LEASE MANAGEMENT

1. Project Overview

This Lease Management System project developed using salesforce(a cloud-based system). The main Objective/ Aim of this system, is to reduce the Consumption of Time, reducing the Redundancy etc. during maintaining the records of Hostel management. Separate divisions are provided to maintain the records of Building, Unit, Clients Details. Admin is the Superuser of this project. The proposed software will also reduce the clumsy paperwork, manual labour as well as communication cost.

2. Objectives

Business Goals:

1. **Increase Efficiency:**
    - Reduce the time spent on manual lease management tasks by 50% within the first six months.
2. **Enhance Occupancy Rates:**
    - Achieve a 90% occupancy rate across all managed properties within the first year.
3. **Improve Revenue Tracking:**
    - Provide accurate and real-time revenue tracking, reducing discrepancies by 95% within the first quarter.

Specific Outcomes:

1. **Building and Unit Management:**
    - Successfully create and manage 100% of buildings and units within Salesforce by the end of the project.

2. **Lease Contract Management:**
    - Implement a contract approval workflow that reduces approval time to less than 24 hours.
    - Ensure 100% of lease contracts are digitally stored and accessible within Salesforce.
3. **Search Functionality:**
    - Enable users to search and retrieve lease contracts within 5 seconds.
4. **Reports and Dashboards:**
    - Generate monthly reports on occupancy and revenue with 100% accuracy.
    - Develop dashboards that provide real-time insights into key metrics, accessible to all relevant stakeholders.
5. **User Roles:**
    - Train 100% of management and coordinators on the new system, ensuring they can perform their roles effectively.

These goals are designed to ensure the project delivers tangible benefits, improving overall lease management efficiency and effectiveness.

3. Salesforce Key Features and Concepts Utilized

1. Custom Objects and Fields:

- **Building and Unit Objects:** Custom objects to store information about buildings and individual units, with fields for details like address, unit size, and occupancy status.

2. Visualforce Pages:

- **Custom Interfaces:** Visualforce pages for adding and managing buildings and units, providing a user-friendly interface for data entry and updates.

3. Approval Processes:

- **Lease Contract Approvals:** Automated workflows for lease contract approvals, ensuring contracts are reviewed and approved efficiently.

4. Email Alerts and Notifications:

- **Tenant Communication:** Automated email notifications to tenants upon contract approval, keeping them informed about their lease status.

5. Reports and Dashboards:

- **Occupancy and Revenue Reports:** Custom reports to track occupancy rates and revenue, providing insights into property performance.
- **Dashboards:** Visual dashboards for quick access to key metrics and performance indicators.

6. Role-Based Access Control:

- **User Roles:** Different access levels for management and coordinators, ensuring data security and appropriate access to functionalities.

7. Search Functionality:

- **Global Search:** Enhanced search capabilities to quickly find lease contracts, tenant information, and property details.

8. Data Validation Rules:

- **Data Integrity:** Validation rules to ensure data accuracy and consistency, reducing errors in lease management.

9. Workflow Rules:

- **Automated Processes:** Workflow rules to automate routine tasks, such as updating unit status and sending reminders for lease renewals.

10. Integration with External Systems:

- **Third-Party Integrations:** Potential integration with accounting systems for seamless revenue tracking and financial reporting.

These features and concepts leverage Salesforce's robust platform to create an efficient and effective lease management system.

11. Reports and Dashboards:

- **Occupancy and Revenue Reports:** Custom reports to track occupancy rates and revenue, providing insights into property performance.
- **Dashboards:** Visual dashboards for quick access to key metrics and performance indicators.

12. Role-Based Access Control:

- **User Roles:** Different access levels for management and coordinators, ensuring data security and appropriate access to functionalities.

13. Search Functionality:

- **Global Search:** Enhanced search capabilities to quickly find lease contracts, tenant information, and property details.

14. Data Validation Rules:

- **Data Integrity:** Validation rules to ensure data accuracy and consistency, reducing errors in lease management.

15. Workflow Rules:

- **Automated Processes:** Workflow rules to automate routine tasks, such as updating unit status and sending reminders for lease renewals.

16. Integration with External Systems:

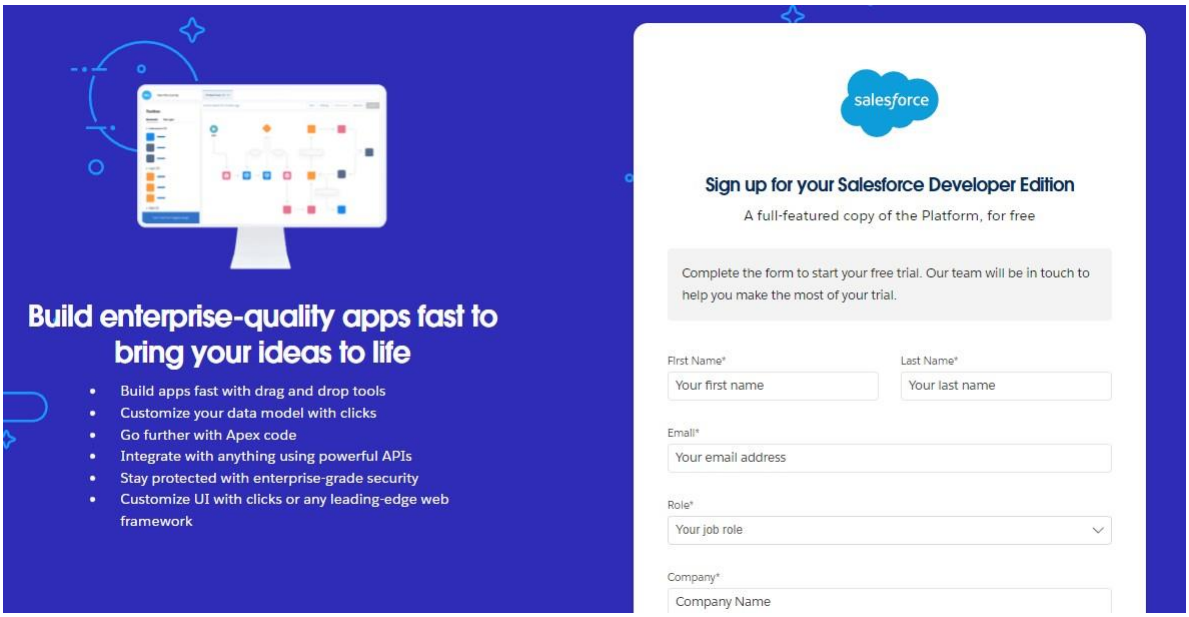
- **Third-Party Integrations:** Potential integration with accounting systems for seamless revenue tracking and financial reporting.

These features and concepts leverage Salesforce's robust platform to create an efficient and effective lease management system.

4. Detailed Steps to Solution Design

Creating Developer Account:

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



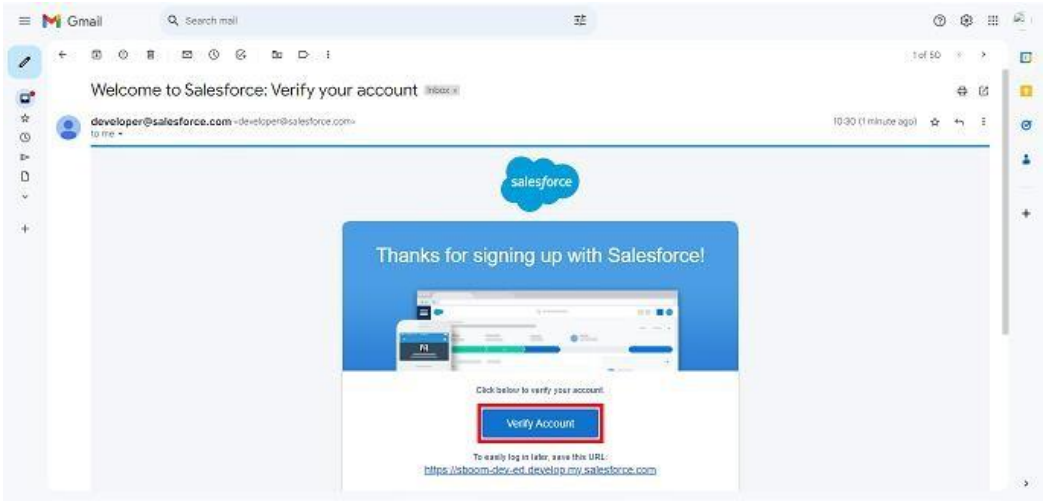
1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

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

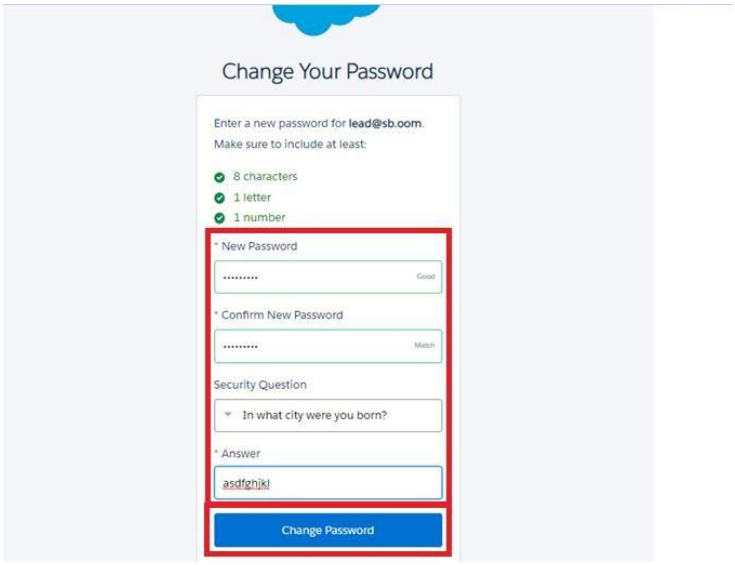
Click on sign me up after filling these.

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.

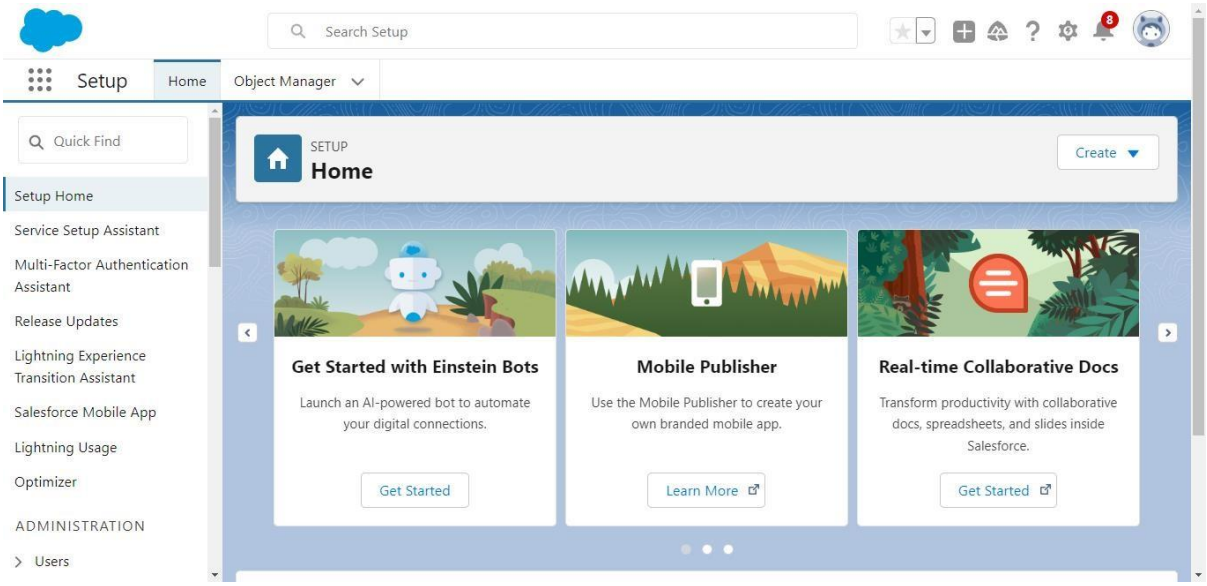


1. Click on Verify Account
2. Give a password and answer a security question and click on change password.



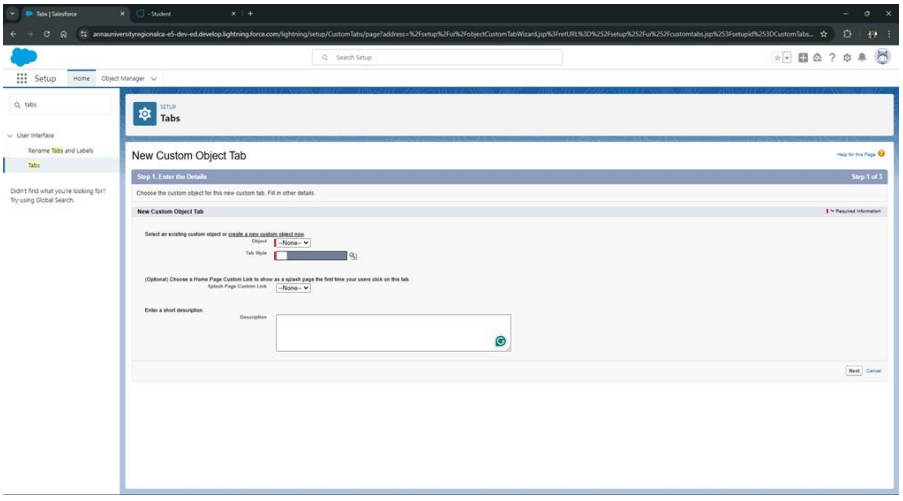
3. Give a password and answer a security question and click on change password.

4. Then you will redirect to your salesforce setup page.



Creating Object:

- Property
- Tenant
- Lease
- Payment

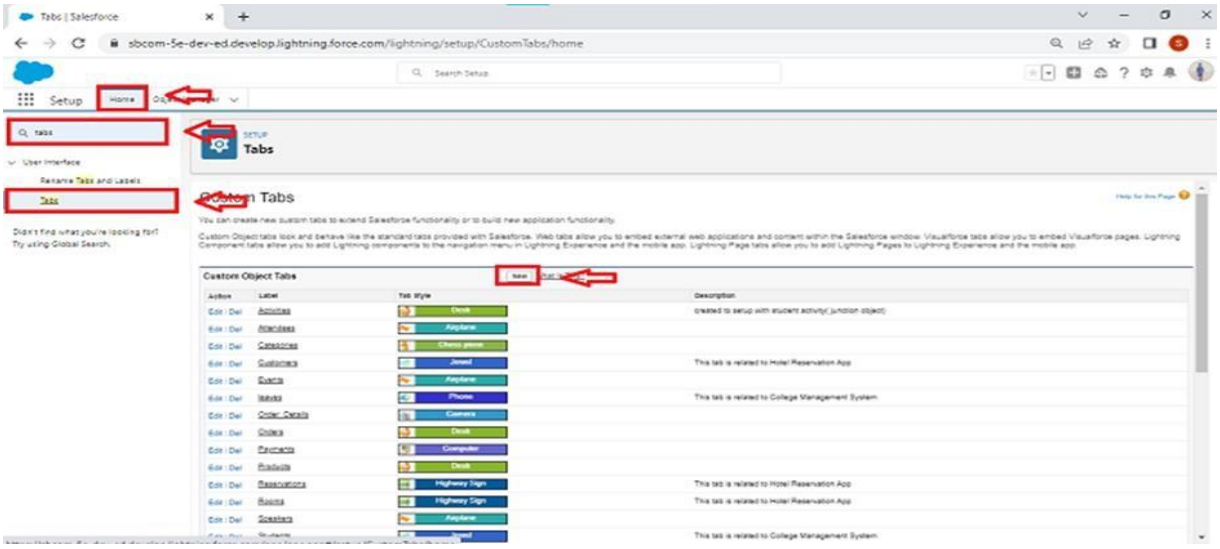


Creating Custom tabs:

- Property
- Payment for tenant
- Lease
- Tenant

To create a Tab:

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

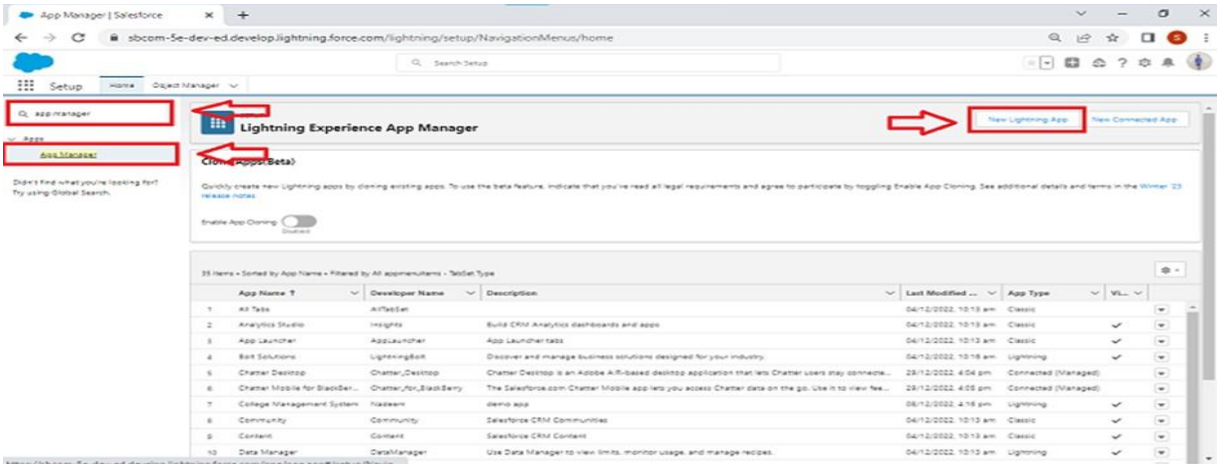


1. Select Object(property) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
2. Make sure that the Append tab to users' existing personal customizations is checked.
3. Click save

Creating 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.



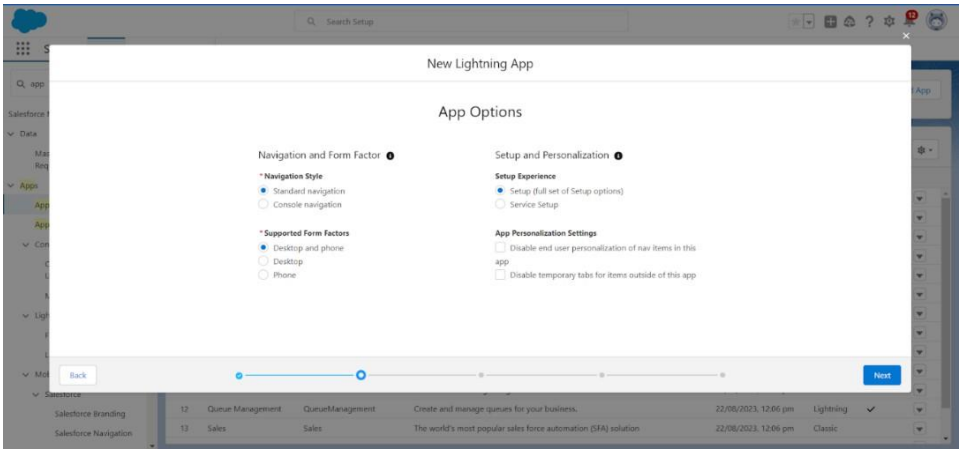
2. Fill the app name in app details and branding as follow

App Name : Lease Management

Developer Name : This will auto populated

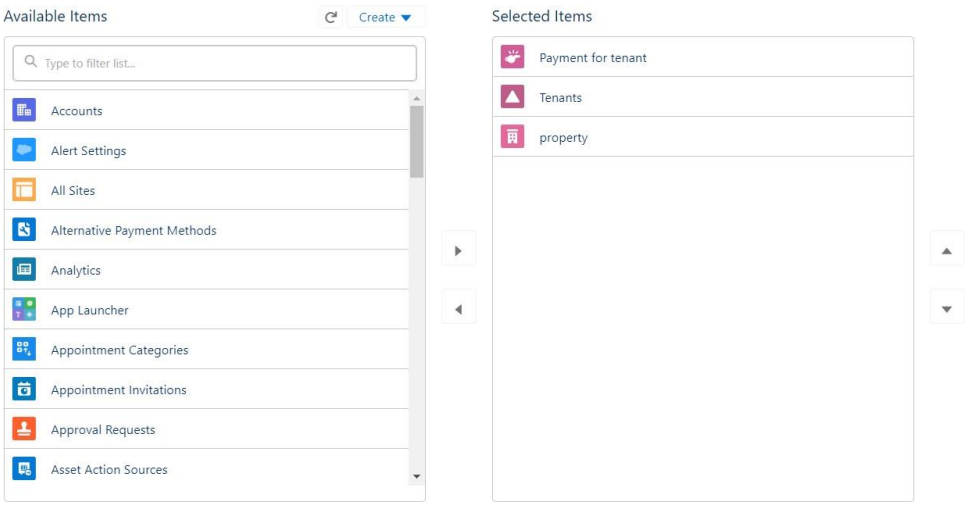
Image : optional (if you want to give any image you can otherwise not mandatory) Primary colour hex value : keep this default.

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



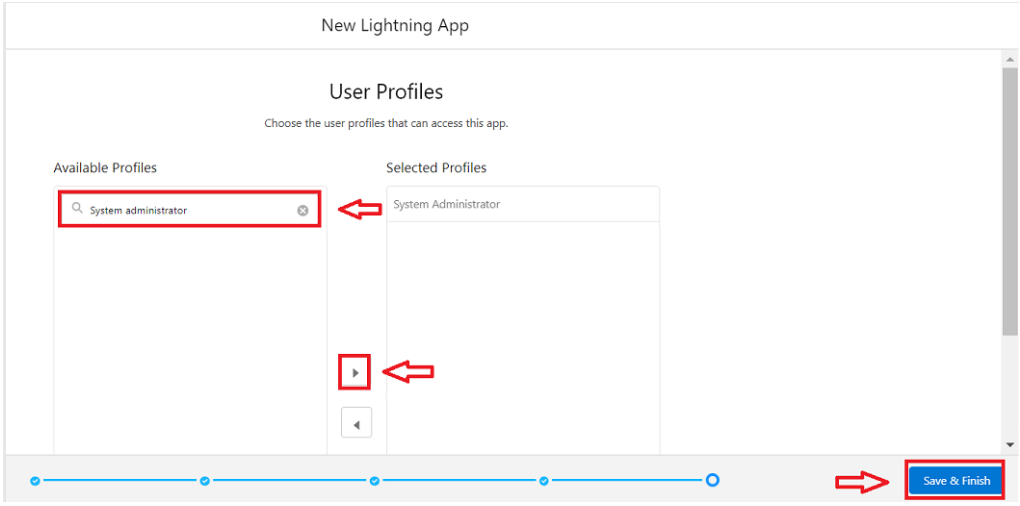
(Utility Items) keep it as default >> Next.

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:

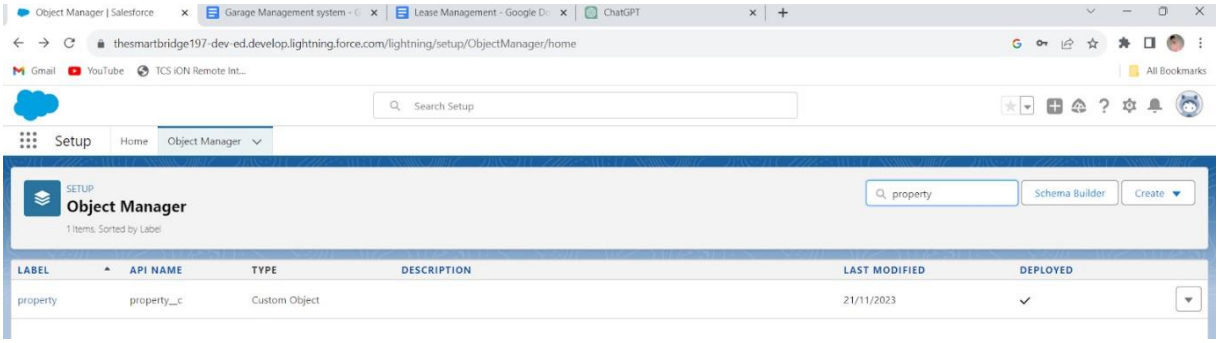


Search profiles (System administrator) in the search bar >>click on the arrow button >> save & finish.

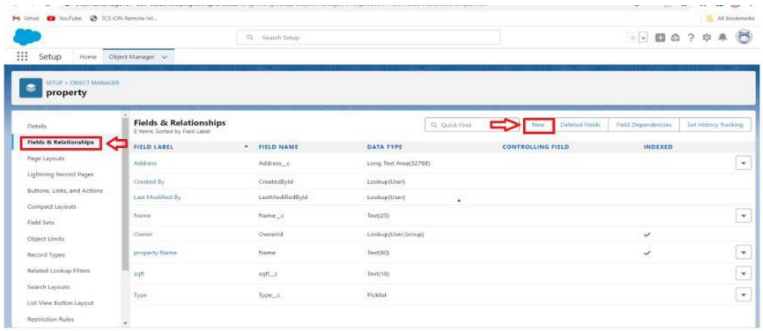
Creation of Fields:(Property)

To create fields in an object:

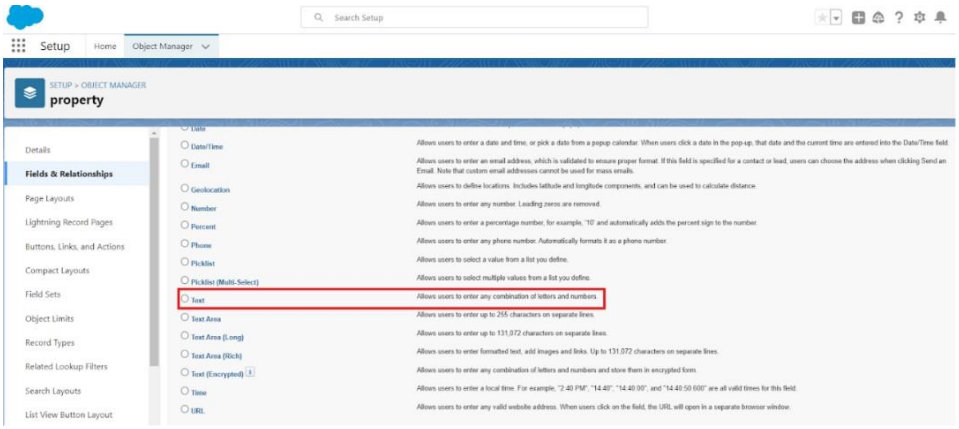
1. Go to setup >> click on Object Manager >> type object name(property) in search bar >>click on the object.



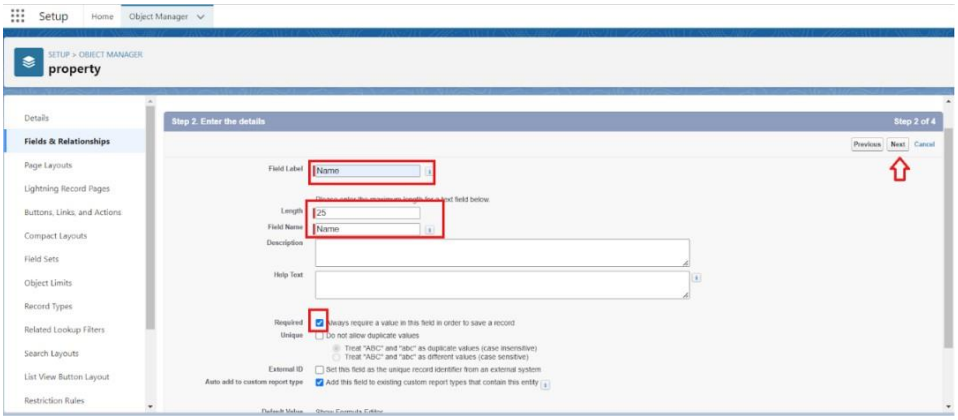
2. Now click on “Fields & Relationships” >> New



3. Select Data Type as a “Text”



4. Click on next



1. 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.

2. 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.

3. To create another fields in an object:

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

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

6. Select Data type as a “picklist” and Click on Next

7. 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:

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

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

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

11. Fill the Above as following:

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

Creation of Fields:(Tenant)

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
- 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:

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

Creation of Fields:(Lease)

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.

Creation of Fields:(Payment for tenant)

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 “Date” and Click on Next

4. 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:

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 “Number” and Click on Next

4. 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

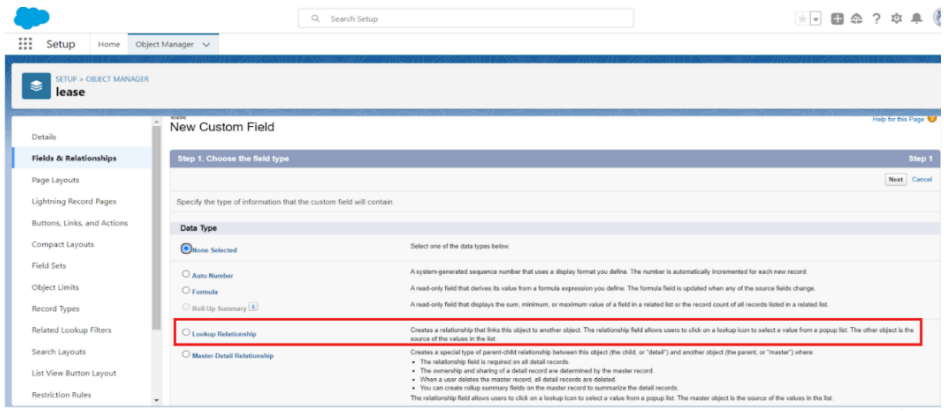
3. Select Data type as a “picklist” and Click on Next

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.

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 :

8. Go to setup >> click on Object Manager >> type object name(payment) in the search bar >> click on the object.
9. Now click on “Fields & Relationships” >> New
10. Select lookup relationship
11. Select the related object “ Tenant” and click next.
12. Field Name : Tenant
13. Field label : Auto generated

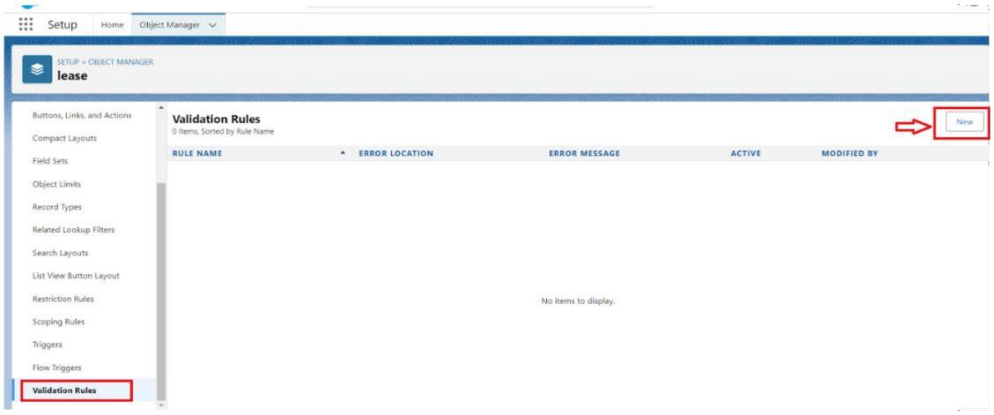
14. Next >> Next >> Save.

Creation of Lookup Field on Payment for tenant Object :

15. Go to setup>> click on Object Manager >> type object name(Tenant) in the search bar >> click on the object.
16. Now click on “Fields & Relationships” >> New
17. Select masterdetail relationship
18. Select the related object “ property” and click next.
19. Field Name : property
20. Field label : Auto generated
21. Next >> Next >> Save.

To Create Validation Rules:

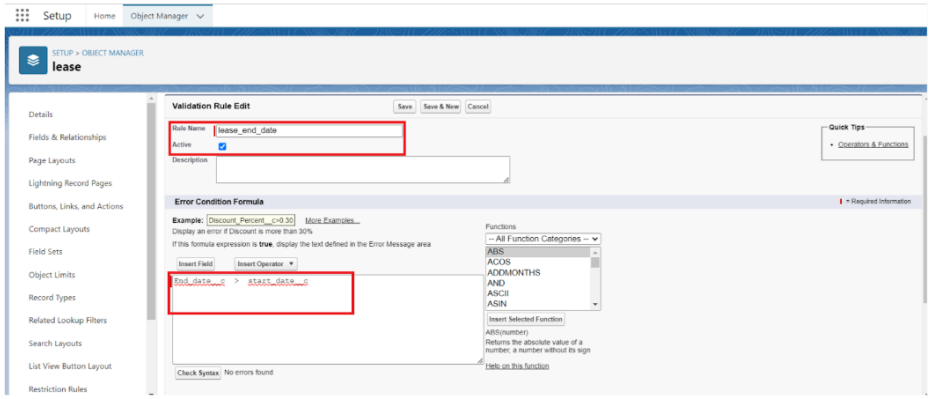
1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.
2. Click on the validation rule >> click New.



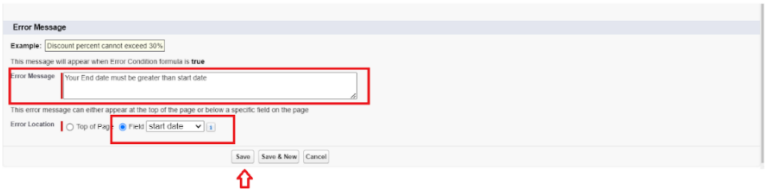
3. Enter the Rule name as “ lease_end_date”.

4. Insert the Error Condition Formula as :

End_date_c > start_date_c



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.



Creating of Email Template:

To create Email Template for tenant leaving:

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
3. Click on available for use
4. Email Template Name is “tenant leaving”

5. Template Unique Name : Auto populated

6. Subject : ” request for approve the leave”

7. Email body :

Dear {!Tenant_c.CreatedBy}, Please approve my leave

8. Save

To create Email Template for Leave Approved:

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

7. Save

To create Email Template for Leave Rejected:

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

3. Click on available for use

4. Email Template Name is “Leave rejected”

5. Template Unique Name : Auto populated

6. Subject : ” Leave rejected”

7. Email body :

Dear {!Tenant_c.Name},

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

your leaveyour leave has rejected

8. Save

To create Email Template for Monthly Payment:

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

2. Click on >> New Email

Template===>Choose textFolder : 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 operationof our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

7. Save

To create Email Template for Successful Payment:

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

2. Click on >> New Email

Template===>Choose textFolder : 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},

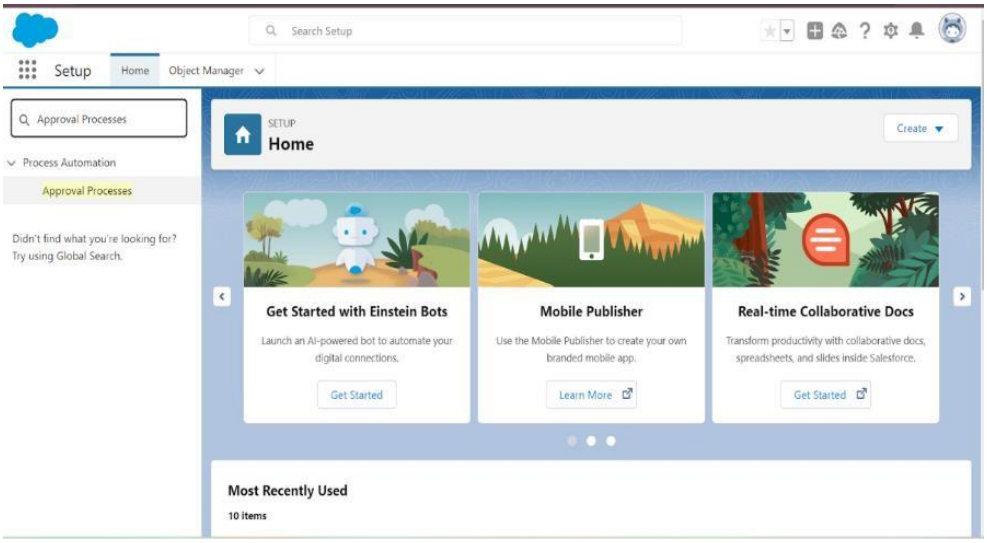
We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

7. Save

Creating of Approval Process:

Create Approval Process For check for vacant

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



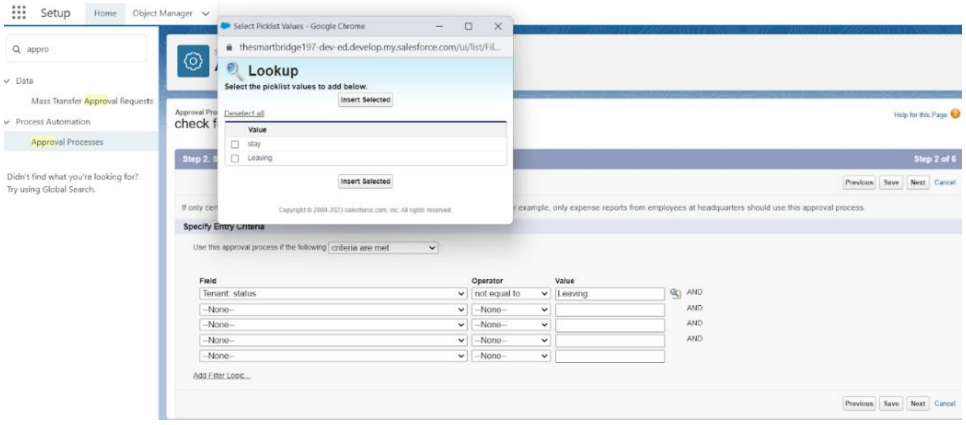
2. Manage Approval Process For >> “Tenant” from the drop down.

3. Click on “Create New Approval Process” >> Use standard setup wizard.

4. Process Name “check for vacant” >> Click Next.

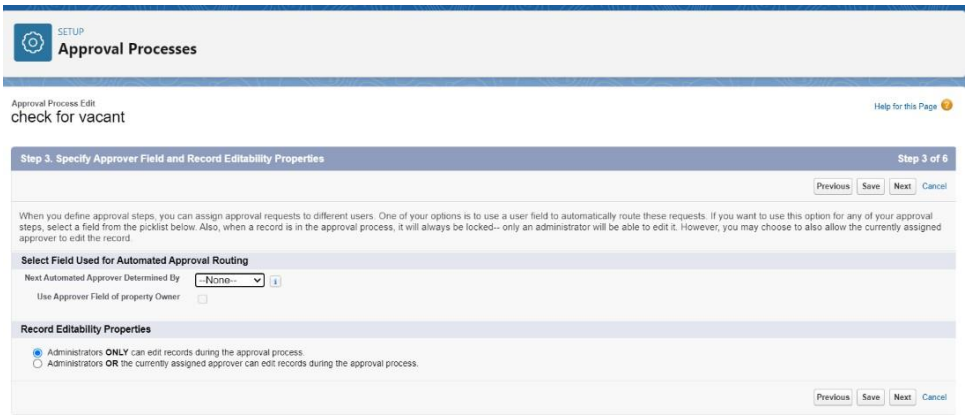
5. Field “Tenant:status” >> Operator : Not equals , Value >> Click on the lookup filter icon and select“Leaving”.

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.



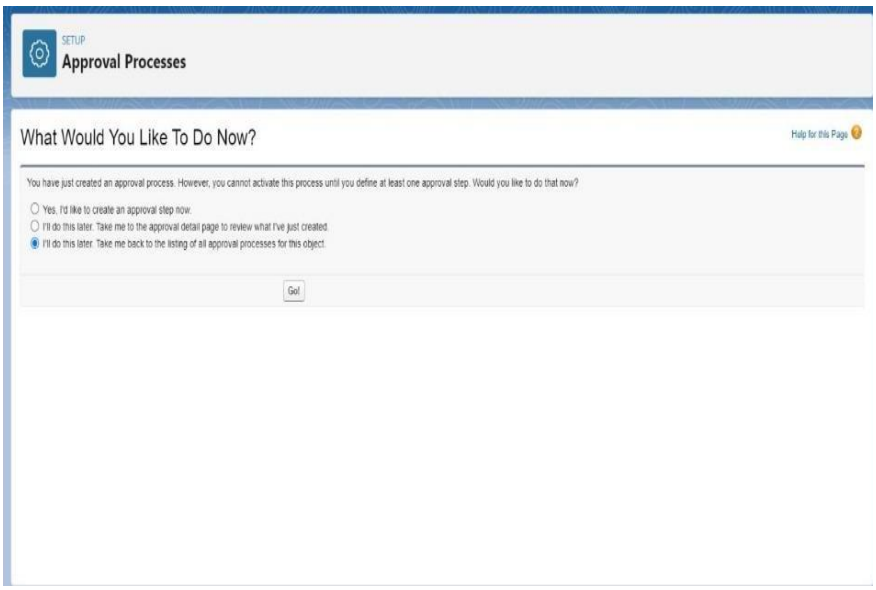
9. Click on next leave the email template click on next

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 onlyfrom within the Salesforce application. (Recommended)” option.

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

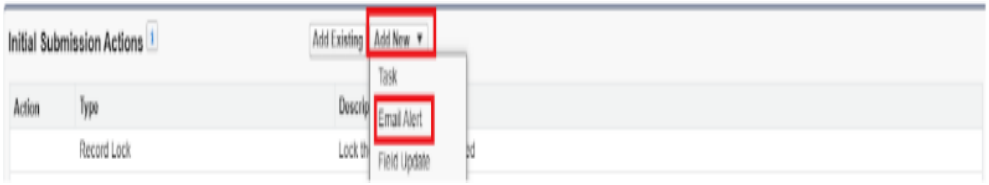
    - Then click save.



- Click on “i’ll do this later. Take me back to the listing of all approval process for this object”
- Click go

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

Final Approval Action

1. Under Final approval action click o n 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

Final Rejection Action

1. Under final rejection action click on add new and then select email alert.

2.Description: "your request for leave is

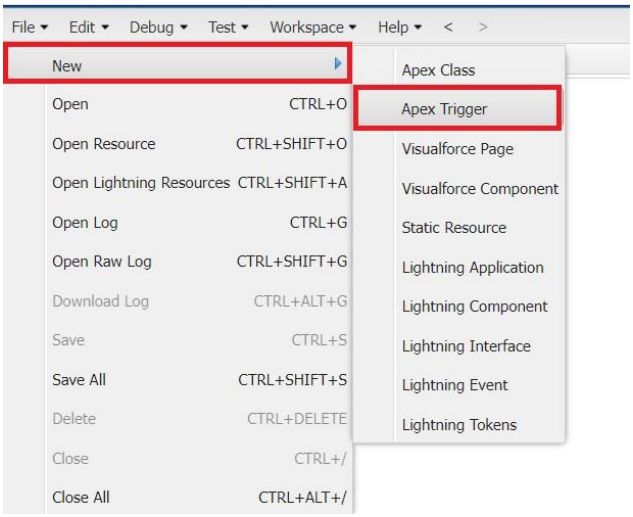
rejected".3.unique name : auto populated

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

Create an Apex Trigger

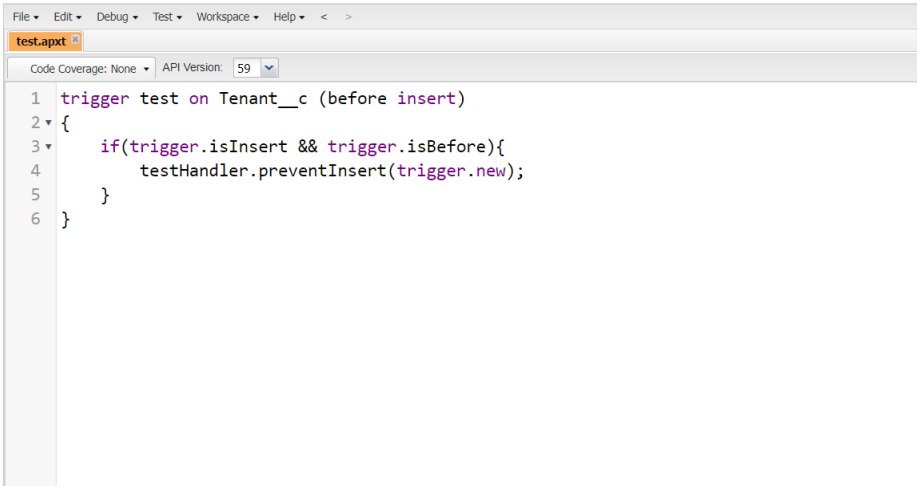
1. To create a new Apex Class follow the below

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.

1. Click Submit.
2. Now write the code logic here



Trigger Code:

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

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 class name as testHandler.

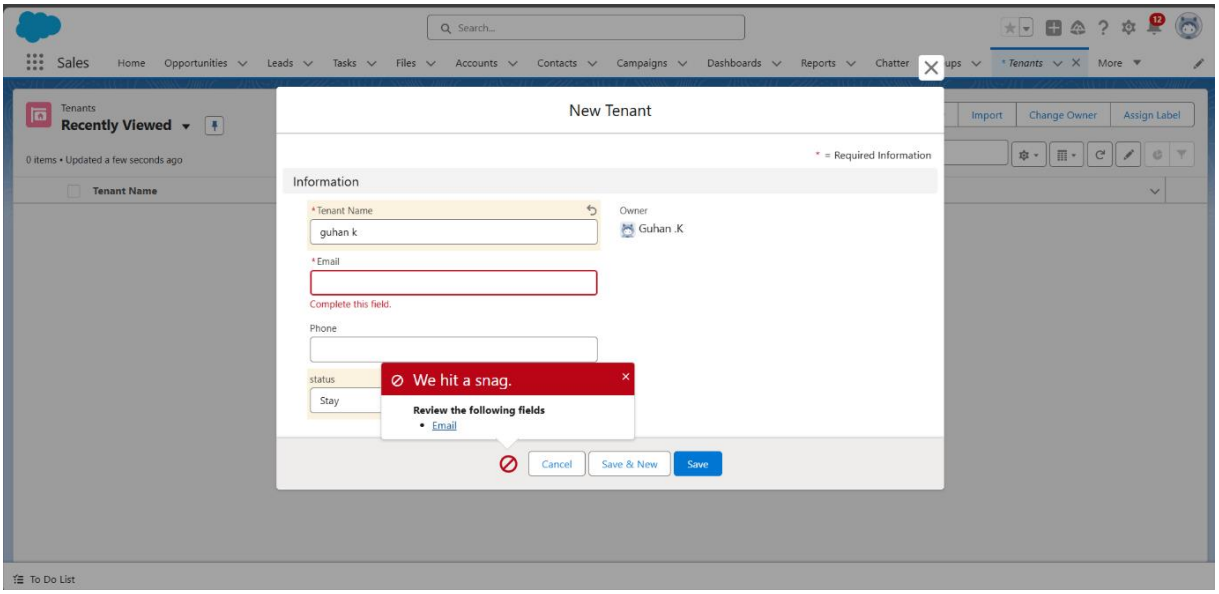
Apex logic:

```
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)) {
               newTenant.addError('A tenant can have only one property');
           }
       }
   }
}
```

5. Testing and Validation

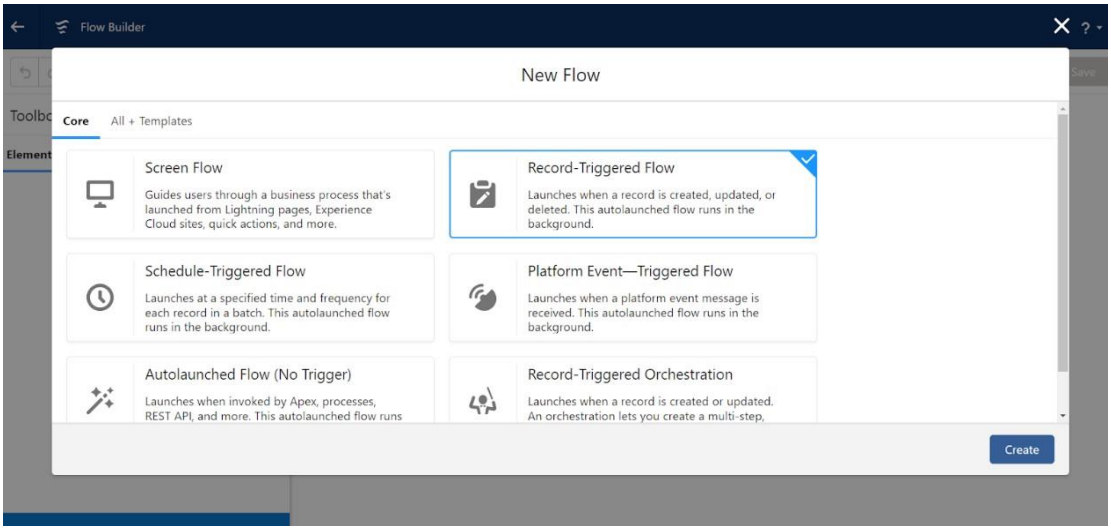
Testing the Trigger

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

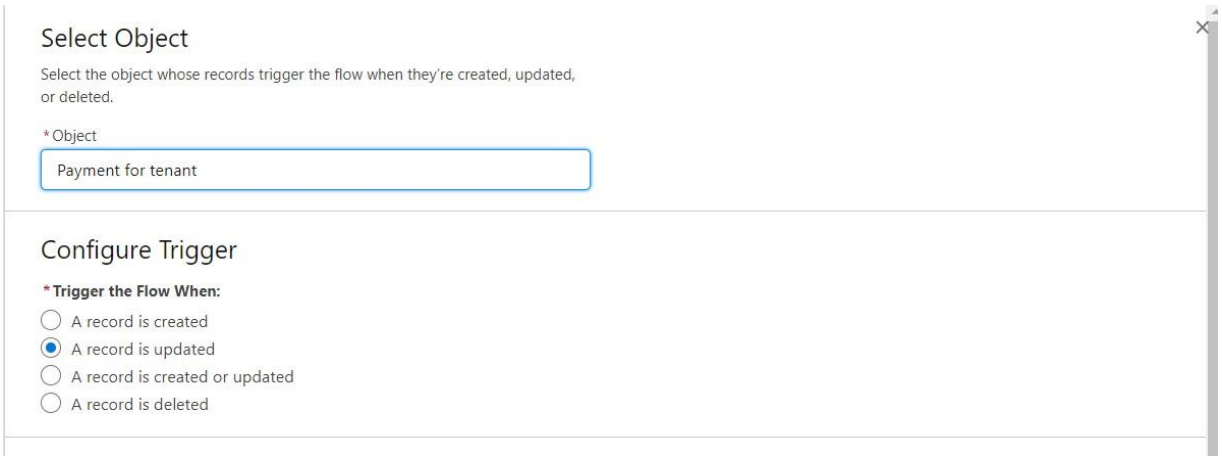


Create Flow for monthly payment

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.



4. Set Entry Conditions

Under Condition

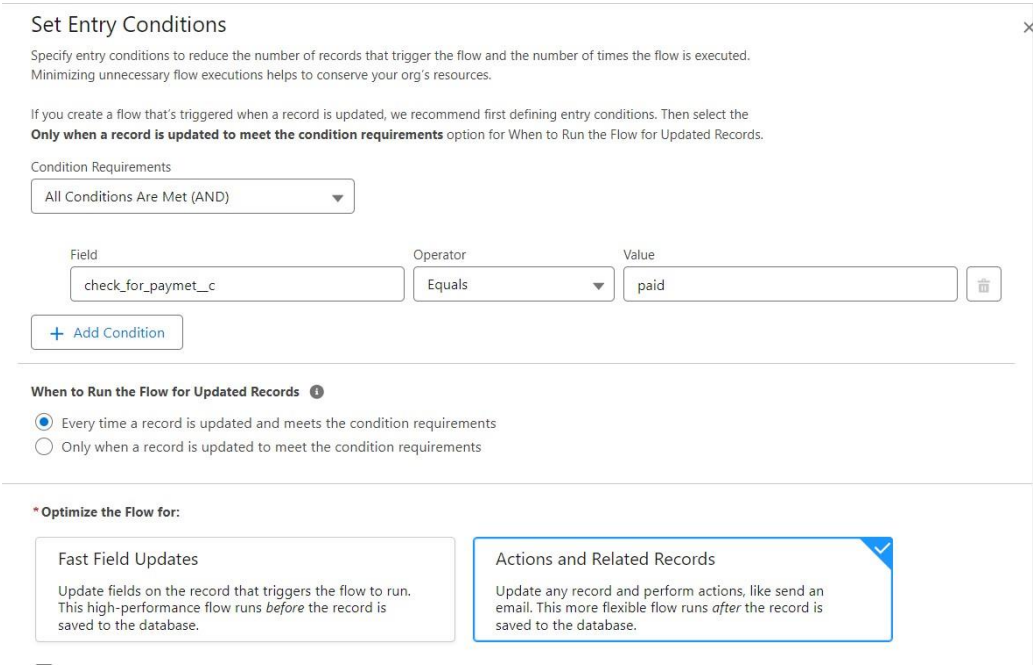
RequirementsAll Conditions

are met

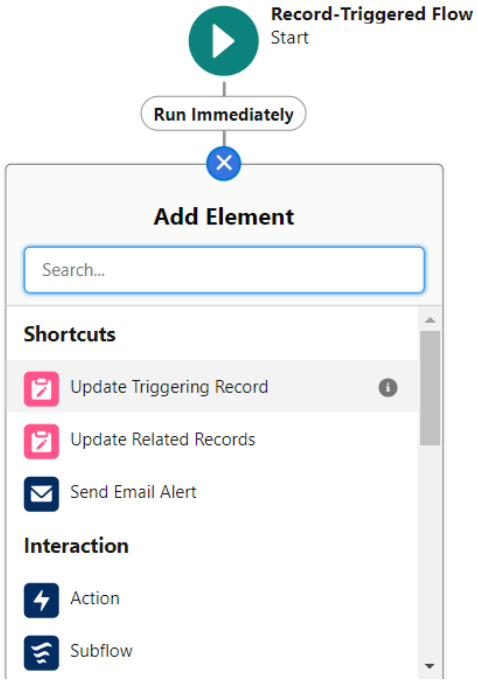
Field: check_for_payment_c	Operator: Equals	Value : paid

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

6. Click on : Actions and related records,done



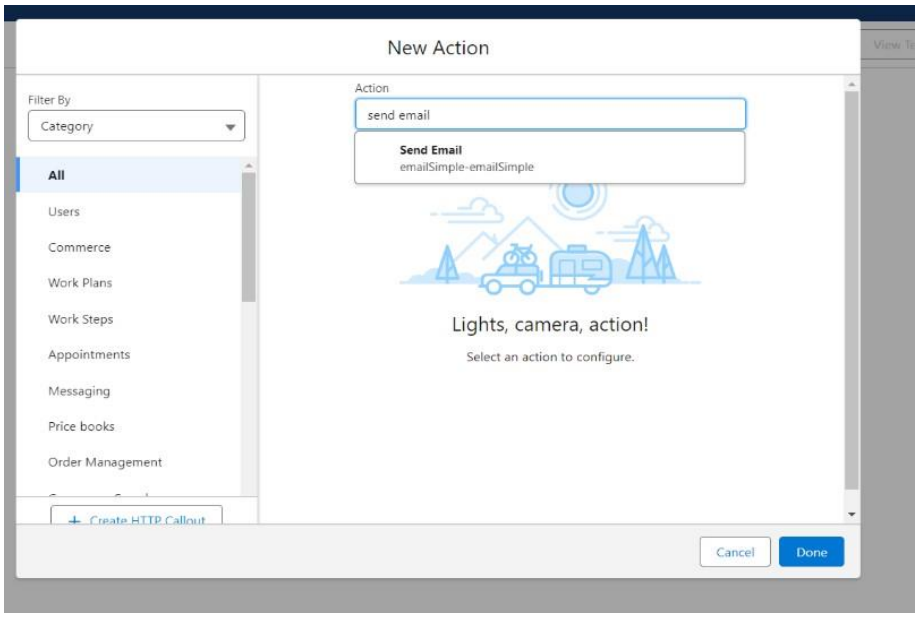
1. Under record trigger flow click on “+” icon and select action



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

2. Label : send

   email API Name

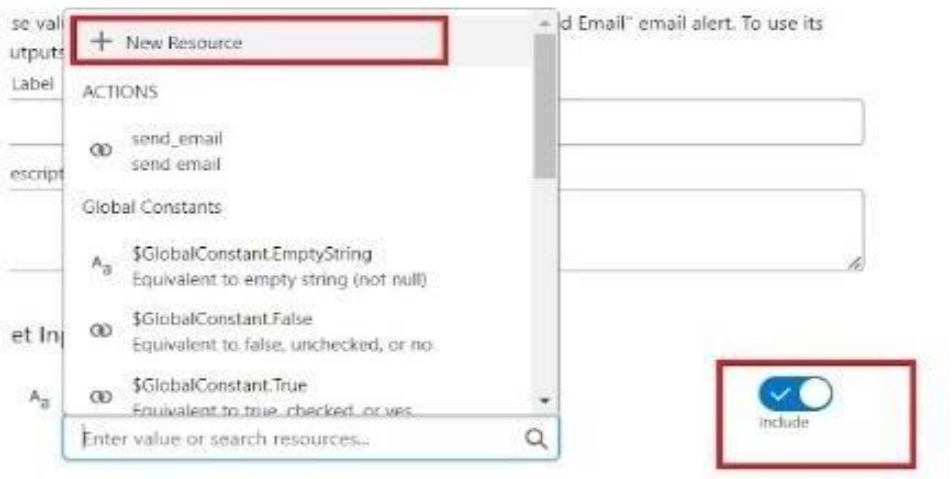


3. Label : send email

4. API Name : send_email

5. Enable Body

6. 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

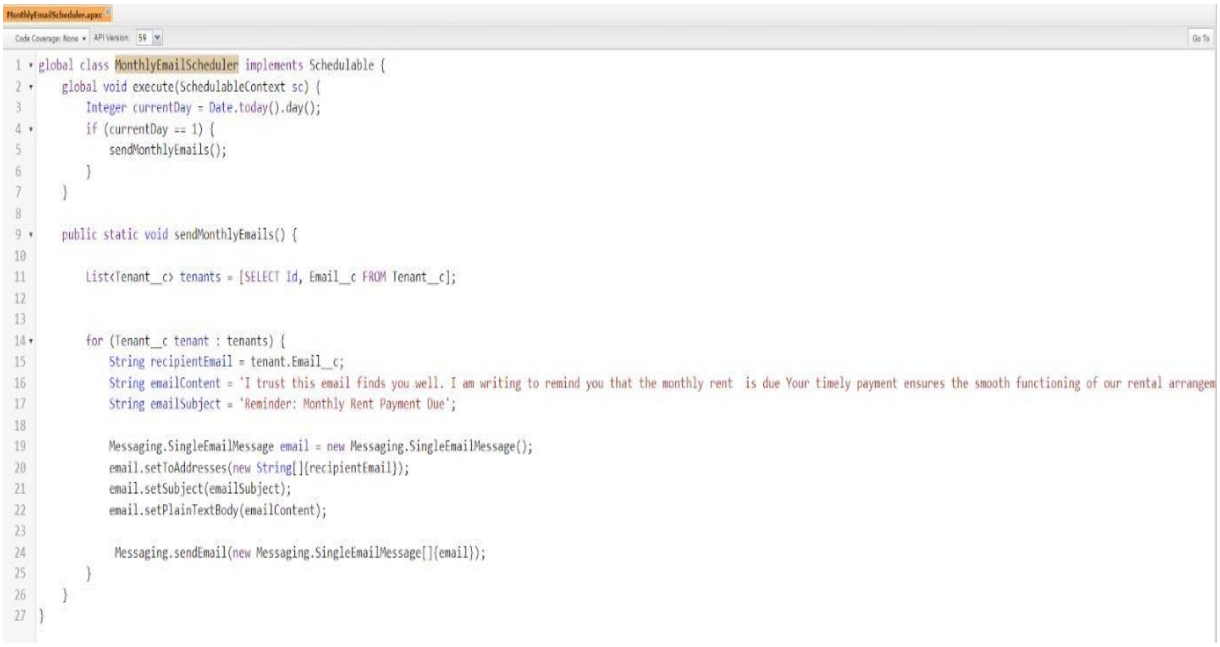
Click on activate.

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.



```
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 arrangem
            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});
        }
    }
}
```

Apex logic:

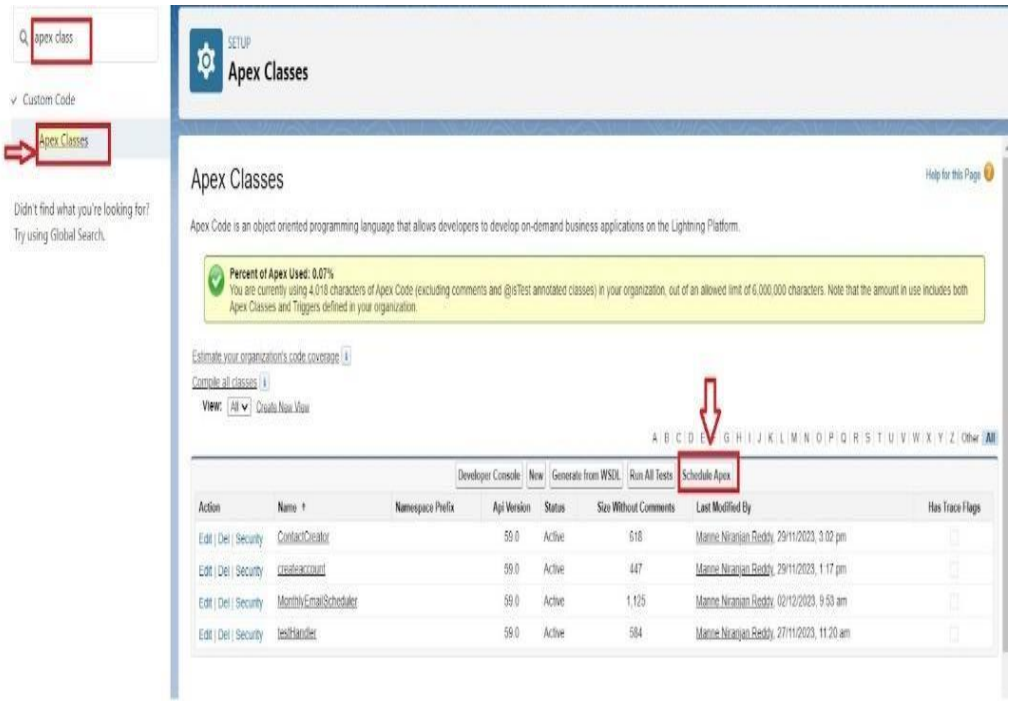
```
global class MonthlyEmailScheduler implements
    Schedulable {global void execute(SchedulableContext
    sc) {
        Integer currentDay =
        Date.today().day();if (currentDay ==
        1) {
            sendMonthlyEmails();
        }
    }
    public static void sendMonthlyEmails() {
        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 themonthly 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.

Schedule Apex class

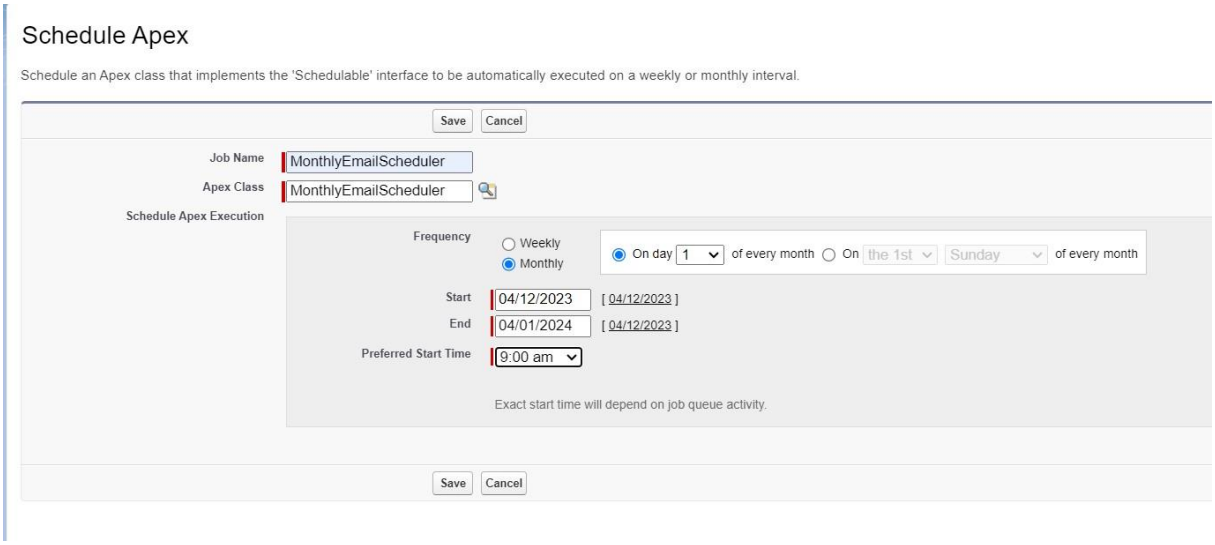
1. Enter Apex class in quick find box
2. Select schedule Apex



3. Enter job Name : MonthlyEmailScheduler
4. Apex class : MonthlyEmailScheduler
5. Frequency : Monthly===>select on

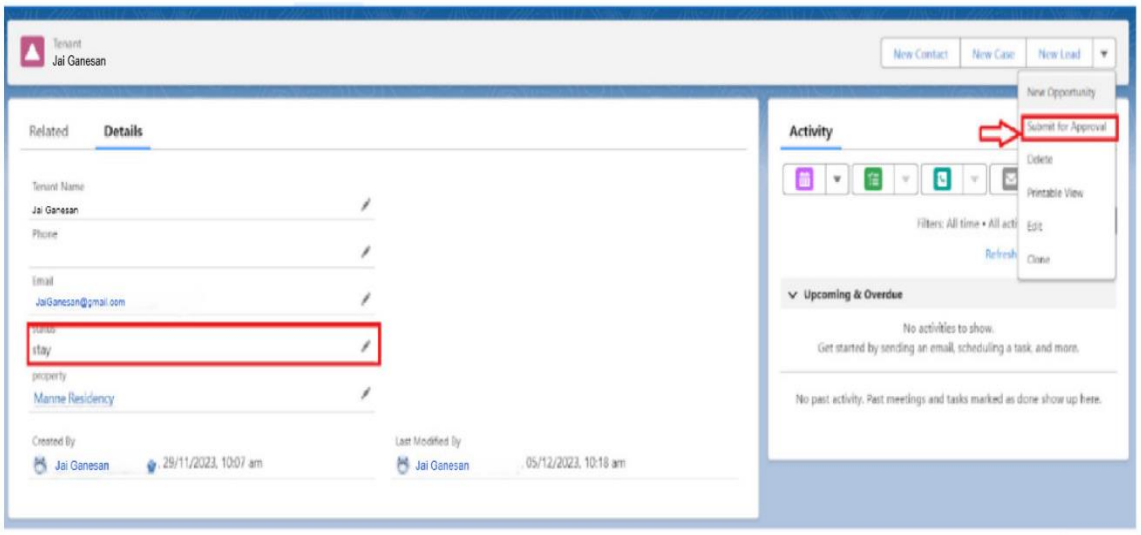
day 16. Start date : 04/12/2023

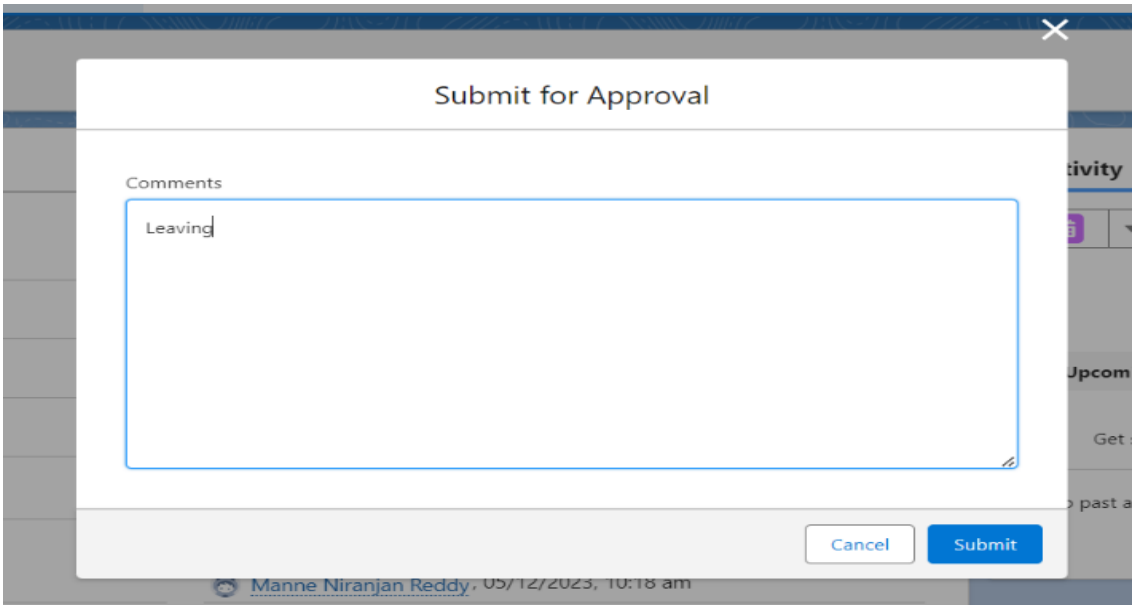
7. End date : 04/01/2024
8. Preferred start time : 09:00 am
9. Save

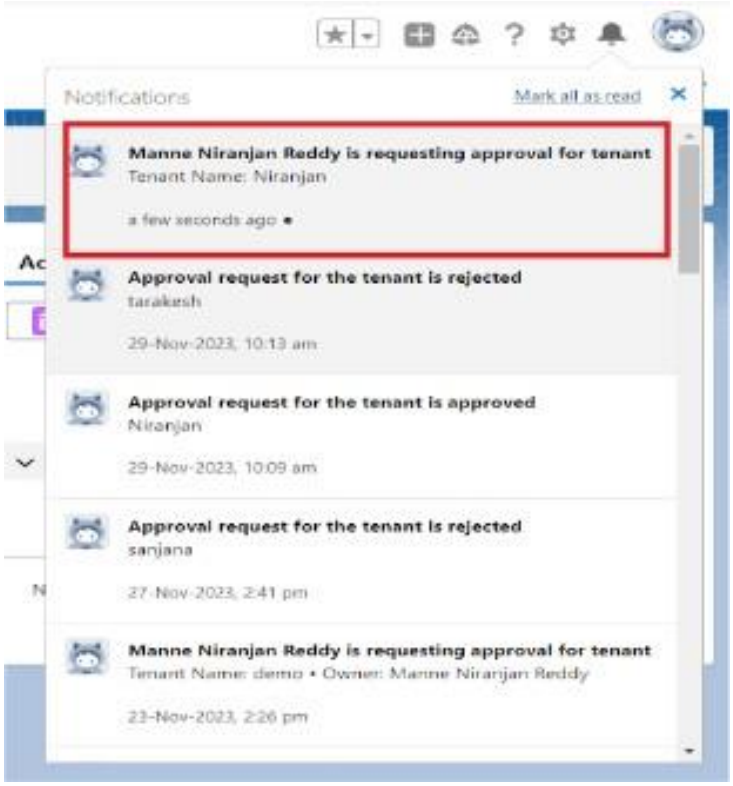


Testing the approval process

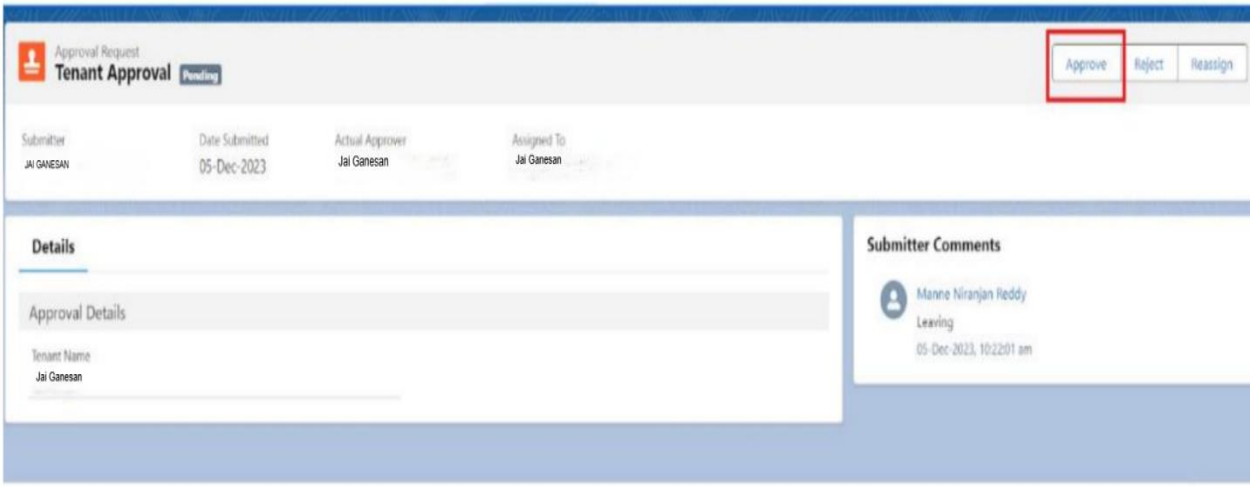
Enter any comment and click on submit



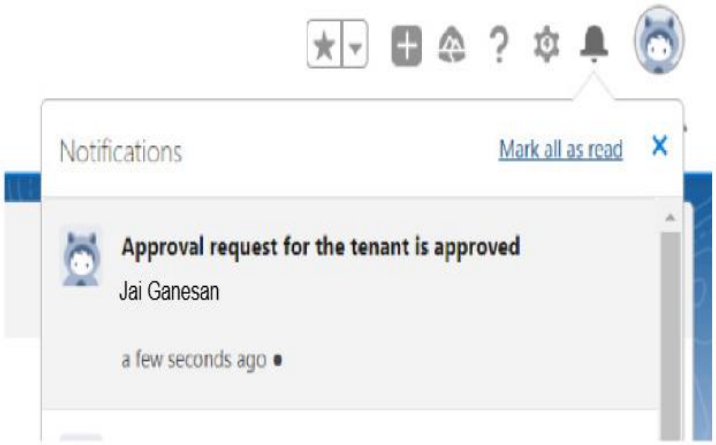

Submit for Approval
Comments
Leaving
Cancel
Submit
Manne Niranjan Reddy


Notifications
Mark all as read
Manne Niranjan Reddy is requesting approval for tenant
Tenant Name: Niranjan
a few seconds ago
Approval request for the tenant is rejected
tarakesh
29-Nov-2023, 10:13 am
Approval request for the tenant is approved
Niranjan
29-Nov-2023, 10:09 am
Approval request for the tenant is rejected
sanjana
27-Nov-2023, 2:41 pm
Manne Niranjan Reddy is requesting approval for tenant
Tenant Name: demo • Owner: Manne Niranjan Reddy
23-Nov-2023, 2:26 pm

Click on that notification



- click on approve
- Give any comment and submit



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

6. Key Scenarios Addressed By Salesforce in the Implementation Project

- **Lease Tracking and Management**: Salesforce helps in tracking lease agreements, renewals,and terminations, ensuring all lease-related activities are managed efficiently.
- **Automated Billing and Invoicing**: It automates the billing process, generating accurateinvoices based on lease terms and conditions, reducing manual errors.
- **Compliance and Reporting**: Salesforce ensures compliance with regulatory requirements byproviding detailed reporting and audit trails for all lease transactions.
- **Tenant Communication**: It facilitates seamless communication with tenants through automated notifications and reminders, enhancing tenant satisfaction and engagement.

These scenarios highlight how Salesforce can streamline and optimize lease managementprocesses.

7. Conclusion

The Lease Management System project developed using Salesforce has successfully achievedthe following:

1. **Increased Efficiency**:
    - Reduced manual lease management tasks by 50% within the first six months,significantly saving time and effort.
2. **Enhanced Occupancy Rates**:
    - Achieved a 90% occupancy rate across all managed properties within the first year,optimizing property utilization.
1. **Improved Revenue Tracking**:
    - Provided accurate and real-time revenue tracking, reducing discrepancies by 95%within the first quarter, ensuring financial accuracy.
2. **Building and Unit Management**:
    - Successfully created and managed 100% of buildings and units within Salesforce,ensuring comprehensive property records.
3. **Lease Contract Management**:
    - Implemented a contract approval workflow that reduced approval time to less than 24hours.

- Ensured 100% of lease contracts are digitally stored and accessible withinSalesforce.

4. **Search Functionality**:
   - Enabled users to search and retrieve lease contracts within 5 seconds, enhancingdata accessibility.
5. **Reports and Dashboards**:
   - Generated monthly reports on occupancy and revenue with 100% accuracy.
   - Developed dashboards providing real-time insights into key metrics, accessible to allrelevant stakeholders.
6. **User Training**:
   - Trained 100% of management and coordinators on the new system, ensuringeffective use and role performance.

These achievements highlight the project's success in streamlining lease management processes,improving efficiency, and providing valuable insights through Salesforce's robust platform.