

TITLE: LEASE MANAGEMENT

Date	07 NOVEMBER 2025
Team ID	NM2025TMID07560
Project Name	Lease Management

Category: Salesforce Developer

Skills Required:

Salesforce Admin, Salesforce Developer

Project Description:

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

Phase 1: Ideation

Problem Statement:

In real estate and asset management companies, handling lease agreements, tenant details, payment schedules, and renewals manually often leads to inefficiency, data loss, and miscommunication. There is a need for a centralized digital platform that automates lease tracking, approval processes, and notifications to improve productivity and accuracy.

Proposed Solution:

The Lease Management System built on Salesforce provides a unified platform to manage leases, tenants, and property details efficiently. Using Salesforce automation tools like **Flows**, **Approval Processes**, and **Email Alerts**, the system ensures streamlined lease operations, reduces manual effort, and provides real-time insights through dashboards and reports.

Objectives:

- Digitize and automate the entire lease lifecycle—from creation to renewal.
- Maintain tenant, property, and payment records in one centralized system.
- Use Salesforce features (objects, triggers, and automation) to minimize manual tasks.
- Enable managers to approve lease requests through an automated approval flow.
- Send automatic email notifications for lease expirations and payment reminders.

Key Features / Functionalities:

- **Custom Objects:** Lease, Tenant, and Property with defined relationships.
- **Validation Rules:** Ensure correct lease terms and data integrity.
- **Approval Process:** Automate lease approval and renewal requests.
- **Email Alerts:** Notify stakeholders about upcoming payments and expirations.
- **Flow Automation:** Streamline lease creation and update processes.
- **Reports & Dashboards:** Track active leases, payment status, and renewals.

Expected Outcome:

A fully automated, user-friendly Salesforce app that simplifies lease management operations, enhances transparency, and provides real-time insights for better decision-making.

Phase 2: Requirement Analysis

Milestone 1: Salesforce Account

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?"

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

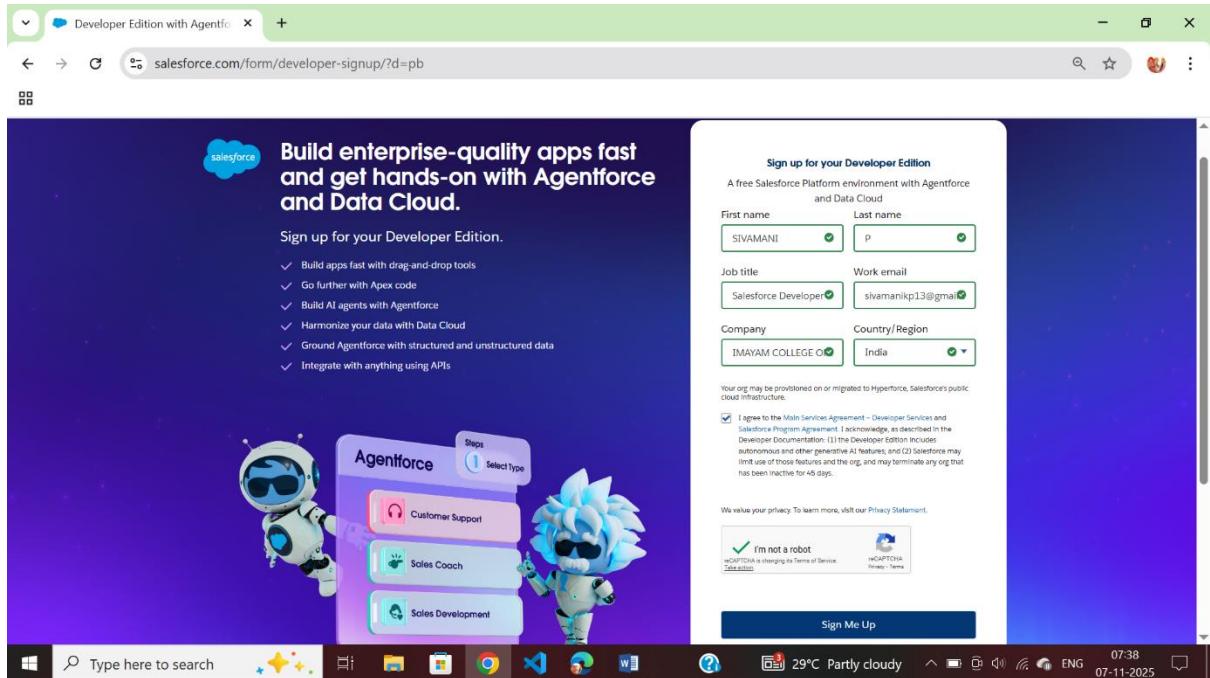
Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

<https://youtu.be/r9EX3lGde5k>

Activity 1: Creating Developer Account

Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :



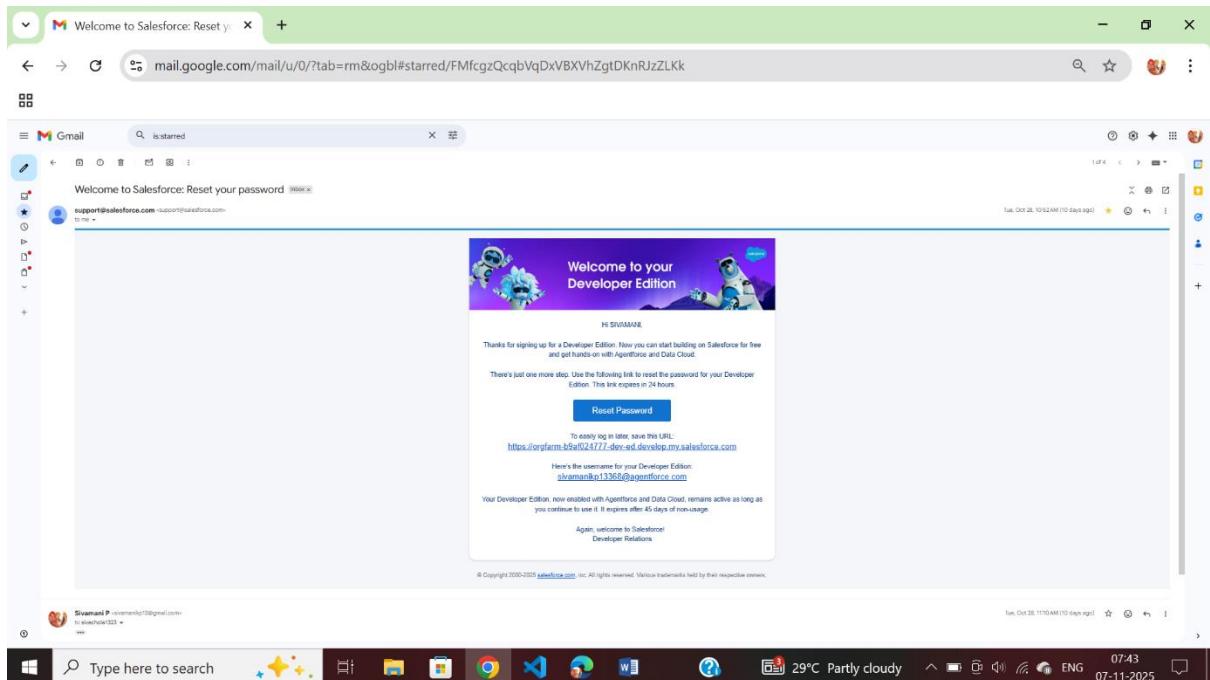
1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format:
username@organization.com

Click on sign me up after filling these.

Activity 2: Account Activation:

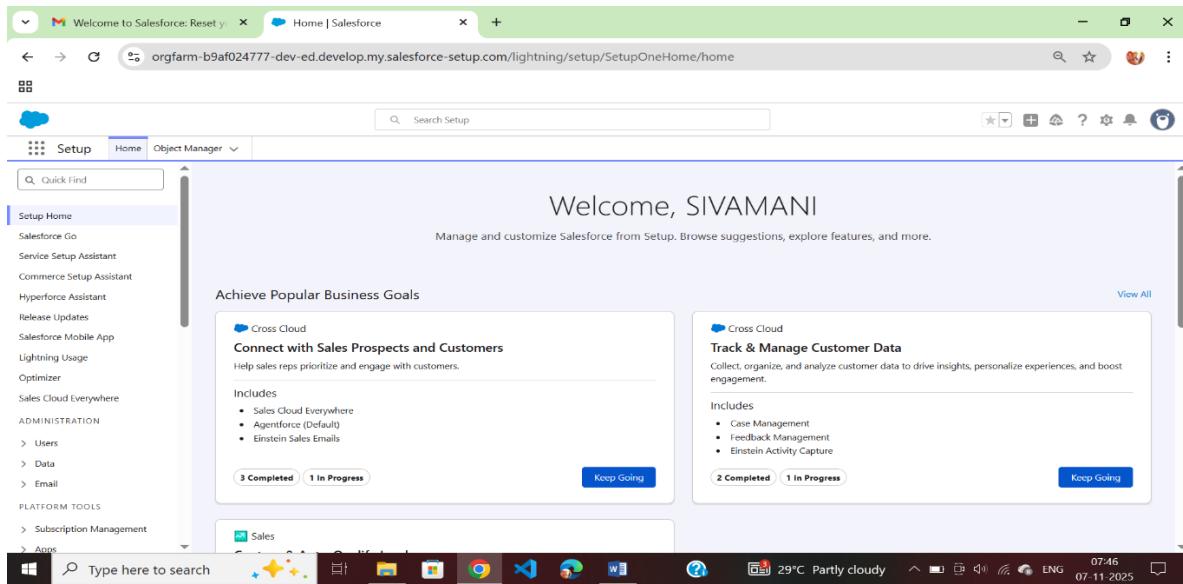
1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



2.

The screenshot shows a "Change Your Password" form. It prompts the user to enter a new password for "lead@sb.oom" and specifies that it must be at least 8 characters long, containing 1 letter and 1 number. The "New Password" field contains "....." and is labeled "Good". The "Confirm New Password" field contains "....." and is labeled "Match". Below these are "Security Question" and "Answer" fields. The "Security Question" dropdown is set to "In what city were you born?". The "Answer" field contains "asdfghjkl". At the bottom is a blue "Change Password" button.

1. Click on Reset Password
2. Give a password and answer a security question and click on change password.
3. Give a password and answer a security question and click on change password.
4. Then you will redirect to your salesforce setup page.



Milestone 2: Object

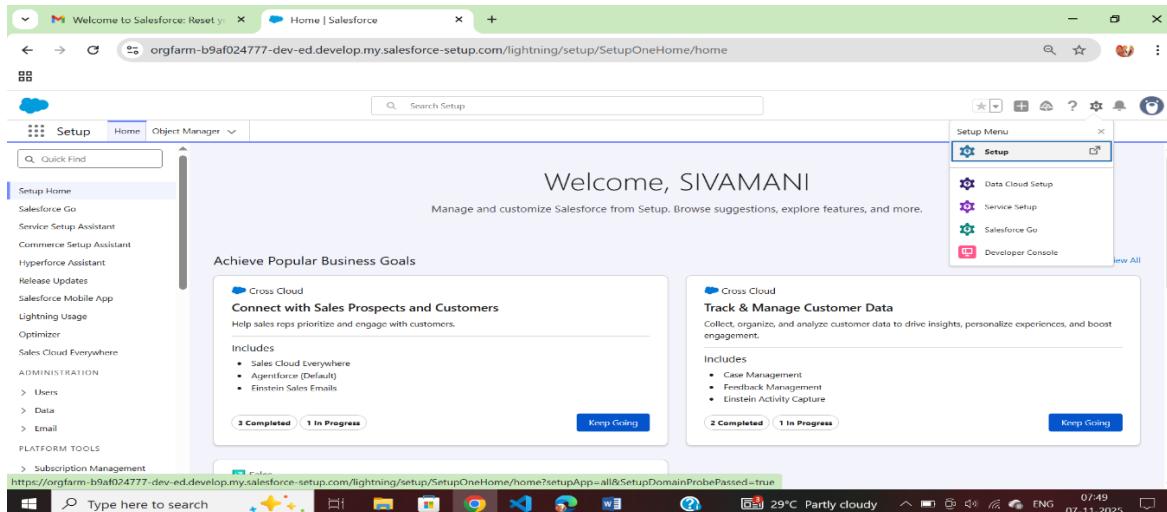
What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects

Salesforce objects are of two types:

1. Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

To Navigate to Setup page:



To create an object:

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.

2. On Custom object defining page:
3. Enter the label name, plural label name, click on Allow reports, Allow search.

4. Click on Save.

Activity 1: Create Property Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name>> property
3. Plural label name>> property
4. Enter Record Name Label and Format
5. Record Name >>property Name
6. Data Type >> Text
7. Click on Allow reports and Track Field History,Allow Activities
8. Allow search >> Save.

The screenshot shows the Salesforce Setup interface with the following details:

- Custom Object Information:**
 - Label:** property (Example: Account)
 - Plural Label:** property (Example: Accounts)
 - Starts with vowel sound:**
 - Object Name:** property (Example: Account)
 - Description:** (Empty text area)
 - Context-Sensitive Help Setting:**
 - Open the standard Salesforce.com Help & Training window
 - Open a window using a Visualforce page
 - Content Name:** None
- Enter Record Name Label and Format:**
 - Record Name:** property Name (Example: Account Name)
 - Data Type:** Text
 - Warning:** If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

The screenshot shows the Salesforce Setup interface with the following details:

- Optional Features:**
 - Allow Reports
 - Allow Activities
 - Track Field History
 - Allow in Chatter Groups
 - Enable Licensing
- Object Classification:**

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#)

 - Allow Sharing
 - Allow Bulk API Access
 - Allow Streaming API Access
- Deployment Status:**
 - In Development
 - Deployed
- Search Status:**

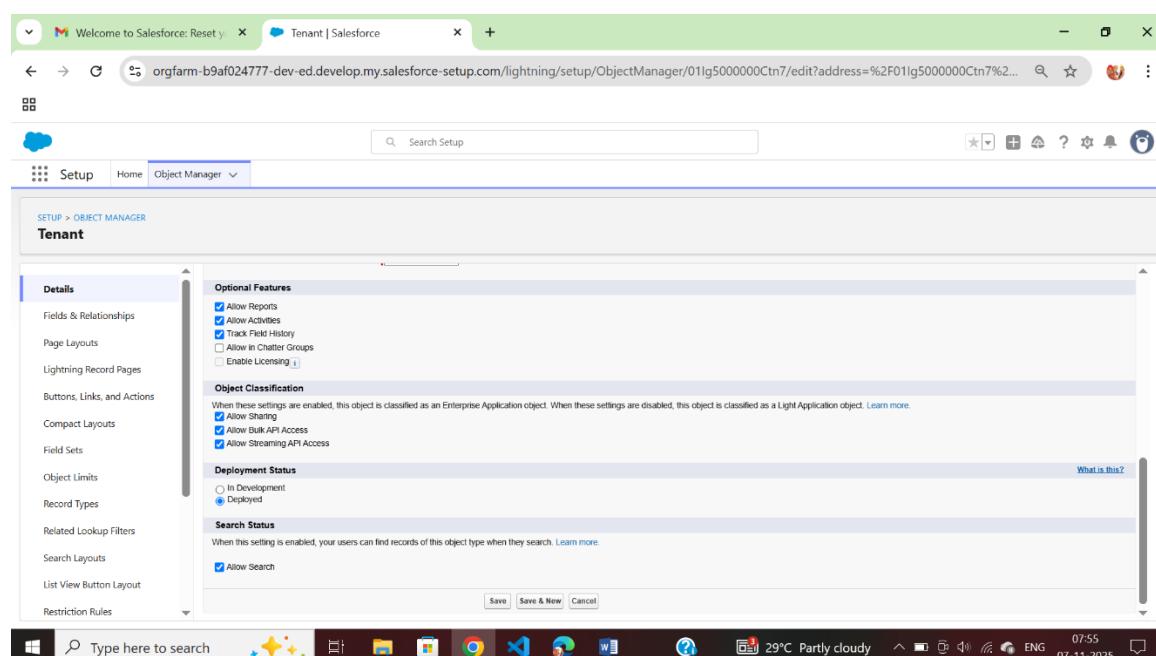
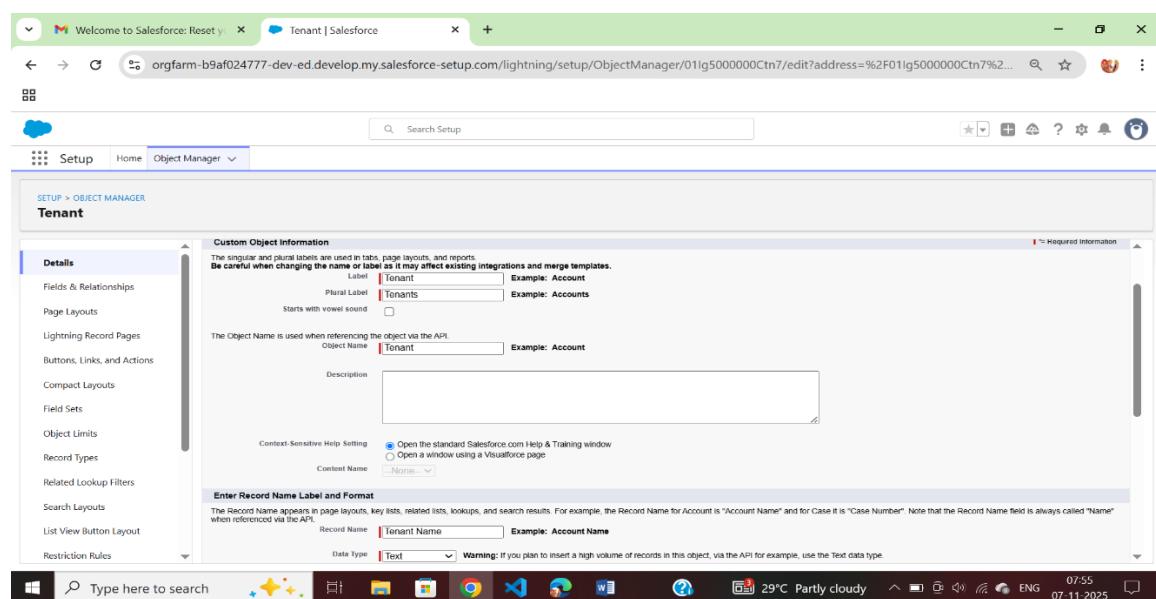
When this setting is enabled, your users can find records of this object type when they search. [Learn more](#).

 - Allow Search

Activity 2: Create Tenant Object:

To create an object:

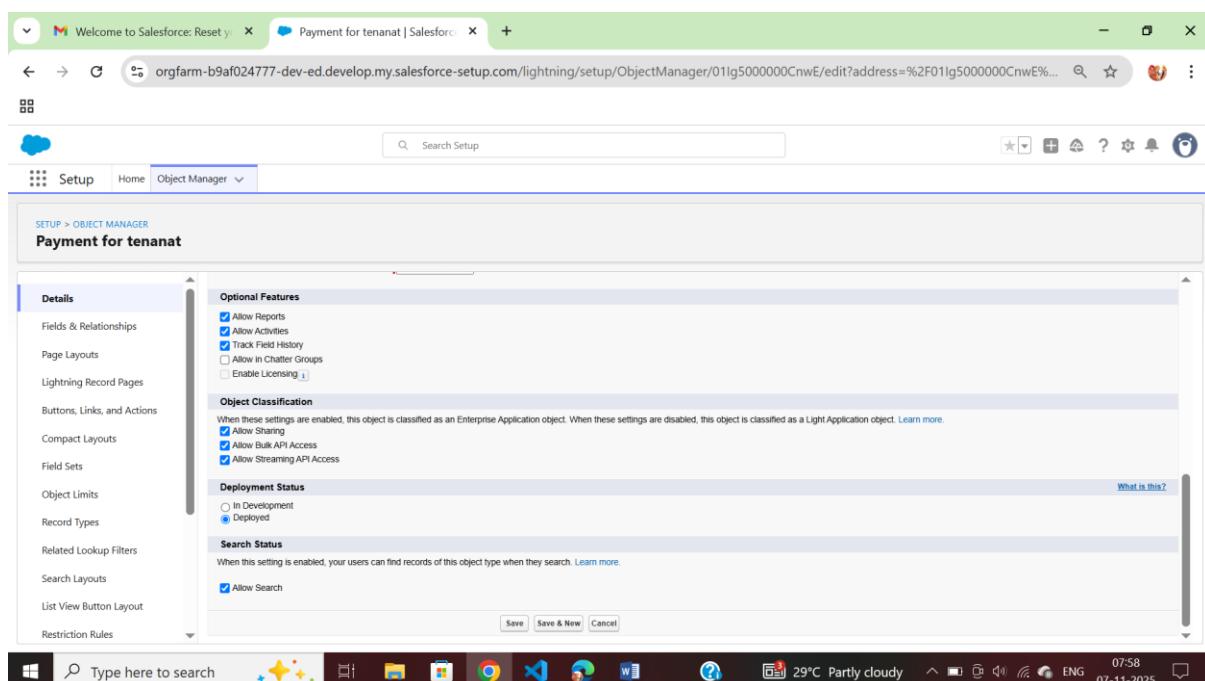
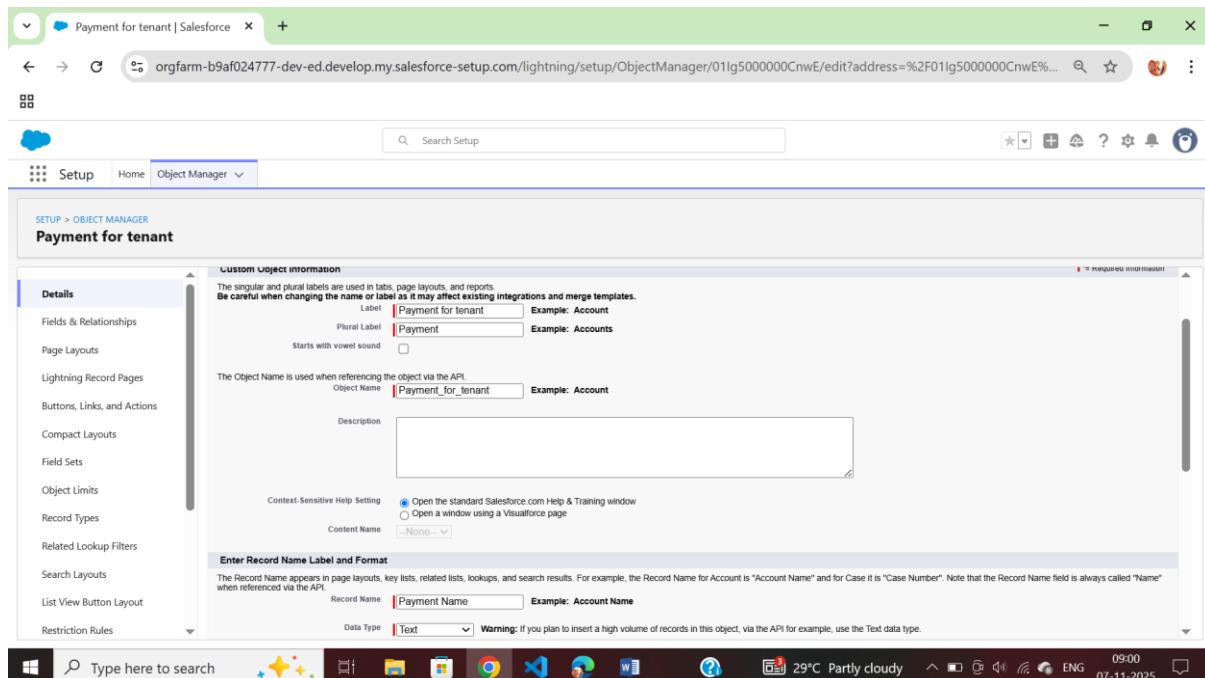
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Tenant
3. Plural label name >> Tenants
4. Enter Record Name Label and Format
5. Record Name >> Tenant Name
6. Data Type >> Text
7. Click on Allow reports and Track Field History, Allow Activities
8. Allow search >> Save.



Activity 3: Create Payment Object:

To create an object:

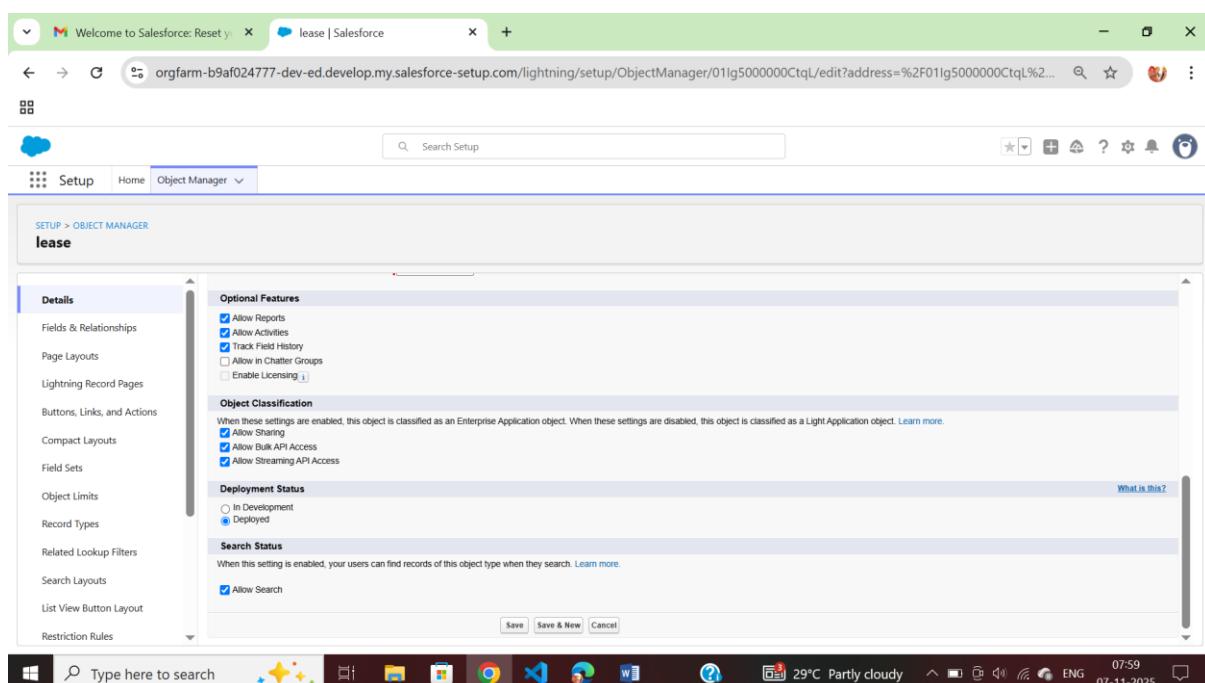
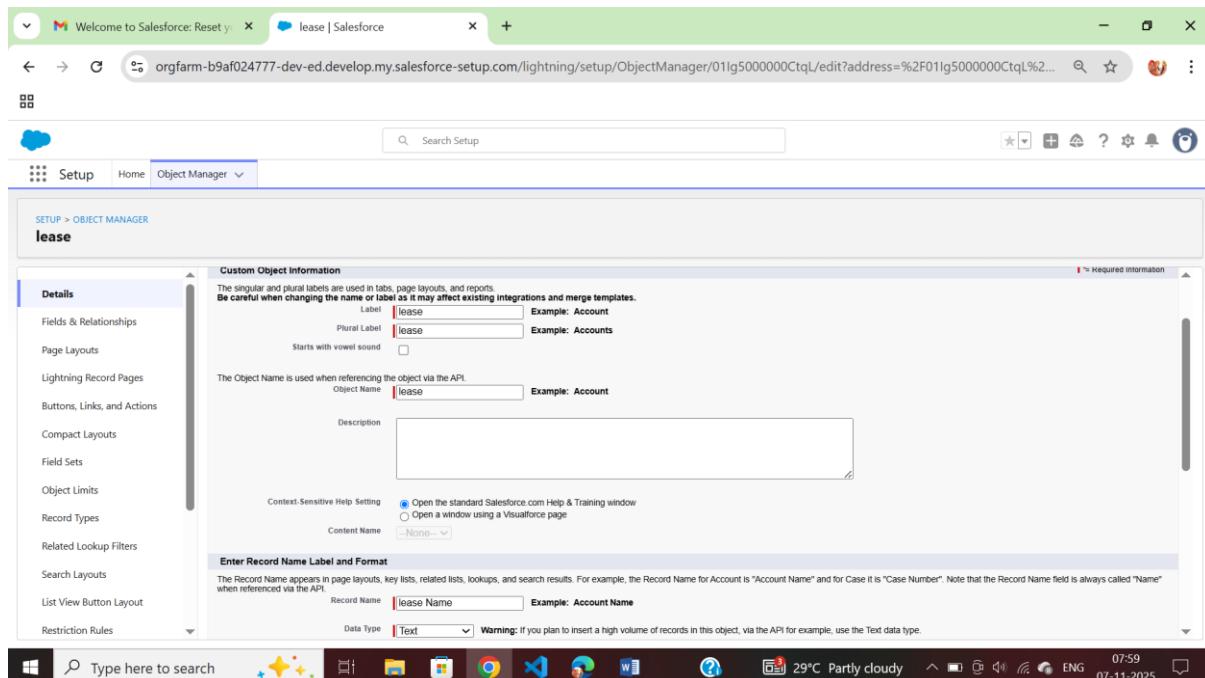
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name>> Payment for tenant
3. Plural label name>> Payment
4. Enter Record Name Label and Format
5. Record Name >> Payment Name
6. Data Type >> Text
7. Click on Allow reports and Track Field History,Allow Activities
8. Allow search >> Save.



Activity 4: Create Lease Object:

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name>> lease
3. Plural label name>> lease
4. Enter Record Name Label and Format
5. Record Name >> lease Name
6. Data Type >> Text
7. Click on Allow reports and Track Field History, Allow Activities
8. Allow search >> Save.



Phase 3: Project Design

Milestone 1: Tabs

What is Tab: A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Web Tabs

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

5. Lightning Page Tabs

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs.

Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

Activity 1: Create A Custom Tab:

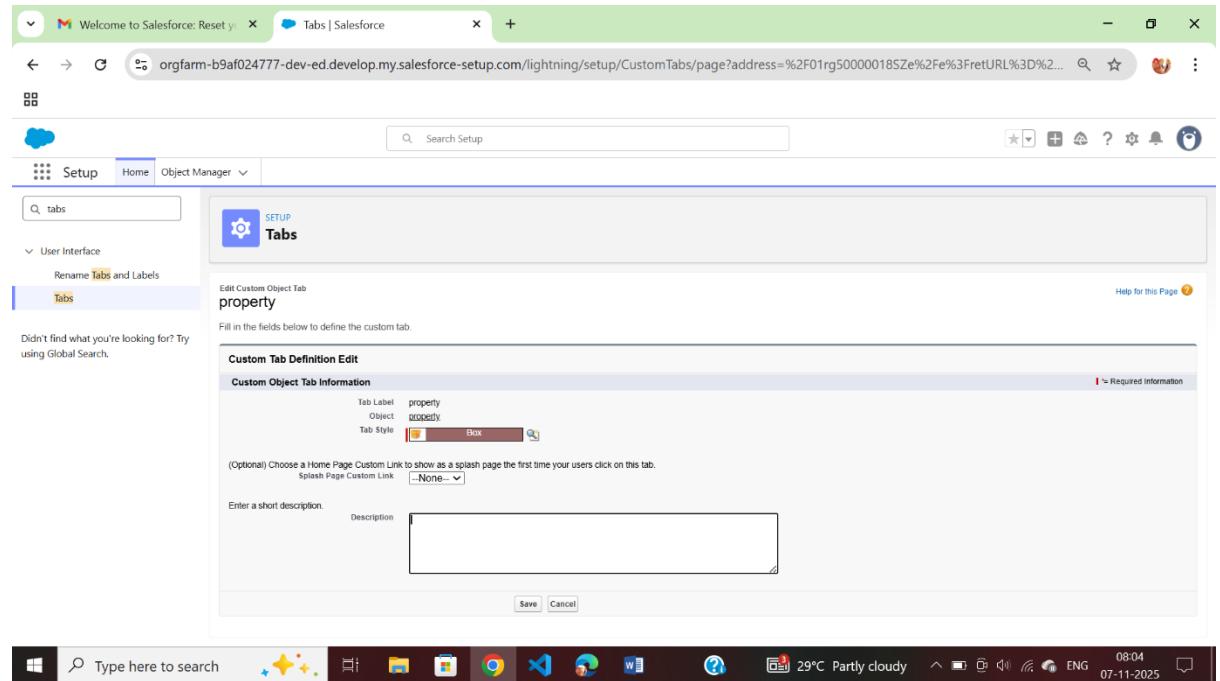
To create a Tab:(Property)

1. Go to setup page >>type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)

Action	Label	Tab Style	Description
Edit Del	Lease	Laptop	
Edit Del	Payment	Credit card	
Edit Del	property	Box	
Edit Del	Tenants	Building Block	

2. Select Object(property) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

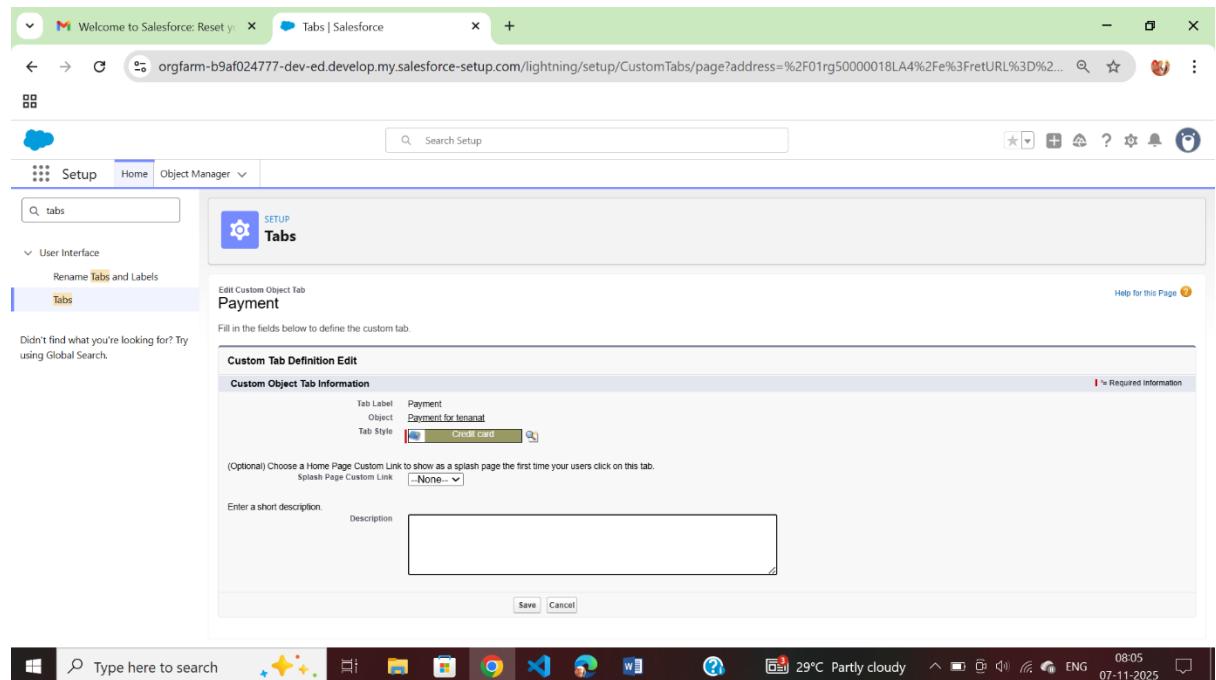
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save.



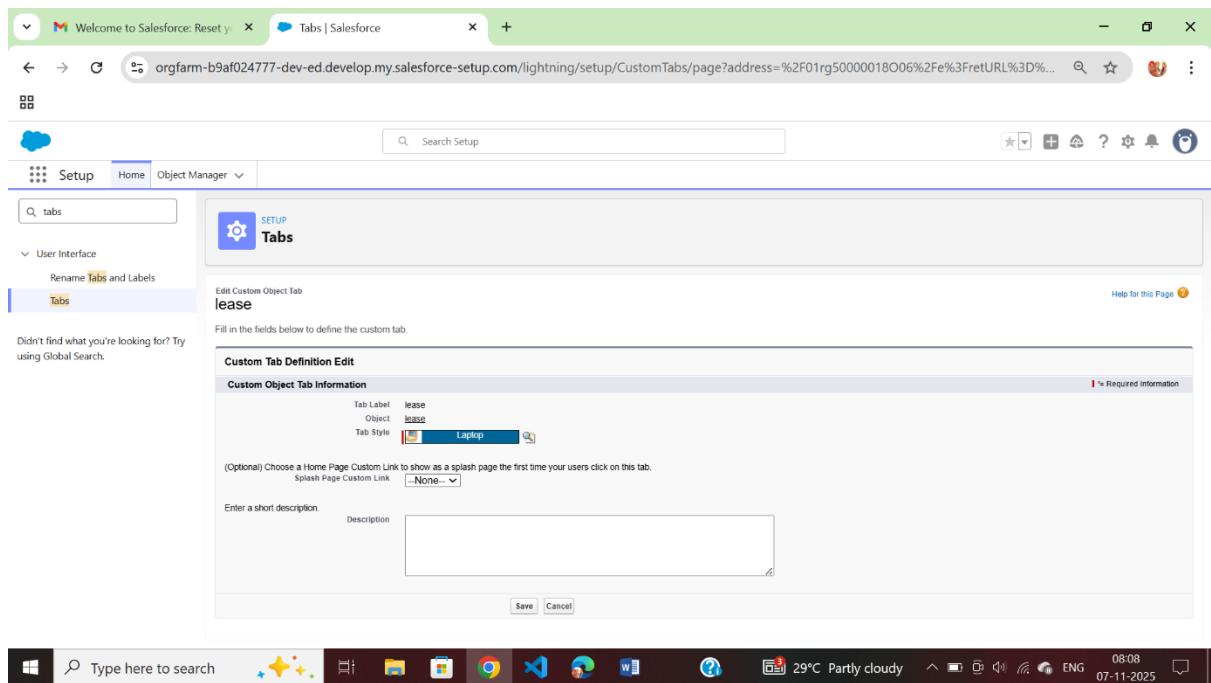
Activity 2: Create Remaining Tabs:

1. Now create the Tabs for the remaining Objects, they are “Payment for tenant, lease, tenant”.
2. Follow the same steps as mentioned in Activity - 1.

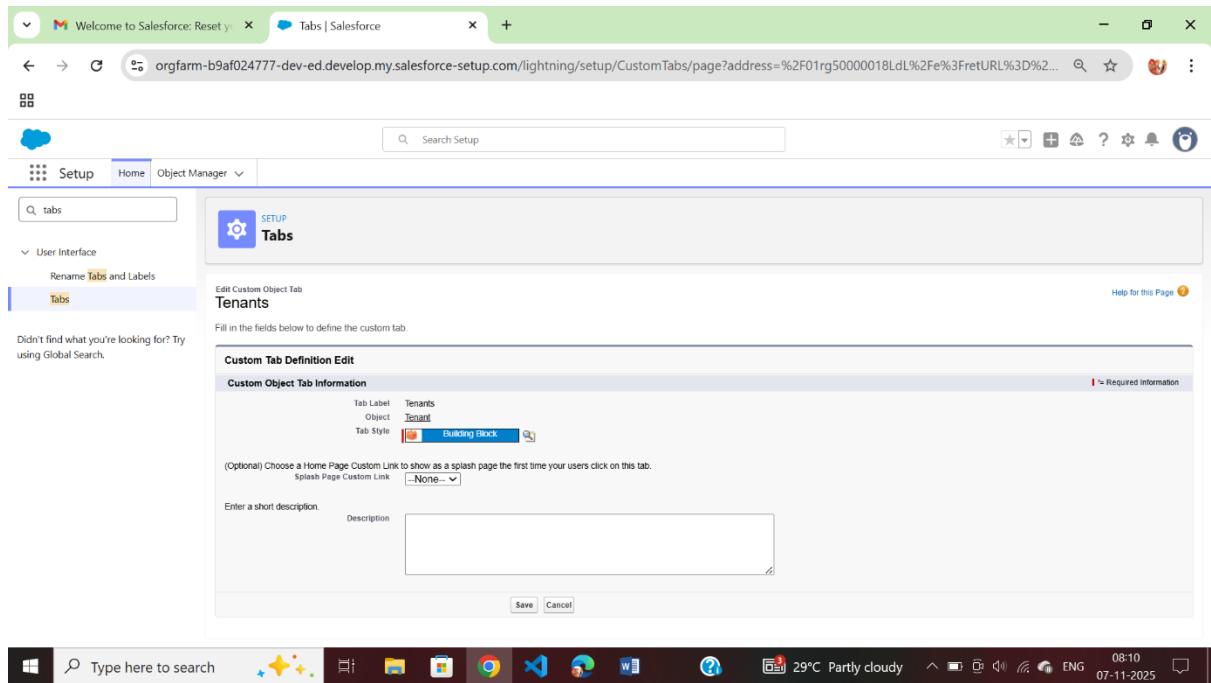
Tab (Payment for tenant)



Tab (lease)



Tab (tenant)



Milestone 2: The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

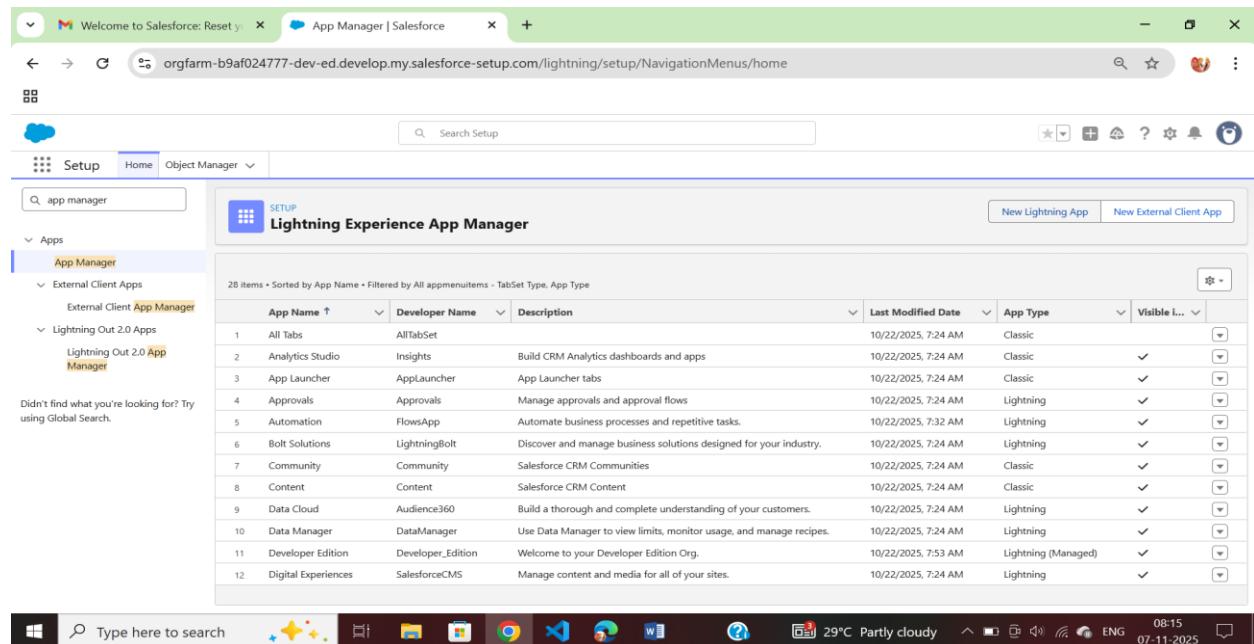
Use Case:

Well done you have reached close to your requirement by creating the objects to store the organisation's data. Making a database for an organisation is just not enough to reach out the requirements, the task is how the users at the organisation can access the objects you have created for them. As an Admin for the organisation it's your duty to make sure every user of the organisation is able to access the data modelling structure.

Activity 1: Create A Lightning App:

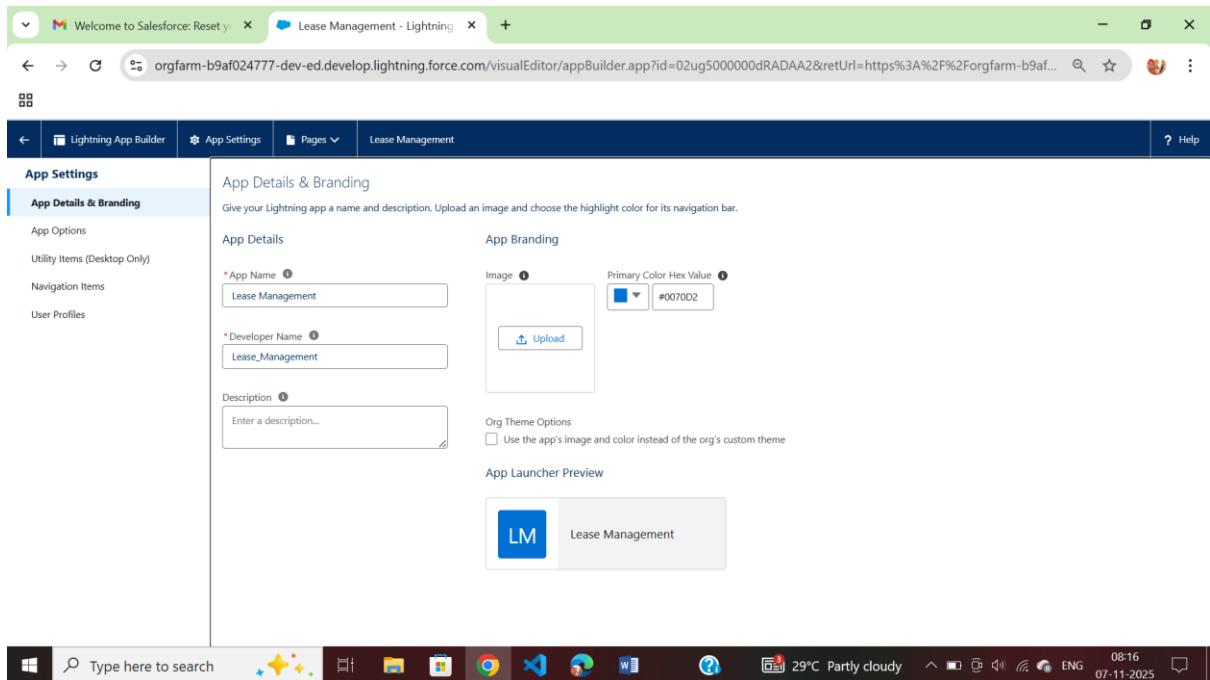
To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

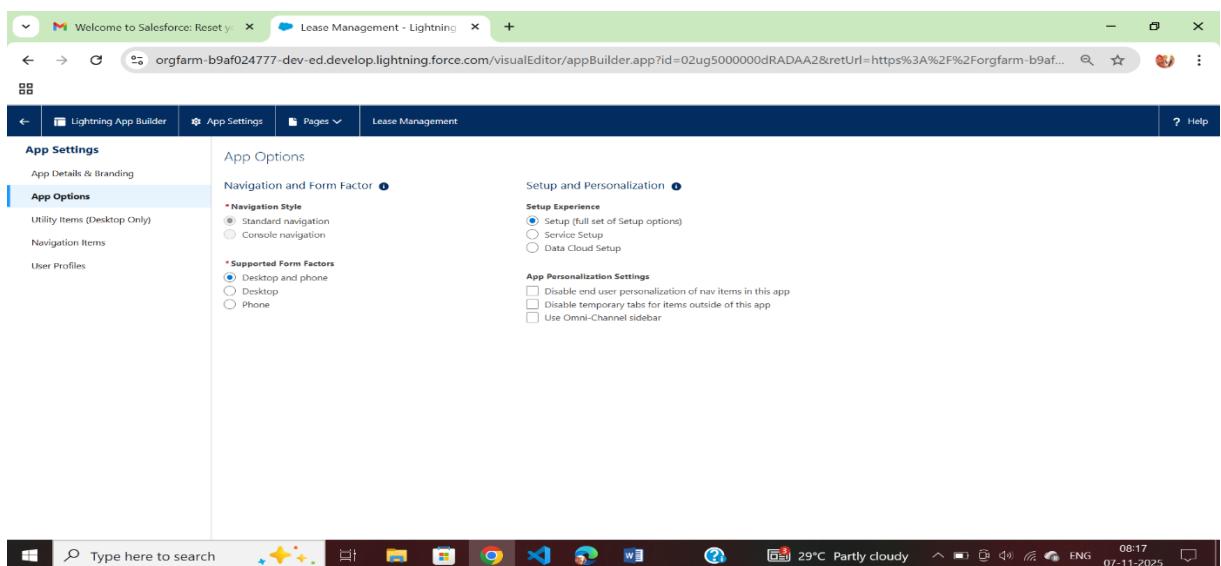


2. Fill the app name in app details and branding as follow

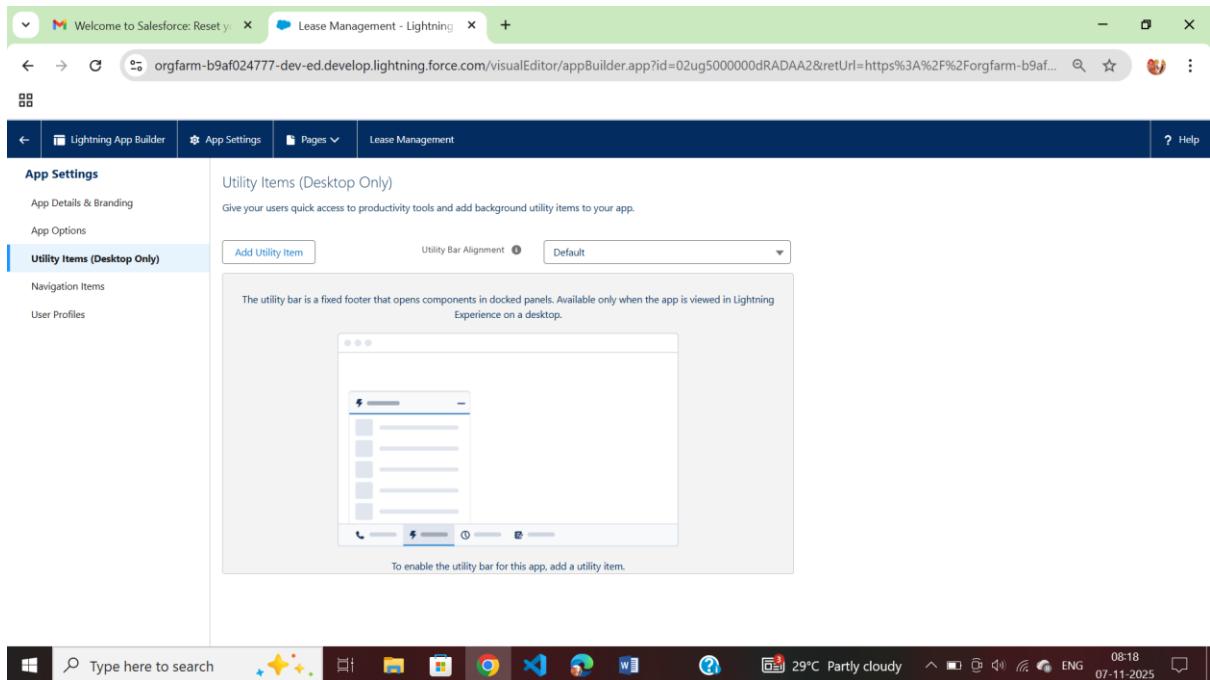
- App Name : Lease Management
- Developer Name : This will auto populated
- Image : optional (if you want to give any image you can otherwise not mandatory)
- Primary colour hex value : keep this default.



3. Then click Next >> (App option page) Set Navigation Style as Standard Navigation >> Next.

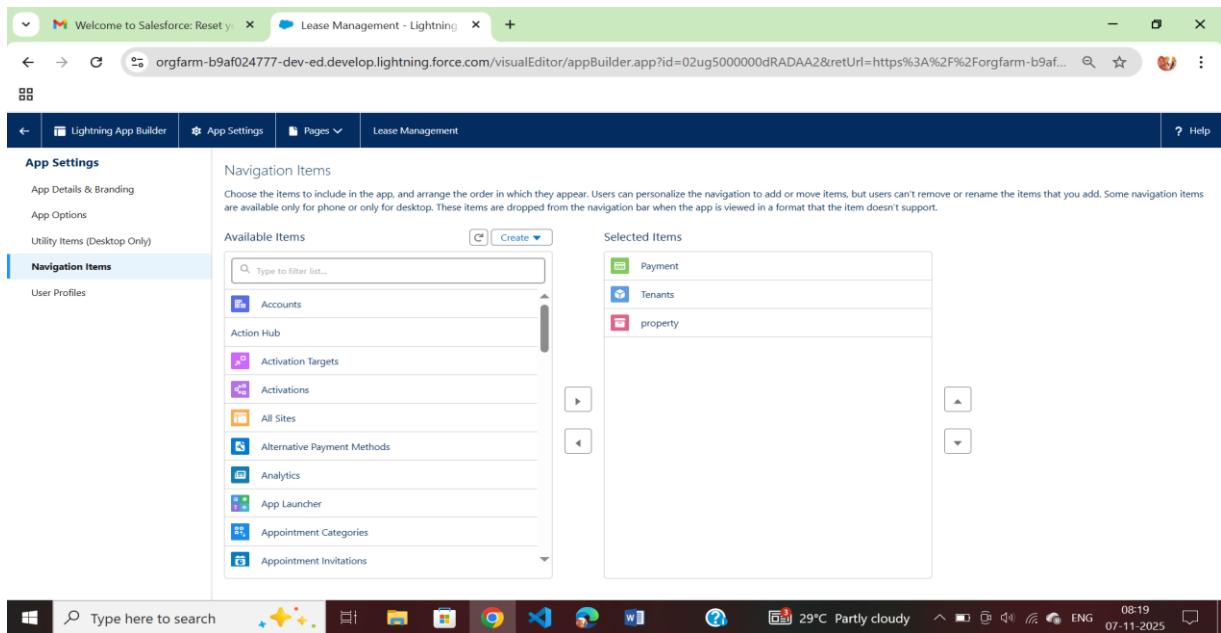


4. (Utility Items) keep it as default >> Next.



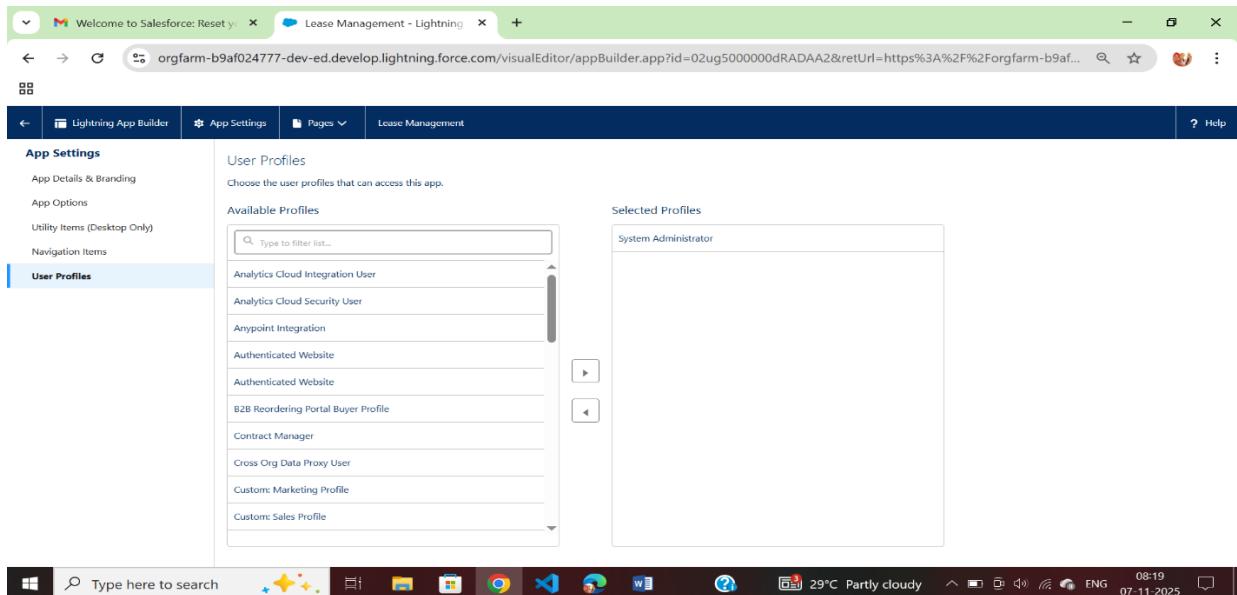
5. To Add Navigation Items:

Search for the item in the (Payment for tenant, Tenants,property,lease) from the search bar and move it using the arrow button ? Next? Next.



6. To Add User Profiles:

Search profiles (System administrator) in the search bar >>click on the arrow button >> save & finish.



Milestone 3: Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields:

1. Standard Fields: As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

- >>Created By
- >>Owner
- >>Last Modified
- >> Field Made During object Creation

2. Custom Fields: On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

Activity 1: Creation Of Fields For The Property Object:

To create fields in an object:

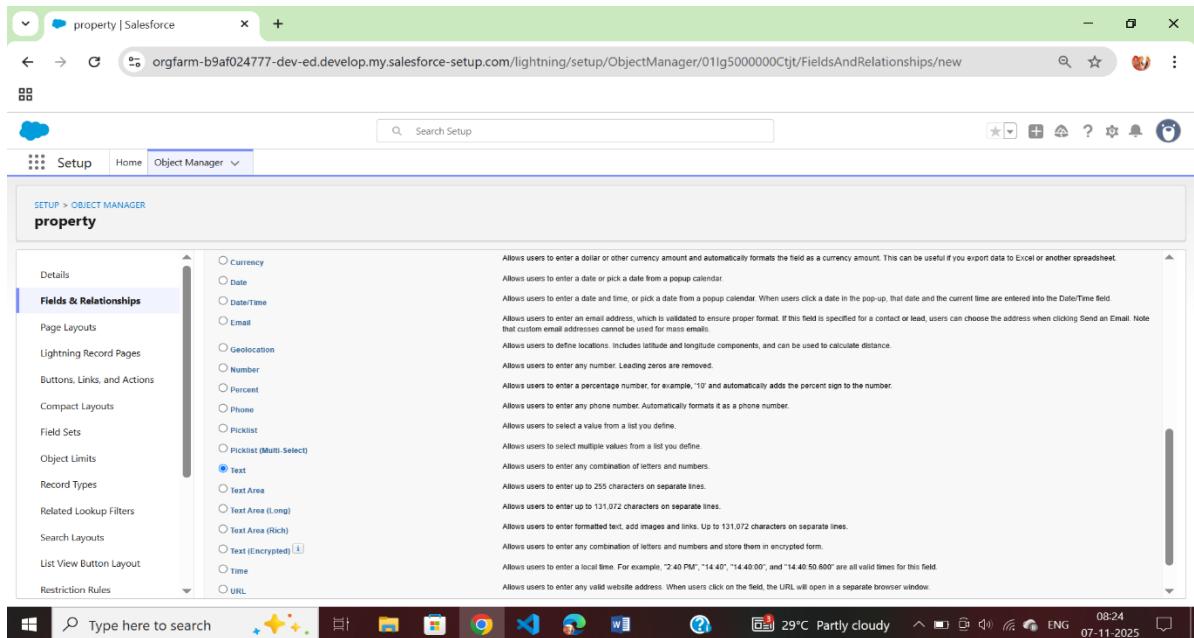
1. Go to setup >> click on Object Manager >> type object name(property) in search bar >>click on the object.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a green header bar with the title "Object Manager | Salesforce". Below it is a browser-like address bar with the URL "orgfarm-b9af024777-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/home". The main content area has a blue header "SETUP Object Manager" with a sub-header "1 Items, Sorted by Label". A search bar at the top right contains the text "property". Below the header is a table with columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. There is one item listed: "property" with API name "property_c", Type "Custom Object", Description "", Last Modified "10/27/2025", and Deployed status checked. The bottom of the screen shows a Windows taskbar with various icons and system status.

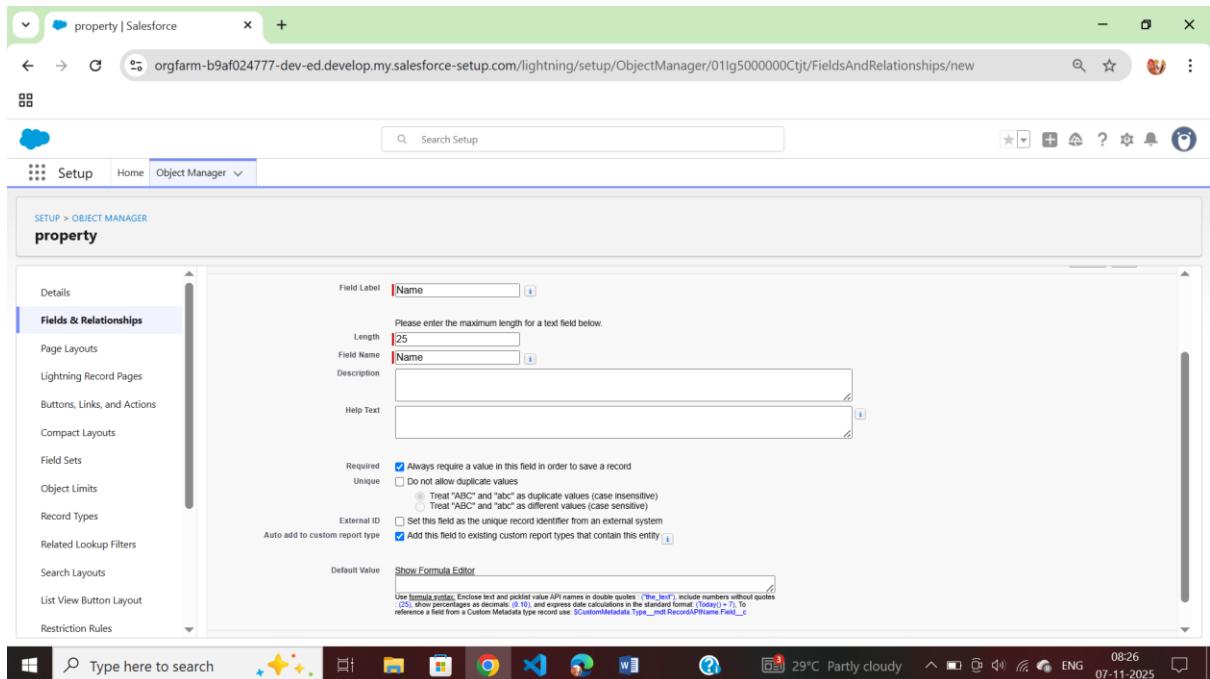
2. Now click on “Fields & Relationships” >> New

The screenshot shows the "Fields & Relationships" page for the "property" object. The top navigation bar includes "SETUP > OBJECT MANAGER" and the object name "property". On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled "Fields & Relationships" with a sub-header "8 Items, Sorted by Field Label". It features a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table lists eight fields: Address (Address_c, Long Text Area(32768)), Created By (CreatedById, Lookup(User)), Last Modified By (LastModifiedById, Lookup(User)), Name (Name_c, Text(25)), Owner (OwnerId, Lookup(User,Group)), property Name (Name, Text(80)), stqt (stqt_c, Text(18)), and Type (Type_c, Picklist). The bottom of the screen shows a Windows taskbar with various icons and system status.

3. Select Data Type as a “Text”

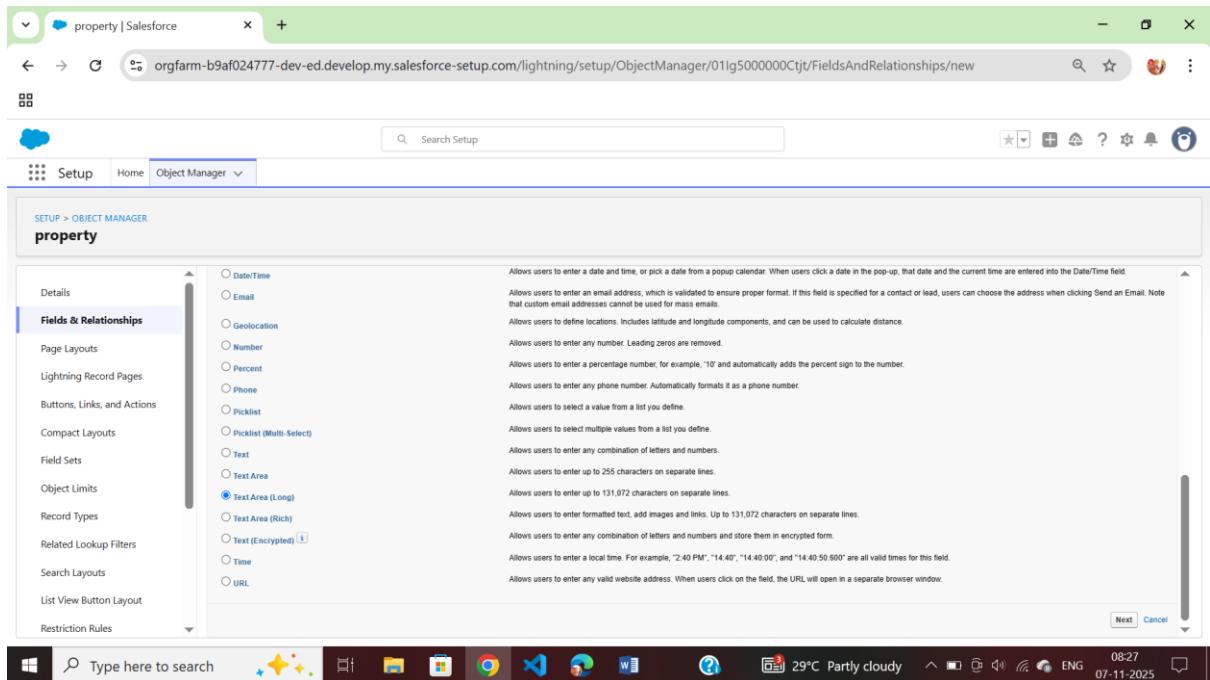


4. Click on next
5. Fill the Above as following:
 - Field Label: Name
 - Field Name : gets auto generated
 - Length : 25
 - Required :check box
 - Click on Next >> Next >> Save and new.



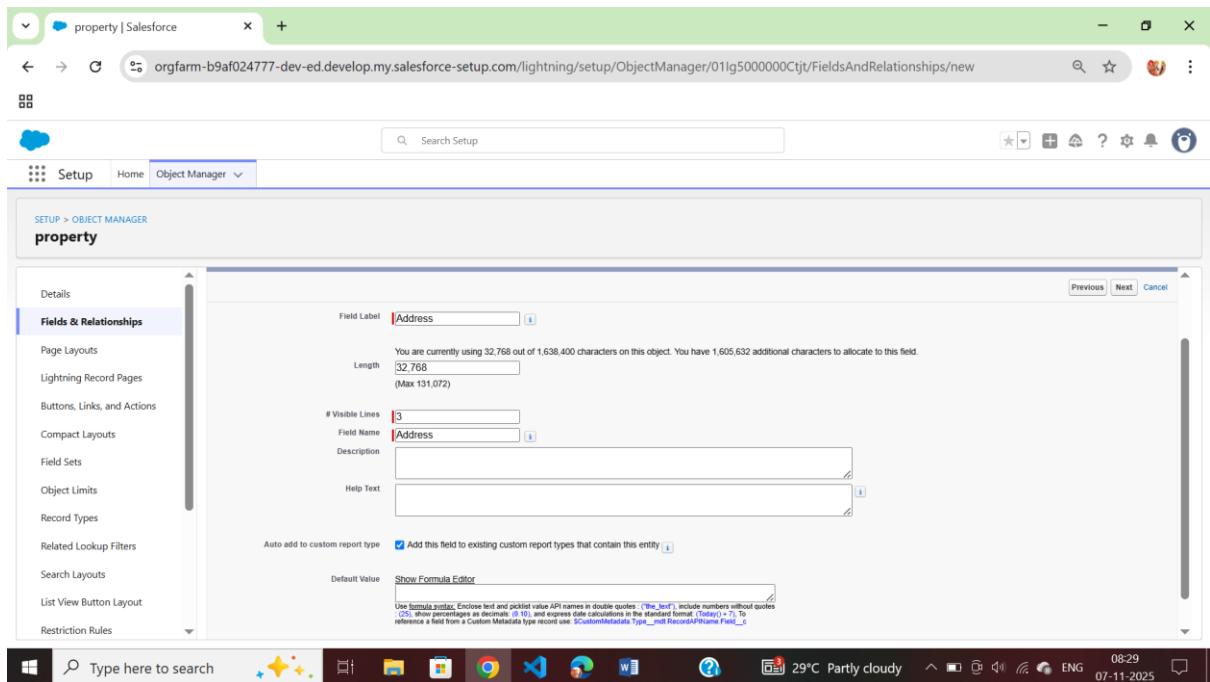
To create another fields in an object:

6. Go to setup >> click on Object Manager >>type object name (property) in search bar >>click on the object.
7. Now click on “Fields & Relationships” >>New
8. Select Data type as a “Long Text” and Click on Next



9. Fill the Above as following:

- Field Label : Address
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.



To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(property) in search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New
3. Select Data type as a "picklist" and Click on Next

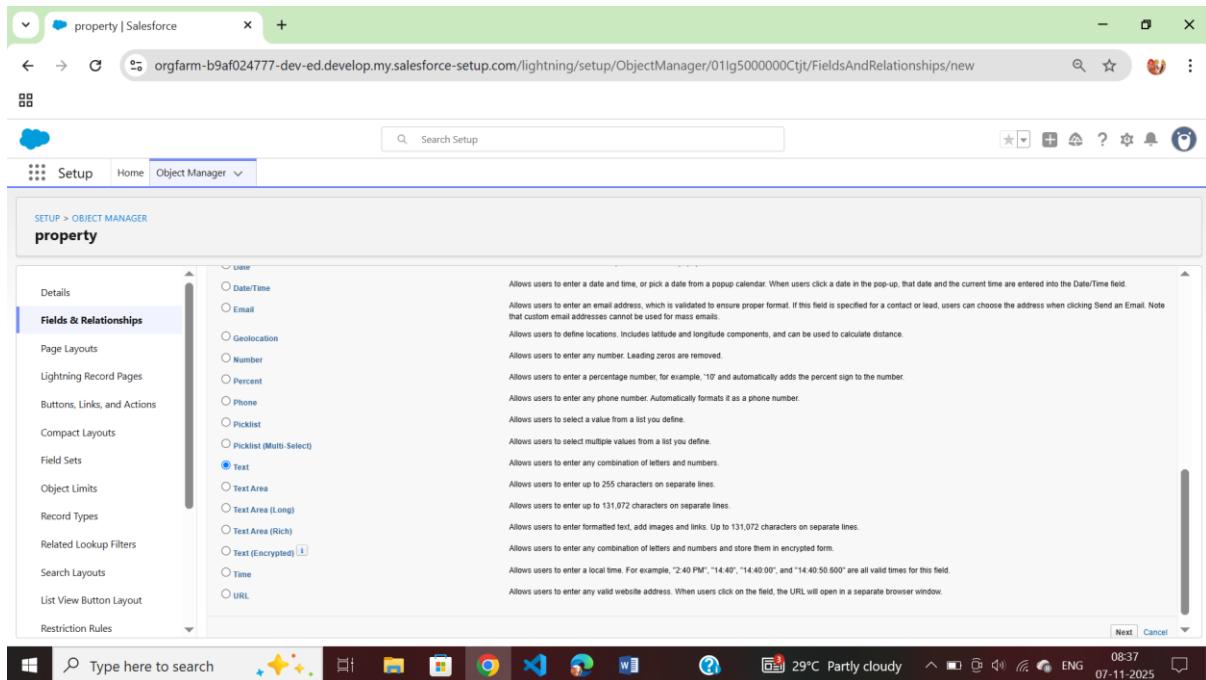
4. Fill the Above as following:

- Field Label : Type
- Field Name : gets auto generated
- Enter values, with each value separated by a new line
- Enter these values
1BHK
2BHK
3BHK
- Click on Next >> Next >> Save and new.

To create another fields in an object:

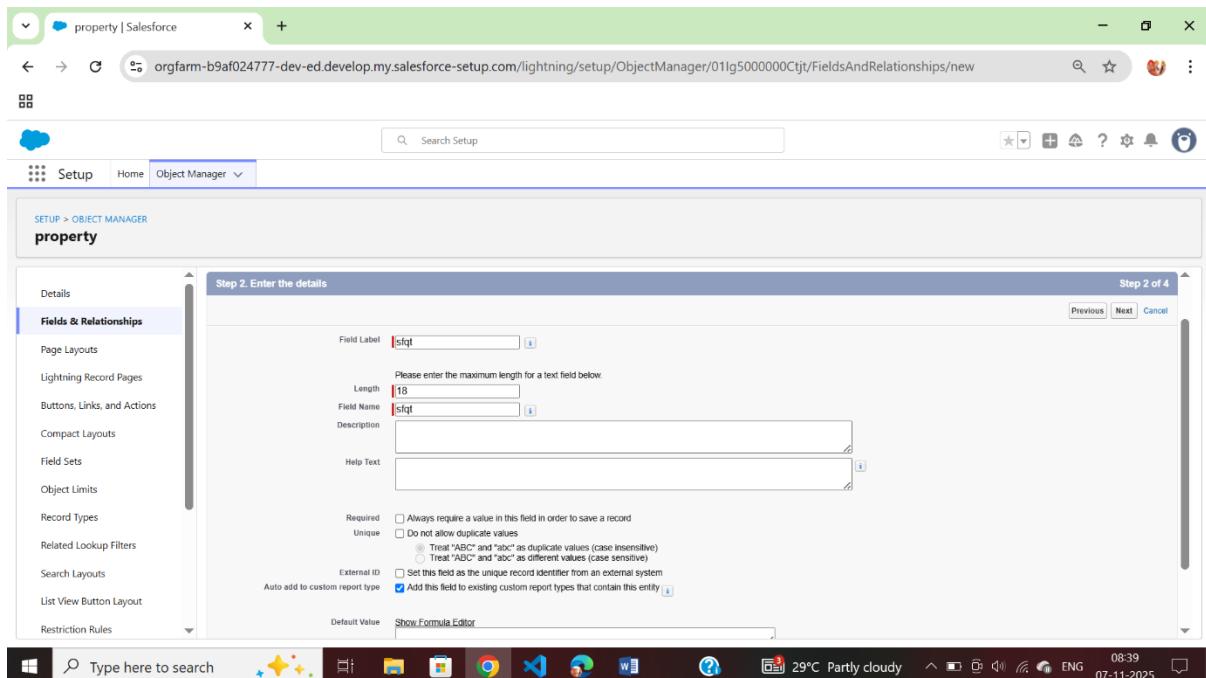
1. Go to setup >> click on Object Manager >> type object name(property) in search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New

3. Select Data type as a “Text” and Click on Next



4. Fill the Above as following:

- Field Label : sfqt
- Field Name : gets auto generated
- Length : 18
- Click on Next >> Next >> Save.



Activity 2: Creation Of Fields For The Tenant Object:

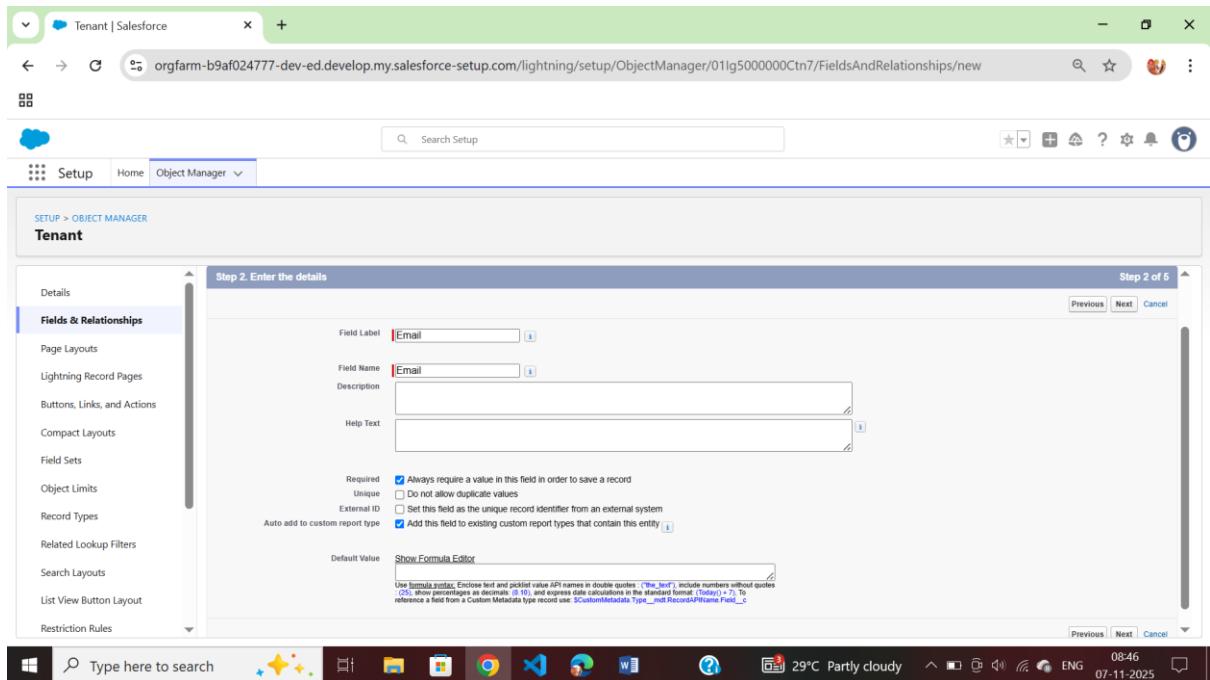
1. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with tabs for 'Setup', 'Home', and 'Object Manager'. Below the navigation bar is a search bar labeled 'Search Setup'. The main content area is titled 'Object Manager' and shows a table with one item. The table columns are 'LABEL', 'API NAME', 'TYPE', 'DESCRIPTION', 'LAST MODIFIED', and 'DEPLOYED'. The single item listed is 'Tenant' with API name 'Tenant_c', Type 'Custom Object', and Last Modified date '10/27/2025'. A deployment status indicator shows a checkmark. At the bottom of the page is a Windows taskbar with various pinned icons.

2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next

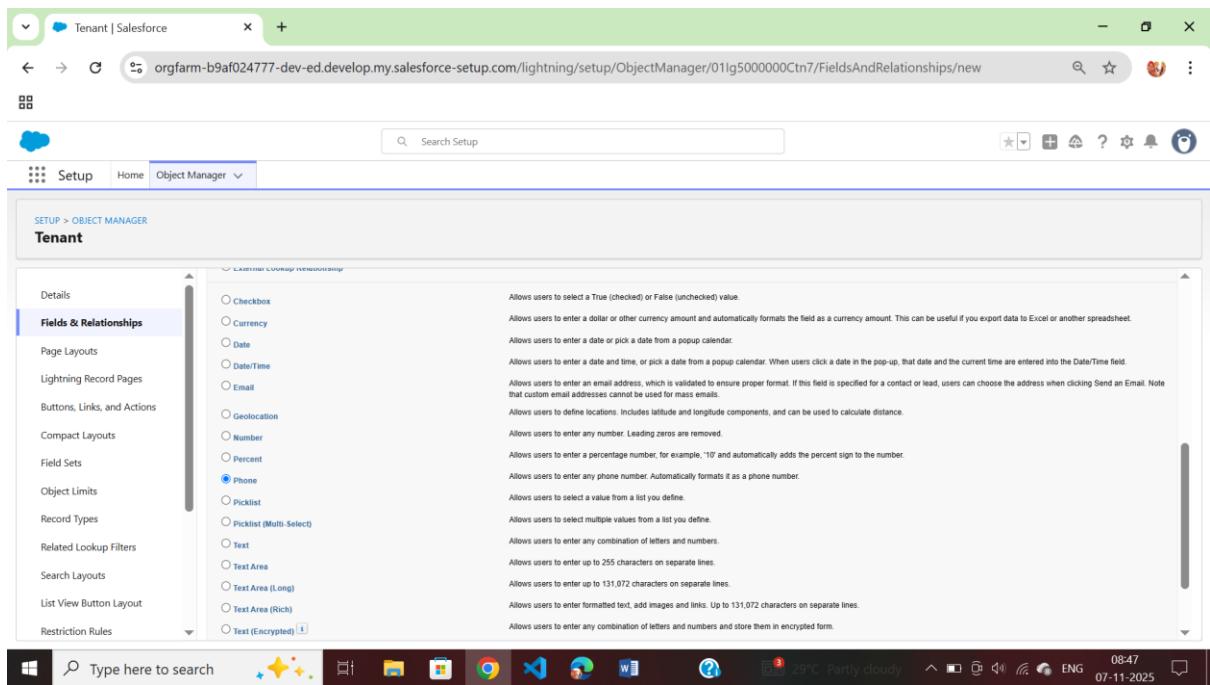
The screenshot shows the 'Fields & Relationships' creation page for the 'Tenant' object. On the left, there's a sidebar with options like 'Details', 'Fields & Relationships', 'Page Layouts', etc. The 'Fields & Relationships' option is selected. In the main pane, under 'External Lookup Relationship', there's a list of field types. The 'Email' type is selected, indicated by a blue circle. A tooltip for 'Email' provides a detailed description: 'Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.' Below the list are several other field types like Checkbox, Currency, Date, etc., each with a brief description. At the bottom of the page is a Windows taskbar.

4. Fill the Above as following:
- Field Label : Email
- Field Name : gets auto generated
- Click on required check box
- Click on Next >> Next >> Save and new.



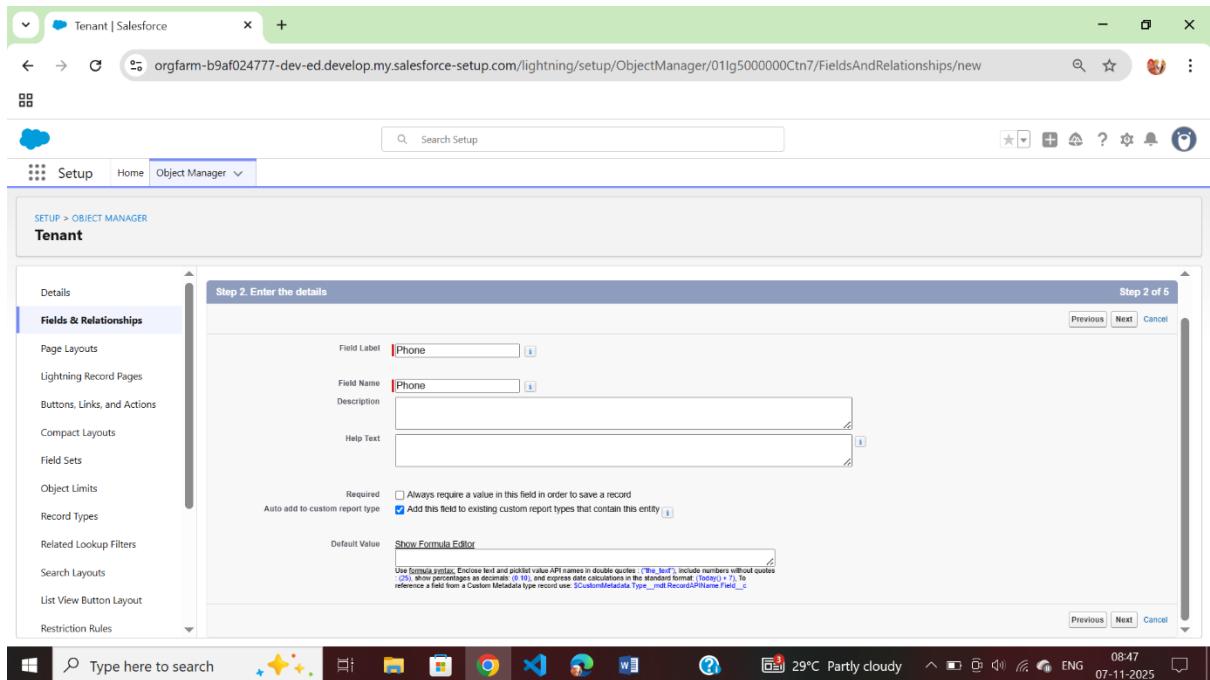
To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “phone” and Click on Next



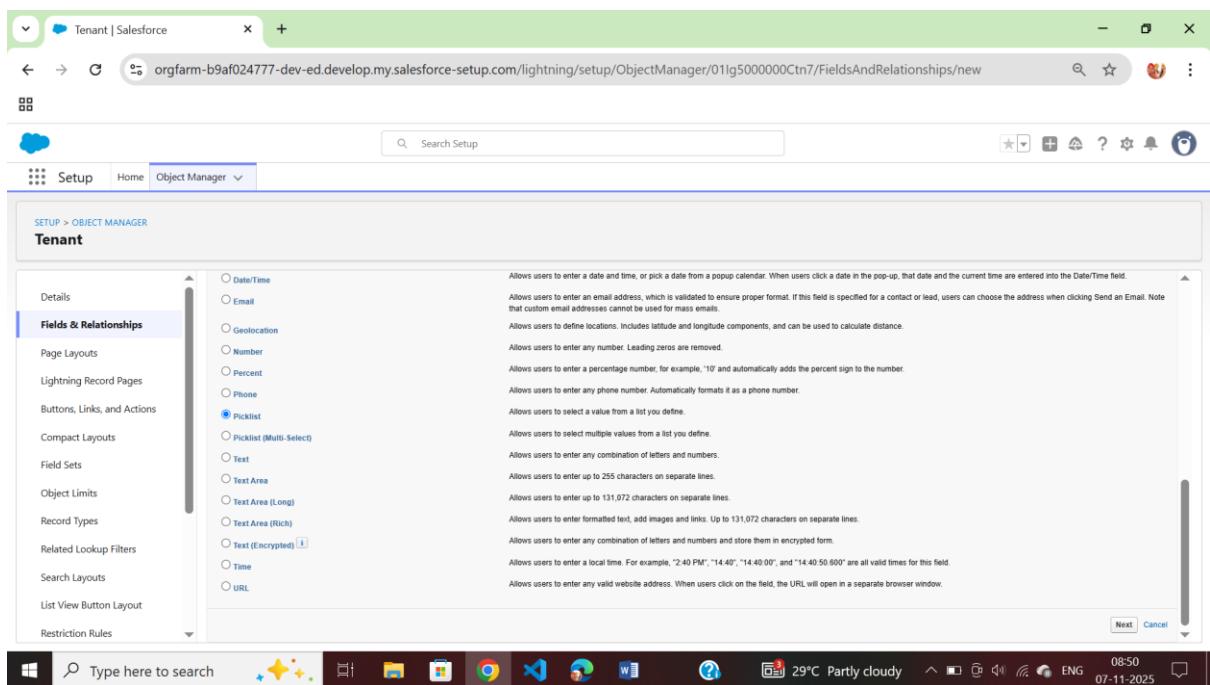
4. Fill the Above as following:

- Field Label : Phone
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.



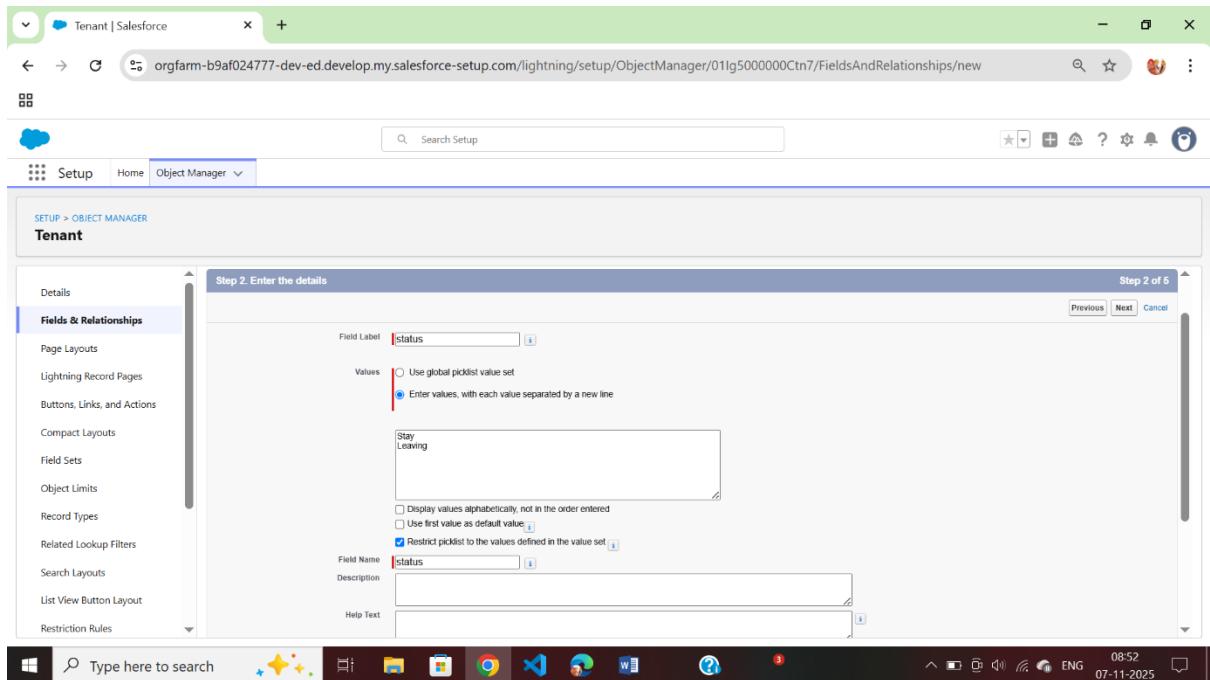
To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >>New
3. Select Data type as a “picklist” and Click on Next



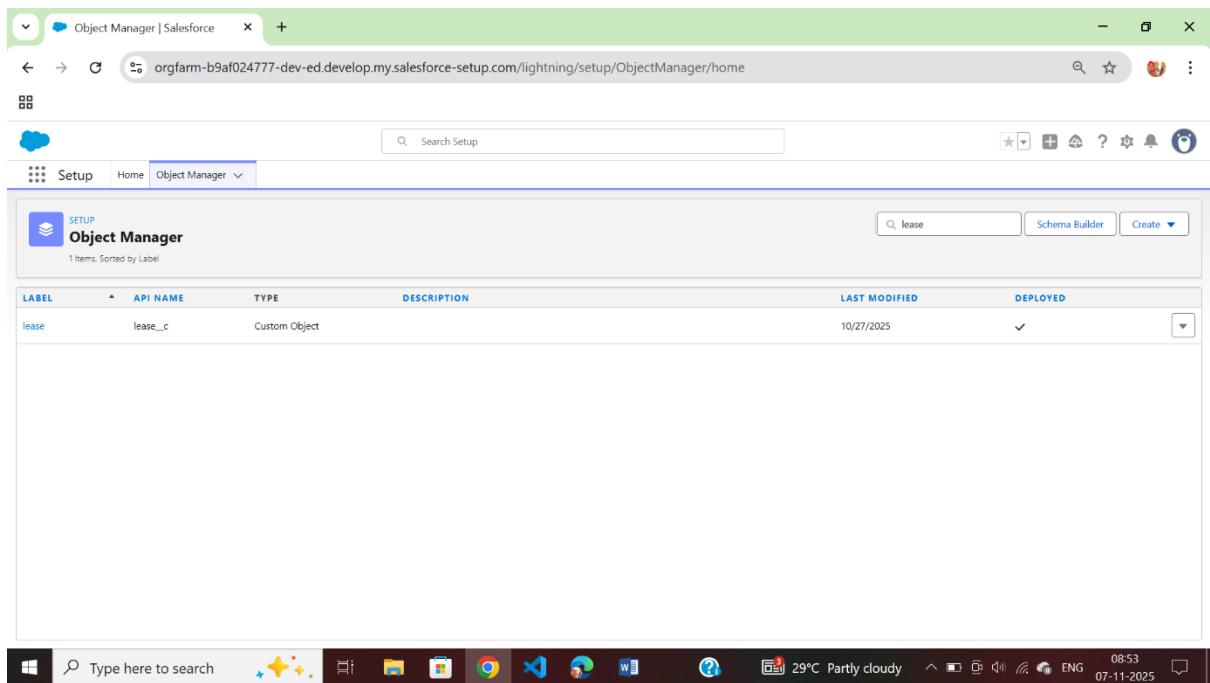
4. Fill the Above as following:

- Field Label : status
- Field Name : gets auto generated
- Enter values, with each value separated by a new line
- Enter these values
Stay
Leaving
- Click on Next >> Next >> Save

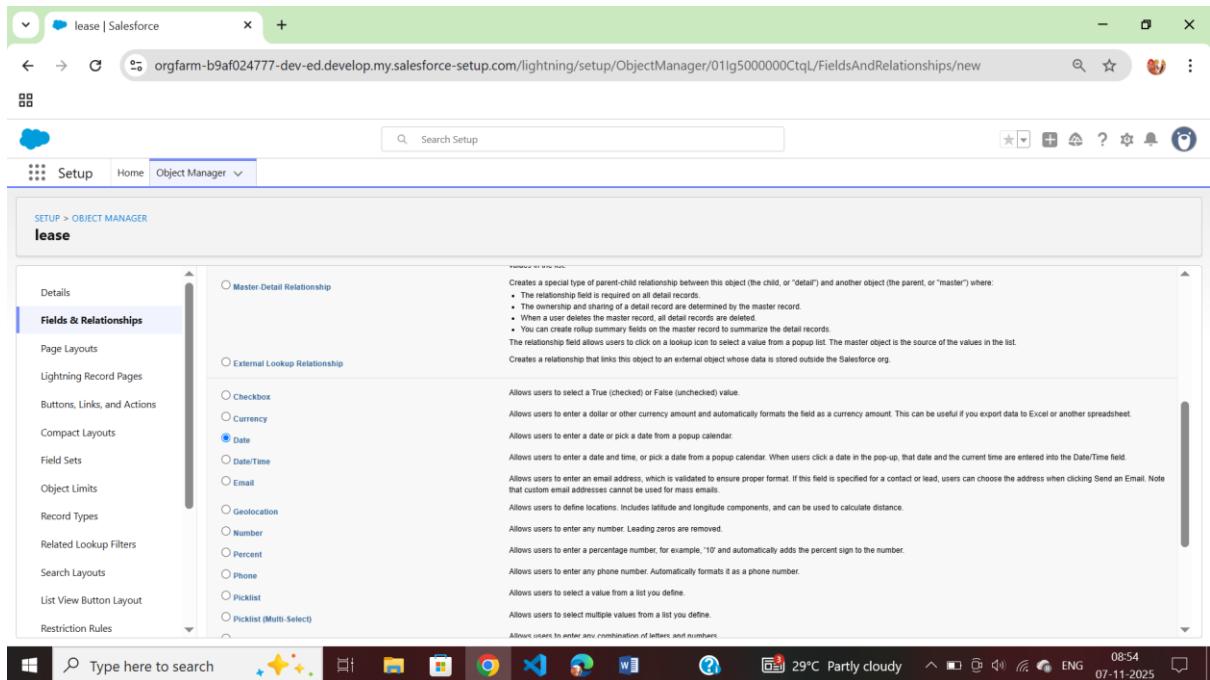


Activity 3: Creation Of Fields For The Lease Object:

1. Go to setup >> click on Object Manager >> type object name(Lease) in search bar >> click on the object.

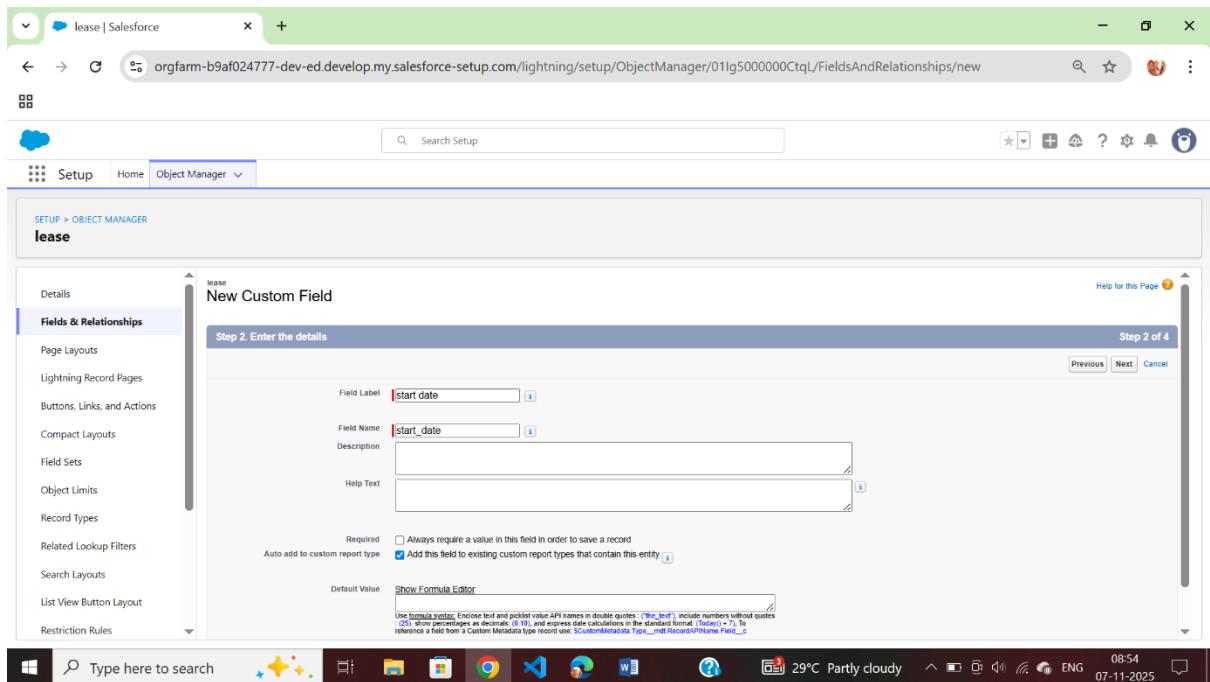


2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Date” and Click on Next



4. Fill the Above as following:

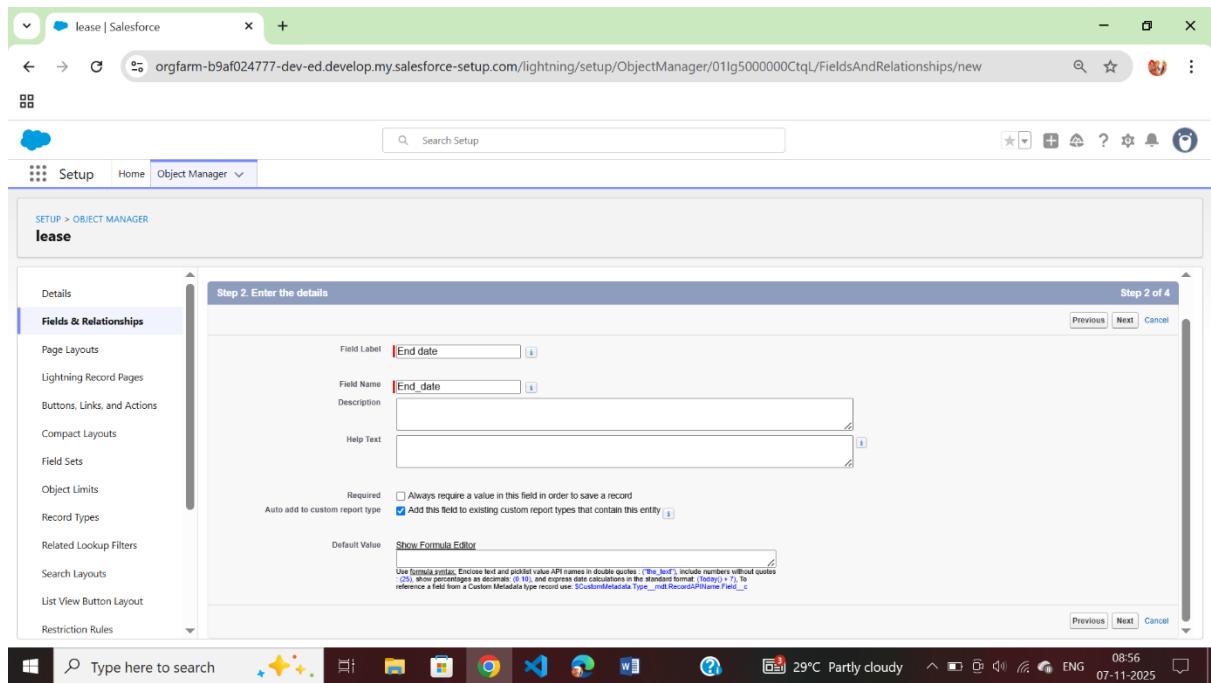
- Field Label : start date
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.



To create another fields in an object:

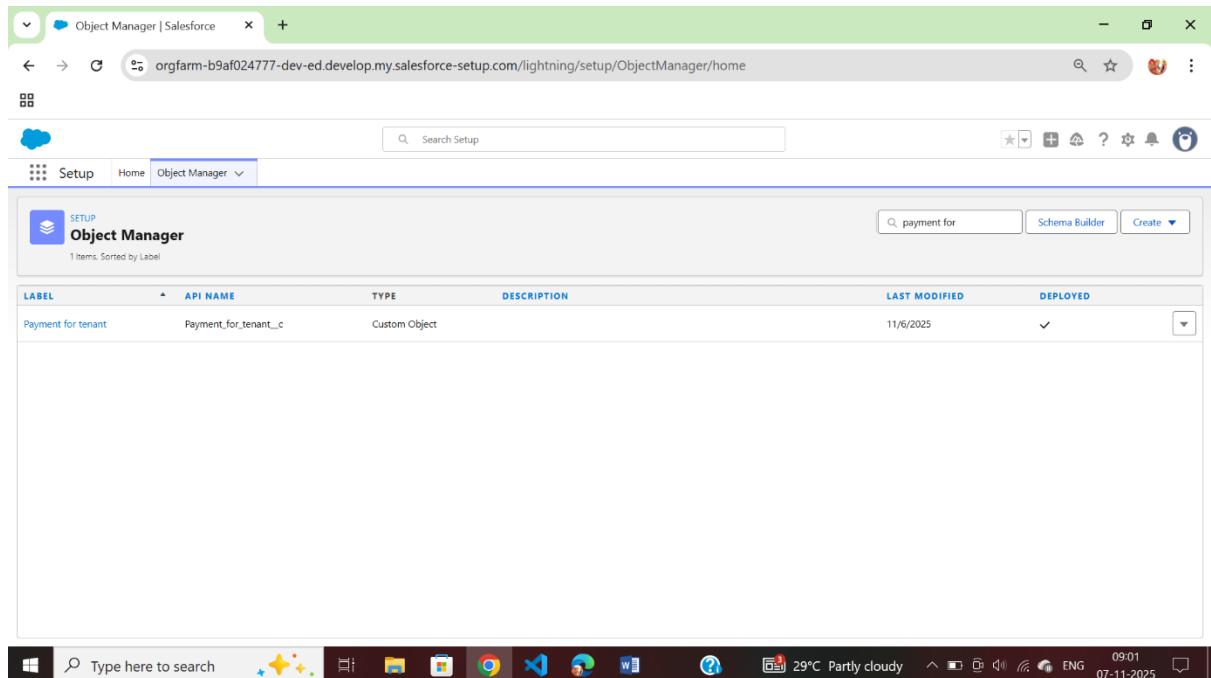
1. Go to setup >> click on Object Manager >> type object name(Lease) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Date” and Click on Next
4. Fill the Above as following:
 - Field Label : End date
 - Field Name : gets auto generated

- Click on Next >> Next >> Save and new.



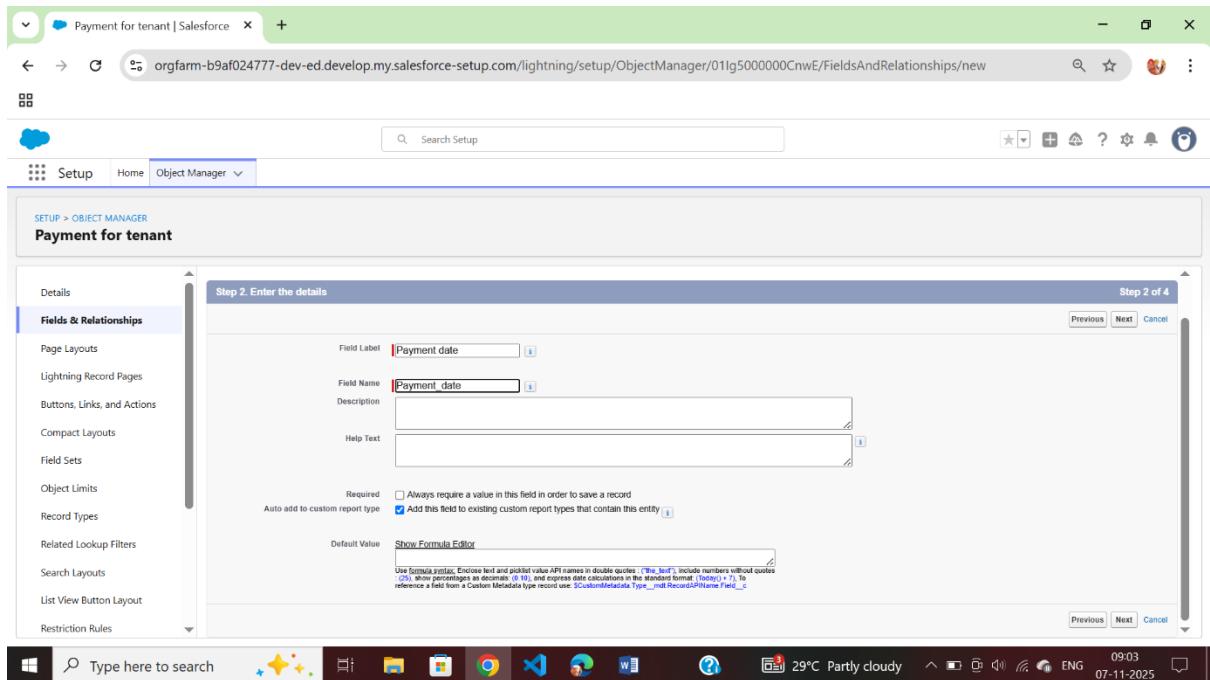
Activity 4: Creation Of Fields For The Payment For Tenant Object:

1. Go to setup >> click on Object Manager >> type object name(Payment for tenant) in search bar >> click on the object.



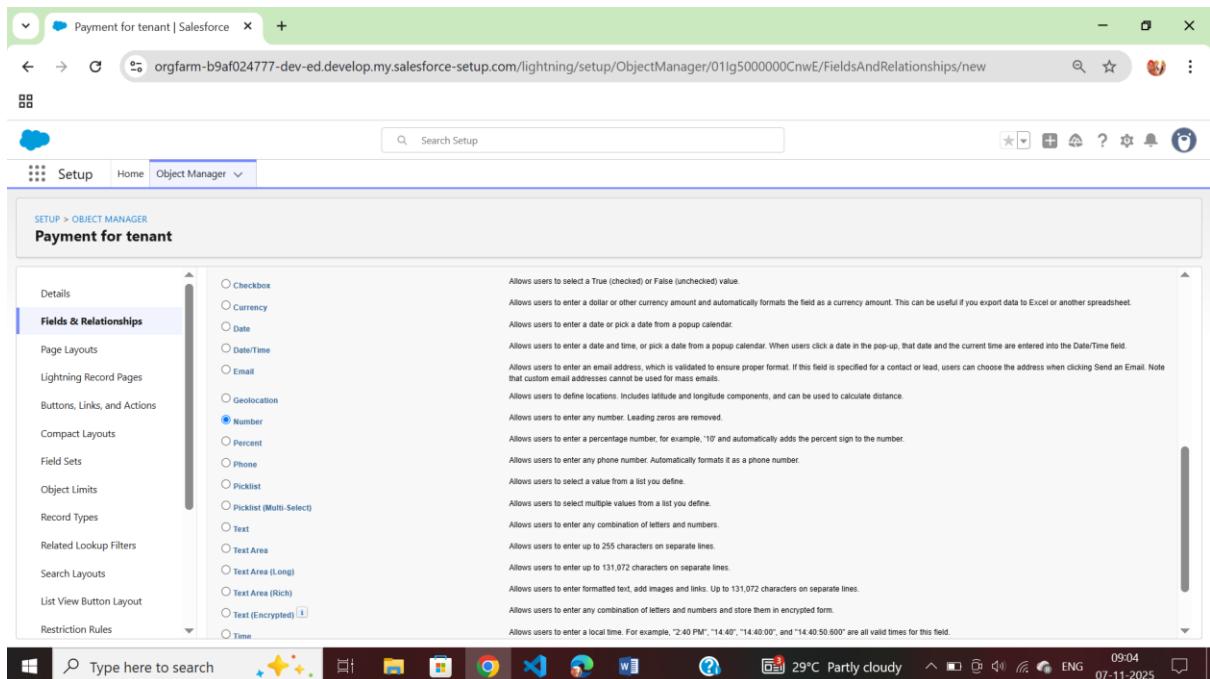
- Now click on "Fields & Relationships" >> New
- Select Data type as a "Date" and Click on Next
- Fill the Above as following:
 - Field Label : Payment date
 - Field Name : gets auto generated

- Click on Next >> Next >> Save and new.



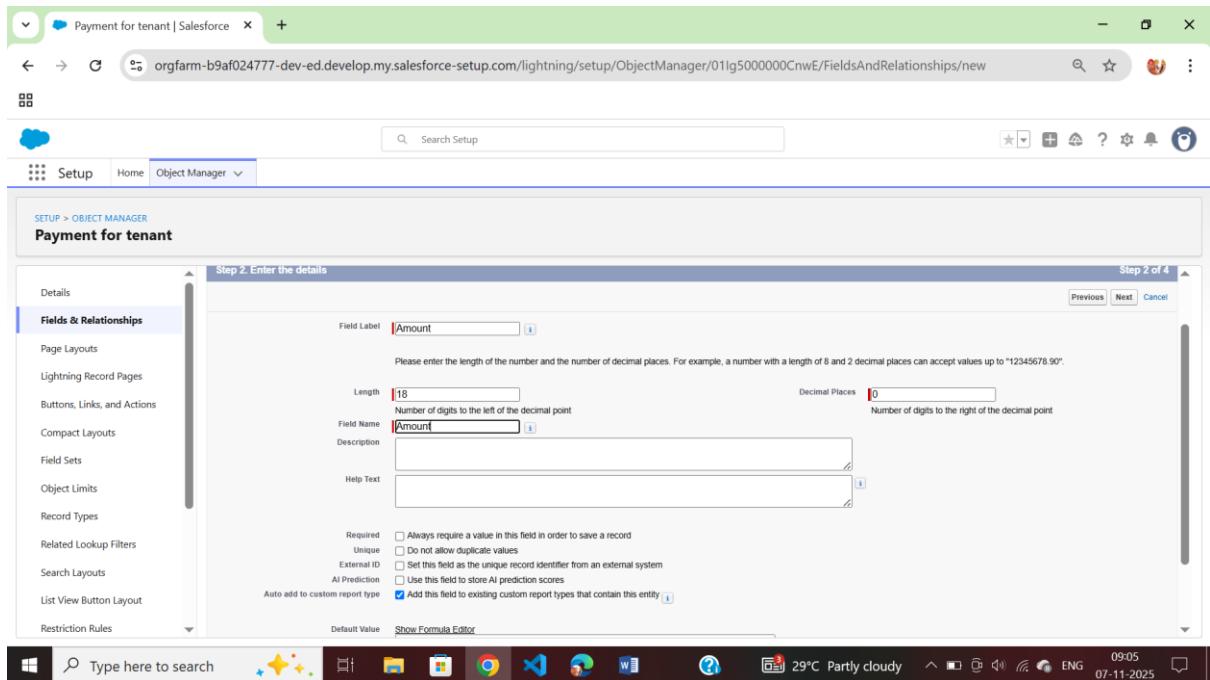
To create another fields in an object:

- Go to setup >> click on Object Manager >> type object name(Payment for tenant) in search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select Data type as a “Number” and Click on Next



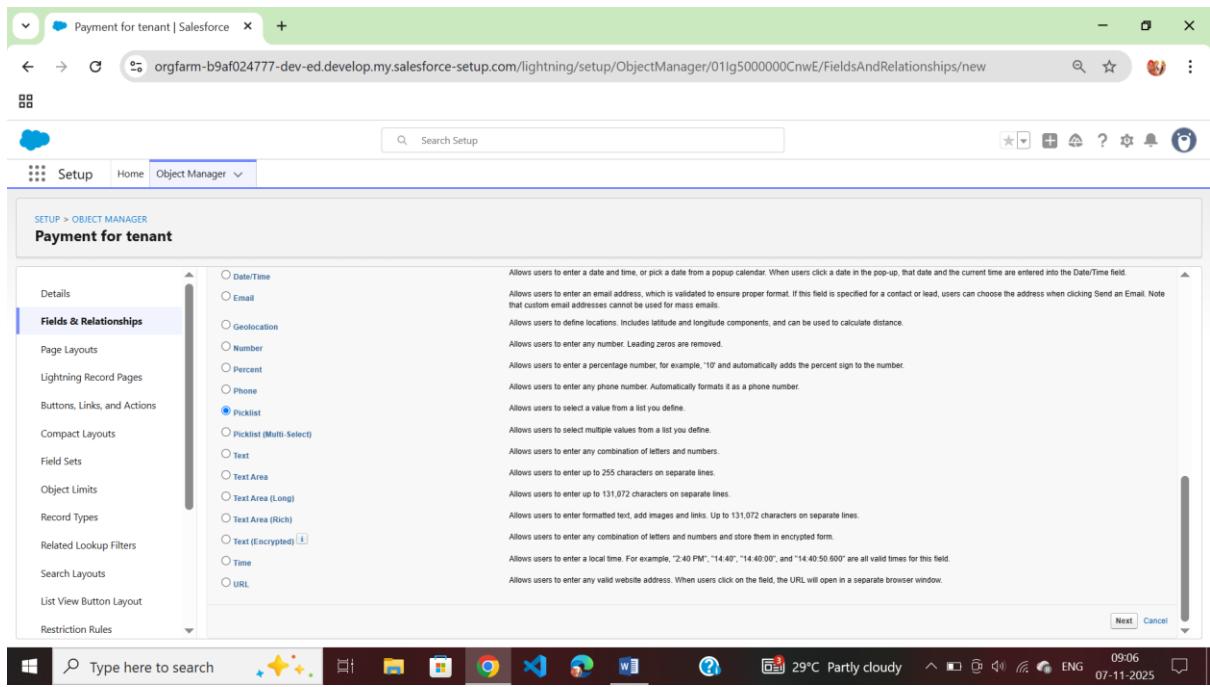
- Fill the Above as following:

- Field Label : Amount
- Length : 18
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.



To create another fields in an object:

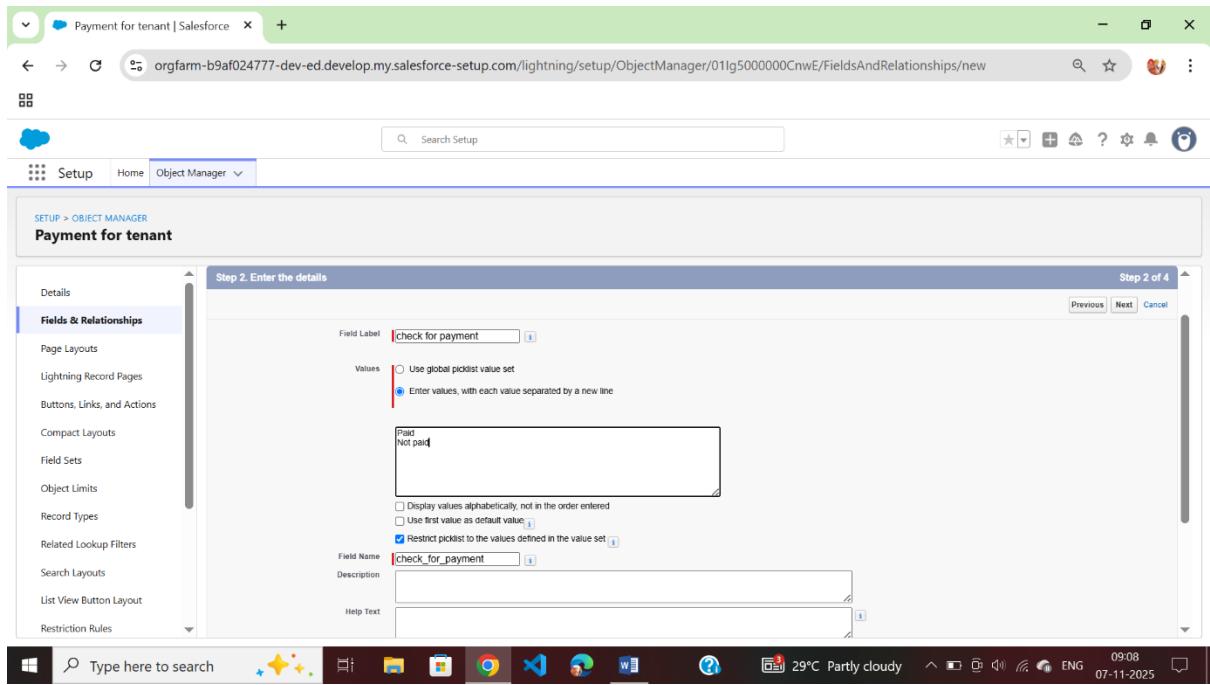
1. Go to setup >> click on Object Manager >> type object name(Payment for tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “picklist” and Click on Next



4. Fill the Above as following:

- Field Label : check for payment
- Field Name : gets auto generated
- Enter values, with each value separated by a new line
- Enter these values
Paid
Not paid

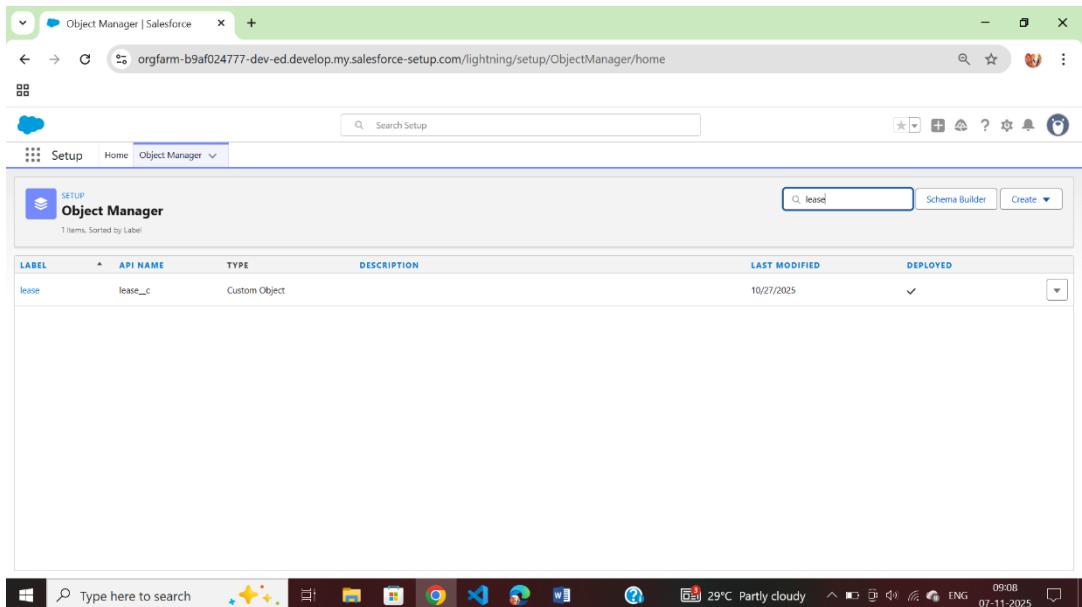
- Click on Next >> Next >> Save and new.



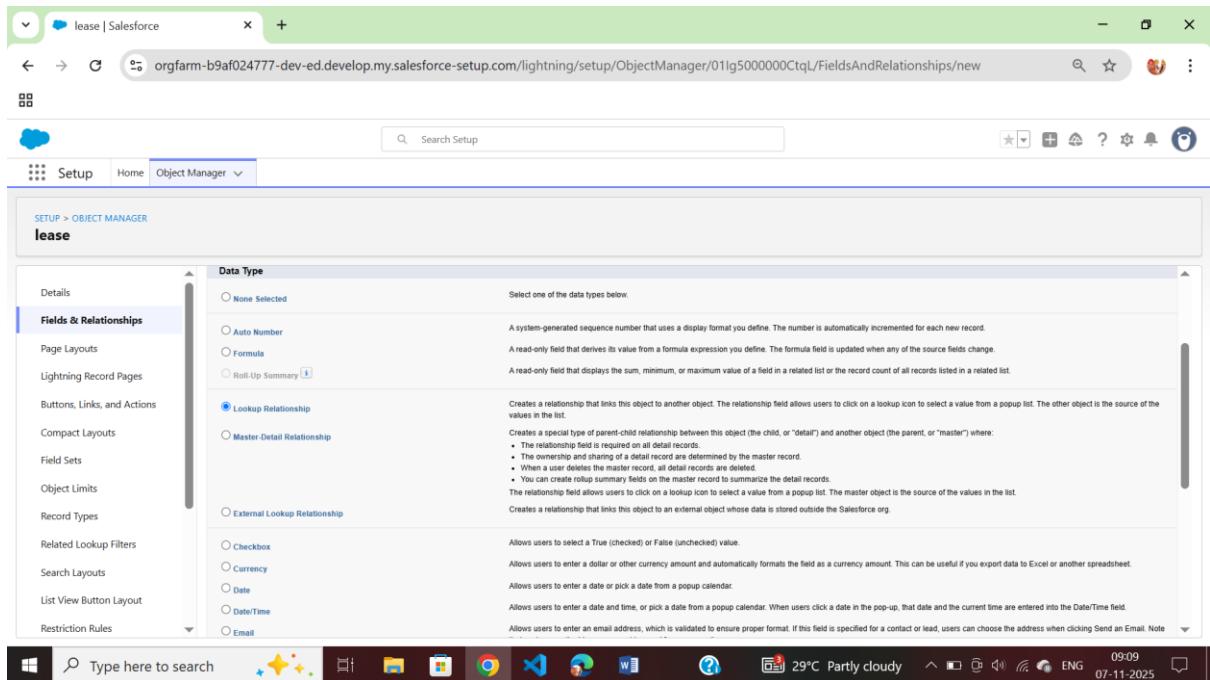
Activity 5: Creation Of Lookup Fields:

Creation of Lookup Field on Lease Object :

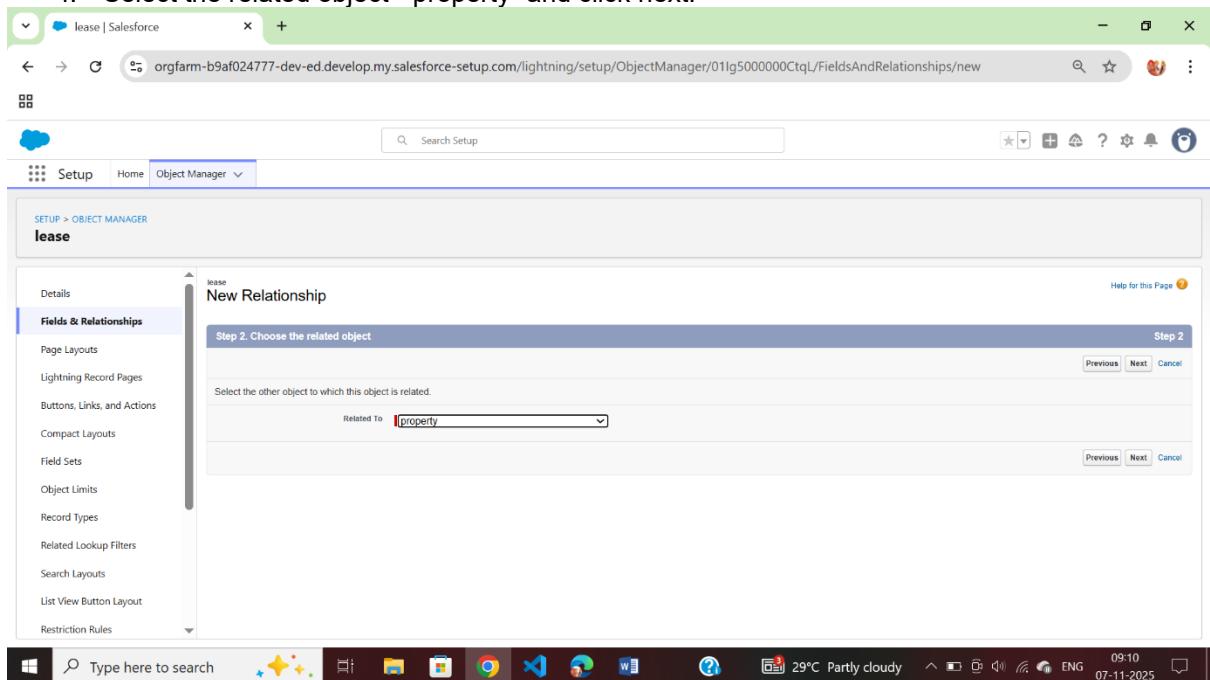
1. Go to setup>> click on Object Manager >> type object name(Lease) in the search bar >> click on the object.



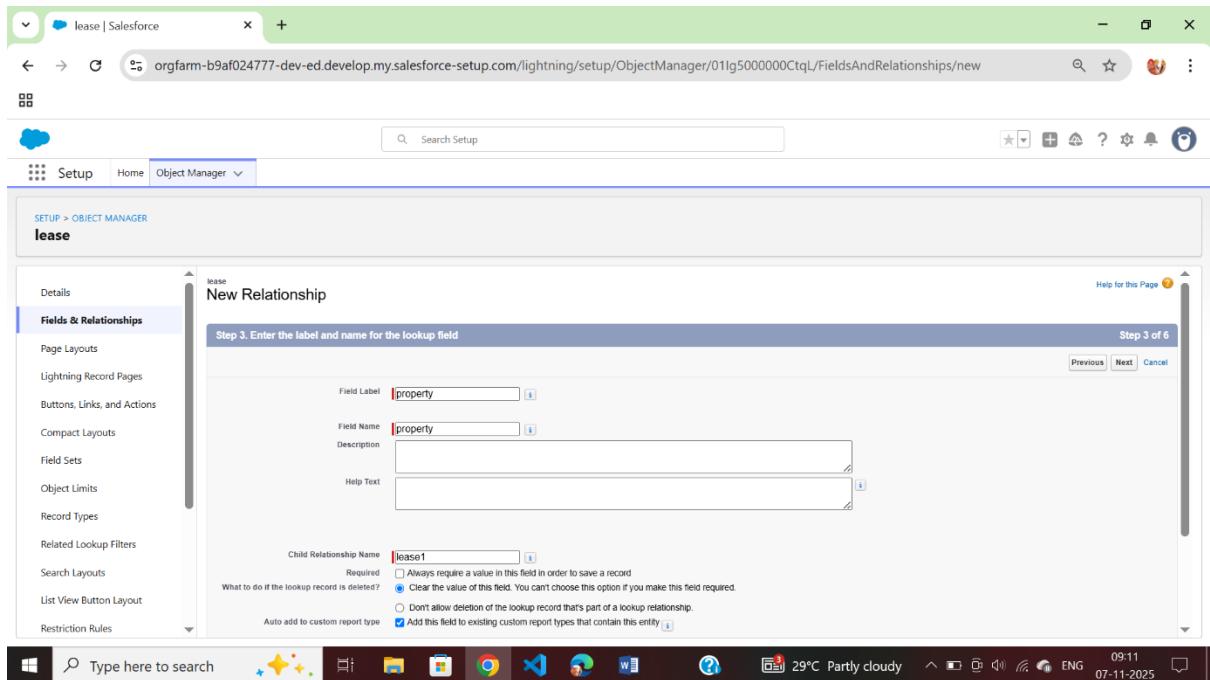
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship



4. Select the related object “property” and click next.

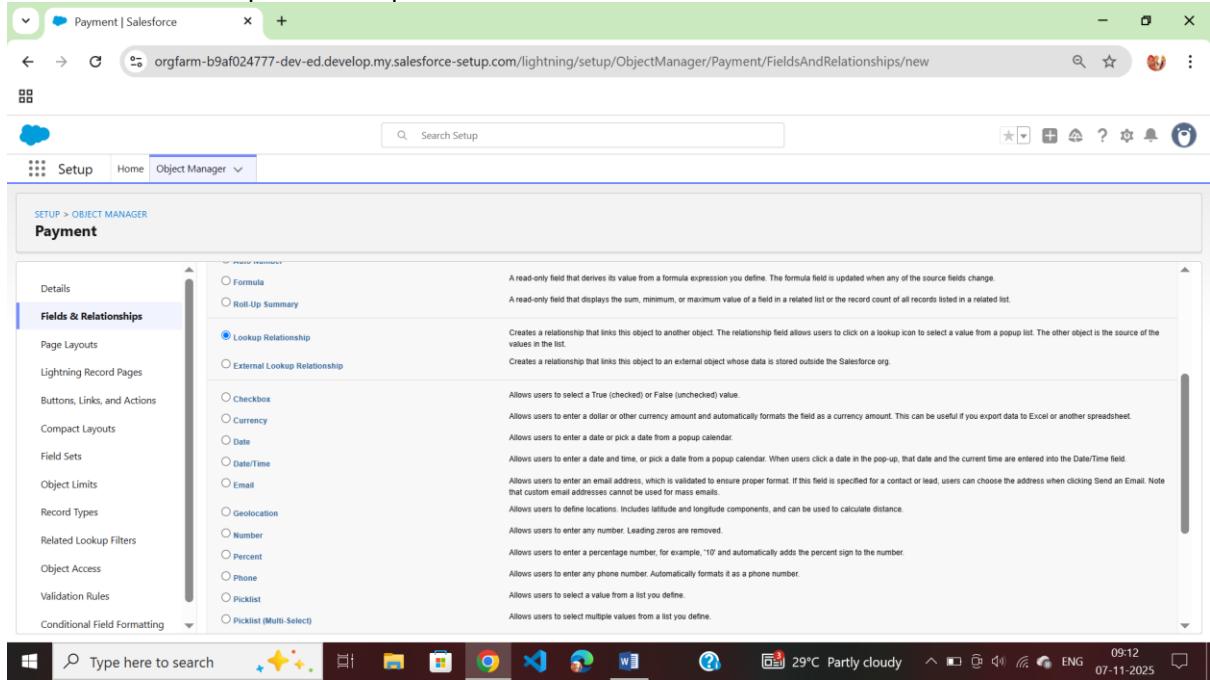


5. Field Name : property
6. Field label : Auto generated
7. Next >> Next >> Save.

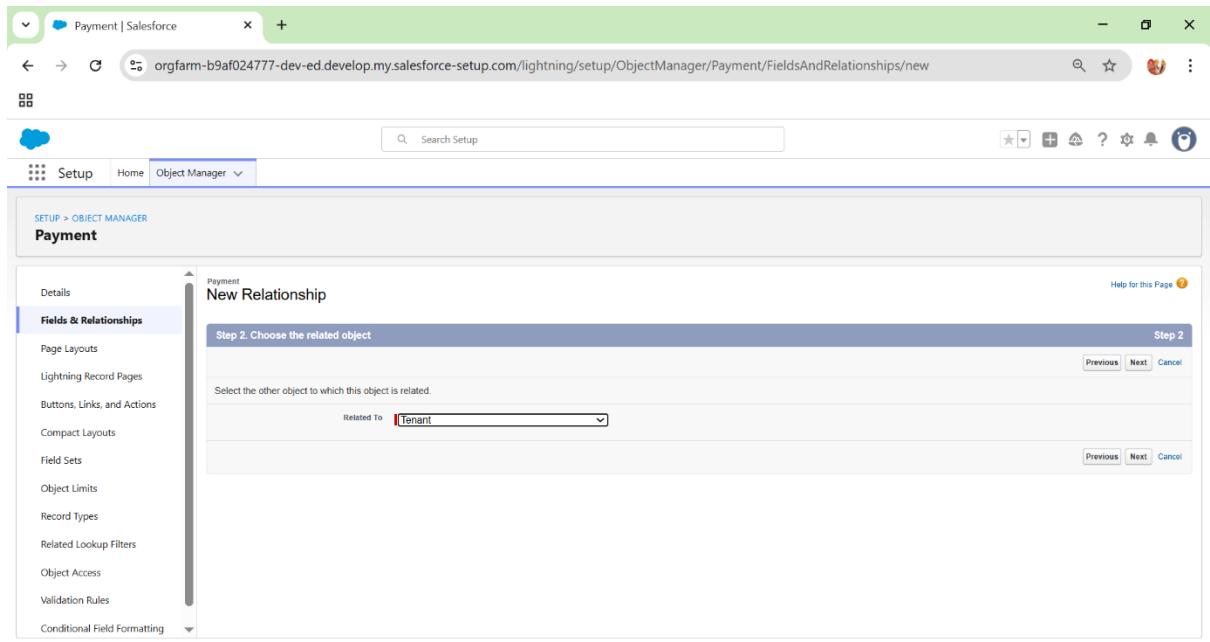


Creation of Lookup Field on Payment Object:

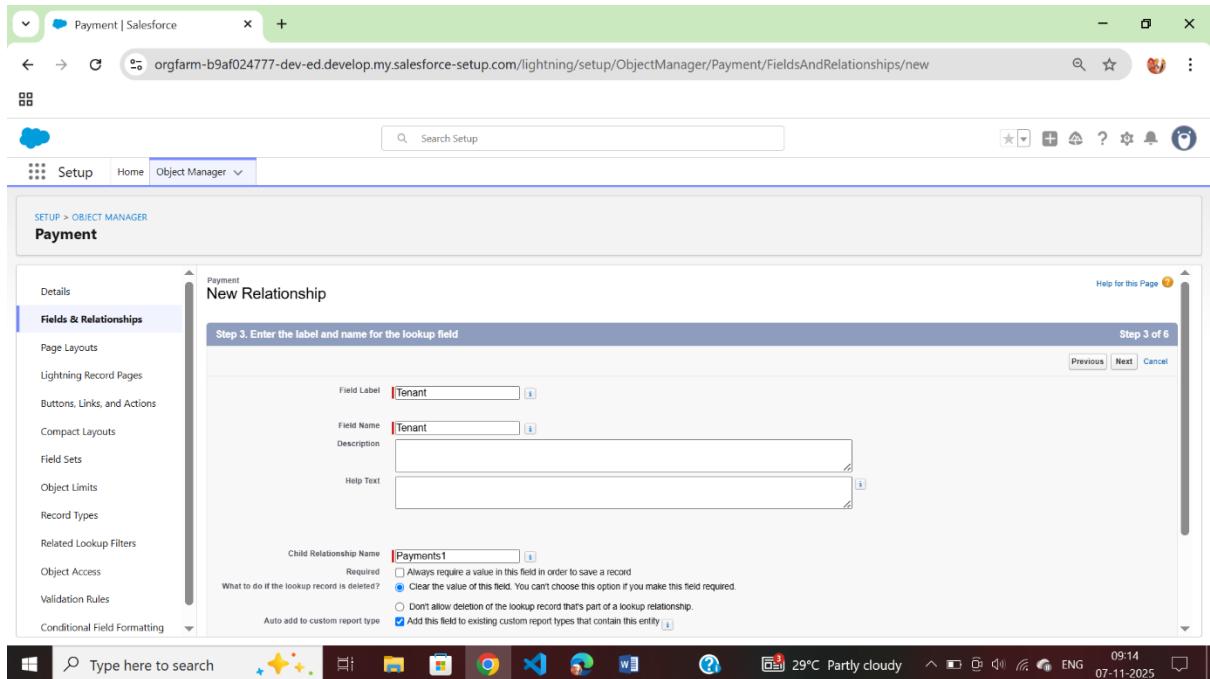
1. Go to setup >> click on Object Manager >> type object name(payment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship



4. Select the related object “Tenant” and click next.

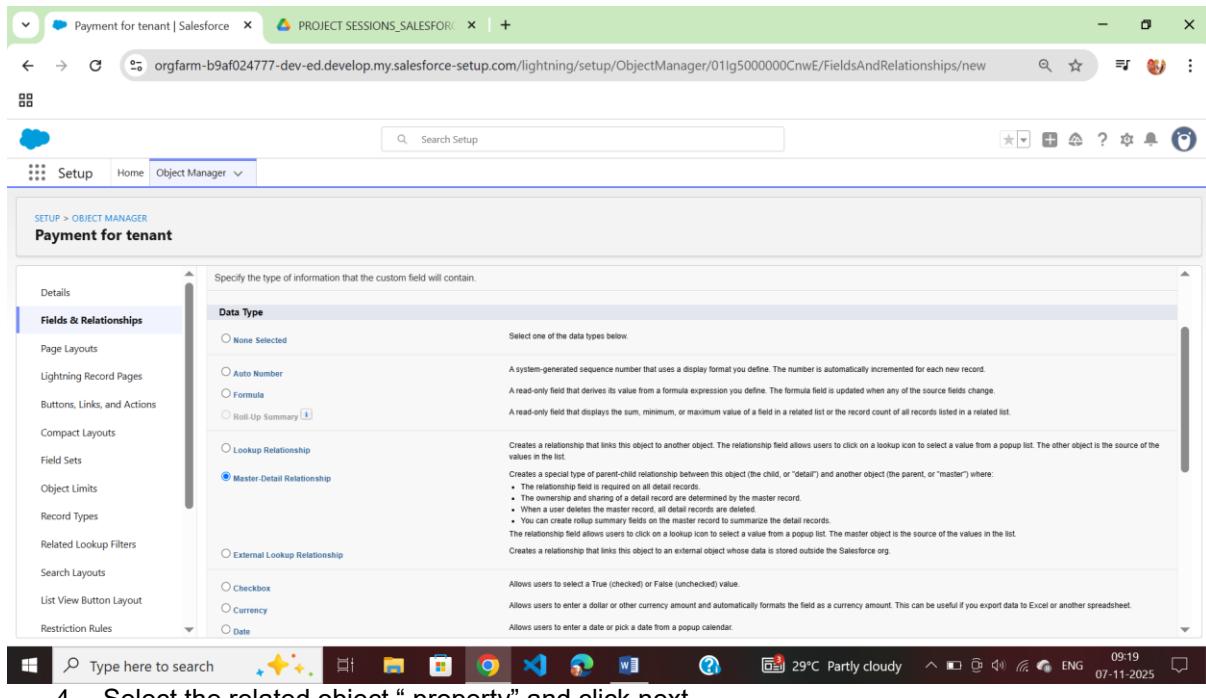


5. Field Name : Tenant
6. Field label : Auto generated
7. Next >> Next >> Save.

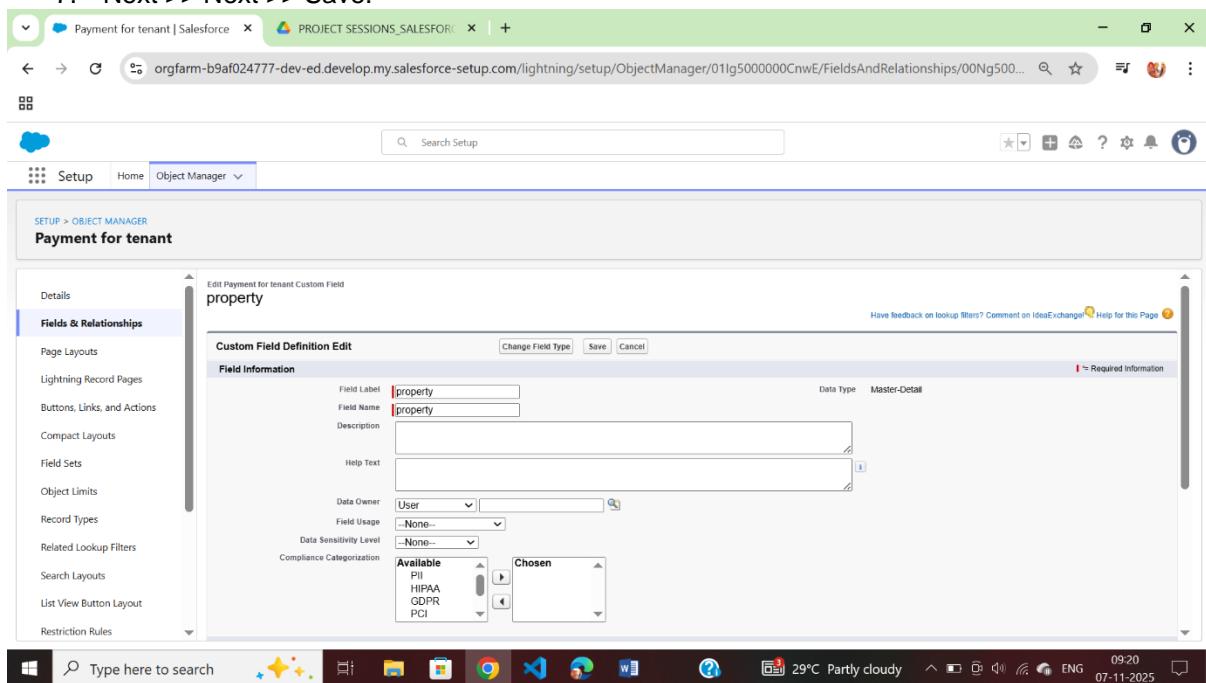


Creation of Lookup Field on Payment for tenant Object :

1. Go to setup>> click on Object Manager >> type object name(Payment for tenant) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select masterdetail relationship



4. Select the related object “ property” and click next.
5. Field Name : property
6. Field label : Auto generated
7. Next >> Next >> Save.



Milestone 4: Validation Rule

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

Activity 1: To Create A Validation Rule To An Lease Object:

1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.
2. Click on the validation rule >> click New.

The screenshot shows the Salesforce Object Manager Validation Rules screen for the 'lease' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main area displays a table titled 'Validation Rules' with one item: 'lease_end_date' (Rule Name), 'start date' (Error Location), and 'Your End date must be greater than start date' (Error Message). The status is 'ACTIVE' and it was modified by 'SIVAMANI P' on 10/27/2025, 11:59 PM.

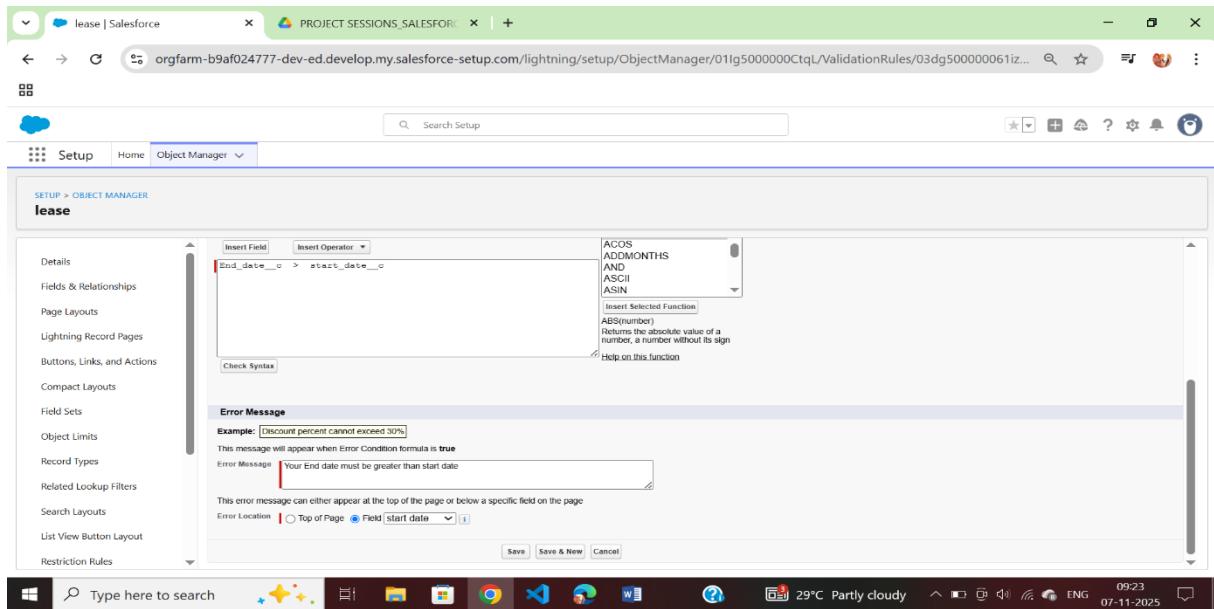
3. Enter the Rule name as “lease_end_date”.

4. Insert the Error Condition Formula as :

`End_date__c > start_date__c`

The screenshot shows the Validation Rule Edit screen for the 'lease_end_date' rule. The 'Rule Name' is set to 'lease_end_date'. The 'Error Condition Formula' field contains the formula `End_date__c > start_date__c`. A dropdown menu for 'Functions' is open, showing various mathematical functions like ABS, ACOS, ADDMONTHS, AND, OR, NOT, and ASIN. The 'Description' field is empty. The status is 'Active'.

5. Enter the Error Message as “Your End date must be greater than start date”, select the Error location as Field and select the field as “start date”, and click Save.



Phase 4: Project Development

Milestone 1: Email Templates

We use email templates to increase productivity and ensure consistent messaging. Email templates with merge fields let you quickly send emails that include field data from Salesforce records like contacts, leads, or opportunities. You can use email templates when emailing groups of people—with list email or mass email—or just one person.

Salesforce email templates are the easiest way to get your emails done. They help you create and send quick emails that include merge fields from Salesforce records like Contacts, Leads, Opportunities, or Custom Objects.

When you have a large number of contacts or leads in Salesforce, it can be difficult to keep track of who needs to be notified about new information. Salesforce email templates allow you to combine all these contacts or leads into one email and then send it out simultaneously.

Activity 1: Create Email Template For Tenant Leaving:

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Appointment for Unauthenticated User using Appointment Types - For Amazon Chime.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with Amazon Chime.	sfdcadmin	10/22/2025
Edit Del	Appointment for Unauthenticated User using Appointment Types - For third party.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with third party video applications.	sfdcadmin	10/22/2025
Edit Del	Appointment for Unauthenticated User using Engagement Channels-For Amazon Chime.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with Amazon Chime.	sfdcadmin	10/22/2025
Edit Del	Appointment for Unauthenticated User using Engagement Channels-For third party.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with third party video applications.	sfdcadmin	10/22/2025
Edit Del	Canceled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm canceling of a service appointment.	sfdcadmin	10/22/2025
Edit Del	Commerce Reorder Portal Invitation	Custom	✓	Invite a contact to a Commerce Reorder Portal.	autoproc	10/22/2025
Edit Del	Group Service Appointments Enrollment Confirmation Email	Custom	✓	Email Template to confirm enrollment of an attendee to a Group service appointment.	sfdcadmin	10/22/2025
Edit Del	Leave approved	Text	✓		SL	10/28/2025
Edit Del	Leave rejected	Text	✓		SL	10/28/2025

2. Click on >> New Email Template==>Choose text
3. Folder : Unfiled public Classic Email templates
4. Click on available for use
5. Email Template Name is “tenant leaving”
6. Template Unique Name : Auto populated
7. Subject : ” request for approve the leave”
8. Email body :
9. Dear {!Tenant__c.CreatedBy},
Please approve my leave

10. Save

Activity 2: Create Email Template For Leave Approved:

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template====>Choose text

Folder : Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is "Leave approved"

4. Template Unique Name : Auto populated

5. Subject : " Leave approved"

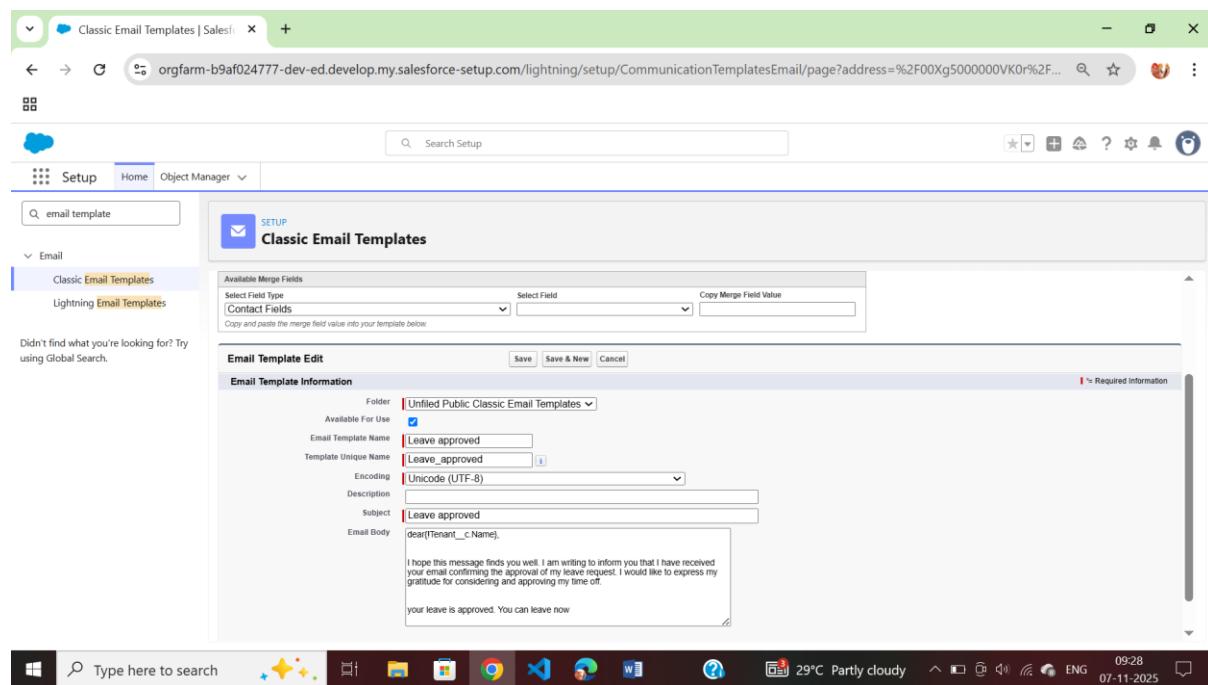
6. Email body :

dear{!Tenant__c.Name},

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

your leave is approved. You can leave now

7. Save



Activity 3: Create Email Template For Rejection For Leave:

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >>New Email Template==>Choose text
Folder : Unfiled public Classic Email templates

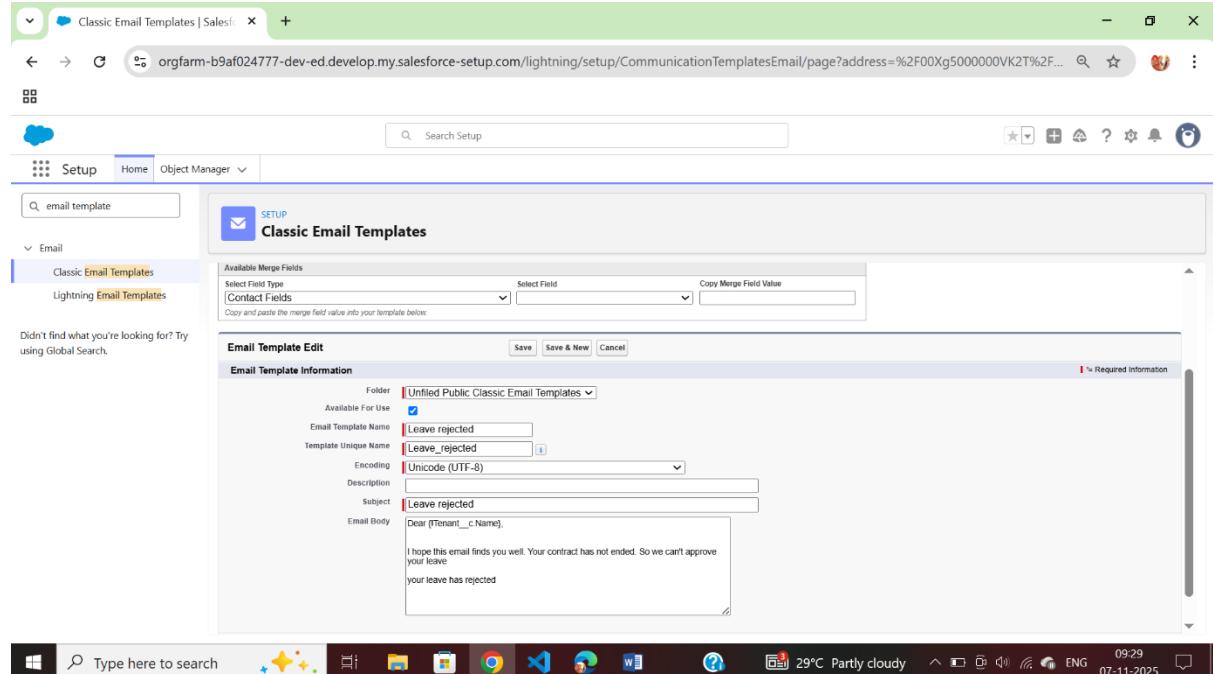
Click on available for use

3. Email Template Name is "Leave rejected"
4. Template Unique Name : Auto populated
5. Subject : " Leave rejected"
6. Email body :

Dear {!Tenant__c.Name},

I hope this email finds you well. Your contract has not ended. So we can't approve your leave
your leave has rejected

7. Save



Activity 4: Create Email Template For Monthly Payment:

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template==>Choose text

Folder : Unfiled public Classic Email templates
Click on available for use

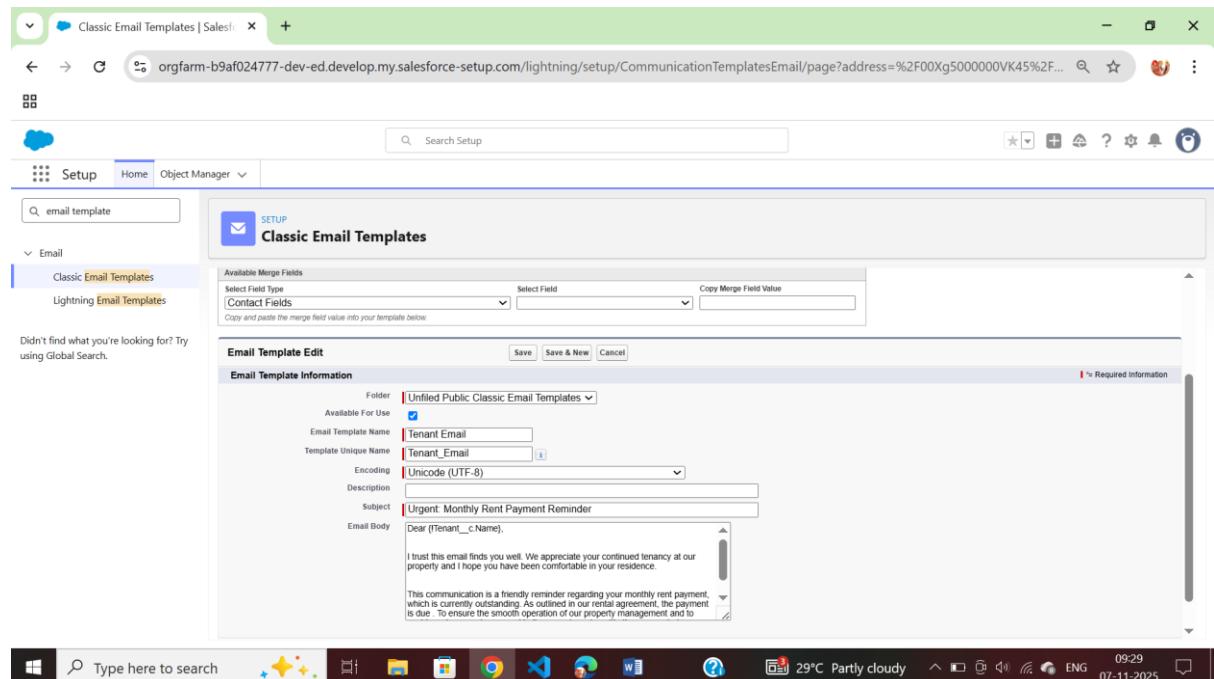
3. Email Template Name is "Tenant Email"
4. Template Unique Name : Auto populated
5. Subject : " Urgent: Monthly Rent Payment Reminder"
6. Email body :

Dear {!Tenant__c.Name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due . To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

7. Save



Activity 5: Create Email Template For Successful Payment:

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >> New Email Template==>Choose text

Folder : Unfiled public Classic Email templates
Click on available for use

3. Email Template Name is "tenant payment"

4. Template Unique Name : Auto populated

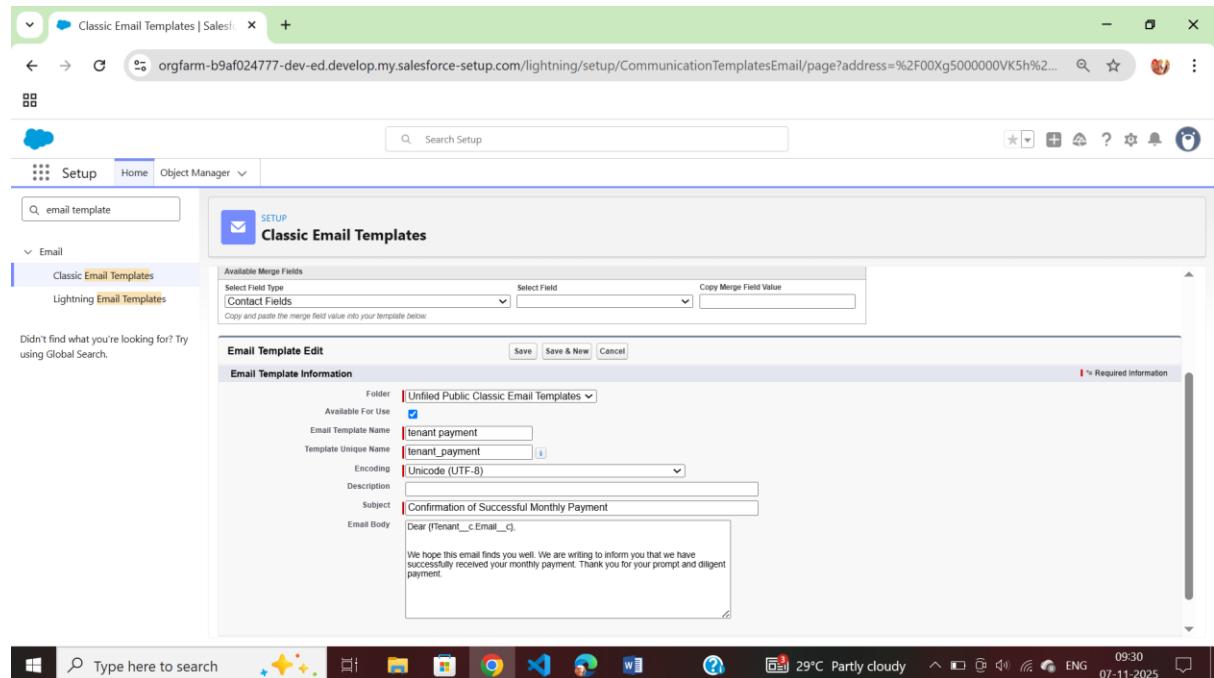
5. Subject : " Confirmation of Successful Monthly Payment"

6. Email body :

Dear {!Tenant__c.Email__c},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

7. Save



Milestone 2: Approval Process

What Is Approval Process In Salesforce?

The Approval Process is an automated process that an org uses to approve records in Salesforce. For example, When In the organization, someone is not able to decide a particular thing then he can ask someone else for approval. So, for such frequent cases or situations, one can define the approval process. So, Users can take benefit of such an approval process whenever needed. Records submitted for approval are approved by the user(s) in the organization. These users are called Approvers. A single Approval process is bound to a single object because when a rule is defined, this object influences the fields that will be available to set the criteria. An approval process consists of finalizing the basic properties of the approval process (as shown in the below image), approval steps, and actions to be executed.

Actions In Salesforce Approval Process

There are 4 actions present except the approval steps which complete an approval process, following are:

1. Initial Submission Actions

Initial submission actions are the actions that occur when a user first submits a record for approval. By default, an action to lock the record runs automatically on initial submission. Initial submission actions can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

2. Final Approval Actions

Final Approval actions are the actions that occur when a record is approved from all the approval steps. It also locks or unlocks the record, as specified. It can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

3. Final Rejection Actions

Final Rejection actions are the actions that occur when a record is rejected from any of the approval steps. It also locks or unlocks the record, as specified. It can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

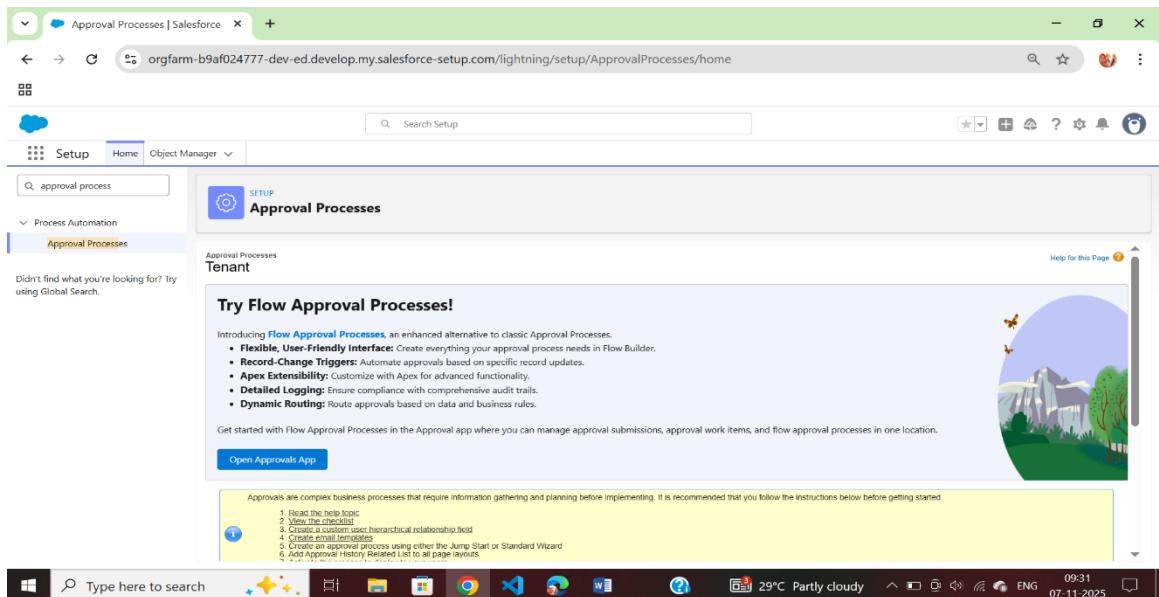
4. Recall Actions

Recall actions are the actions that occur when a record is recalled after submission for approval. It can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

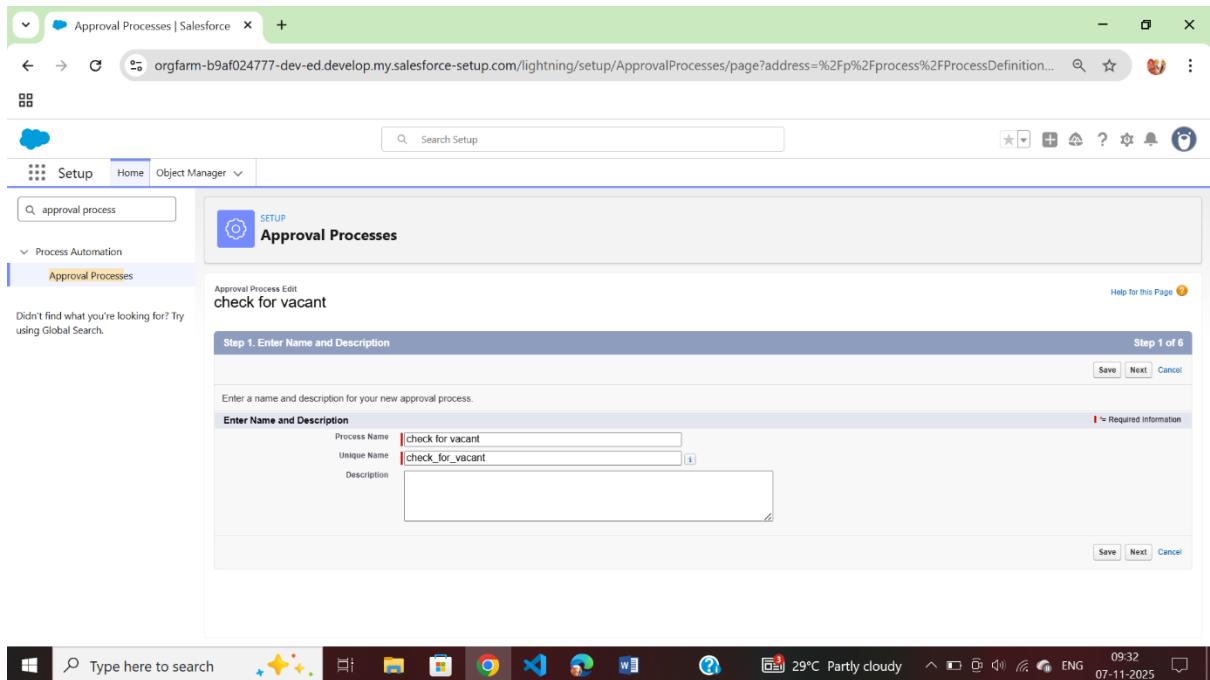
Activity 1: Create Approval Process For Check For Vacant:

To create fields in an object:

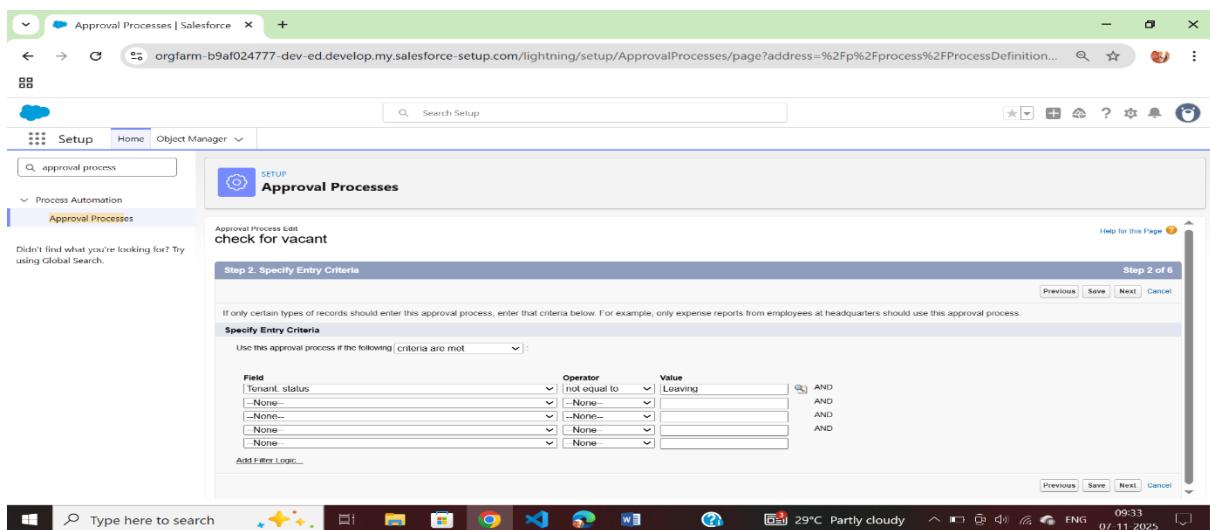
1. Go to setup >> Approval Processes in quick find bar>>click on it.



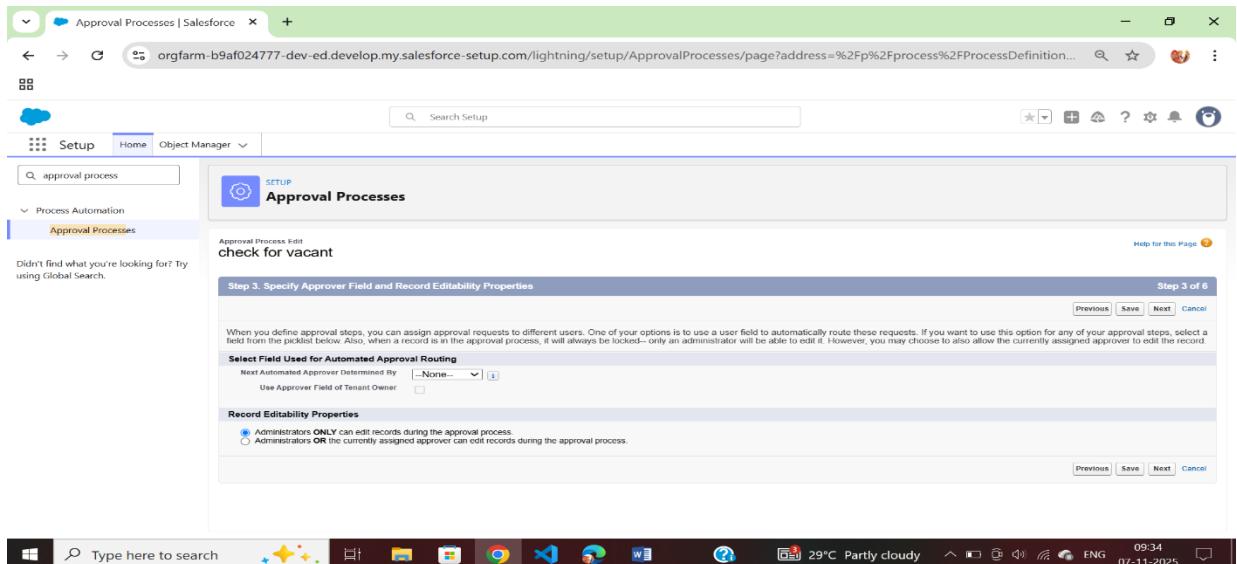
2. Manage Approval Process For >> “Tenant” from the drop down.
3. Click on “Create New Approval Process” >> Use standard setup wizard.
4. Process Name “check for vacant” >> Click Next.



5. Field “Tenant:status” >> Operator : Not equals , Value >> Click on the lookup filter icon and select “Leaving”.
6. Click insert field,then click Next.



7. Next Automated Approver determined by “None” from the drop down.
8. Select the “Administrators ONLY can edit records during the approval process”. Then Next

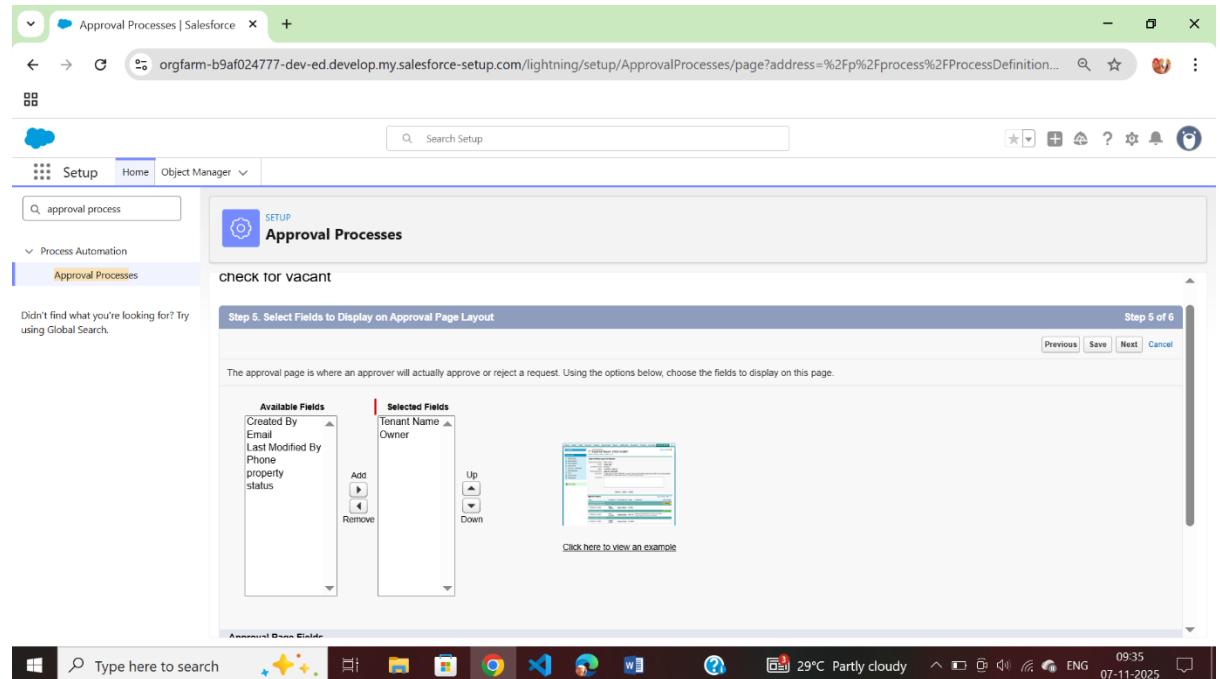


9. Click on next leave the email template click on next

10.From the available fields select >> Tenant Name, and then add >>Add it to the selected.Then Next.

- Make sure Display approver history is checked.
- And under security settings check the “Allow approvers to access the approval page only from within the Salesforce application. (Recommended)” option.

11. Submitter type Search>>Owner, Allowed Submitters>>Property Owner.Then Next.



- Then click save.

What Would You Like To Do Now?

You have just created an approval process. However, you cannot activate this process until you define at least one approval step. Would you like to do that now?

- Yes, I'd like to create an approval step now.
- I'll do this later. Take me to the approval detail page to review what I've just created.
- I'll do this later. Take me back to the listing of all approval processes for this object.

Go!

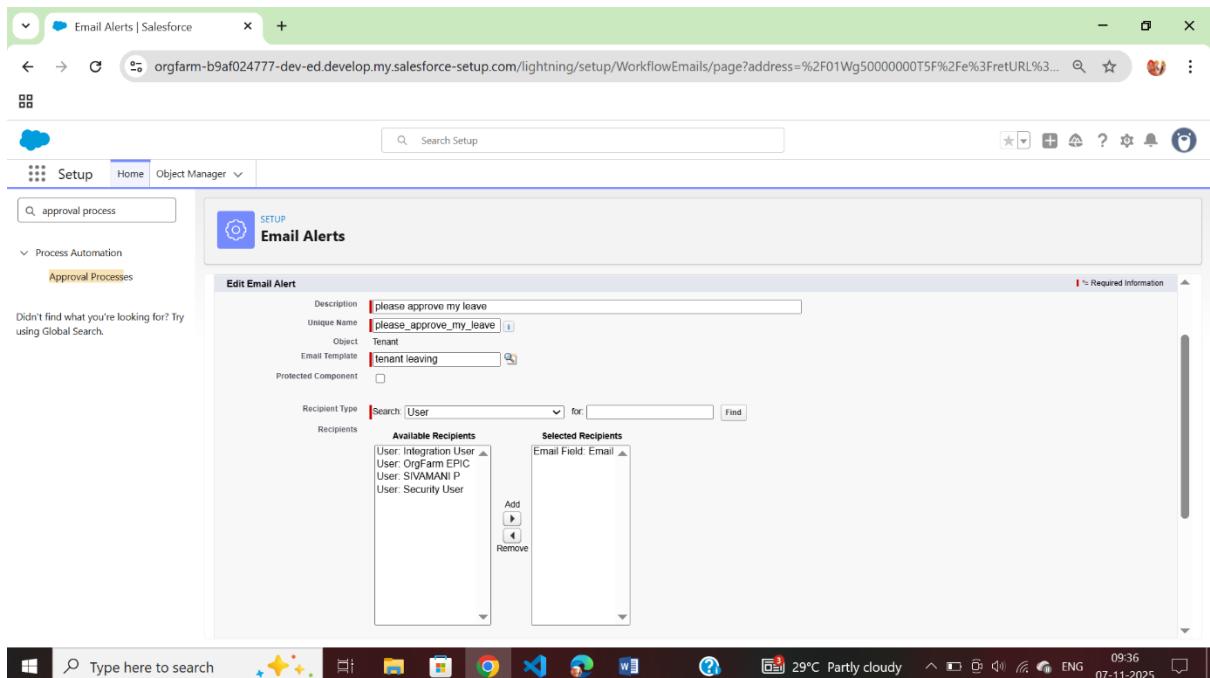
- Click on "I'll do this later. Take me back to the listing of all approval process for this object"
- Click go

Activity 2: Initial Submission Action:

1. Under initial submission action click on add new and then select email alert.

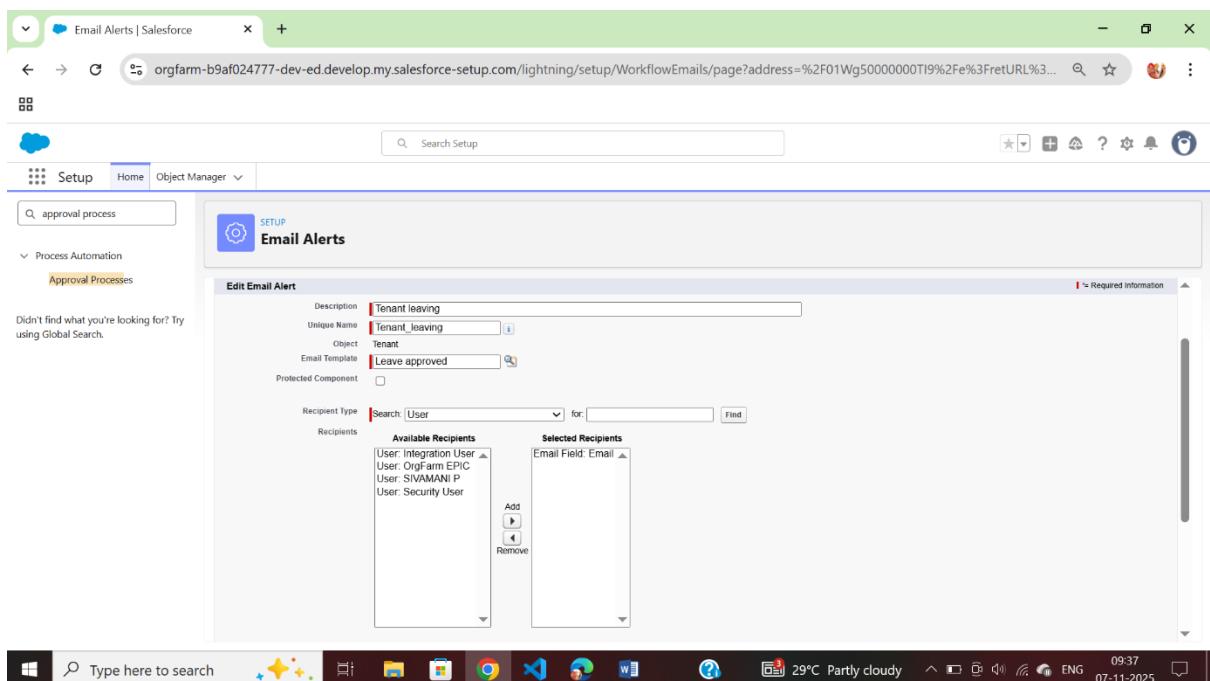
Action	Type	Description	Task
Record Lock		Email Alert	Lock the record
			Field Update

2. Description: "please approve my leave".
3. unique name : auto populated
4. Email template : tenant leaving
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user's email
8. Click save.



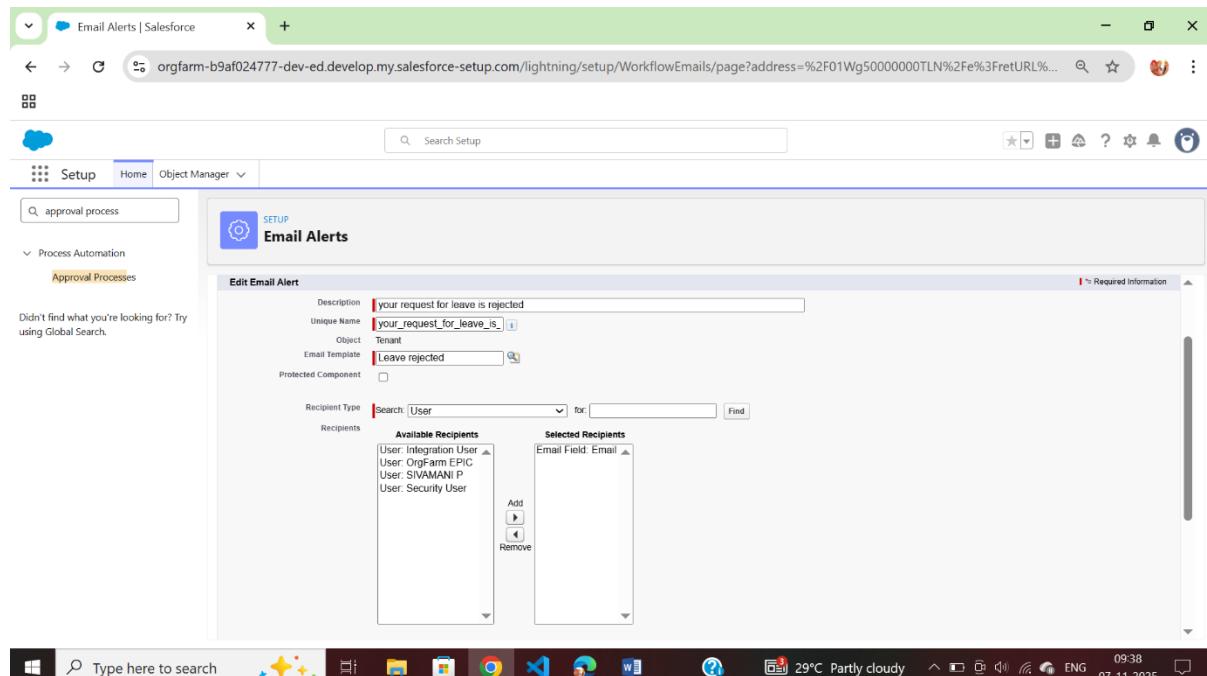
Activity 3: Final Approval Action:

1. Under Final approval action click on new and then select email alert.
2. Description: "Tenant leaving".
3. unique name : auto populated
4. Email template : Leave approved
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user's email
8. Click save.



Activity 4: Final Rejection Action:

- Under final rejection action click on add new and then select email alert.
- Description: "your request for leave is rejected".
- unique name : auto populated
- Email template : leave rejected
- Recipient type : Email field
- Available Recipients : Email field : Email
- From Email address : Current user's email
- Click save



Milestone 3: Apex Trigger

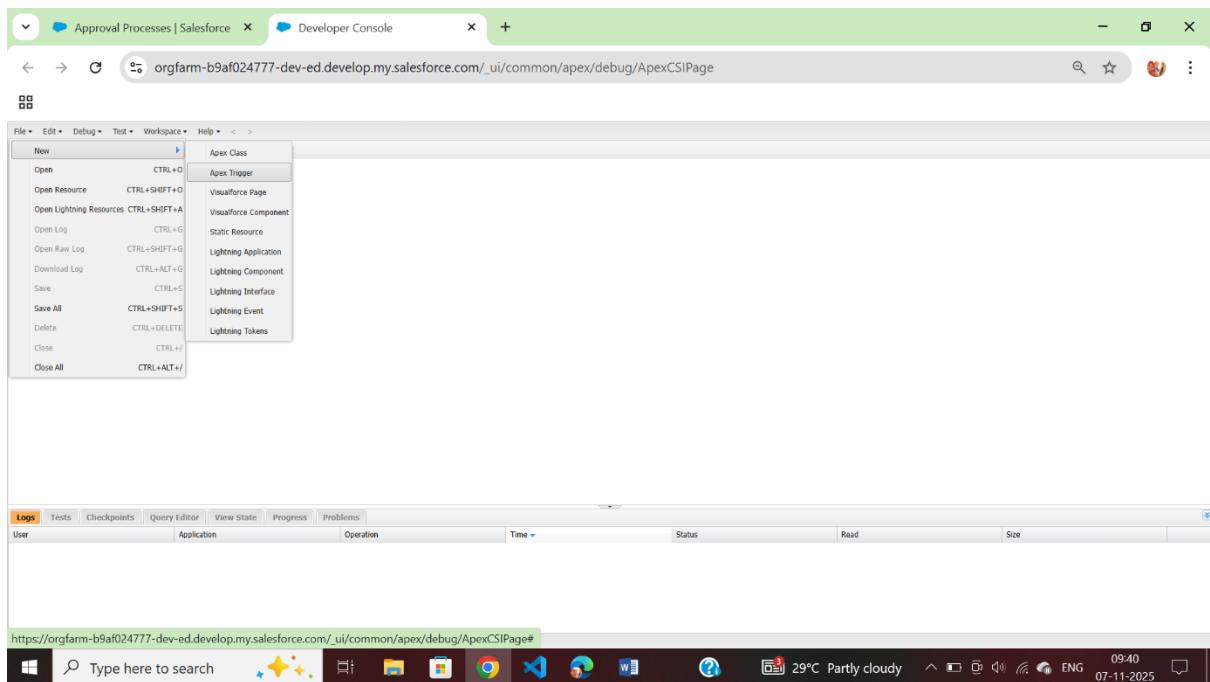
Use case:

The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

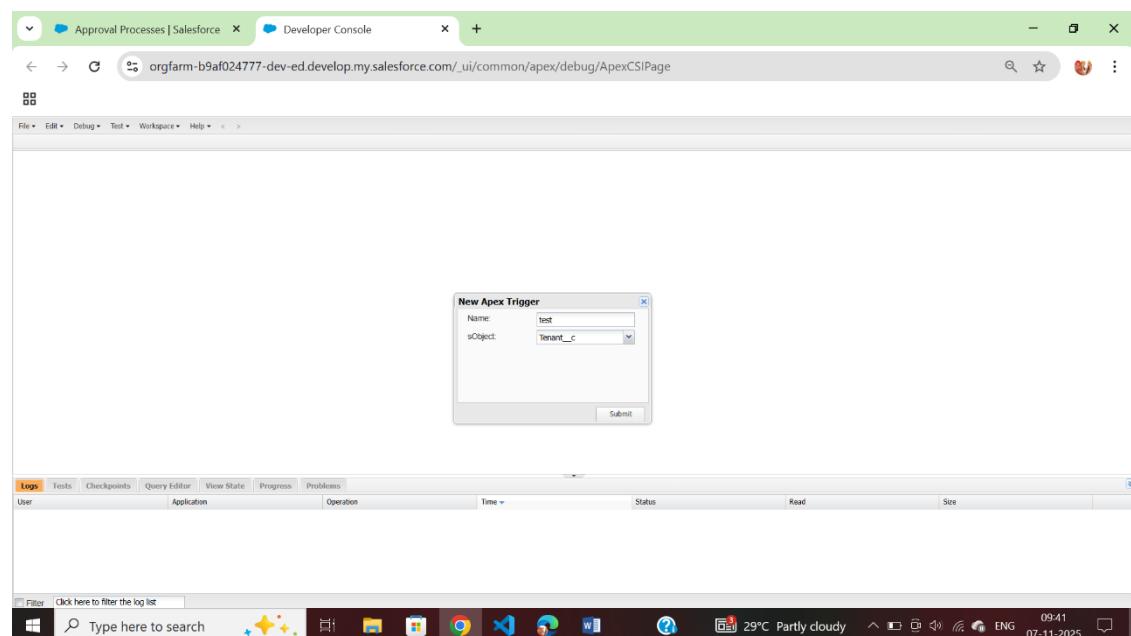
Write a code to achieve this requirement using Salesforce developer skills to fulfill the Managers requirement.

Activity 1: Create An Apex Trigger:

- To create a new Apex Class follow the below steps:
Click on the file >> New ? Apex Class.



2. Give the Apex Trigger name as “test”, and select “Tenant__c” from the dropdown for sObject.



3. Click Submit.

4. Now write the code logic here

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Approval Processes | Salesforce' and 'Developer Console'. The main area displays an Apex trigger named 'test.apxt' with the following code:

```

1 trigger test on Tenant__c (before insert)
2 {
3     if(trigger.isInsert && trigger.isBefore){
4         testHandler.preventInsert(trigger.new);
5     }
6 }

```

Below the code editor, there is a tabs bar with 'Logs' selected, followed by 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. At the bottom of the screen, a Windows taskbar shows various application icons and the system tray.

Trigger Code:

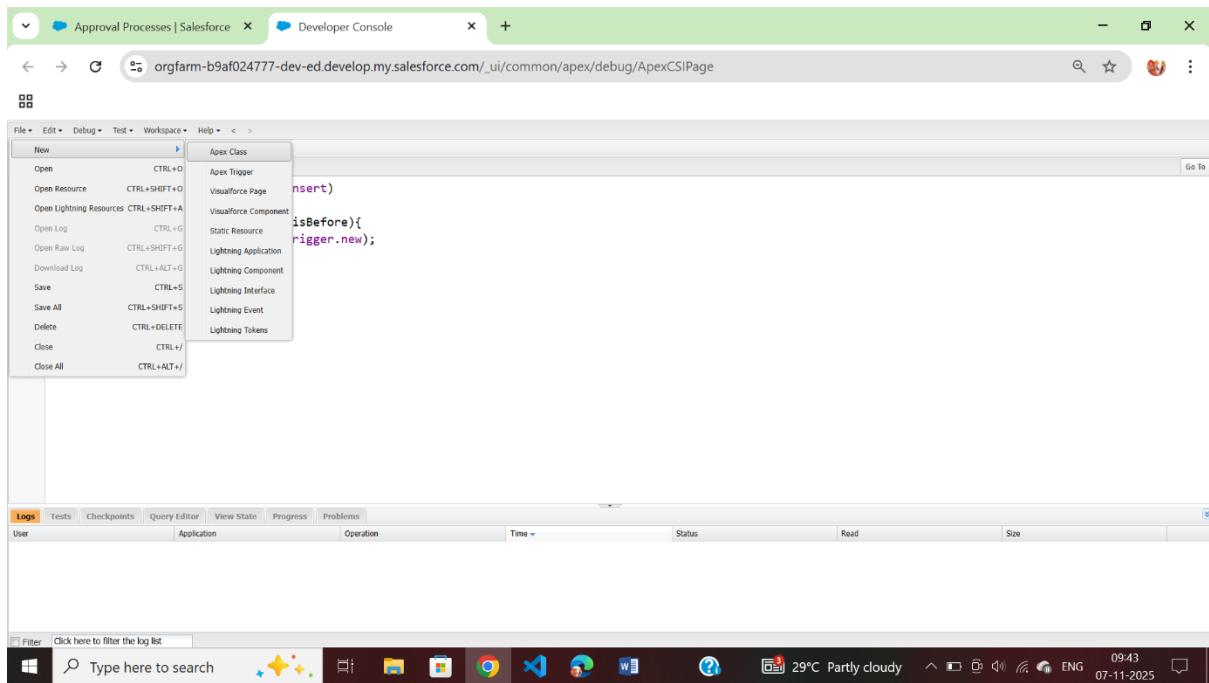
```

trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}

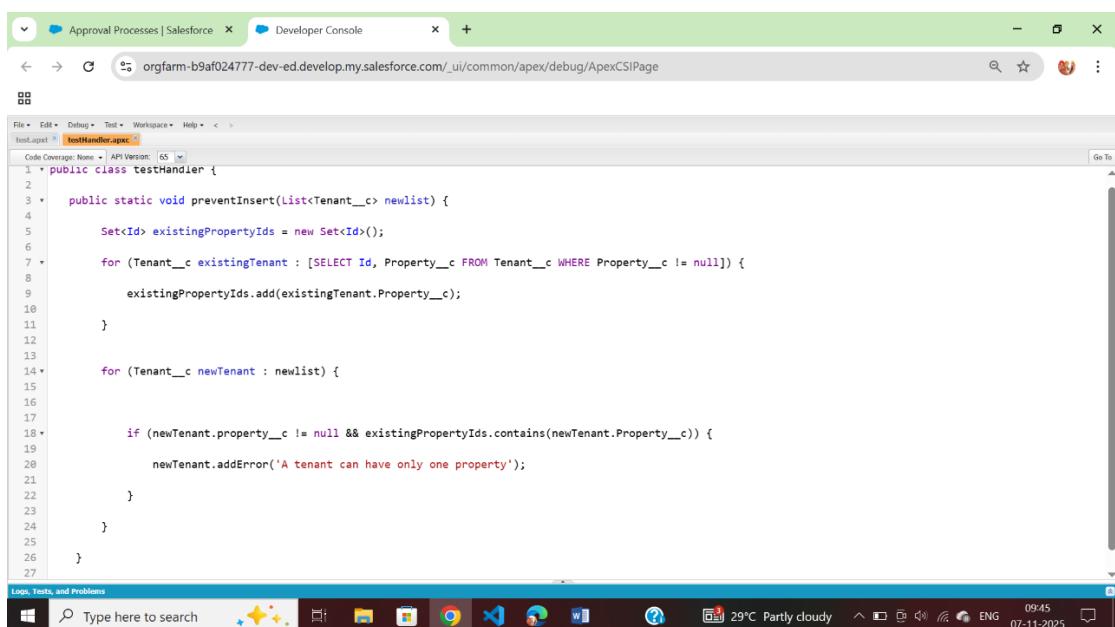
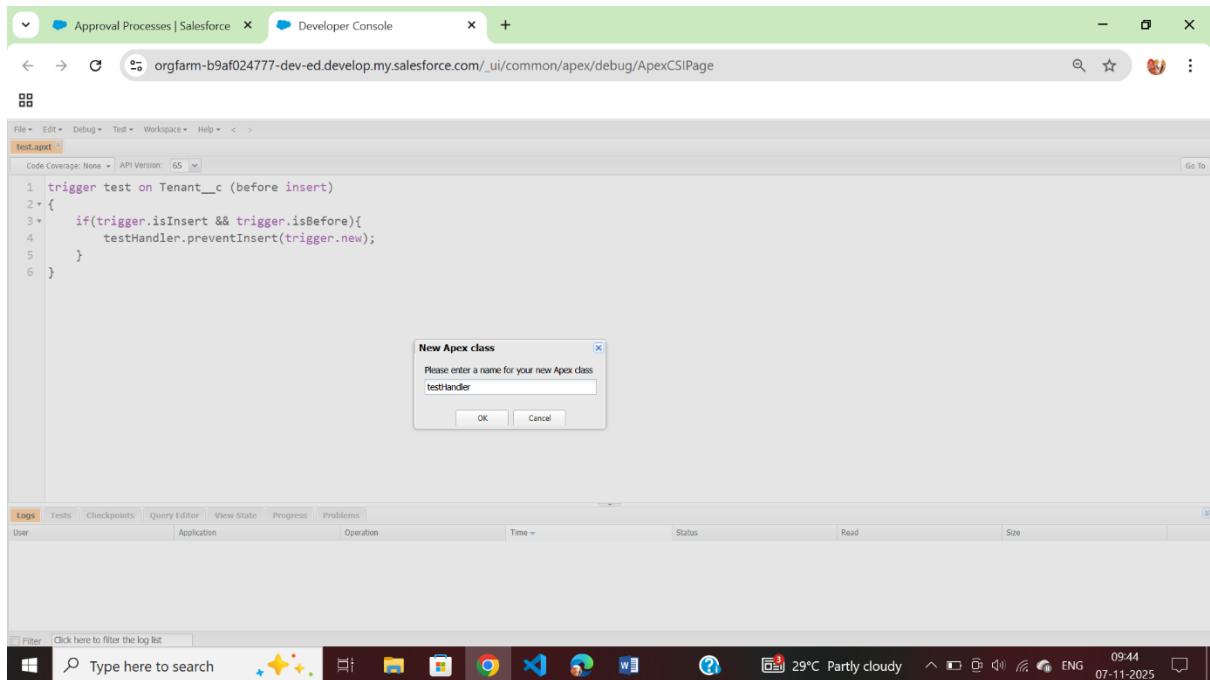
```

Activity 2: Create An Apex Handler Class:

1. To create a new Apex Class follow the below steps:
Click on the file >> New >>Apex Class.



2. Enter class name as testHandler.



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)) {
                newTenantaddError('A tenant can have only one property');
            }
        }
    }
}

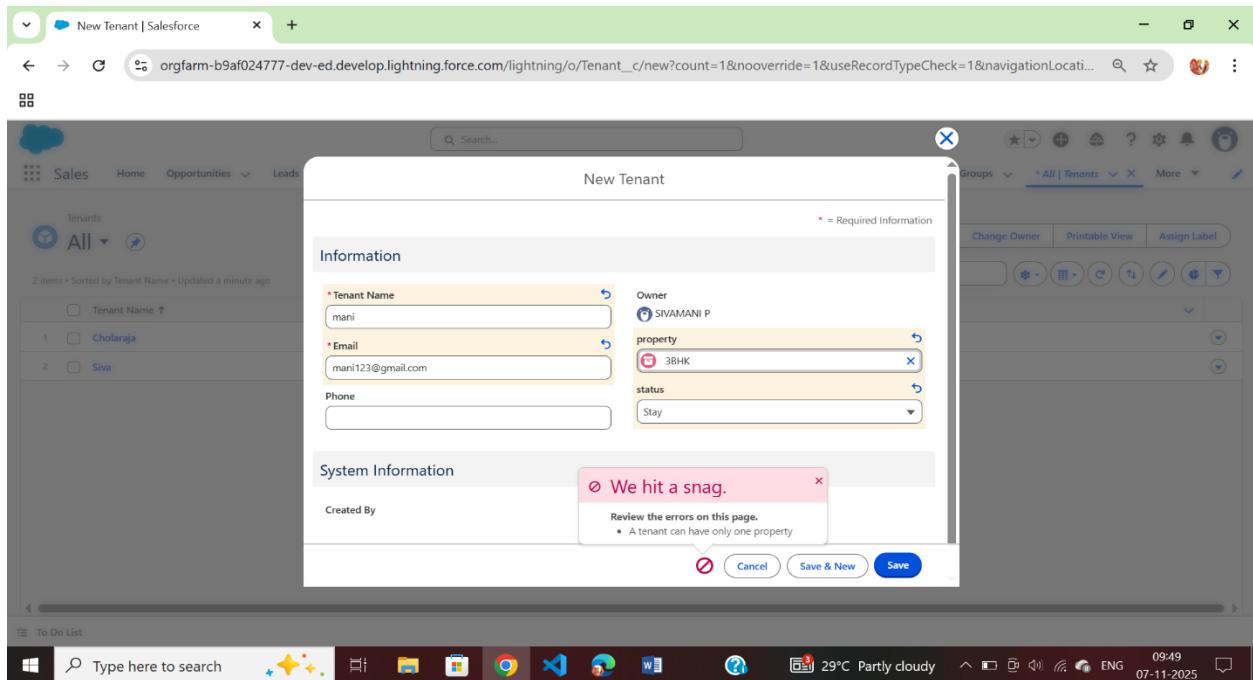
```

Activity 3: Testing The Trigger:

Try to create new tenant with the existing property then it shows the error

The screenshot shows the Salesforce Setup Home page. The top navigation bar includes 'Home | Salesforce', a search bar, and various icons. The main area features a 'Welcome, SIVAMANI' message and a 'Manage and customize Salesforce from Setup. Browse suggestions, explore features, and more.' banner. On the left, there's a sidebar with sections like 'App Launcher', 'Setup', 'Home', 'Object Manager', 'Tenants' (with a search bar), 'Popular Business Goals', and various system settings. A central panel displays two cards: 'Cross Cloud Connect with Sales Prospects and Customers' and 'Cross Cloud Track & Manage Customer Data'. At the bottom, there's a task bar with links like 'Subscription Management' and the URL 'https://orgfarm-b9af024777-dev-ed.develop.my.salesforce-setup.com/lightning/o/Tenant_c/home'.

The screenshot shows the Salesforce Tenant list page. The top navigation bar includes 'All | Tenants | Salesforce', a search bar, and various icons. The main area features a 'Sales' tab and a 'Tenants' section. The 'Tenants' section shows a list of two items: 'Cholaraja' and 'Siva', both with a checkbox next to them. There are buttons for 'New', 'Import', 'Change Owner', 'Printable View', and 'Assign Label'. Below the list is a search bar and a toolbar with various icons. The URL at the bottom is 'https://orgfarm-b9af024777-dev-ed.develop.lightning.force.com/lightning/o/Tenant__c/list?filterName=All'.



Milestone 4: Flow

What is a flow ?

In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data. Flow Builder is the declarative interface used to build individual flows.

Flows fall into five categories:

Screen Flows: These are flows that have a UI element and require input from users. These types of flows are either launched as an action or embedded as an element on a Lightning page.

Schedule-Triggered Flows: These autolaunched flows launch at a specified time and frequency for each record in a batch, and they run in the background.

Autolaunched Flows: Run automated tasks with this flow type. Autolaunched flows can be invoked from other flows (subflow), process builder, from within an Apex class, from a set schedule, from record changes, or from platform events.

Record-Triggered Flows: These autolaunched flows run in the background either before a record save or after the record is saved when a record is created, updated, or deleted.

Platform Event-Triggered Flows: When a platform event message is received, these autolaunched flows run in the background.

When and why should we use a flow

If you need to generate a new automated business process, or user guided experience that does not reach the complexity threshold for Apex Code, then flow is your go-to tool. If you are modifying an existing process that was built with Process Builder or workflow, then you should consider a number of factors when deciding whether to modify the existing process or migrate it to Flow. Flows are able to create, edit, and delete records in Salesforce, send emails, show relevant data and gather input from users, and generate outbound messages.

Activity 1: Create Flow For Monthly Payment:

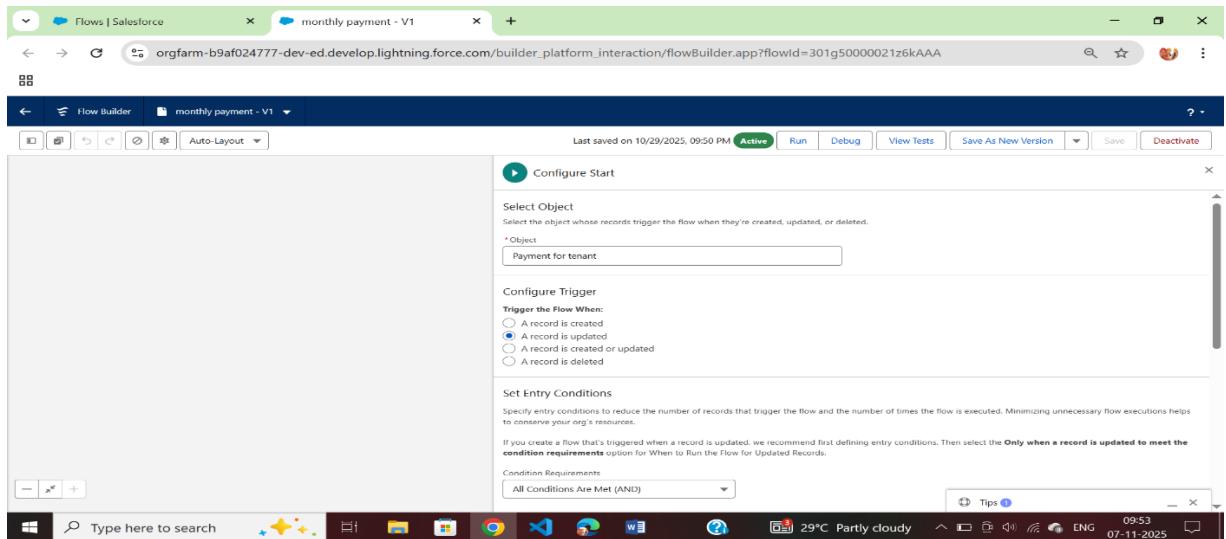
1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

The screenshot shows the Salesforce Flows page. The left sidebar has sections for Process Automation (Flows), Identity (Login Flows), and a search bar. The main area displays a table of flow definitions with columns for Flow Label, Process Type, Active, Template, Package State, and Last Modified Date. The table lists various flows like 'Add or Modify Service Appointment Attendees', 'Approvals Workflow: Evaluate Approval Requests', and 'Basic Approval Request'. A message at the top says 'Didn't find what you're looking for? Try using Global Search.'

2. Select the record Triggered flow. Click on create.

The screenshot shows the Flow Builder 'New Automation' page. It features a 'Get Started with Automations' section with a search bar and categories for Scheduled, Screen, and Autolaunched automations. Below this are sections for 'Frequently Used' automations: Record-Triggered Flow (selected), Screen Flow, Schedule-Triggered Flow, and Autolaunched Flow (No Trigger). The status bar at the bottom shows it's 09:50 on 07-11-2025.

3. Under Object select "Payment for tenant". Click on A record is updated.



4. Set Entry Conditions

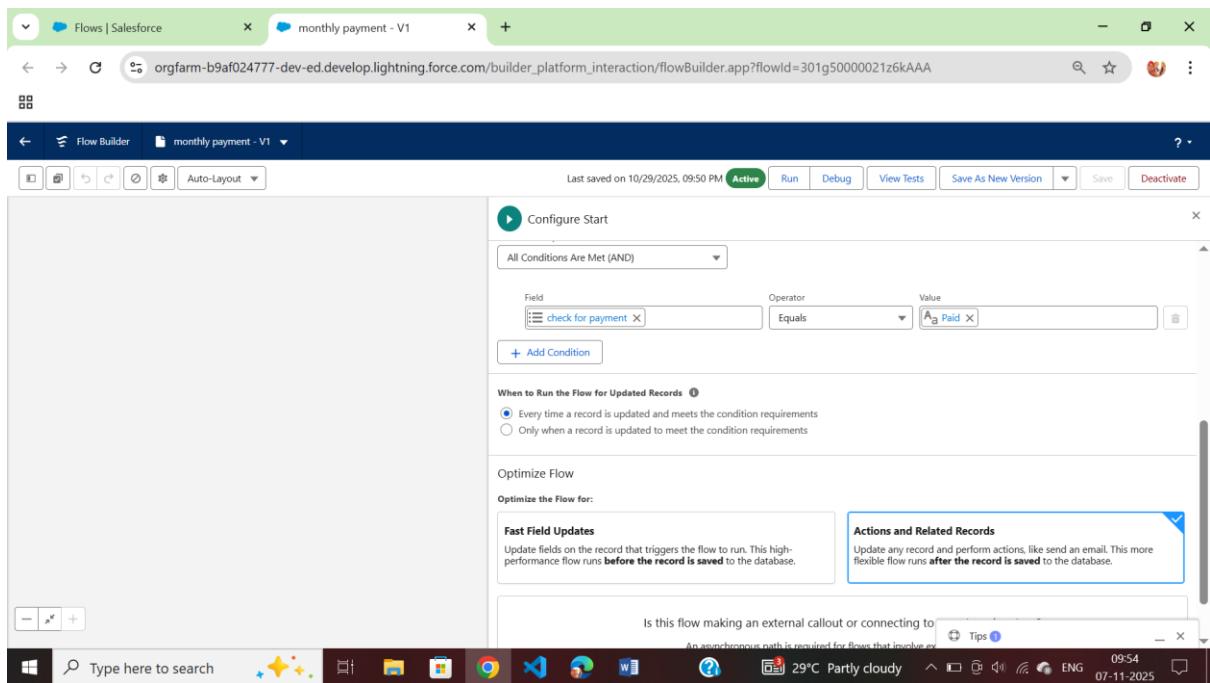
Under Condition Requirements

All Conditions are met

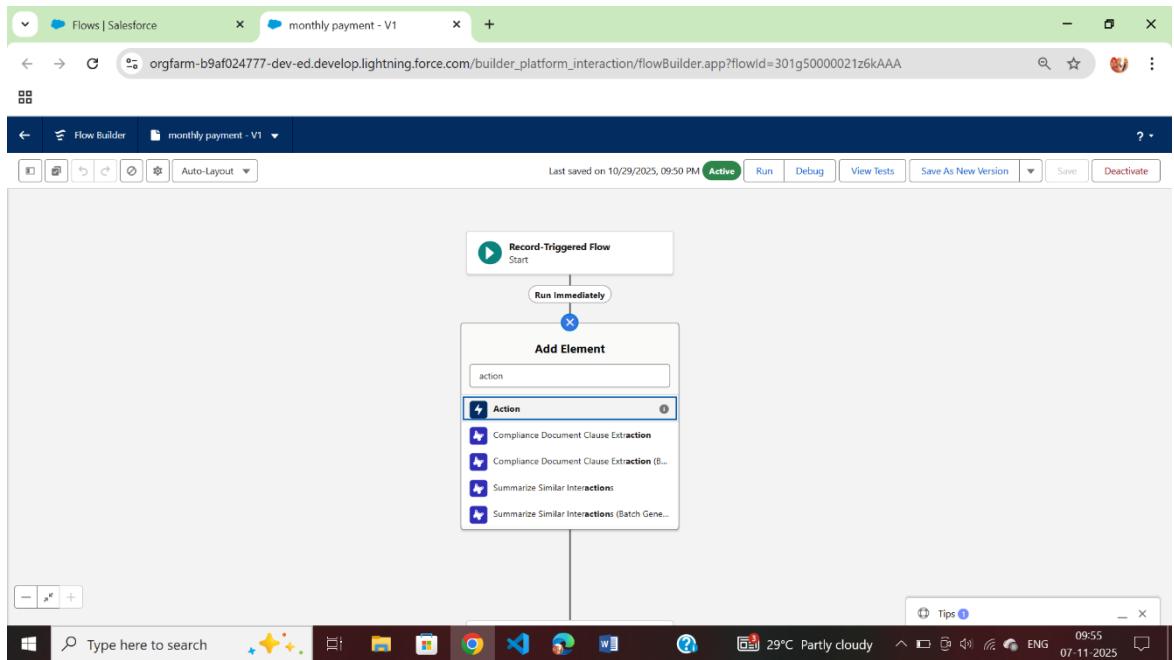
Field: check_for_payment__c	Operator: Equals	Value : paid
-----------------------------	------------------	--------------

5. Click on : Every time a record is updated and meets the condition requirements

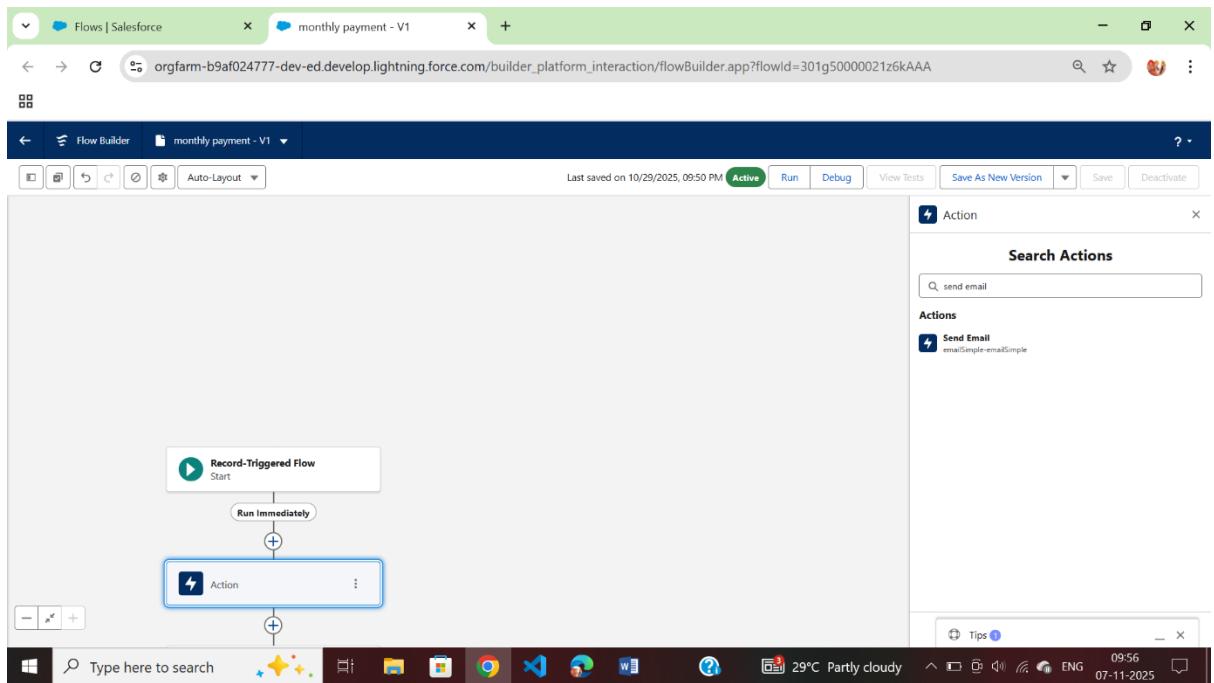
6. Click on : Actions and related records, done.



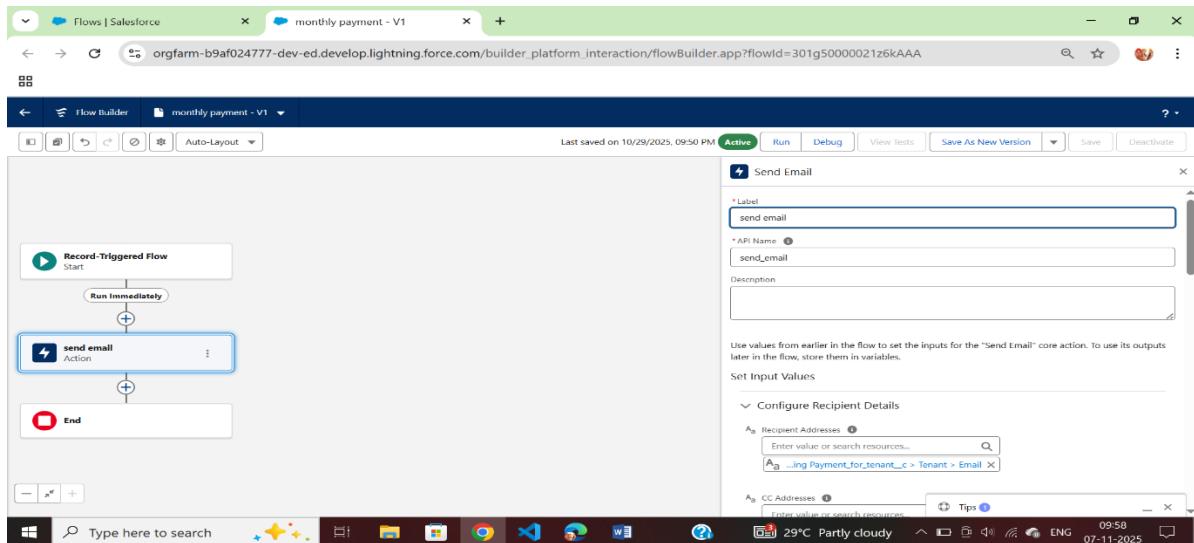
7. Under record trigger flow click on "+" icon and select action.



In action search for send email then click on send email (check below image)



8. Label : send email
10. API Name : send_email
11. Enable Body
12. Click on new resource



Under resource type select “Text Template”

API Name : emailbody

Under body: (paste the below text)

Dear {!\$Record.Tenant__r.Name},

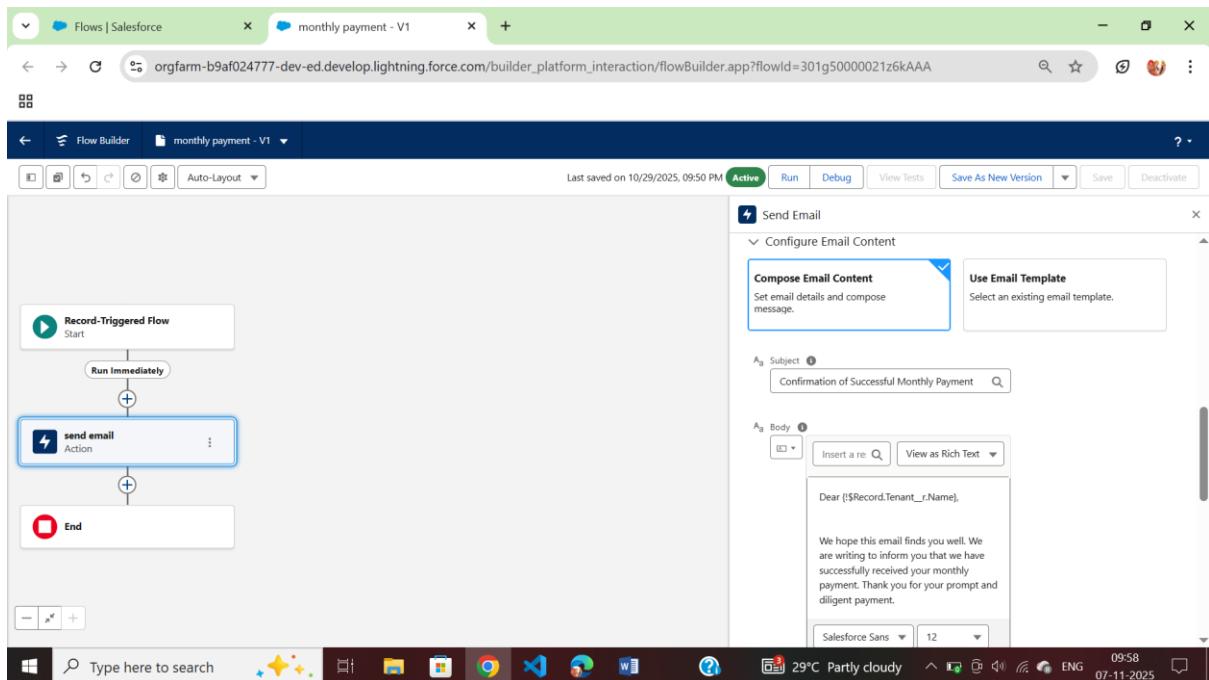
We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

14. Click Done.

15. Enable recipient Address List

Paste this ? {!\$Record.Tenant__r.Email__c}

16. Click Done



17. Enable subject

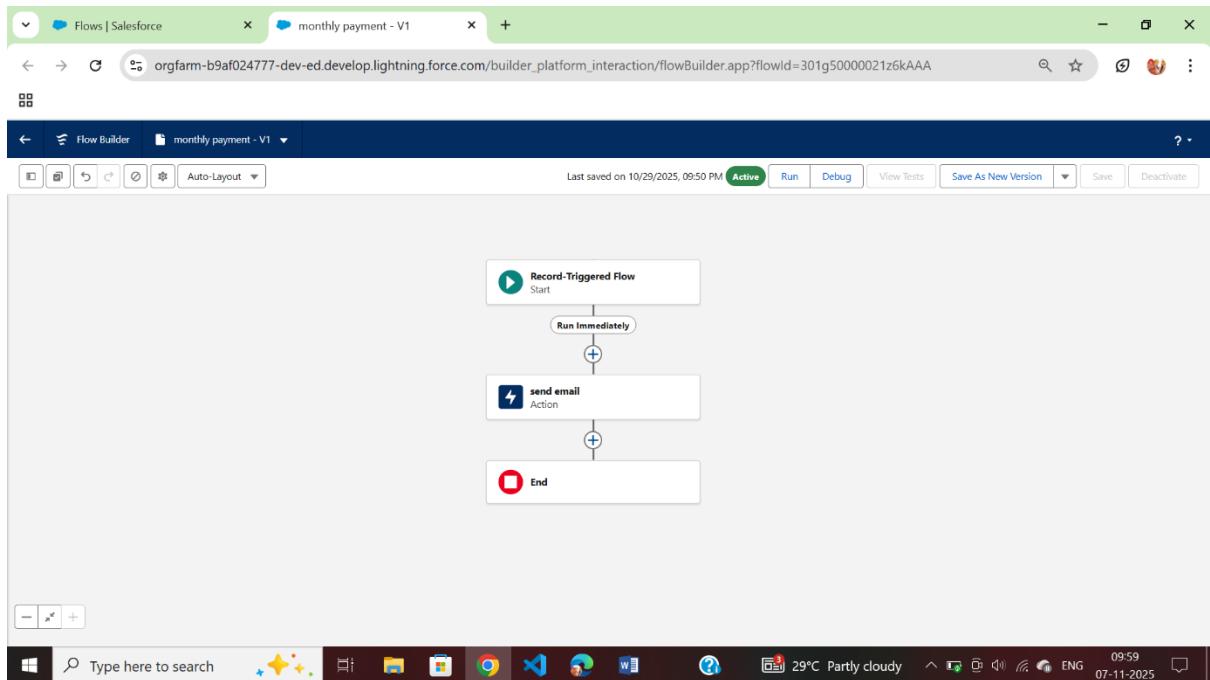
Pate this >> Confirmation of Successful Monthly Payment

18. Click on save

Flow label : monthly payment

Flow API Name : monthly_payment

Click on activate



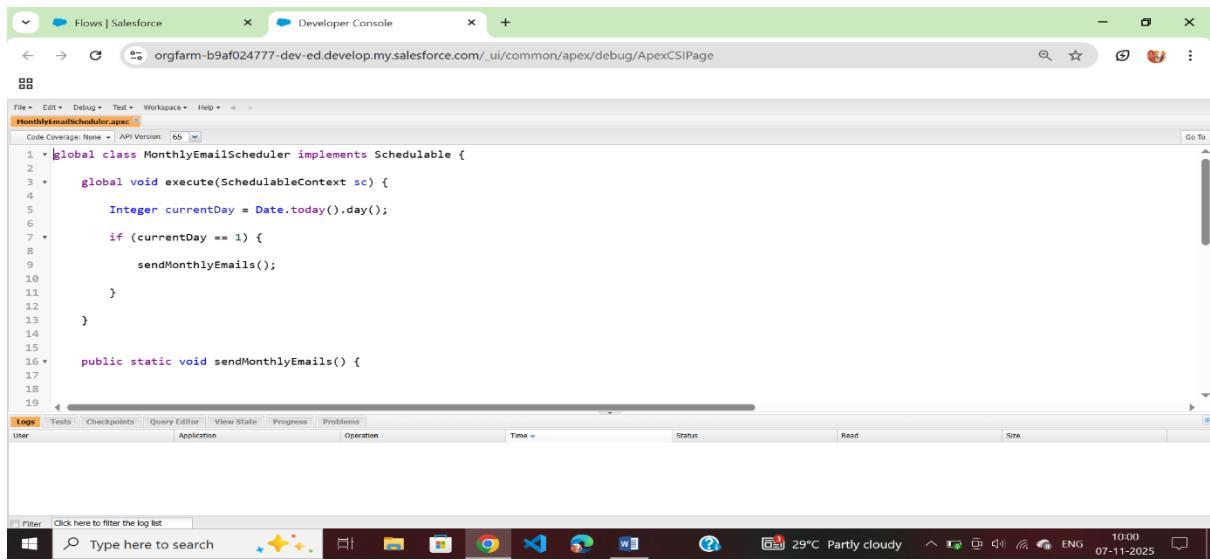
Milestone 5: Schedule Class

Activity 1: Create An Apex Class:

1. To create a new Apex Class follow the below steps:

Click on the file >> New >> Apex Class.

2. Enter class name as MonthlyEmailScheduler.



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});
    }
}
}

```

Save the code.

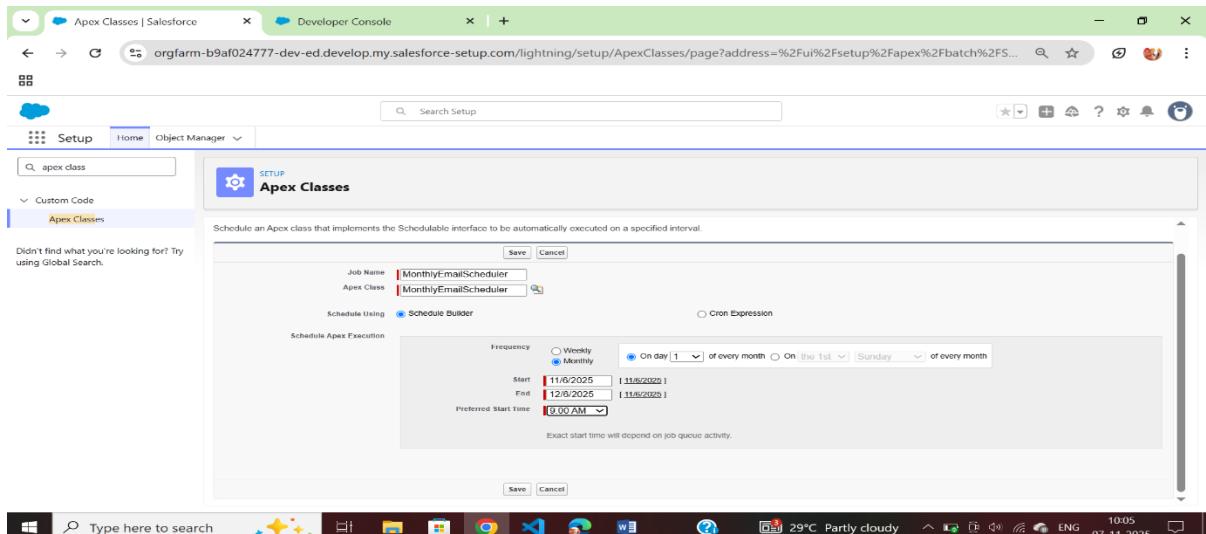
Activity 2: Schedule Apex Class:

1. Enter Apex class in quick find box
2. Select schedule Apex

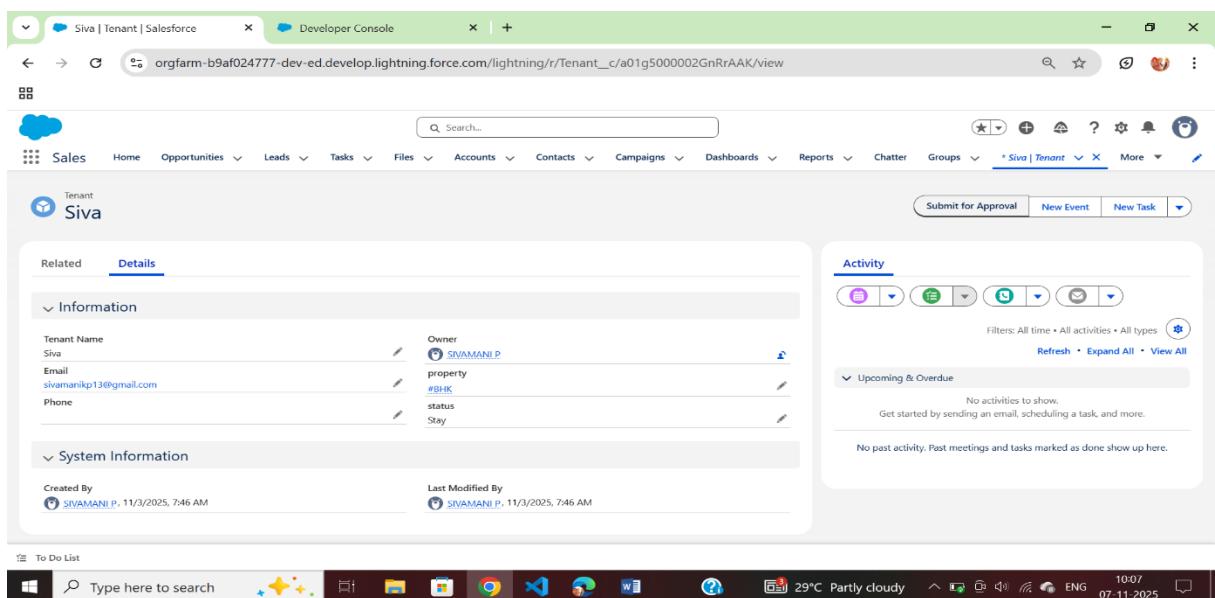
The screenshot shows the Salesforce Setup Apex Classes page. The URL is <https://orgfarm-b9af024777-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApexClasses/home>. The page title is "Apex Classes". It displays a table of Apex classes with columns: Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table contains several entries, including "DeveloperEditionUtils", "DeveloperEditionUtilsTest", "MonthlyEmailScheduler", "PostinstallScript", "PostinstallScriptTest", and "IsHandler". A green banner at the top indicates "Percent of Apex Used: 0.03%" and "You are currently using 1,856 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." Below the table, there are buttons for "Create New View" and "Schedule Apex". The status bar at the bottom shows the URL, a search bar, taskbar icons, weather (29°C Partly cloudy), system status (ENG 07-11 2025), and a timestamp (10:01).

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Security	DeveloperEditionUtils	devedapp	64.0	Active	164	OrgFarm EPIC, 10/22/2025, 7:53 AM	<input type="checkbox"/>
Edit	DeveloperEditionUtilsTest	devedapp	64.0	Active	261	OrgFarm EPIC, 10/22/2025, 7:53 AM	<input type="checkbox"/>
Edit Security	MonthlyEmailScheduler		68.0	Active	1,125	SIVAMANI.P, 10/29/2025, 9:23 AM	<input type="checkbox"/>
Edit Security	PostinstallScript	devedapp	64.0	Active	2,175	OrgFarm EPIC, 10/22/2025, 7:53 AM	<input type="checkbox"/>
Edit	PostinstallScriptTest	devedapp	64.0	Active	781	OrgFarm EPIC, 10/22/2025, 7:53 AM	<input type="checkbox"/>
Edit Del Security	IsHandler		65.0	Active	584	SIVAMANI.P, 10/28/2025, 2:43 AM	<input type="checkbox"/>

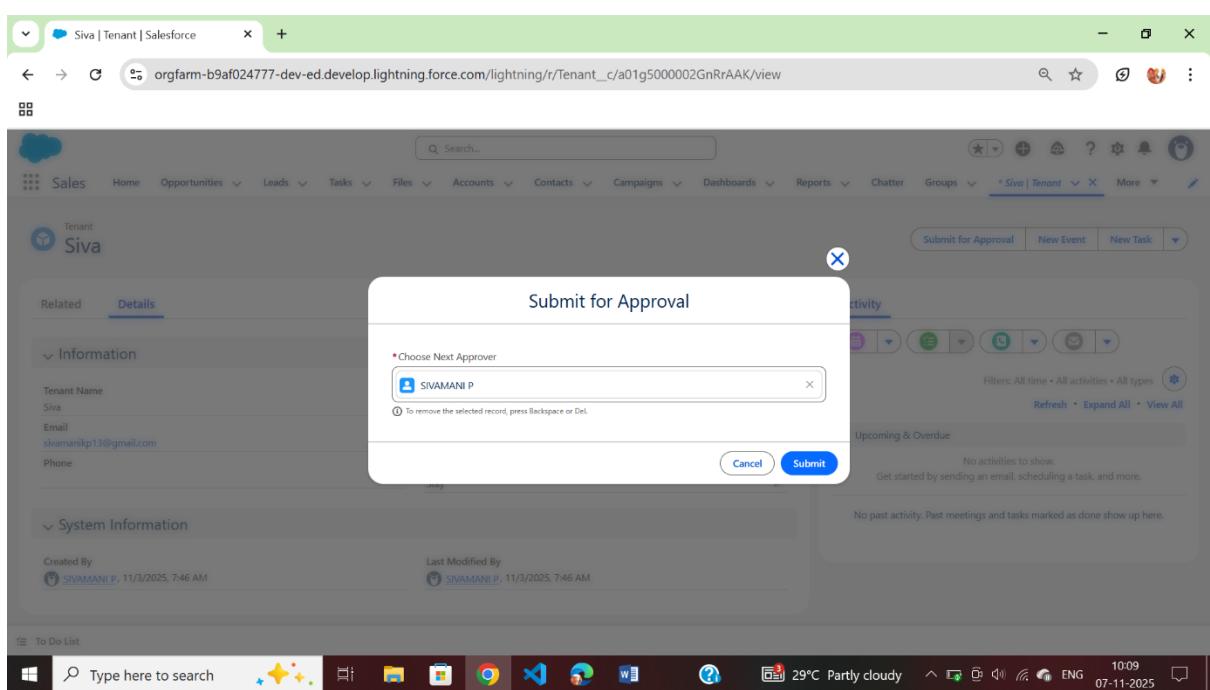
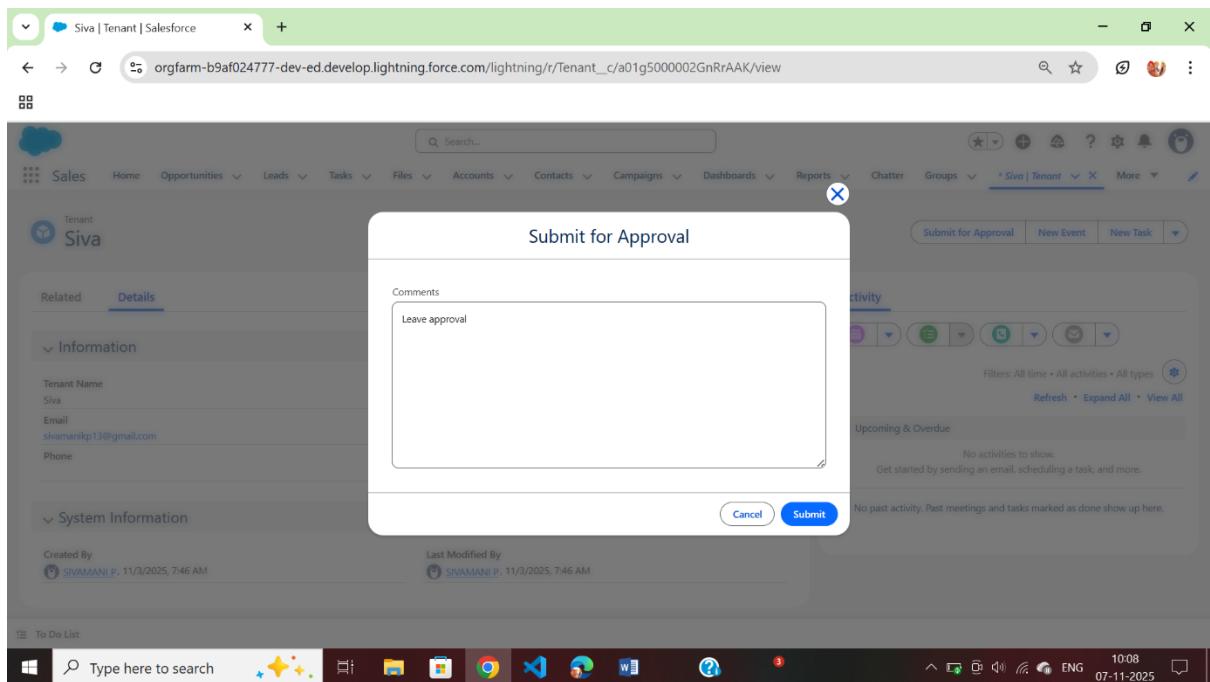
3. Enter job Name : MonthlyEmailScheduler
4. Apex class : MonthlyEmailSchedule
5. Frequency : Monthly==>select on day 1
6. Start date : 04/12/2023
7. End date : 04/01/2024
8. Preferred start time : 09:00 am
9. save



Testing the approval process



Enter any comment and click on submit



Siva | Tenant | Salesforce

orgfarm-b9af024777-dev-ed.develop.lightning.force.com/lightning/r/Tenant__c/a01g5000002GnRrAAK/view

Tenant Siva

Related Details

Information

Tenant Name: Siva
Email: sivamanikp13@gmail.com
Phone:

Owner: SIVAMANI P
property: #BHK
status: Stay

System Information

Created By: SIVAMANI P, 11/3/2025, 7:46 AM
Last Modified By: SIVAMANI P, 11/3/2025, 7:46 AM

Notifications

- SIVAMANI P is requesting approval for tenant
Tenant Name: Siva • Owner: SIVAMANI P
3 minutes ago
- Approval request for the tenant is approved
Siva
Nov 3, 2025, 9:18 PM
- Approval request for the tenant is approved
Cholaraja
Nov 3, 2025, 9:04 PM
- Approval request for the tenant is approved
Cholaraja
Oct 28, 2025, 2:27 PM
- New Guidance Center learning resource available
Define Your Sales Process
Learn how to guide reps through the sales process.
Oct 28, 2025, 2:10 PM

javasciptvoid()

Type here to search

29°C Partly cloudy

10:10 07-11-2025

Click on that notification

Siva Approval | Salesforce

orgfarm-b9af024777-dev-ed.develop.lightning.force.com/lightning/r/ProcessInstanceWorkitem/04ig50000021cjAAA/view

Approval Request
Tenant Approval Pending

Submitter: SIVAMANI P Date Submitted: Nov 6, 2025 Actual Approver: SIVAMANI P Assigned To: SIVAMANI P

Approve Reject Reassign

Details

Approval Details

Tenant Name: Siva Owner: SIVAMANI P

Submitter Comments

SIVAMANI P
Leave approval
Nov 6, 2025, 8:36:54 PM

javasciptvoid()

Type here to search

29°C Partly cloudy

10:10 07-11-2025

click on approve
Give any comment and submit

The screenshot shows a Salesforce Lightning interface. At the top, the URL is `orgfarm-b9af024777-dev-ed.develop.lightning.force.com/lightning/r/ProcessInstanceWorkitem/04ig500000021cjAAA/view`. The main page displays an 'Approval Request' for a 'Tenant Approval' step. The status is 'Pending'. The submitter is SIVAMANI P, and the date submitted is Nov 6, 2025. The actual approver is also SIVAMANI P. A comment section contains the message 'you are approved for leave'. On the right, there's a sidebar for 'Submitter Comments' from SIVAMANI P, dated Nov 6, 2025, at 8:56:54 PM, stating 'Leave approval'. Below the main content is a 'Comments' section with a 'Cancel' and 'Approve' button. The bottom of the screen shows a Windows taskbar with various icons and system status.

The screenshot shows the same Salesforce Lightning interface after the approval has been completed. The status is now 'Approved'. The notifications sidebar on the right lists four approval requests for the tenant, each with a timestamp and user name: 'Approval request for the tenant is approved Siva' (Nov 6, 2025, 9:18 PM), 'Approval request for the tenant is approved Cholanja' (Nov 6, 2025, 9:04 PM), 'Approval request for the tenant is approved Cholanja' (Oct 26, 2025, 2:27 PM), and 'New Guidance Center learning resource available Define Your Sales Process' (Oct 26, 2025, 2:10 PM). The bottom of the screen shows a Windows taskbar with various icons and system status.

You will find notification like this and you will get an email check
 Note: similarly do for reject also you will get mail and notification

