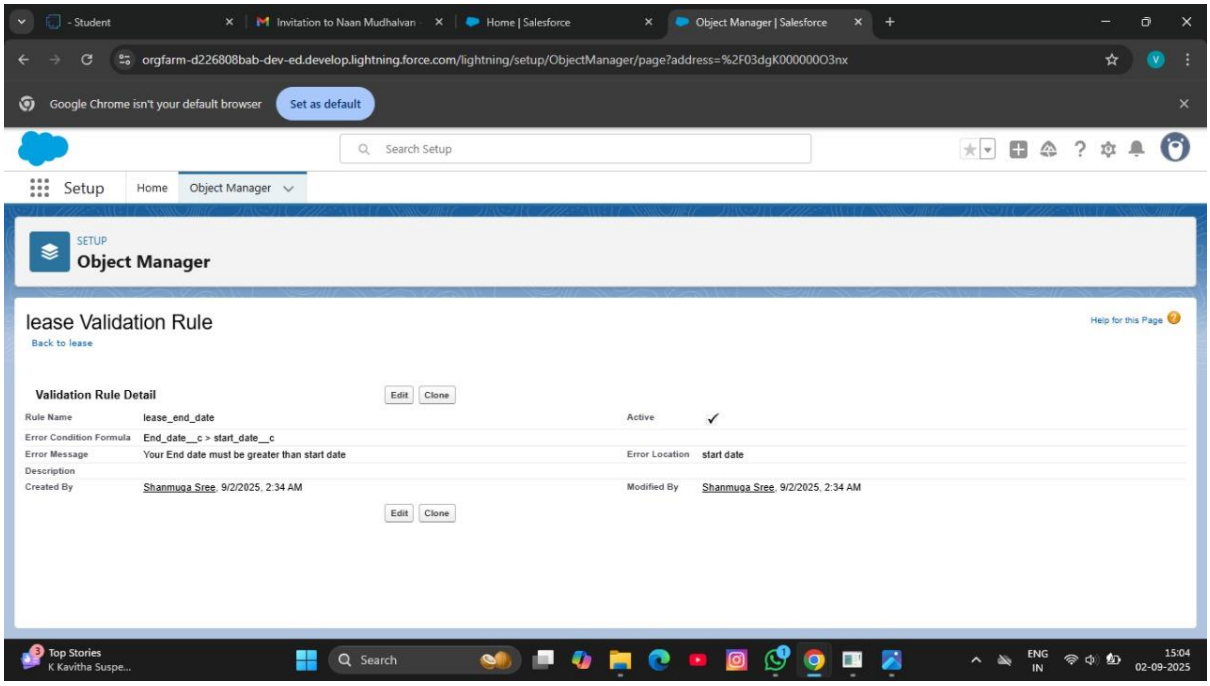

orgfarm-d226808bab-dev-ed.develop.lightning.force.com/lightning/setup/ApexClasses/home
Search Setup
Setup
Home
Object Manager
apex
Email
Apex Exception Email
Custom Code
Apex Classes
Apex Settings
Apex Test Execution
Apex Test History
Apex Triggers
Environments
Jobs
Apex Flex Queue
Apex Jobs
Didn't find what you're looking for?
Try using Global Search.
SETUP
Apex Classes
Apex Classes
Help for this Page
Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.
Percent of Apex Used: 0.04%
You are currently using 2,178 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.
Estimate your organization's code coverage
Compile all classes
View: All Create New View
Developer Console
New
Generate from WSDL
Run All Tests
Schedule Apex
Action
Name
Namespace Prefix
Api Version
Status
Size Without Comments
Last Modified By
Has Trace Flags
Edit | Del | Security
MonthlyEmailScheduler
64.0
Active
1,125
Shanmuga Sree, 9/2/2025, 10:05 AM
Edit | Del | Security
testHandler
64.0
Active
881
Shanmuga Sree, 9/2/2025, 8:49 AM
Dynamic Apex Classes
Dynamic Apex extends your programming reach by interacting with Lightning Platform components.
View: All Create New View
Class Name
Namespace Prefix
Api Version
Created By
Last Modified By
No records to display.
Type here to search
26°C Partly cloudy
ENG
21:38
05-09-2025



TESTING APPROACH

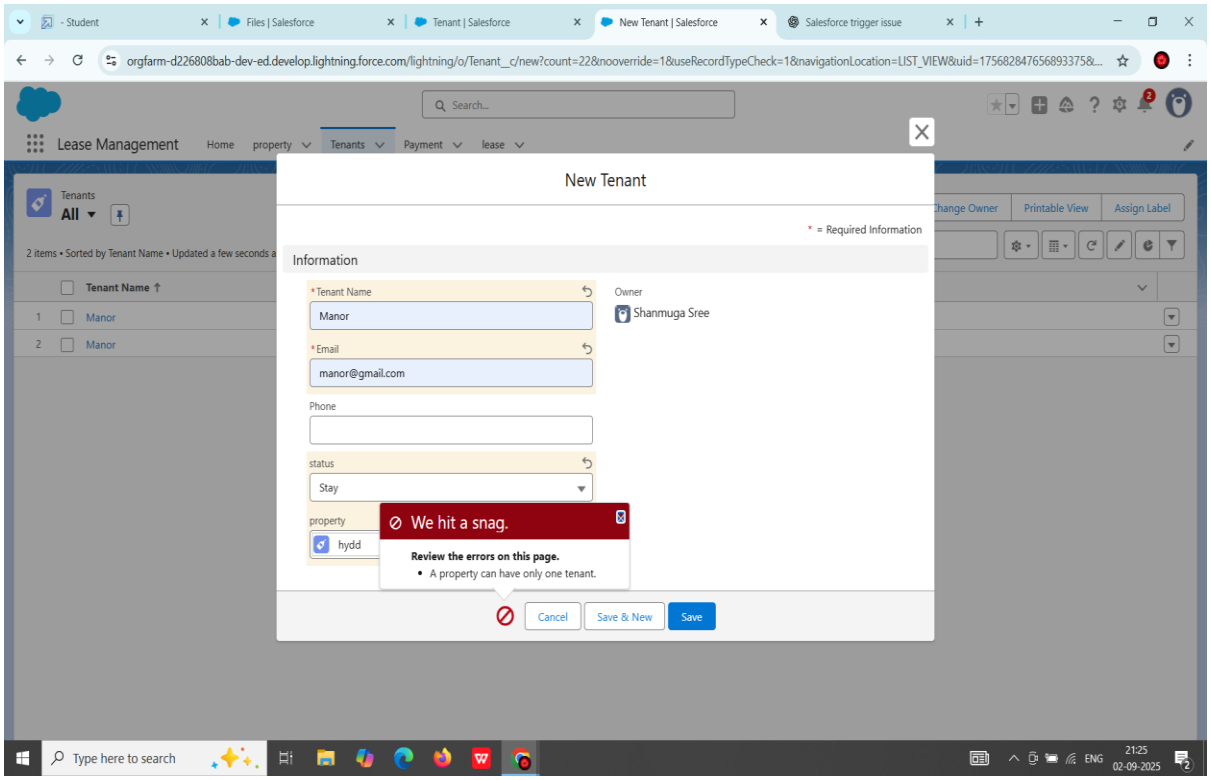
☐ This project was tested mainly for validation rules, approval processes, flows, triggers, and reports. The testing ensured that data is accurate, automation works correctly, and system behavior meets requirements. Issues (snags) found during testing were corrected to improve reliability.

- **Functional Testing:**



- The validation rule was tested to ensure correct data entry. In this example, the system successfully prevented saving a record if the lease end date was earlier than the start date, proving that functional testing was achieved.

- **Performance Testing:**



- It verified that system responses were quick, and reports generated without delay for sample data, which proved that performance testing was achieved.
- During testing, an Apex trigger was executed to check business logic. At this stage, a snag was encountered, showing that the validation correctly restricted invalid data entry. This confirms that the system behaves as expected under rule conditions.

FUTURE ENHANCEMENTS

- ☐ **Mobile App Integration:** Provide mobile access for tenants and property owners.
- ☐ **Online Payment Gateway:** Enable secure digital rent and lease payments.

- **AI Suggestions:** Use AI to recommend lease renewals, detect risks, and give smart insights.
- **Chatbot Integration:** Add a chatbot for quick tenant and landlord support.
- **Advanced Reporting:** Build interactive dashboards for better decision-making.

CONCLUSION

This project shows how Salesforce tools like custom objects, validation rules, flows, and triggers can make lease management easier. Using Salesforce, the system became more accurate, faster, and ready to handle future needs.