

Lease Management

User Story:

The objective of the Lease Management System is to streamline the process of managing property leases, ensuring efficient handling of lease agreements, renewals, and terminations. It aims to provide a centralized platform for tracking tenant information, payment schedules, and property details. The system enhances transparency by offering real-time updates and automated reminders for key lease events. Additionally, it reduces manual errors and administrative overhead, ensuring compliance with legal and financial obligations. Ultimately, the project aims to improve productivity, tenant satisfaction, and overall lease lifecycle management.

Project Overview:

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.

Project Flow:

Milestone 1: Salesforce Developer Account Creation

Milestone 2: Object

Milestone 3: Tabs

Milestone 4: The Lightning App

Milestone 5: Fields

Milestone 6: Validation Rule

Milestone 7: Email Template

Milestone 8: Approval Processes

Milestone 9: Apex Trigger

Milestone 10: Flows

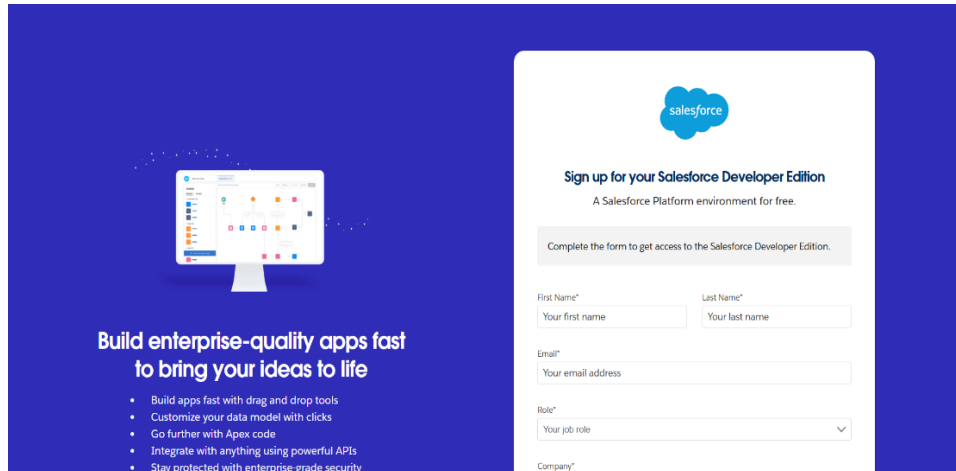
Milestone 11: Schedule class

Milestone 1: Salesforce Developer Account Creation:

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:

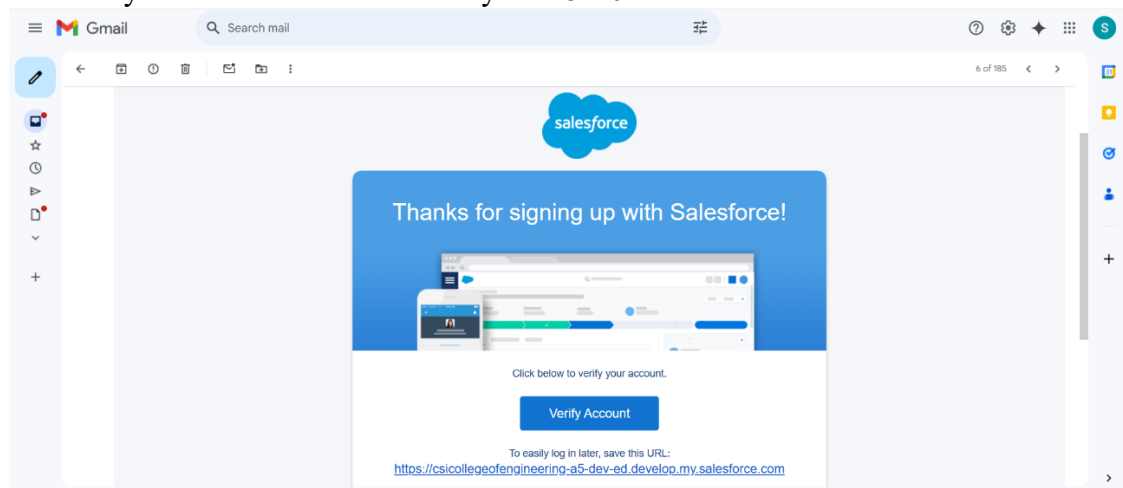
The image shows the Salesforce Developer Edition sign-up page. On the left, there's a blue background with a white monitor icon displaying a Salesforce interface. Text on the left says "Build enterprise-quality apps fast to bring your ideas to life" followed by a bulleted list of features. On the right, there's a white sign-up form with the Salesforce logo at the top. The form title is "Sign up for your Salesforce Developer Edition" with a subtitle "A Salesforce Platform environment for free." Below this is a grey box saying "Complete the form to get access to the Salesforce Developer Edition." The form fields include: "First Name*" (placeholder: "Your first name"), "Last Name*" (placeholder: "Your last name"), "Email*" (placeholder: "Your email address"), "Role*" (a dropdown menu with "Your job role" selected), and "Company*" (a text field).

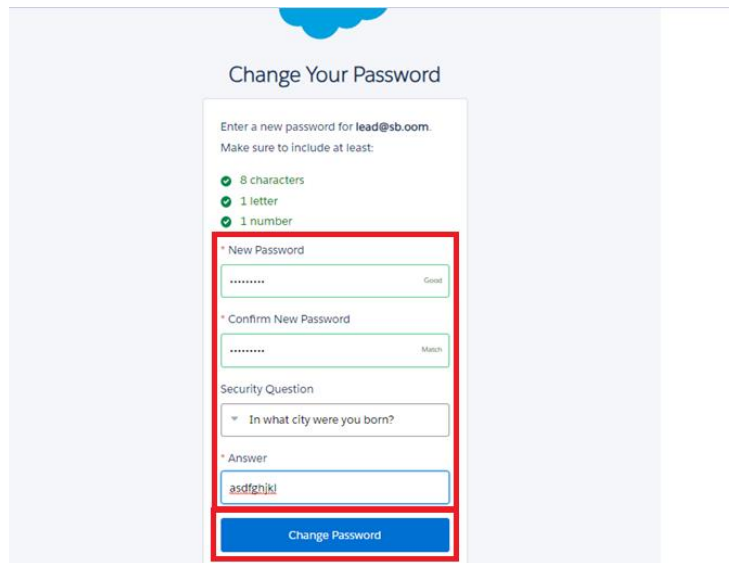
1. First name & Last name
2. Email
3. Role: Developer
4. Company: College Name
5. Country: India
6. Postal Code: Pin Code
7. Username: should be a combination of your name and company

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 5-10 mins.





Change Your Password

Enter a new password for **lead@sb.oom**.
Make sure to include at least:

- ✓ 8 characters
- ✓ 1 letter
- ✓ 1 number

* New Password
 Good

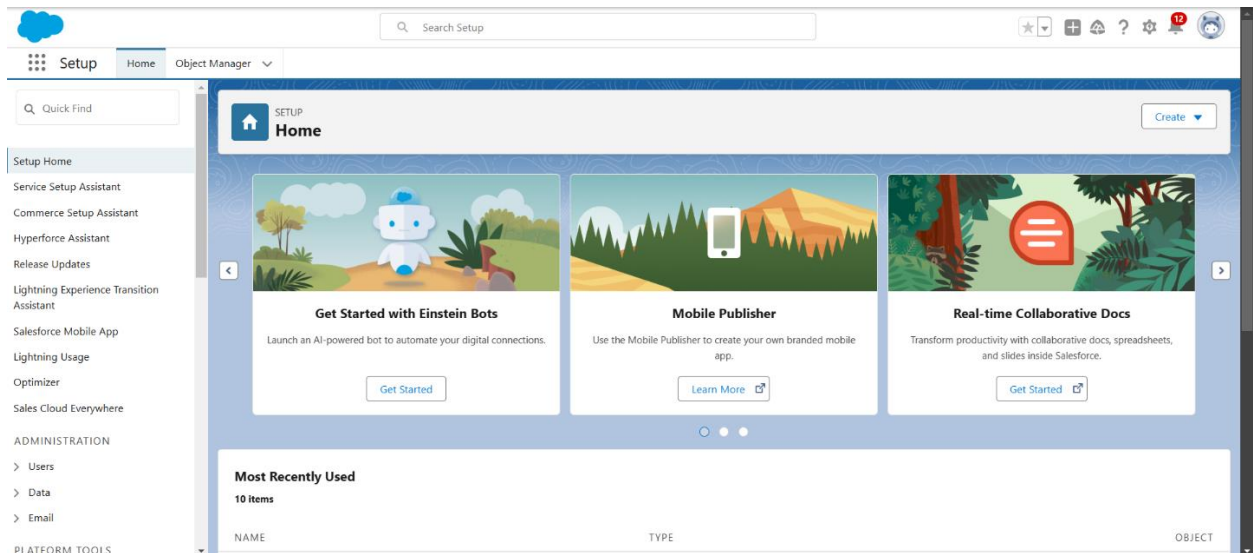
* Confirm New Password
 Match

Security Question
 ▼ In what city were you born?

* Answer

Change Password

1. Click on Verify Account
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.



The image shows the Salesforce Setup Home page. The top navigation bar includes the Setup icon, a search bar, and various utility icons. The left sidebar contains a 'Quick Find' search bar and a list of setup categories: Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, ADMINISTRATION (with sub-items Users, Data, Email), and PLATFORM TOOLS.

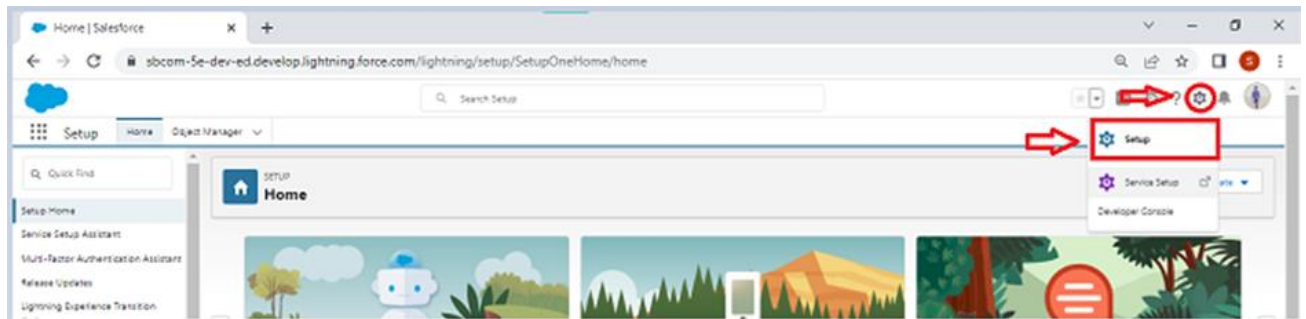
The main content area is titled 'SETUP Home' and features three featured cards:

- Get Started with Einstein Bots**: Launch an AI-powered bot to automate your digital connections. [Get Started](#)
- Mobile Publisher**: Use the Mobile Publisher to create your own branded mobile app. [Learn More](#)
- Real-time Collaborative Docs**: Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce. [Get Started](#)

Below these cards is a section titled 'Most Recently Used' showing 10 items in a table with columns for NAME, TYPE, and OBJECT.

Milestone 2: Object

To navigate to setup page:

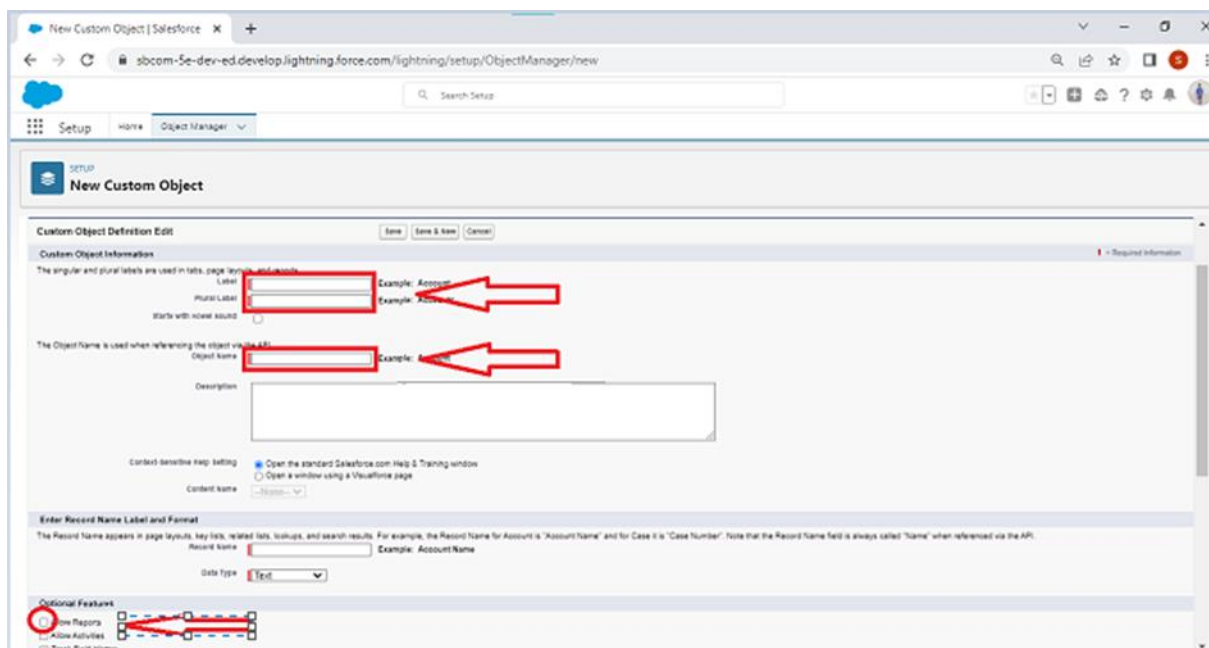


To create an object:

1. From the setup page→Click Object Manager→Click Create→Click Custom Object.



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



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

[What is this?](#)

- ☐ 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

Object Creation Options (Available only when custom object is first created)

- ☐ Add Notes and Attachments related list to default page layout
- ☐ Launch New Custom Tab Wizard after saving this custom object

Buttons: Save, Save & New, Cancel

4. Click on Save

Activity 1: Create Property Object

To create an object:

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

1. Enter the label name→property
2. Plural label name→property
3. Enter Record Name Label and Format
4. Record Name→property Name
5. Data Type→Text
6. Click on Allow reports and Track Field History, Allow Activities
7. Allow search→Save.

Activity 2: Create Tenant Object

To create an object:

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

1. Enter the label name→Tenant
2. Plural label name→Tenants
3. Enter Record Name Label and Format
4. Record Name→Tenant Name
5. Data Type→Text
6. Click on Allow reports and Track Field History, Allow Activities
7. Allow search→Save.

Activity 3: Create Payment Object

To create an object:

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

1. Enter the label name→Payment for tenant
2. Plural label name→Payment
3. Enter Record Name Label and Format
4. Record Name→Payment Name
5. Data Type→Text
6. Click on Allow reports and Track Field History, Allow Activities
7. Allow search→Save.

Activity 4: Create Lease Object

To create an object:

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

1. Enter the label name→lease
2. Plural label name→lease
3. Enter Record Name Label and Format
4. Record Name→lease Name
5. Data Type→Text
6. Click on Allow reports and Track Field History, Allow Activities
7. Allow search→Save.

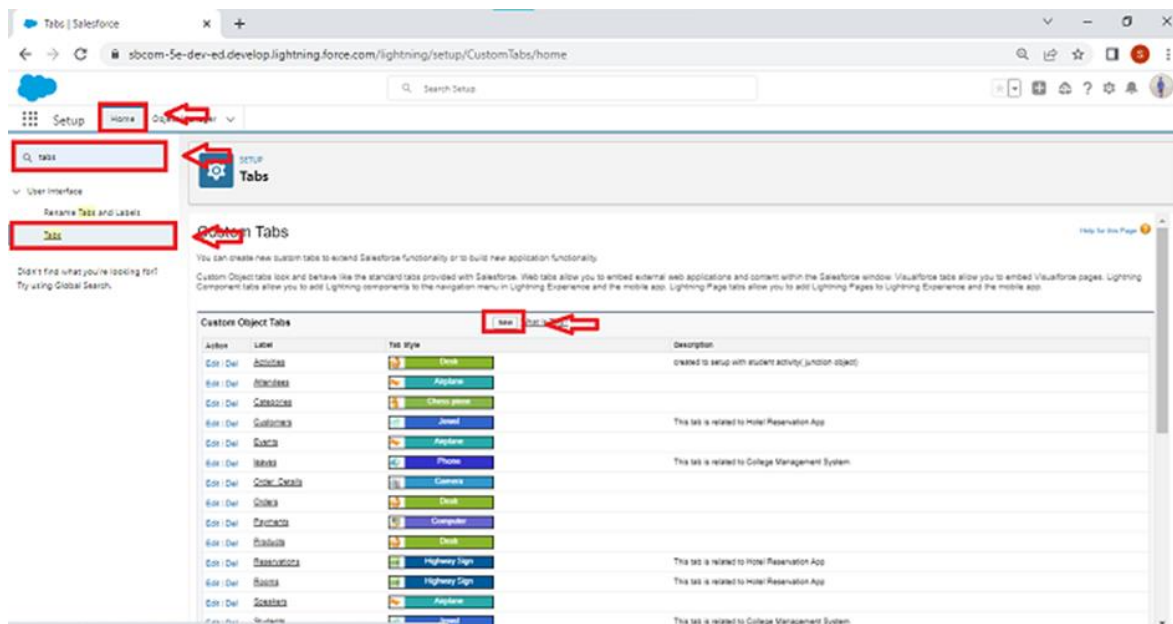
Milestone 3: Tabs

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

Activity 1: Creating 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)



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: Creating Remaining Tabs

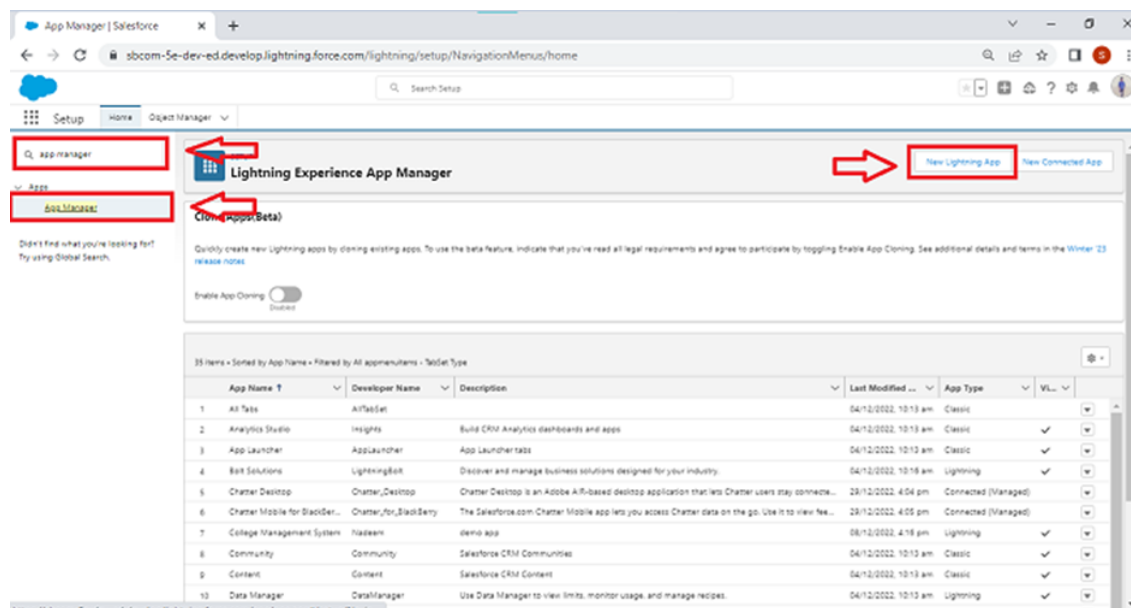
1. Now create the Tabs for the remaining Objects, they are “Payment for tenant, lease, Tenant”.
2. Follow the same steps as mentioned in Activity -1.

Milestone 4: The Lightning App

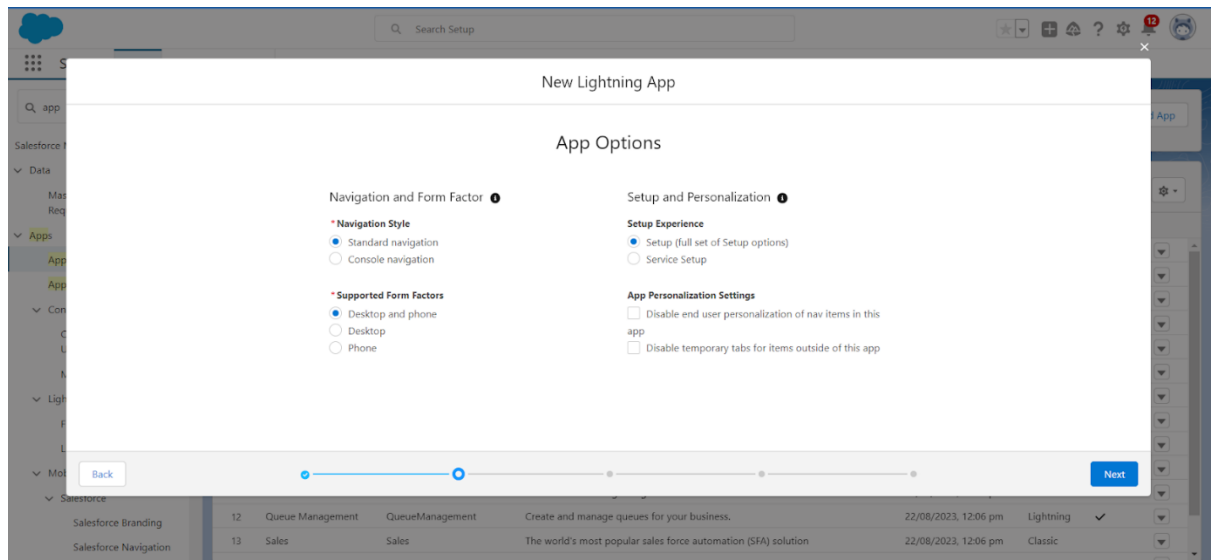
Activity 1: Create A Lightning App

To create a lightning app page:

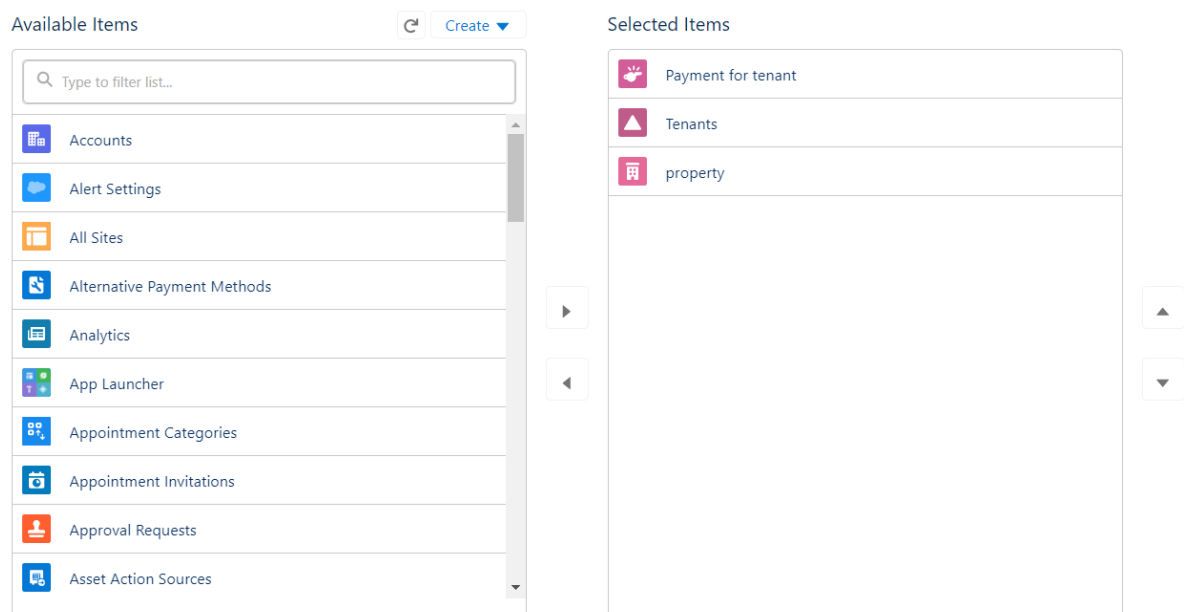
1. Go to setup page→search “app manager” in quick find→select “app manager” →click on New lightning App.



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.



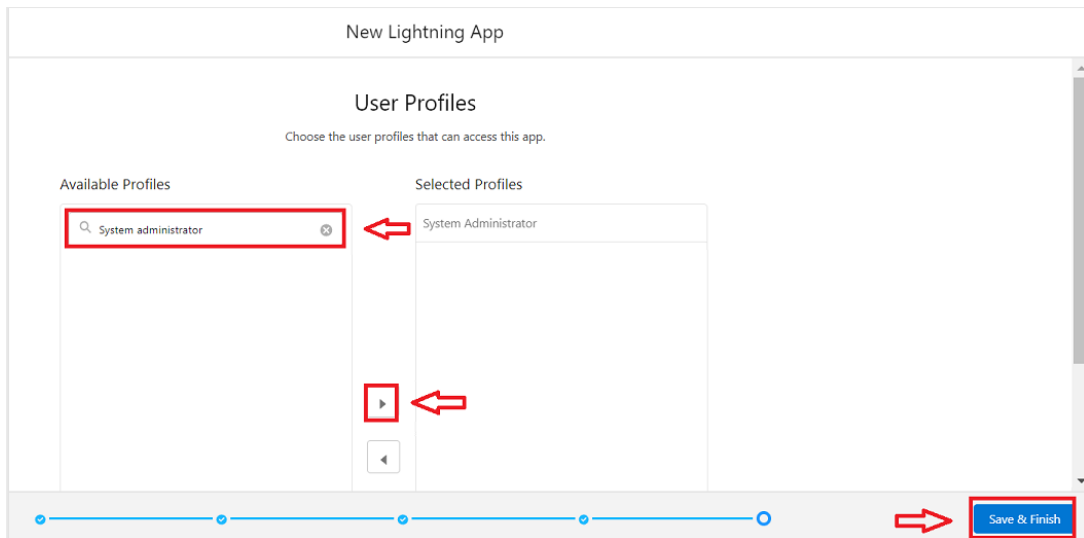
4. (Utility Items) keep it as default → Next
5. To Add Navigation Items:



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.

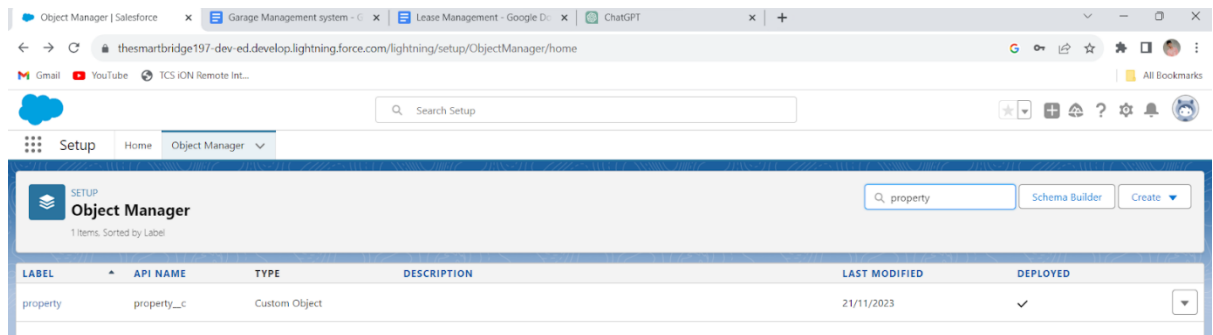


Milestone 5: Fields

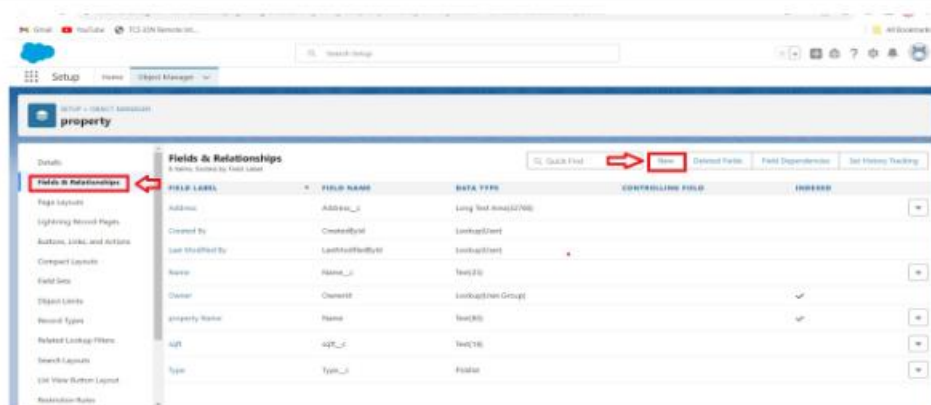
Activity 1: Creation of Fields for the Property Object

To create fields in an object:

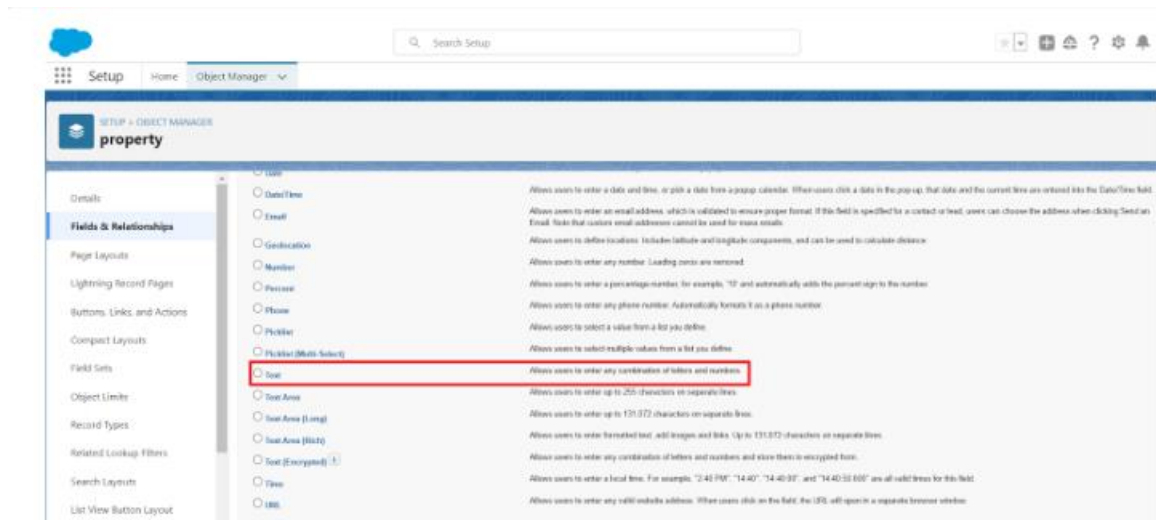
1. Go to setup→click on Object Manager→type object name(property) in search bar→click on the object.



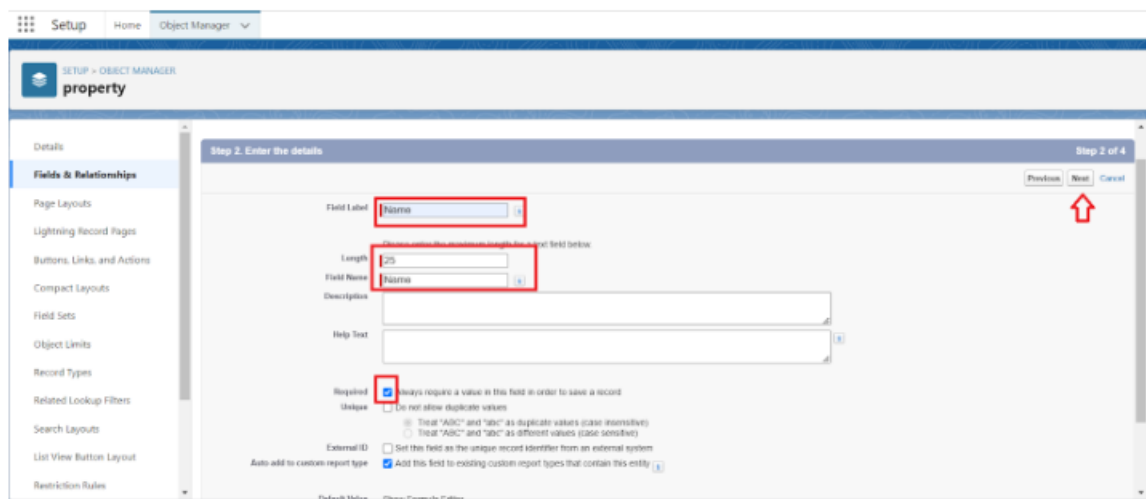
2. Now click on “Fields & Relationships”→New



3. Select Data Type as a “Text”



4. Click on Next



5. Fill the Above as following:
6. Field Label: Name
7. Field Name: gets auto generated
8. Length: 25
9. Required: check box
10. 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 “Long Text” and Click on Next
4. 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: sqft
 - Field Name: gets auto generated
 - Length: 18
5. 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.
2. Now click on “Fields & Relationships”→New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:
 - Field Label: Email
 - Field Name: gets auto generated
 - Click on required check box
5. 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:
5. Field Label: status
6. Field Name: gets auto generated
7. Enter values, with each value separated by a new line
8. Enter these values:
9. Stay
10. Leaving
11. 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 (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.

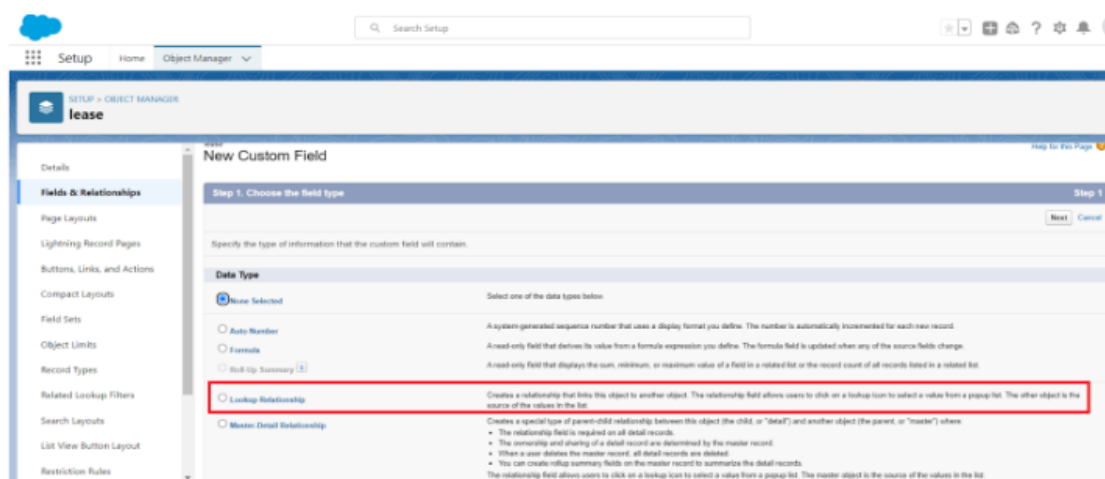
To create another fields in an object:

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 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.

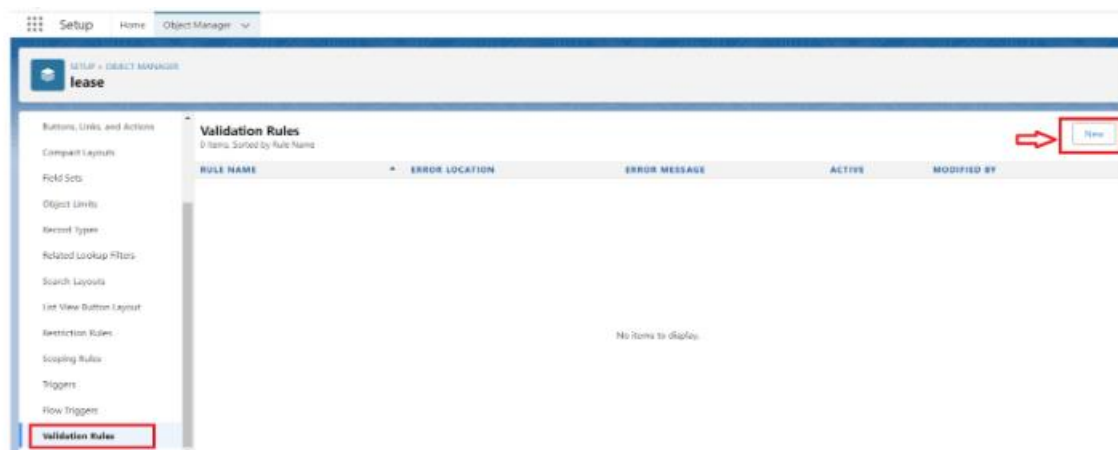
Creation of Lookup Field on Payment for tenant Object :

1. Go to setup→click on Object Manager→type object name (property) in the search bar→click on the object.
2. Now click on “Fields & Relationships”→New
3. Select master-detail relationship
4. Select the related object “property” and click next.
5. Field Name: property
6. Field label: Auto generated
7. Next→Next→Save.

Milestone 6: Validation Rule

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.



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' page for a rule named 'lease_end_date'. The 'Rule Name' field is highlighted with a red box. The 'Error Condition Formula' section shows the formula 'start_date < end_date' with a red box around it. The 'Check Syntax' button indicates 'No errors found'.

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.

The screenshot shows the 'Error Message' configuration page. The 'Error Message' field is highlighted with a red box and contains the text 'Your End date must be greater than start date'. The 'Error Location' is set to 'Field' and the 'Field' is set to 'start_date'. A red arrow points to the 'Save' button.

Milestone 7: Email Template

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
2. Click on→New Email Template→choose text
Folder: Unified public classic Email templates
Click on available for use
3. Email Template Name is “Tenant leaving”
4. Template unique name: Auto populated
5. Subject: “request for approve the leave”
6. Email body:
Dear {!Tenant_c.CreatedBy},
Please approve my leave
7. 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 Template
Click 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: Unified 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: Unified public classic Email templates
Click on available for use
3. Email Template Name is “Template Email”
4. Template unique name: Auto populated
5. Subject: “Urgent: Monthly Rent Payment Remainder”

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

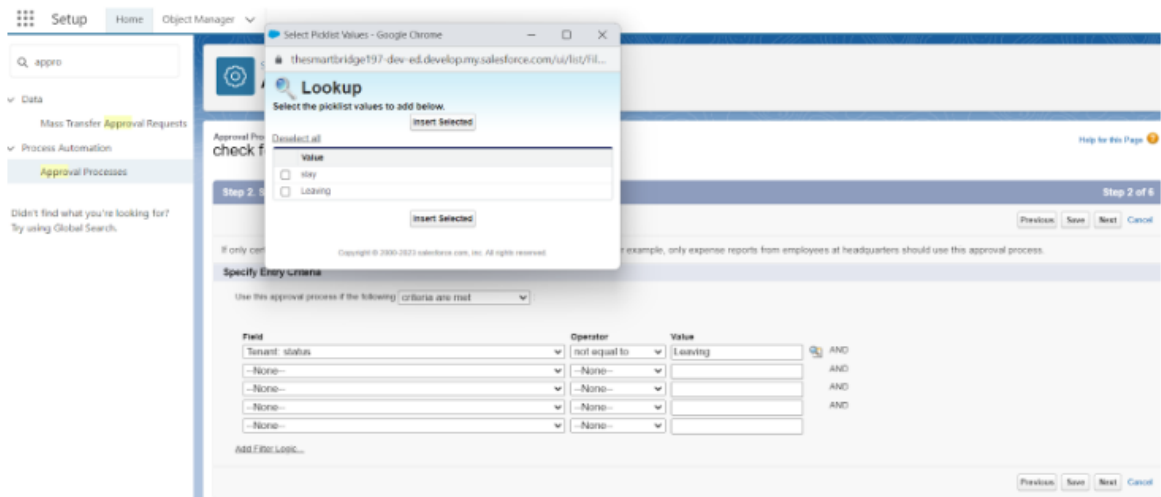
1. Go to setup in quick find box enter email template→click on classic Email Template
2. Click on→New Email Template→choose text
Folder: Unified public classic Email templates
Click on available for use
3. Email Template Name is “Tenant Payment”
4. Template unique name: Auto populated
5. Subject: “Confirmation 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 8: Approval Process

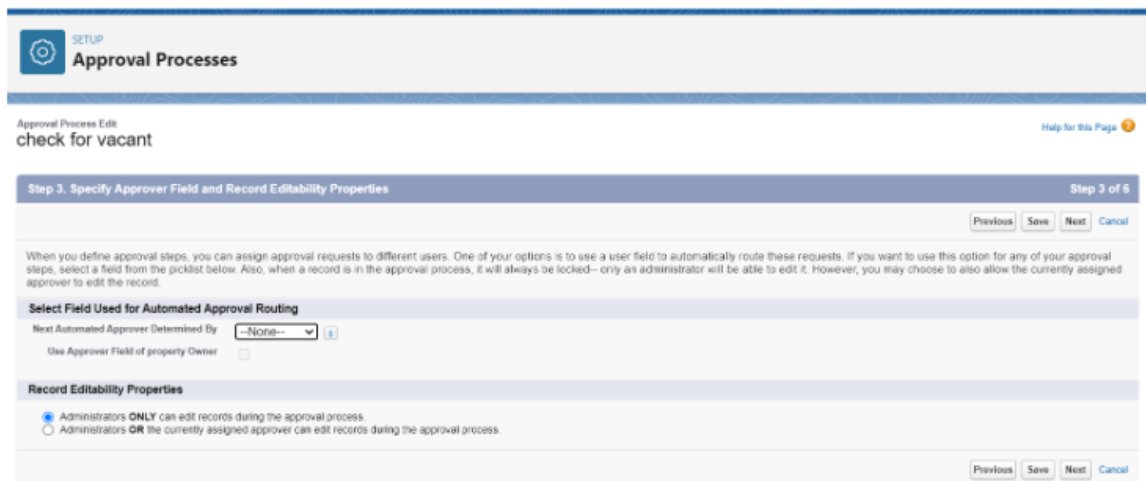
Activity 1: Create Approval Process for Check for Vacant

To create fields in an object:

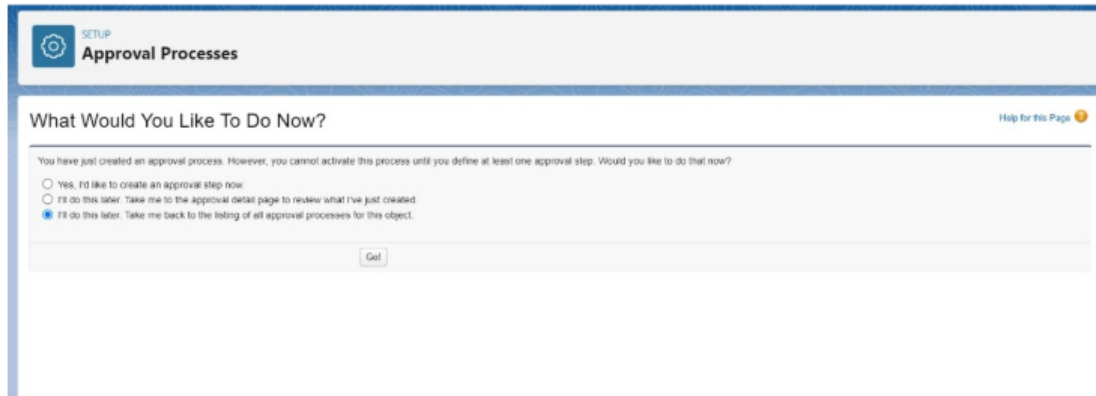
1. Go to setup→Approval Processes in quick find bar→click on it
2. Manage Approval Processes for→”Tenant” from the drop down.
3. Click on “Create New Approval Process”→Use standard setup wizard.
4. Process Name ”check for vacant”→Click Next.
5. Field “Tenant: status”→Operator: Not equals, Value→Click on the lookup filter icon and select ”Leaving”.
6. Click insert fields, 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 setting 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.
 - 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.



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

1. Under Final rejection action click on new and then select email alert.
2. Description: "Your request for leave is rejected"
3. unique name: auto populated

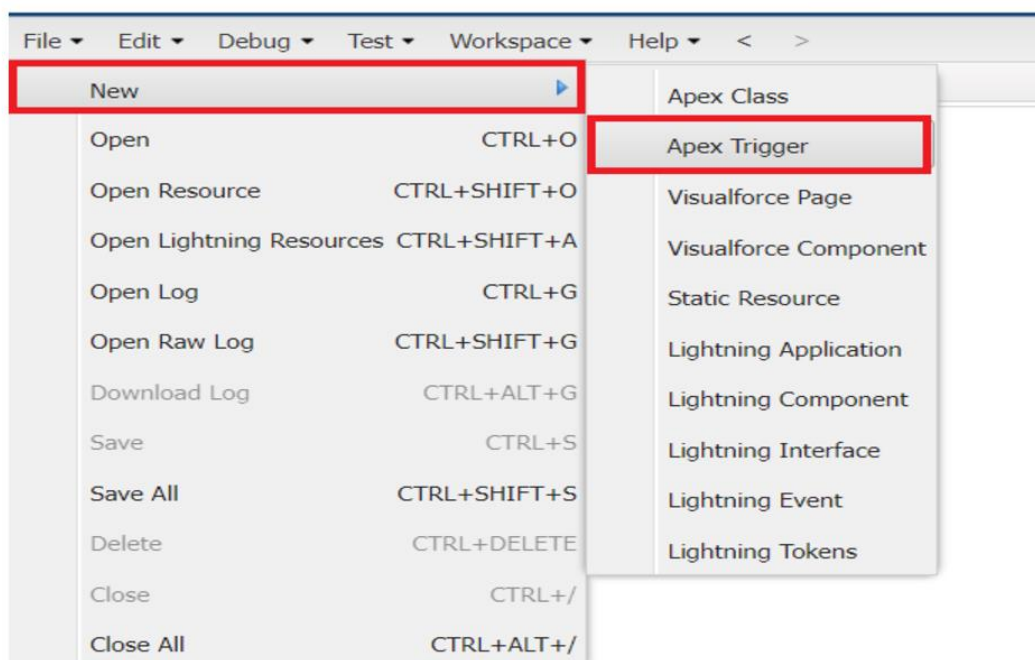
4. Email template: Leave rejected
5. Recipient type: Email field
6. Available Recipients: Email field: Email
7. From Email address: Current user's email
8. Click save

Milestone 9: Apex Trigger

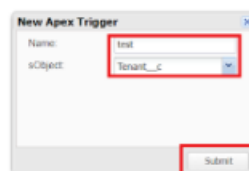
Activity 1: Create an Apex Trigger

To create a new Apex Class follow the below steps:

1. Click on the File→ New→Apex Class



2. Give the Apex name as “Test”, and select “Tenant_c” from the dropdown for sObject.



3. Click Submit
4. Now write the code logic here

```

File Edit Debug Test Workspace Help < >
test.apxt
Code Coverage: None API Version: 59
1 trigger test on Tenant__c (before insert)
2 {
3     if(trigger.isInsert && trigger.isBefore){
4         testHandler.preventInsert(trigger.new);
5     }
6 }

```

Trigger Code:

```

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

```

Activity 2: Create an Apex Handler class

To create a new Apex Class follow the below steps:

1. Click on the File→New→Apex Class
2. Enter the class name as testHandler

```

testHandler.apex
Code Coverage: None API Version: 59
1 public class testHandler {
2     public static void preventInsert(List<Tenant__c> newList) {
3         Set<Id> existingPropertyIds = new Set<Id>();
4         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
5             existingPropertyIds.add(existingTenant.Property__c);
6         }
7
8         for (Tenant__c newTenant : newList) {
9
10            if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
11                newTenant.addError('A tenant can have only one property');
12            }
13        }
14    }
15 }

```

Apex Logic:

```

public class TestHandler {
    public static void preventInsert(List<Tenant__c> newList) {
        Set<Id> existingPropertyIds = new Set<Id>();
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE
Property__c != null]) {
            existingPropertyIds.add(existingTenant.Property__c);
        }
    }
}

```

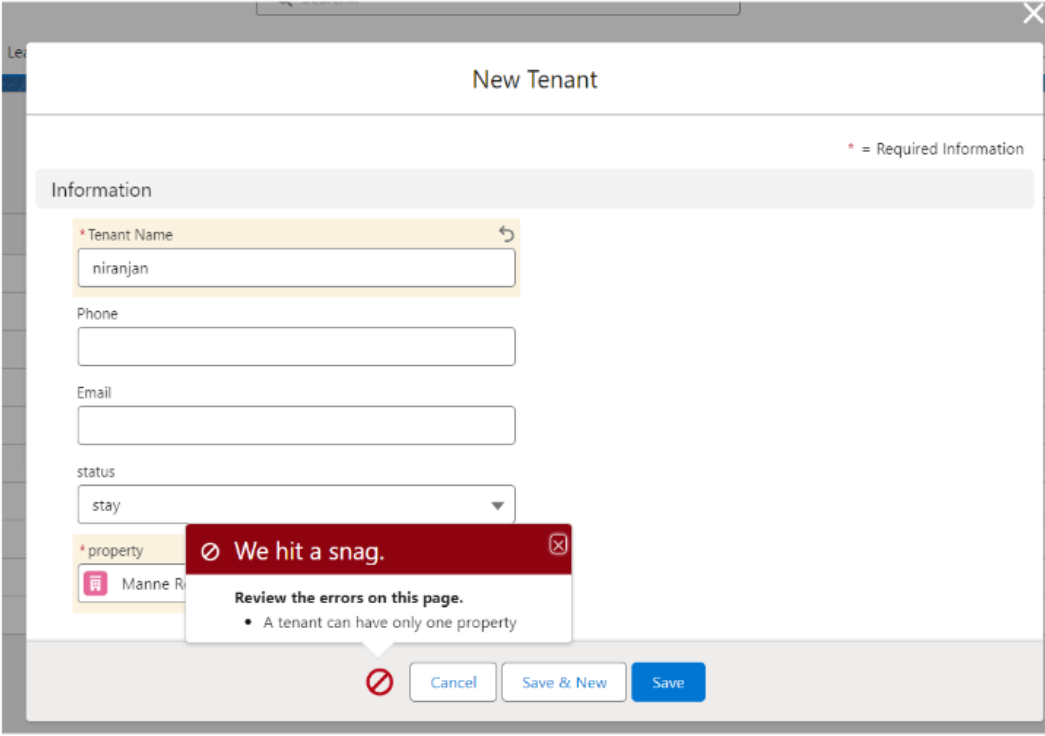
```

        for (Tenant__c newTenant : newList) {
            if (newTenant.Property__c != null &&
                existingPropertyIds.contains(newTenant.Property__c)) {
                newTenant.addError('A tenant can have only one property');
            }
        }
    }
}

```

Activity 3: Testing the Trigger

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

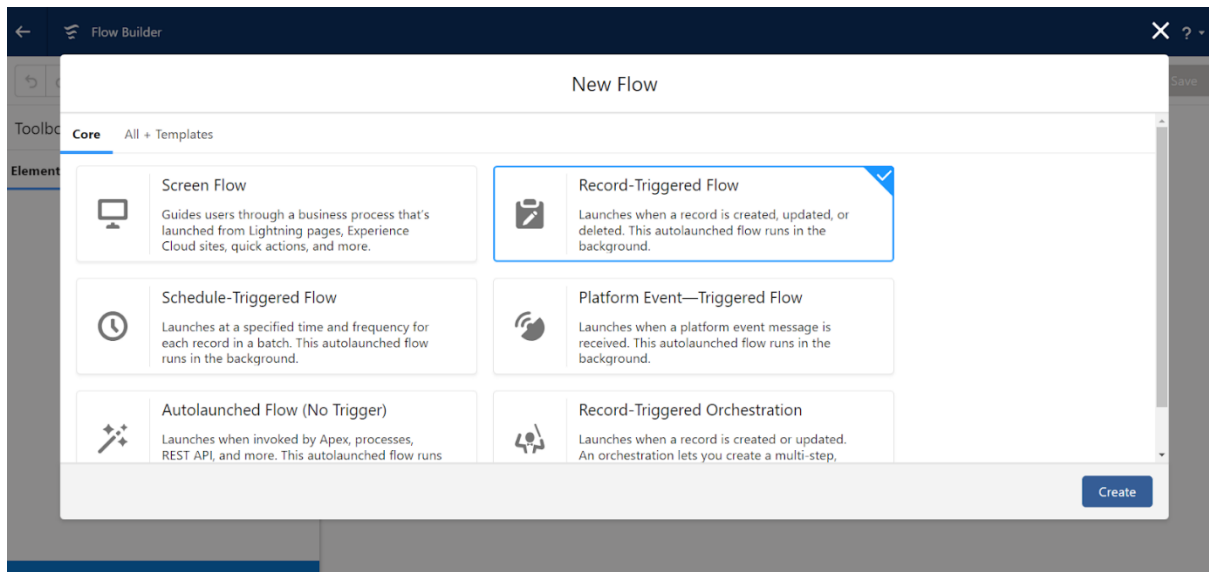


The screenshot shows the 'New Tenant' form in Salesforce. The form has a title bar 'New Tenant' and a legend '* = Required Information'. The form fields are: 'Tenant Name' (required, value 'niranjan'), 'Phone', 'Email', 'status' (dropdown, value 'stay'), and 'property' (required, value 'Manne R'). A red error message box is displayed over the 'property' field, stating 'We hit a snag. Review the errors on this page. • A tenant can have only one property'. At the bottom of the form are three buttons: 'Cancel', 'Save & New', and 'Save'.

Milestone 10: Flows

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



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

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

* Object

Configure Trigger

* Trigger the Flow When:

☐ A record is created
☒ A record is updated
☐ A record is created or updated
☐ A record is deleted

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

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

All Conditions Are Met (AND)

Field

Operator

Value

check_for_paymet__c

Equals

paid

+ Add Condition

When to Run the Flow for Updated Records

☒ Every time a record is updated and meets the condition requirements

☐ Only when a record is updated to meet the condition requirements

* Optimize the Flow for:

Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.

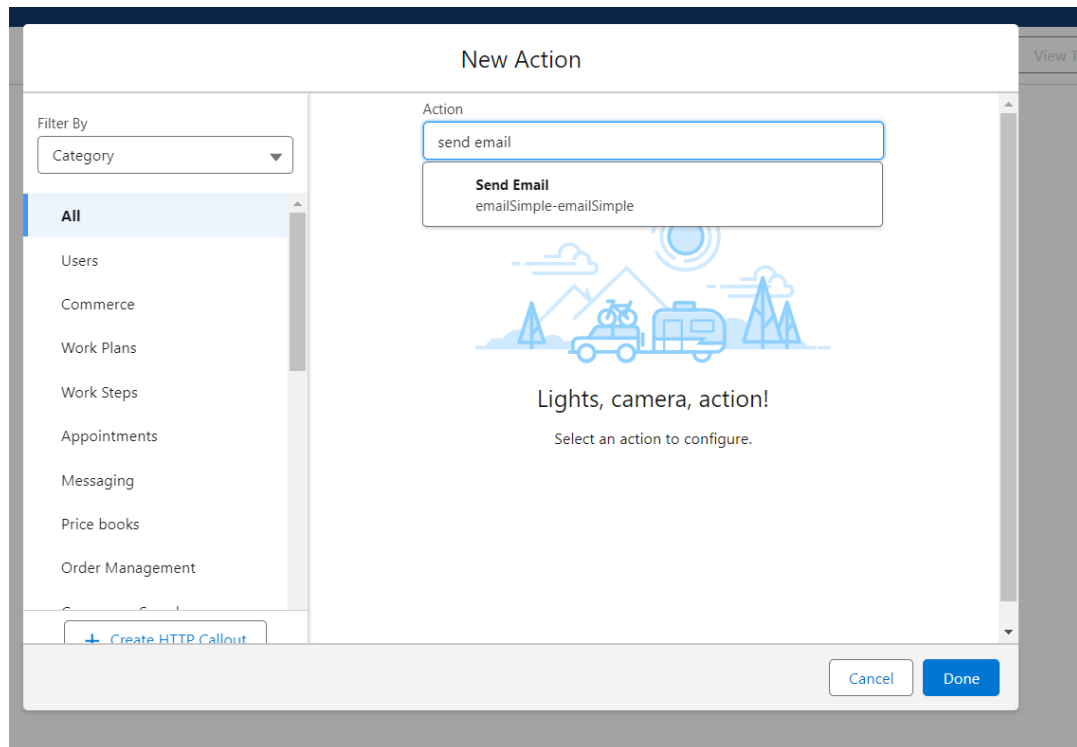
Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

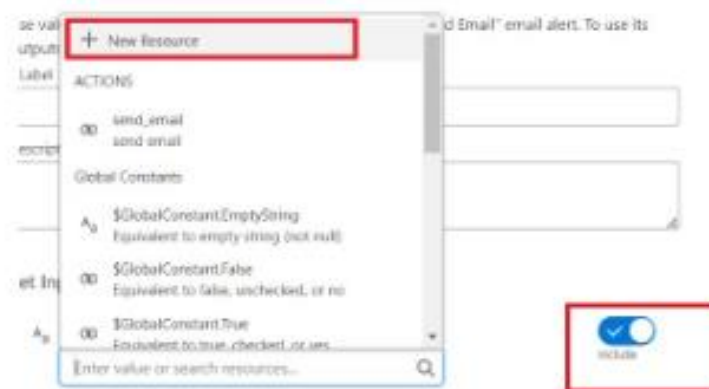
7. Under record trigger flow click on “+” icon and select action

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

8. Label: send Email
API Name: send_Email



9. Label: send Email
10. API Email: send_Email
11. Enable Body
12. Click on new resource



13. 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?

16. Click Done

17. Enable subject

Paste this → Confirmation of Successful Monthly Payment

18. Click on Save

Flow label: monthly payment

Flow API Name: monthly_payment

19. Click on Activate

Milestone 11: Schedule Apex Class

Activity 1: Create an Apex Class

To create a new Apex Class follow the below steps:

1. Click on the file >> New >> Apex Class.
2. Enter class name as MonthlyEmailScheduler.



```
1 • global class MonthlyEmailScheduler implements Schedulable {
2 •     global void execute(SchedulableContext sc) {
3 •         Integer currentDay = Date.today().day();
4 •         if (currentDay == 1) {
5 •             sendMonthlyEmails();
6 •         }
7 •     }
8 •
9 •     public static void sendMonthlyEmails() {
10 •
11 •         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
12 •
13 •
14 •         for (Tenant__c tenant : tenants) {
15 •             String recipientEmail = tenant.Email__c;
16 •             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.';
17 •             String emailSubject = 'Reminder: Monthly Rent Payment Due';
18 •
19 •             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
20 •             email.setToAddresses(new String[]{recipientEmail});
21 •             email.setSubject(emailSubject);
22 •             email.setPlainTextBody(emailContent);
23 •
24 •             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
25 •         }
26 •     }
27 • }
```

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) {
```

```

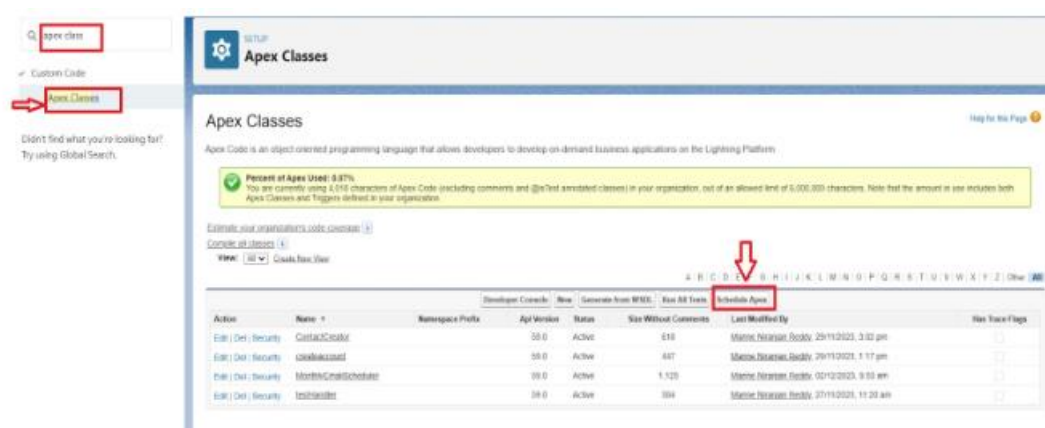
String recipientEmail = tenant.Email__c;
String emailContent = 'I trust this email finds you well. I am writing to remind you that
the monthly rent is due. Your timely payment ensures the smooth functioning of our rental
arrangement and helps maintain a positive living environment for all.';
String emailSubject = '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});
}
}
}

```

3. Save the code.

Activity 2: Schedule Apex Class

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



3. Enter job Name: MonthlyEmailScheduler
4. Apex class: MonthlyEmailScheduler
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

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

SaveCancel

Job NameMonthlyEmailScheduler

Apex ClassMonthlyEmailScheduler

Schedule Apex Execution

Frequency

☐ Weekly

☒ Monthly

On day1of every month

☐ On the 1stSundayof every month

Start04/12/202304/12/2023

End04/01/202404/12/2023

Preferred Start Time9:00 am

Exact start time will depend on job queue activity.

SaveCancel

Testing the Approval Process

Lease_Management

property

Tenants

Payment

lease

Search...

New Contact

Edit

New Opportunity

Related

Details

Tenant Name

swetha

Email

swethasrinivas4282@gmail.com

Phone

1234567890

status

Stay

property

Hill Residency

Created By

swetha v s, 20/11/2024, 8:28 pm

Last Modified By

swetha v s, 21/11/2024, 11:11 am

Activity

Filters: All time • All activities • All types

Refresh • Expand All • View All

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

Entry any comment and click on submit

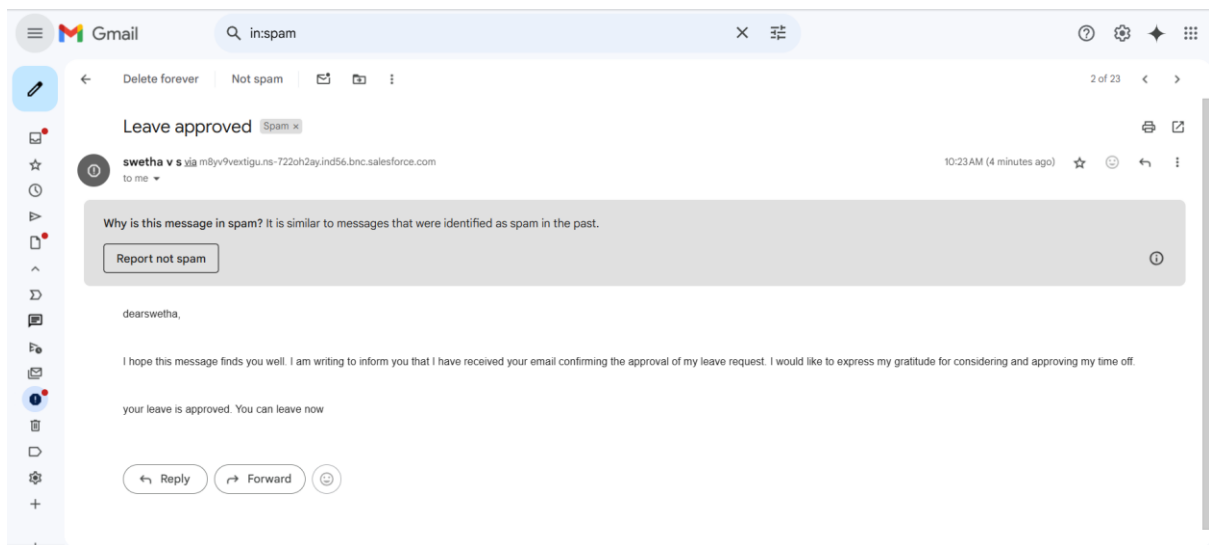
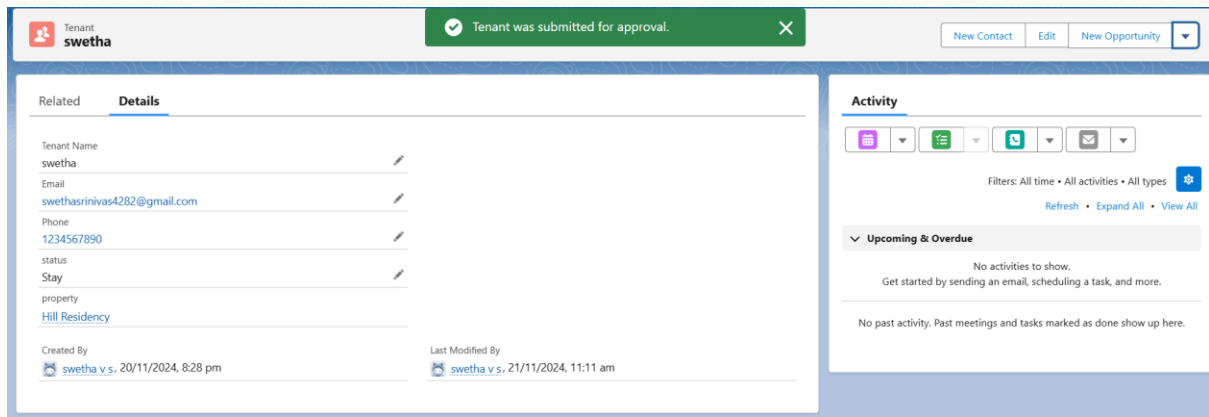
Submit for Approval

Comments

Leaving

Cancel

Submit



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.