A CRM Application to Manage the Mall

Submiited

By

SARANRAJ K (712921104048)

HARIKRISHNAN G V (712921104014)

SUBHASH SHANKAR V (712921104053)

KRISHNA PERUMAL (712921104901)

Project Description:

The CRM (Customer Relationship Management) application is designed to enhance and streamline the management of a shopping mall. It focuses on three key areas: tenant management, tenant issue tracking, and lease management. This project aims to centralize information, automate processes, and provide insightful analytics to improve operational efficiency and tenant satisfaction.

1. **Objectives**

List the specific, measurable goals the project intends to achieve.

Examples: Business Goals:

- Centralized Tenant and Lease Management:
- Improved Issue Tracking and Resolution:
- Optimized Lease Renewal and Expiry Tracking:
- Enhanced Reporting and Data Insights:
- Increased Efficiency and Collaboration:

Specific Outcomes:

- Tenant Details Management:
- Tenant Issue Tracking:
- Lease Tracking & Expiry Alerts:
- Customized Reporting & Dashboards:
- Process Automation and Workflow Optimization:
- Collaboration Tools:

2. **Data Models (Objects, Fields, and Relationships)**

A. Custom Objects

To represent the key elements of the CRM, we will create the following custom objects in Salesforce:

- Tenant Details Object: Stores information related to each tenant.
- Tenant Issues Object: Manages the tracking of issues reported by tenants.
- Lease Tracking Object: Handles lease agreements and their status.

Steps to Create Custom Objects:

1. **Navigate to Object Manager in Salesforce**:
   - Go to Setup → Object Manager → Create → Custom Object.
2. **Create the Tenant Details Object**:
   - Name: Tenant Details
   - Label: Tenant Details
   - API Name: Tenant_Details
   - Description: Object for storing all tenant-related information (tenant profile, contact, lease history).
   - Fields:
     - Tenant Name (Text)
     - Contact Info (Phone, Email)
     - Tenant Type (Picklist: Retail, Service, etc.)
     - Unit Number (Text)
     - Lease Start Date (Date)
     - Lease End Date (Date)
     - Payment History (Currency)
3. **Create the Tenant Issues Object**:
   - Name: Tenant Issues
   - Label: Tenant Issues

- API Name: Tenant_Issues
- Description: Object to track issues or complaints reported by tenants.

- Fields:
  - Issue Type (Picklist: Maintenance, Security, Customer Service)
  - Issue Description (Long Text)
  - Priority (Picklist: Low, Medium, High)
  - Status (Picklist: Open, In Progress, Resolved)
  - Issue Report Date (Date)
  - Resolution Date (Date)
  - Assigned Team (Lookup to Users)

4. **Create the Lease Tracking Object**:
   - Name: Lease Tracking
   - Label: Lease Tracking
   - API Name: Lease_Tracking
   - Description: Object for managing lease agreements and renewal statuses.
   - Fields:
     - Lease Start Date (Date)
     - Lease End Date (Date)
     - Rent Amount (Currency)
     - Renewal Status (Picklist: Active, Pending, Expired)
     - Rent Payment Status (Picklist: Paid, Pending, Overdue)
     - Tenant (Lookup to Tenant Details Object)
5. **Define Relationships Between Objects**:
   - Tenant Details → Lease Tracking: One-to-many relationship (A tenant can have multiple leases).
   - Tenant Details → Tenant Issues: One-to-many relationship (A tenant can report multiple issues).

- Lease Tracking → Tenant Issues: Indirect relationship (Leases may relate to specific tenant issues).

3.Testing and Validation

Testing and validation are critical steps to ensure that the CRM application for managing mall-related tenant details, tenant issues, and lease tracking is functioning as intended. Here's an approach for testing, focusing on Unit Testing (for Apex Classes and Triggers) and User Interface (UI) Testing:

4. Unit Testing (Apex Classes, Triggers)

Unit testing is essential in Salesforce to validate the business logic implemented in Apex Classes and Triggers. These tests ensure that the system is correctly processing tenant data, lease information, and tenant issues.

1 .Apex Classes Testing

Apex classes will contain the business logic that manages tenant details, lease agreements, and tenant issues. Unit tests for these classes will focus on ensuring that each method performs as expected under various scenarios.

Approach:

- Test Class Creation: For each Apex class, a corresponding test class will be created to simulate the logic with test data.
- Test Coverage: Each method within the Apex class should be tested to ensure 100% code coverage. Salesforce requires at least 75% code coverage for deployment to production.

- Mock Data: Use test data that closely resembles real-world tenant, lease, and issue records (e.g., tenant names, lease start/end dates, issue descriptions).
- Edge Case Scenarios: Validate edge cases such as lease start and end dates that are identical, invalid tenant issue types, and lease violations.
- Bulk Testing: Test the ability of the class to handle bulk records to avoid hitting governor limits (e.g., processing multiple tenants or issues in one transaction).

Example Unit Test for Apex Class (Tenant Management)

Apex:

```
@isTest
private class TenantManagementTest {
    @isTest
    static void testTenantCreation() {
        // Create mock data
        Tenant__c testTenant = new Tenant__c(Name='Mall Tenant',
Lease_Start_Date__c=Date.today(), Lease_End_Date__c=Date.today().addYears(1));
        insert testTenant;

        // Call the method that processes tenant data
        TenantManagement.createTenant(testTenant);

        // Validate results
        Tenant__c result = [SELECT Name FROM Tenant__c WHERE Id =
:testTenant.Id];
        System.assertEquals('Mall Tenant', result.Name);
    }
}
```

2.Apex Trigger Testing

Apex triggers are used to execute logic automatically when records are inserted, updated, or deleted. In the context of this CRM, triggers might handle tenant creation, updates to lease information, or triggering workflows when an issue is logged.

5. User Interface (UI) Testing

UI testing focuses on ensuring that the front-end interface of the CRM application (where users interact with tenant details, issues, and lease information) is intuitive, functional, and responsive.

Testing Tools:

- Apex Tests: Salesforce provides built-in testing tools for Apex, and you can use the Developer Console or Visual Studio Code with Salesforce extensions to run tests and check code coverage.
- Selenium/Provar: For UI testing, tools like Selenium or Provar (a Salesforce-specific automation testing tool) can be used for automated UI tests.
- Browser Developer Tools: These tools help test UI responsiveness and troubleshoot any frontend issues.
- Jest (for LWC testing): If the CRM includes Lightning Web Components (LWC), Jest is a framework for testing JavaScript code for UI components.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

In a CRM application designed for mall management, which focuses on handling tenant details, tenant issues, and lease tracking, Salesforce can address a wide variety of business use cases and scenarios to streamline operations, improve tenant relationships, and ensure smooth lease management. Below are the key scenarios that Salesforce can handle during the implementation of such a CRM system:

1. Tenant Onboarding and Management

Scenario: **A new tenant is interested in leasing space in the mall.**

- Use Case: Automating the onboarding process for new tenants, including collecting necessary tenant details (e.g., contact information, business type, lease terms), storing lease agreements, and setting up account records.
- Salesforce Solution: Salesforce allows the creation of custom objects (e.g., Tenant__c) and fields for managing tenant records. Using Lightning Forms and Flow Automation, onboarding tasks can be automated, such as assigning a property manager, sending welcome emails, and scheduling lease start dates.

Key Features Addressed:

- Custom Objects: Create a Tenant__c object with custom fields to capture detailed information about each tenant.
- Process Automation: Using Process Builder or Flows, automate the tenant onboarding process, including notifications and task assignments.
- Record Types: Differentiate between types of tenants (e.g., retail, food court, etc.) by using Record Types for custom layouts and fields.

2. Lease Management and Tracking

Scenario: Managing the lease agreements, renewals, and amendments for tenants.

- Use Case: Managing lease terms (e.g., start/end dates, rent amounts, renewal clauses), tracking lease milestones (e.g., renewal dates, rent increases), and automating lease renewal reminders.
- Salesforce Solution: Salesforce enables the management of lease data via custom objects and related lists to track key lease milestones. Lease dates can trigger automatic reminders for renewals, rate adjustments, and other critical events via Workflow Rules, Process Builder, or Salesforce Flow.

Key Features Addressed:

- Custom Objects & Relationships: Create a Lease__c object related to Tenant__c to store lease details, such as start and end dates, renewal options, and rent terms.

- Reminder Automation: Use Apex Triggers, Process Builder, or Salesforce Flow to send automated reminders or create tasks for lease renewals, escalations, and negotiations.
- Document Management: Store lease agreements and contracts as attachments or Salesforce Files, ensuring that all related documentation is easily accessible and linked to the tenant's record.

---

3. Tenant Issue Management (Service Requests & Maintenance)

Scenario: Tenants face issues with their leased spaces (e.g., maintenance requests, security concerns).

- Use Case: Logging tenant issues, assigning service requests to maintenance teams, and tracking resolution progress. This ensures that tenants' concerns are addressed promptly.
- Salesforce Solution: Salesforce Service Cloud enables issue management by capturing Service Requests in the form of Cases. Workflow automation can route cases to the appropriate team, set priorities, and track case resolution timelines.

Key Features Addressed:

- Case Management: Leverage Salesforce Cases to log and track tenant issues, including details on the issue type, priority, and status.
- Assignment Rules: Use Assignment Rules to automatically assign service requests to the relevant teams (e.g., maintenance, security, or facilities management).
- SLAs and Escalations: Set Service Level Agreements (SLAs) for issue resolution and create Escalation Rules to ensure critical issues are addressed in a timely manner.
- Knowledge Base: Provide a self-service portal or knowledge base for tenants to view FAQs or check the status of open issues.

Summary of Key Scenarios Addressed by Salesforce:

Salesforce addresses critical business scenarios in the mall management CRM application, including:

- Tenant onboarding and management
- Lease tracking and renewals
- Tenant issue management
- Tenant communications and notifications
- Billing and payment tracking
- Reporting and analytics
- Document and contract management
- Compliance and security

What you'll learn

1. Real Time Salesforce Project
2. Creating Custom Objects
3. Creating a custom tab for tenant object
4.Object & Relationship in Salesforce
5. Formula fields and Validation rules
6. Create the Lightning App
7. Record Insertion
8. Creating Flows
9. Apex Triggers
10.Asynchronous Apex
11.Reports & Dashboards.

System Requirements:

1. Operating System: Windows 11
2. Processor: 12th Gen Intel(R) Core(TM) i3-1215U 1.20 GHz
3. System type: 64-bit operating system, x64-based processor

4. Installed RAM: 8.00 GB
5. Bandwidth of 30mbps

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?

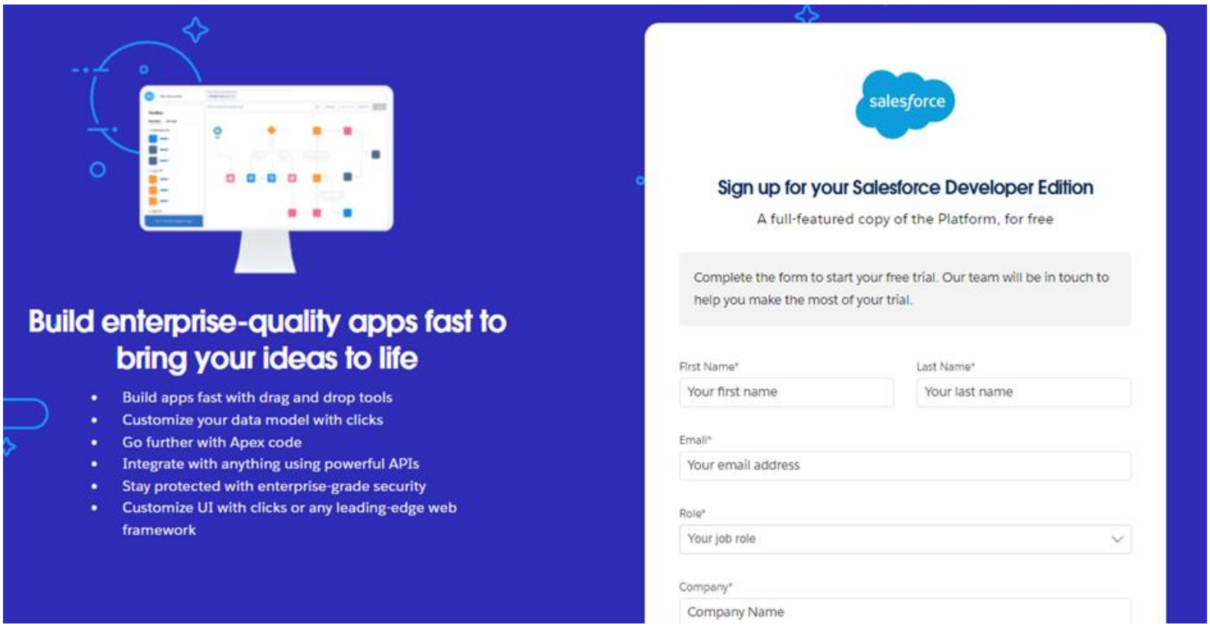
Salesforce is your customer success platform, designed to help you sell, service, market, analyse, and connect with your customers. Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

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

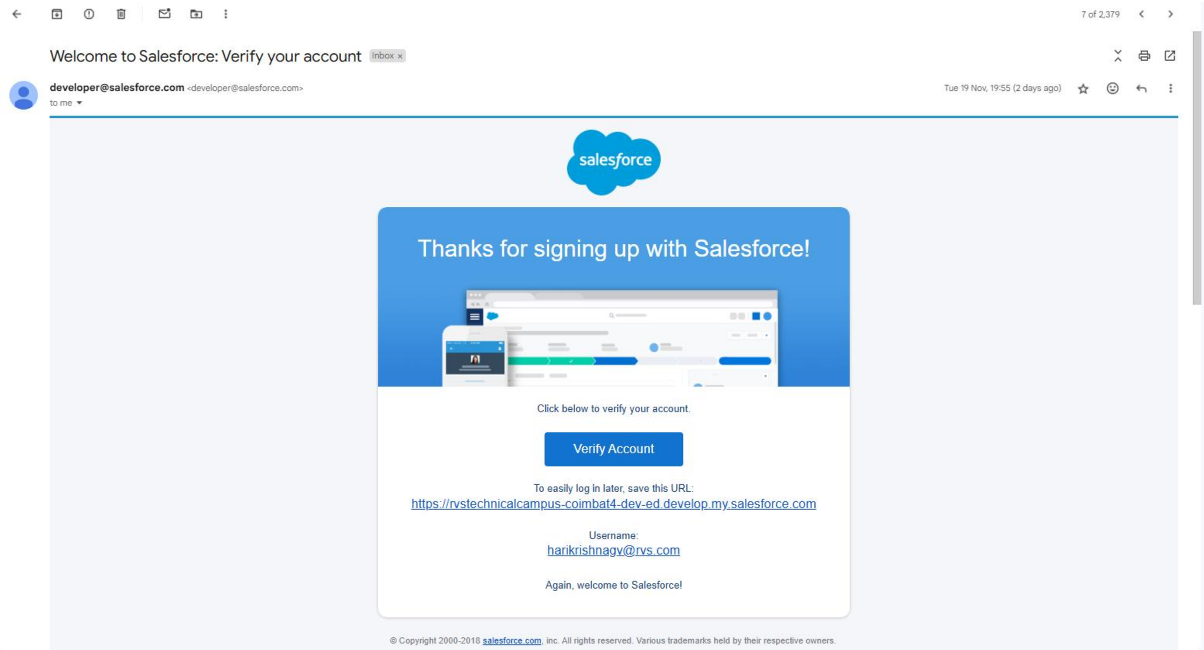
7) Username: should be a combination of your name and company This need not be an actual email id, you can give anything in the format :

username@organization.com

Click on sign me up after filling these.

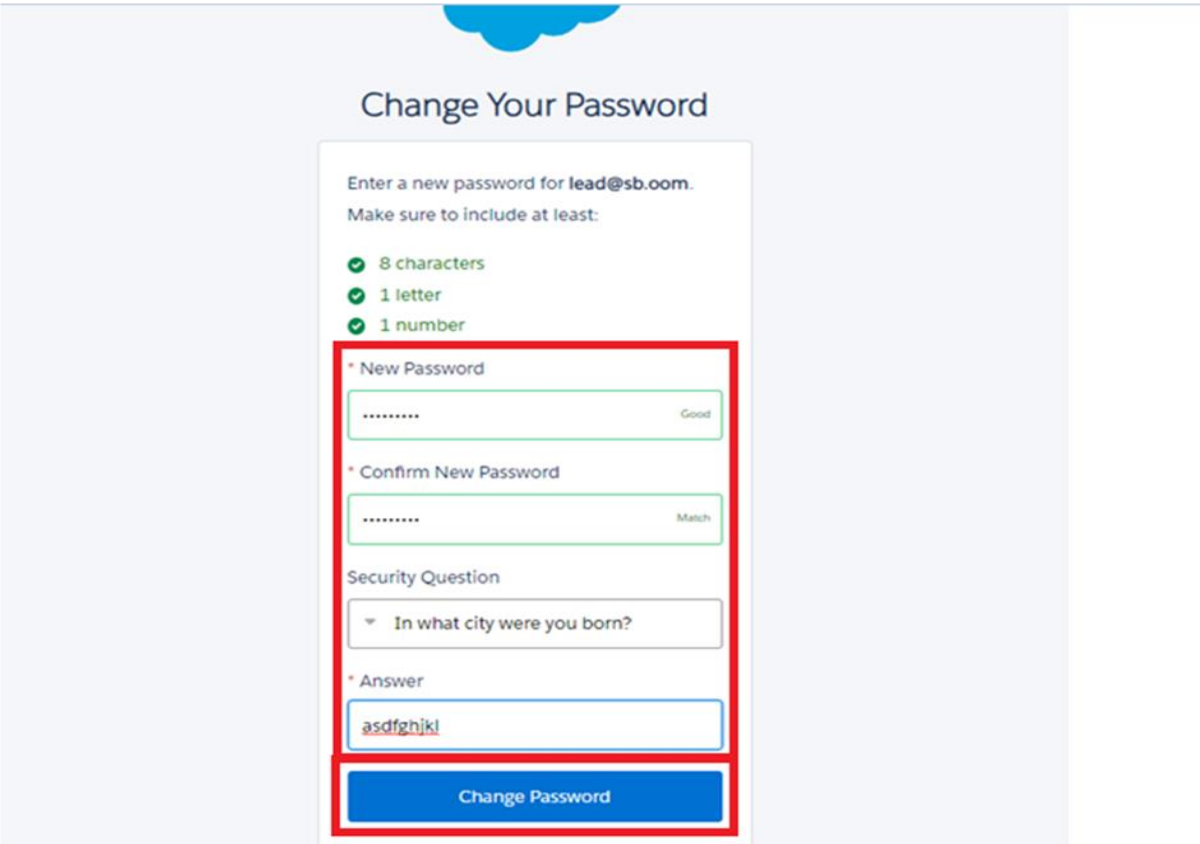
Activity 2: Account Activation:

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.

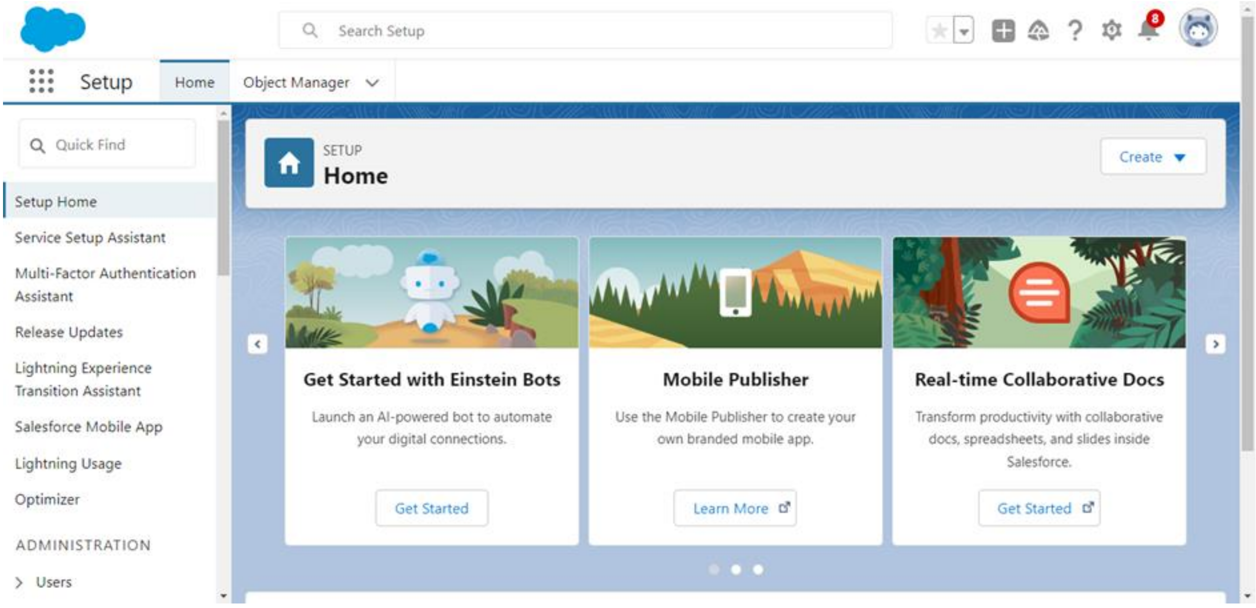


2. Click on Verify Account

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



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



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.

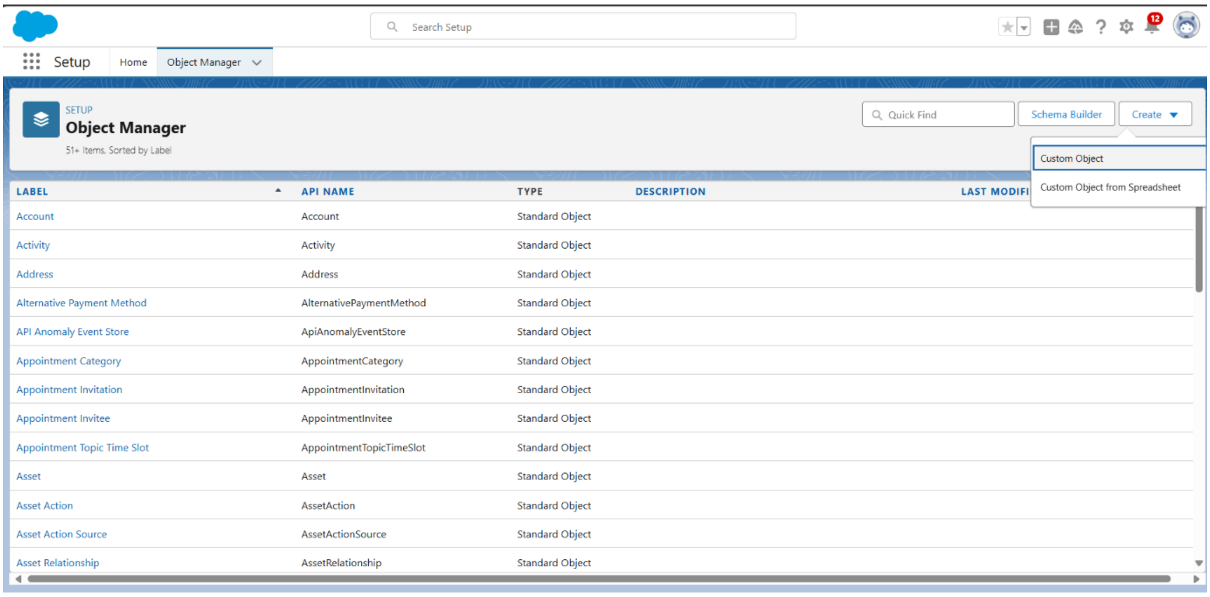
Use Case: Creating an object in Salesforce organization is essential for efficient data management and process automation. By defining custom objects, businesses can structure and store data specific to their needs, enabling streamlined workflows, personalized reporting, and enhanced user experiences. Objects serve as the foundation for organizing and leveraging critical information within Salesforce.

To create an object:

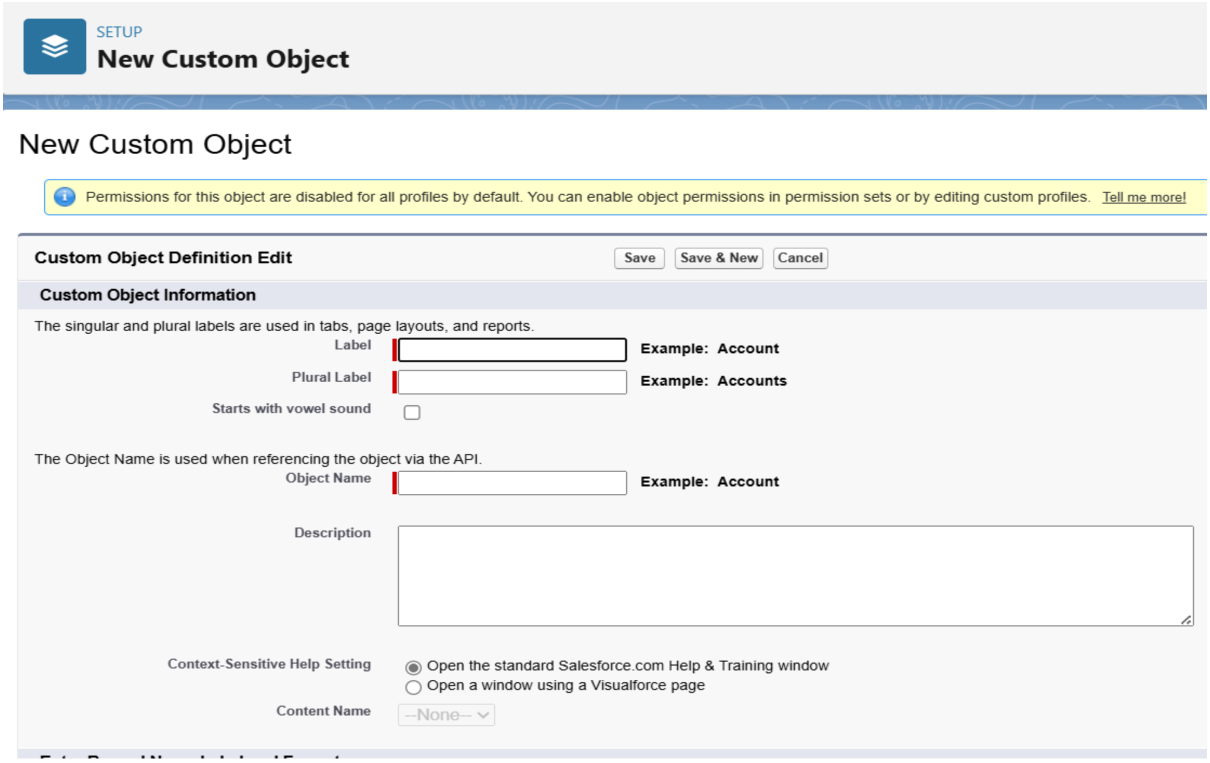
Session 1:Create Custom Objects

Activity 1:Create Tenant 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
    - Record Name : Tenant Name
    - Data Type: Text
4. Click on Allow reports



5. Allow search and Save.

Following the previous steps: Activity 2 ,Activity 3

Activity 2: Create Lease Tracking Object

1. Enter the label name ==>Lease Tracking

2. Plural label name ==> Leases Tracking

3. Enter Record Name Label and Format

- Record Name : Lease Tracking No
- Data Type : Auto Number
- Display Format - TT - {000000}

4. Click on Allow reports

5. Allow search and Save.

Activity 3: Create Tenant Issues Object

1. Enter the label name ==> Tenant Issue

2. Plural label name ==> Tenant Issues

3. Enter Record Name Label and Format

- Record Name : Issues
- Data Type : Auto number

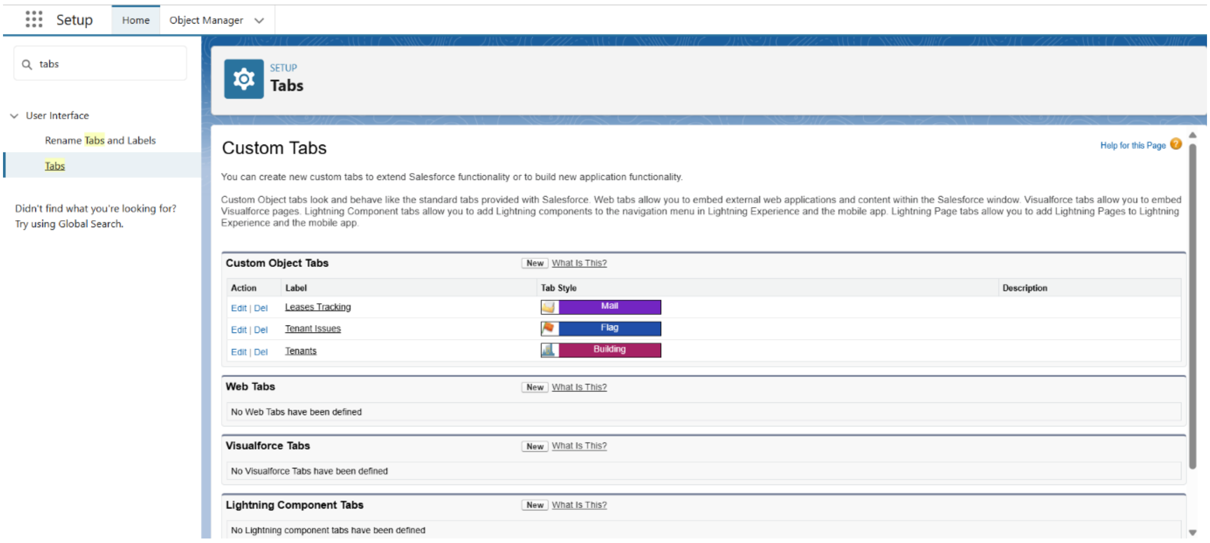
4. Click on Allow reports.

5. Allow search and Save.

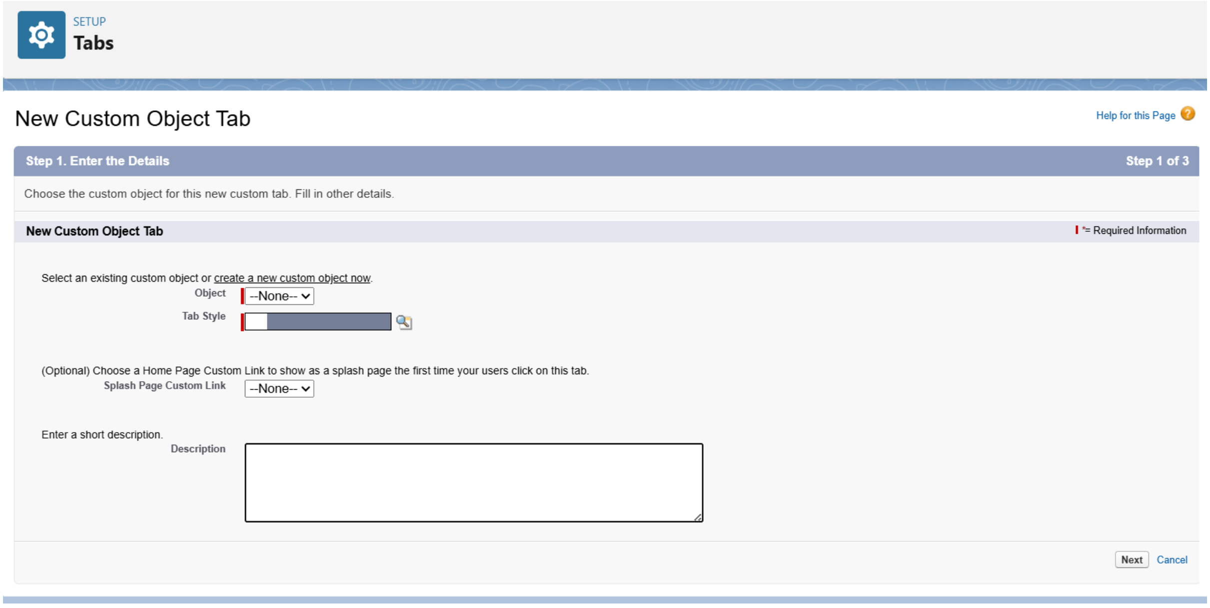
Session 2:Tabs

Activity 1: Create a custom tab for tenant object

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



2. Select Object(Tenant) ==> Select the tab style ==> Next (Add to profiles page) keep it as default ==> Next (Add to Custom App) uncheck the include tab ==> Save.
3. Make sure to append tab to users' existing personal customizations is checked.



Note : Similarly, create tabs for Lease Tracking and Tenant issues.

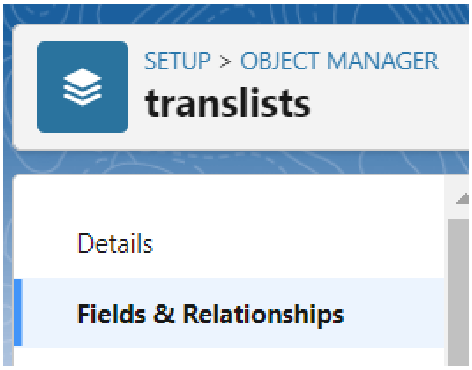
Session 3:Create Fields and Relationships

Activity 1:Create Fields on Tenant object

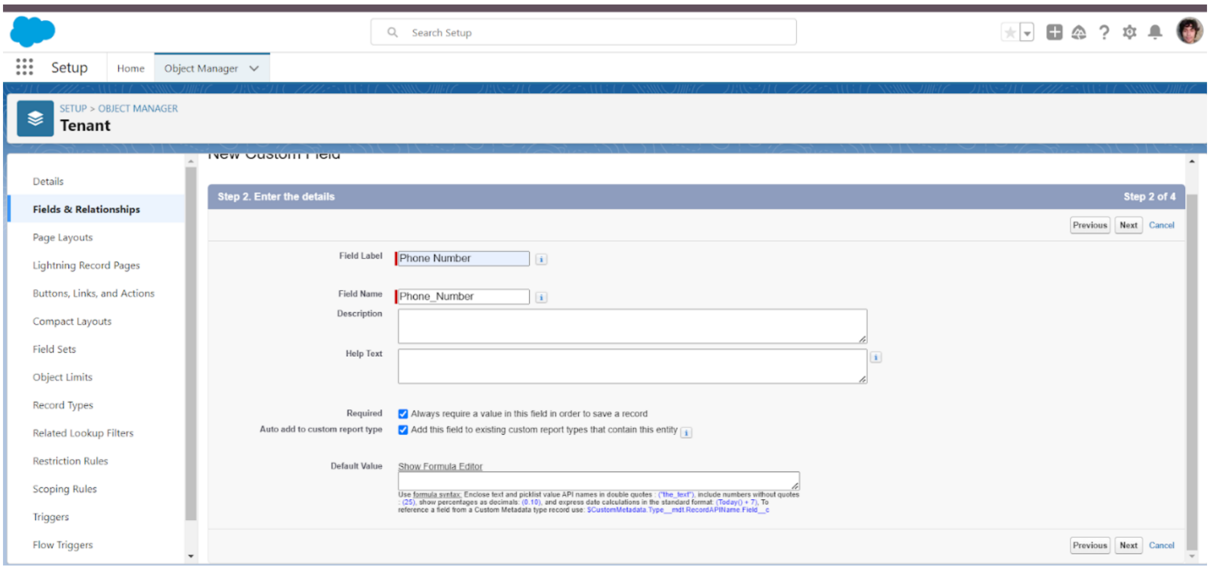
1.While still on your salesforce account, navigate to the gear icon present in the top right corner. You will notice Setup and click on setup.

2.You will now be navigated to the setup page, and click on object manager and search for object "Tenant".

3.Click on "Fields & Relationships" in the left panel.



1. Click on New and choose the data type Phone and first name : Phone Number.
2. Click next and fill the following details in the mentioned.



3. Click Next, Next and click on “Save and New”.

- Note : Repeat the same steps to create the fields :

S No	Field Label	Data Type
1	Address	Text Area(255)
2	PAN Card	Text
3	Date of Reg	Date
4	Email	Email
5	GST No	Text
6	Registered License No	Text
7	Shop Act license No	Text
8	Status of possession	Pick List Pending Hand Overed Renewal Needed Closed

Activity 2: Create Validation Rules for Tenant Object:

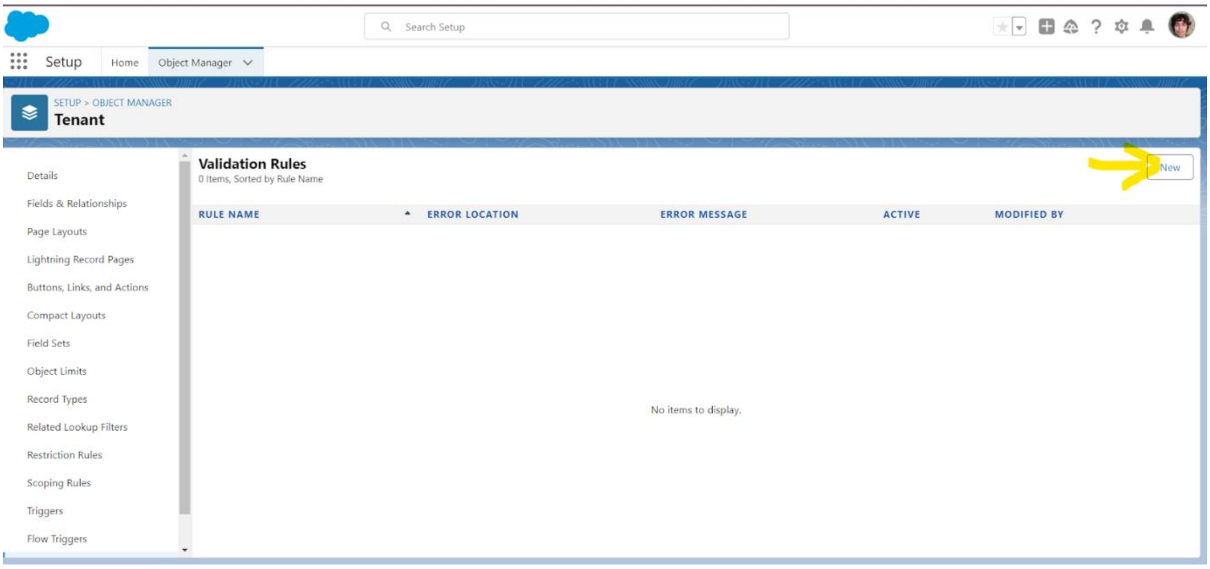
A) Create validation rule for Phone Number

Note:- check if the Phone Number is valid having 10 digits if not then show error.

1] Go to setup ==> click on Object Manager ==> type object name(Tenant) in quick find bar==>click on the object.

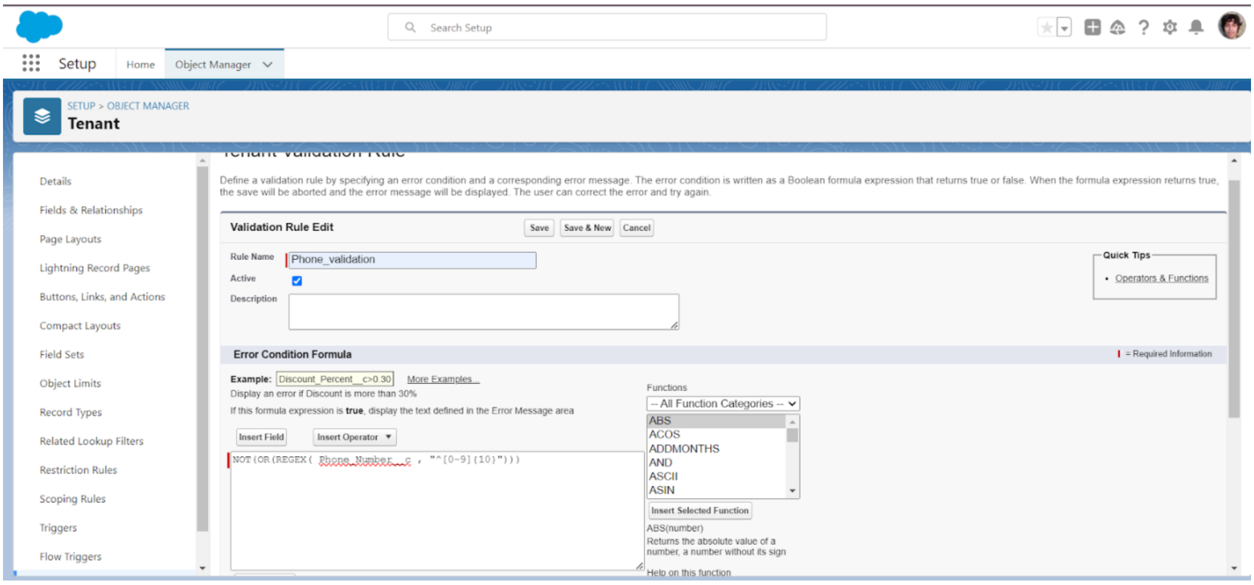
2] Click on the validation rule ==> click New.

3] Enter the Rule name as “Phone Validation “.

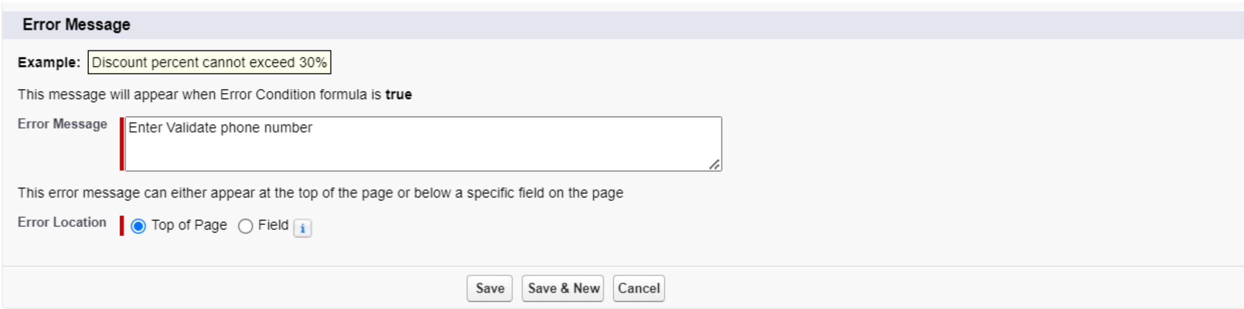


4] Insert the Error Condition Formula as : -

NOT(OR(REGEX(Phone_Number__c , "^[0-9]{10}")))



5] Enter the Error Message as “Enter Valid 10 digit Phone number”, select the Error location as Field and select the field as “Phone Number”, and click Save



B) Create Validation rule for Date of Reg :

Note:- check if the DateofReg is valid and is not a Date in the past.

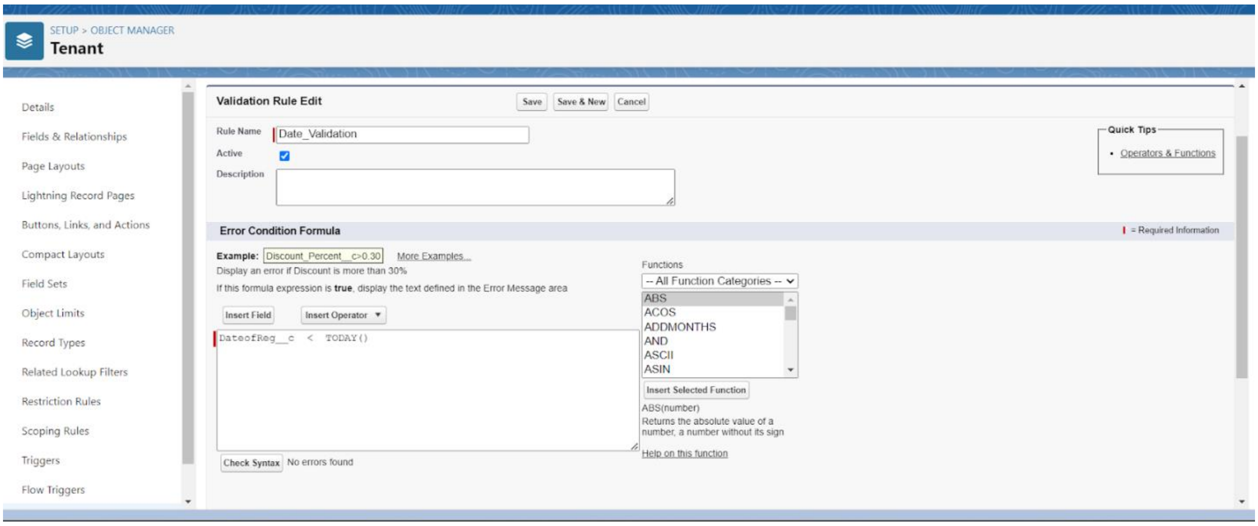
1] Go to setup ==> click on Object Manager ==> type object name(Tenant) in quick find bar==> click on the object.

2] Click on the validation rule==> click New.

3] Enter the Rule name as "Date Validation ".

4] Insert the Error Condition Formula as : -

5] Formula : DateofReg__c < TODAY()



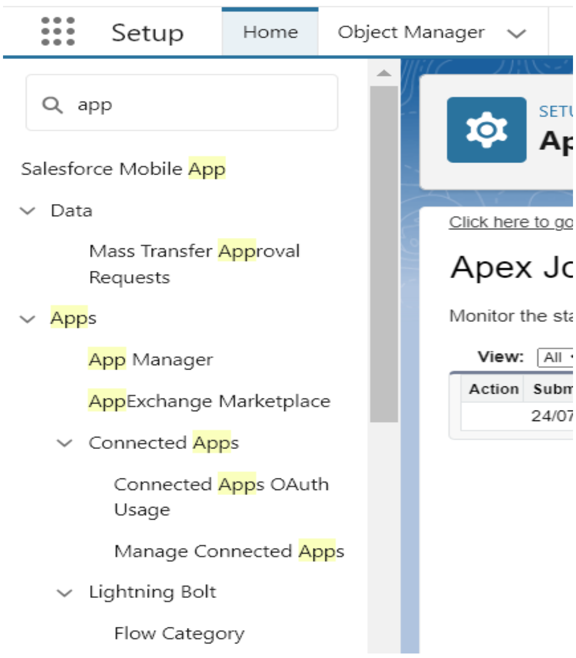
6] Enter the Error Message as "Enter Valid Date", select the Error location as Field and select the field as "DateOfReg", and click Save.

Note: Similarly, create for Lease Tracking and Tenant issues.

<u>Session 4 : Create the Lightning App</u>

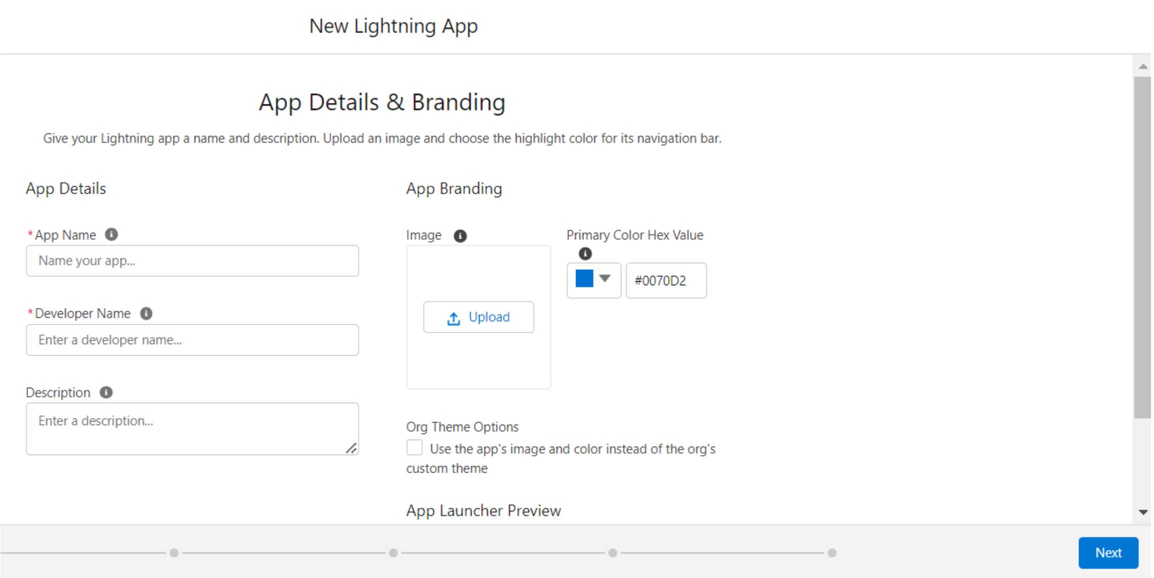
Activity 1: Steps to create custom app in salesforce

1. Go to setup, by clicking the gear icon present in the top right corner.
2. Navigate to the Home bar and in the quick find box, search for App.



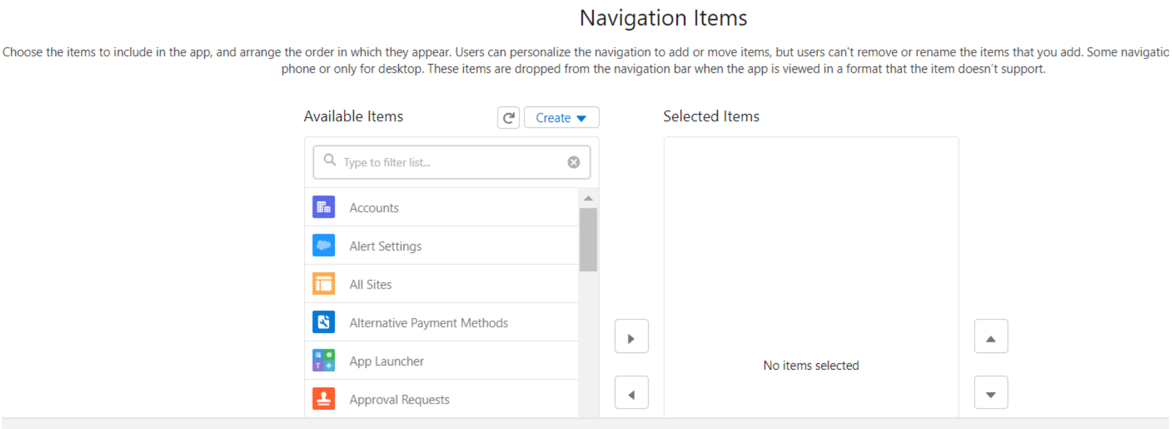
3.Click on APP MANAGER.

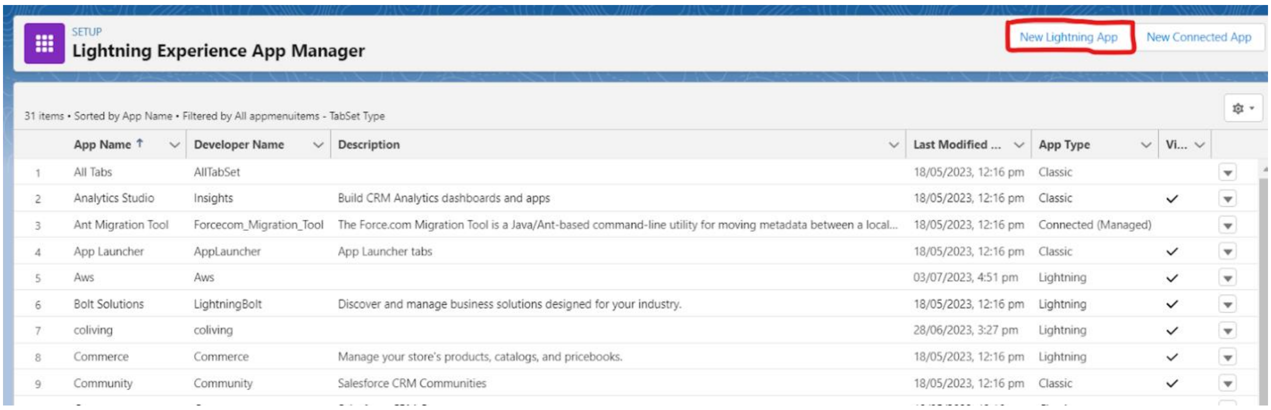
4.You can notice the screen like this. Now click on New Lightning App .You will find like this below.



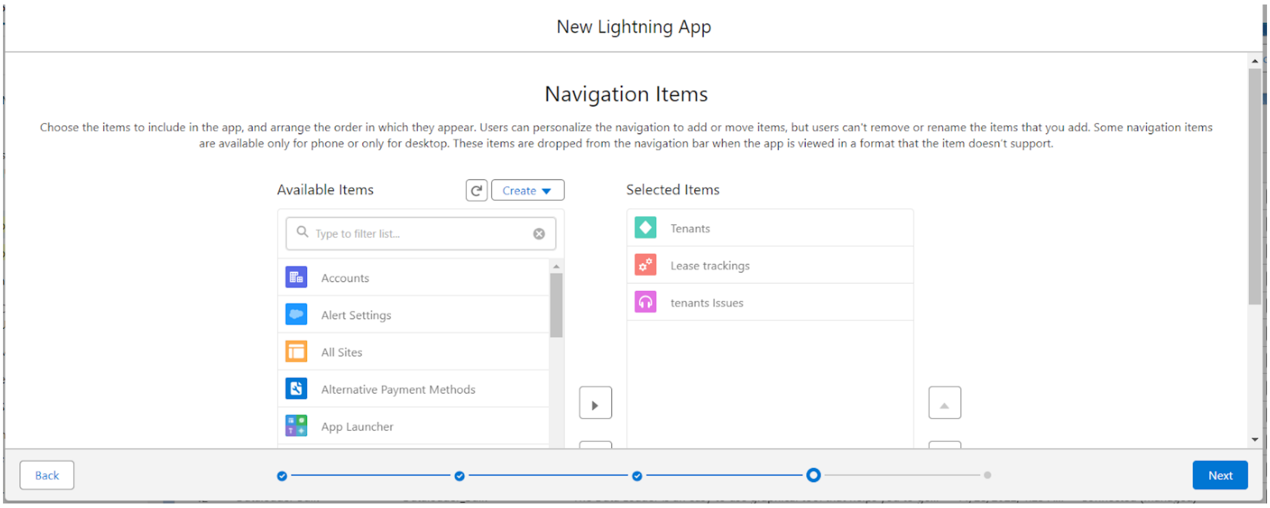
5.Enter the App name(Here we entered ‘SmartMall’),the developer name gets automatically populated. If an image is required, you can browse the image and upload it.

6.Click Next, Next and you can see a Navigation Items window like this:

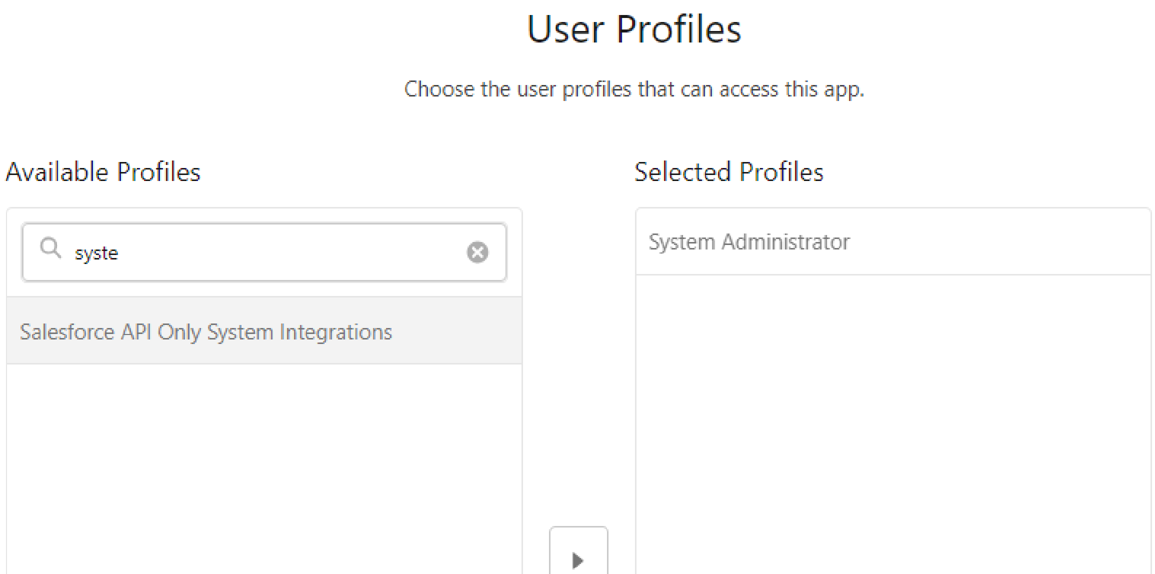




7.In the filter list, enter Tenant , Lease Tracking, Tenant issues and move them in the Selected items from Available items.



1. Click on Next , and you can see User Profiles. This option is used when we want only certain profiles to access them.
2. Enter System Administrator in the filter box and add the system Administrator to the selected profile list.



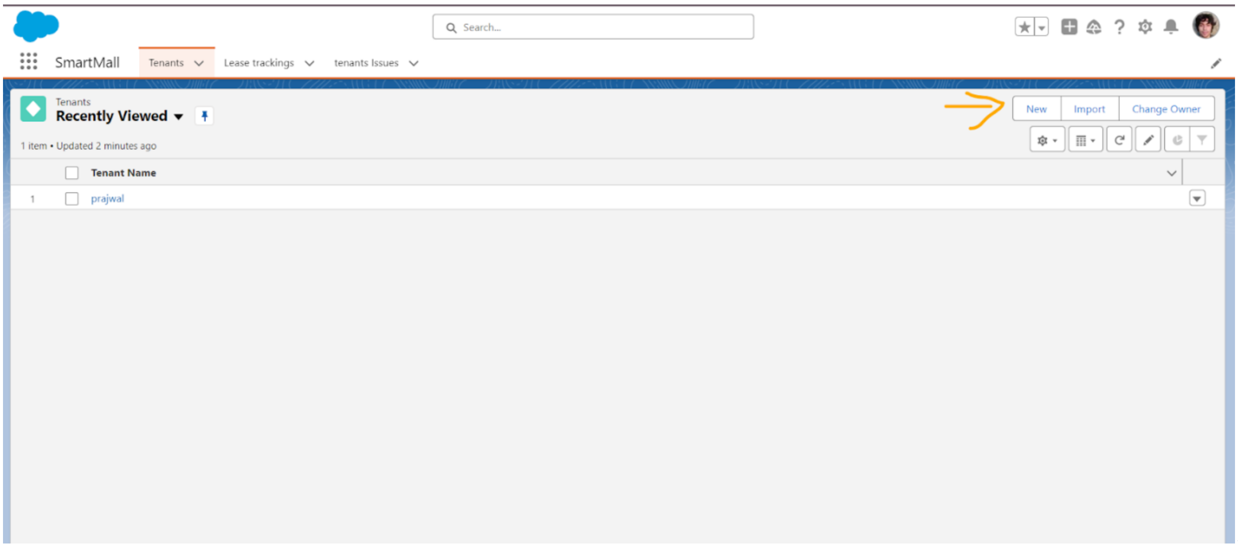
3.Click on Save and Finish.

4.Now navigate to the App launcher and search for SmartMall and you can find the SmartMall app.

Session 5: Record Insertion

Activity 1: Inserting Records in Tenant Object

1) Click on the App Launcher and search Tenant Object then click New in the right corner to create a record

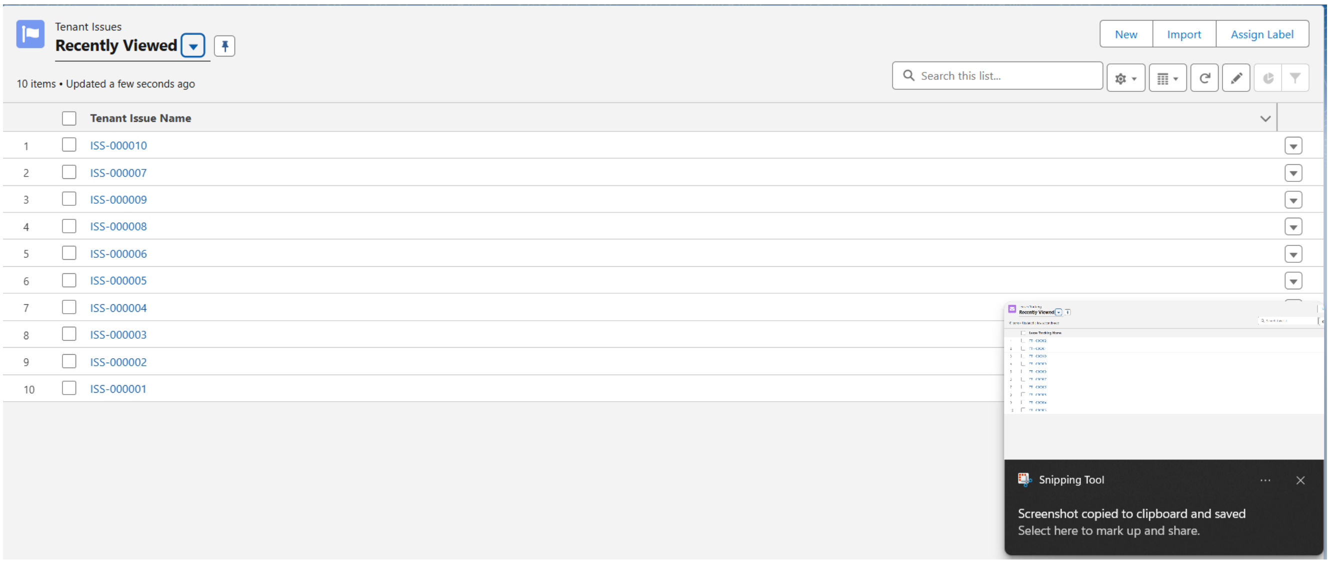


2) Fill every field with valid data, especially the fields on which you have created a validation rule.

3) If you enter Phone Number more or less than 10 digits it will show an error.

4) Similarly, if you enter DateofReg a Past date it will show an error.

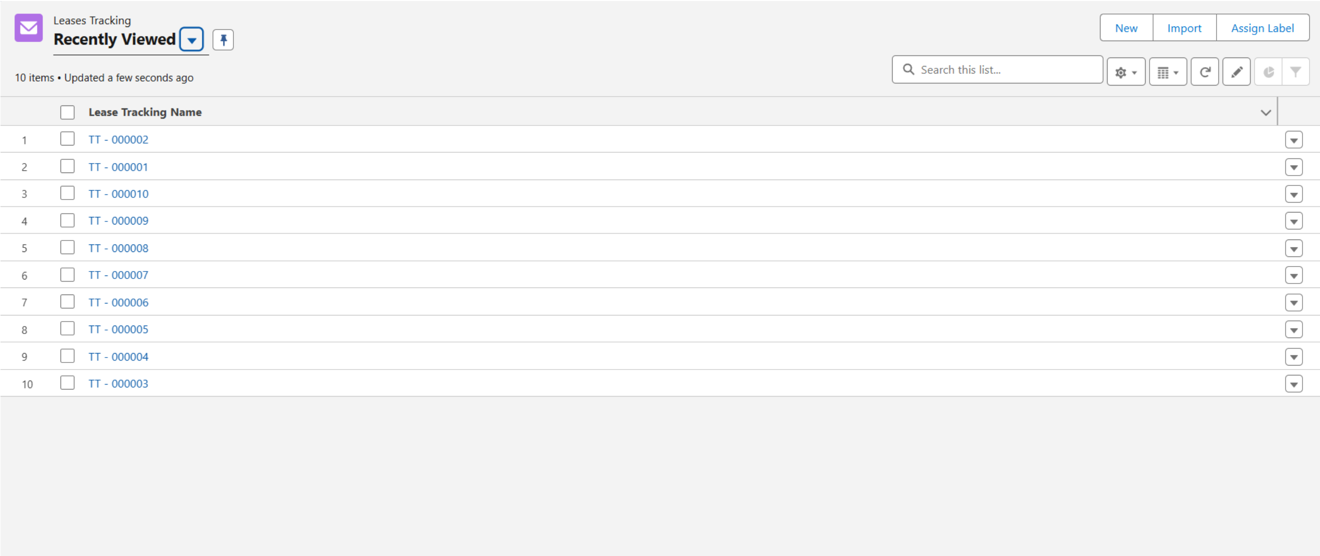
5) After creating a record the page will look like this.



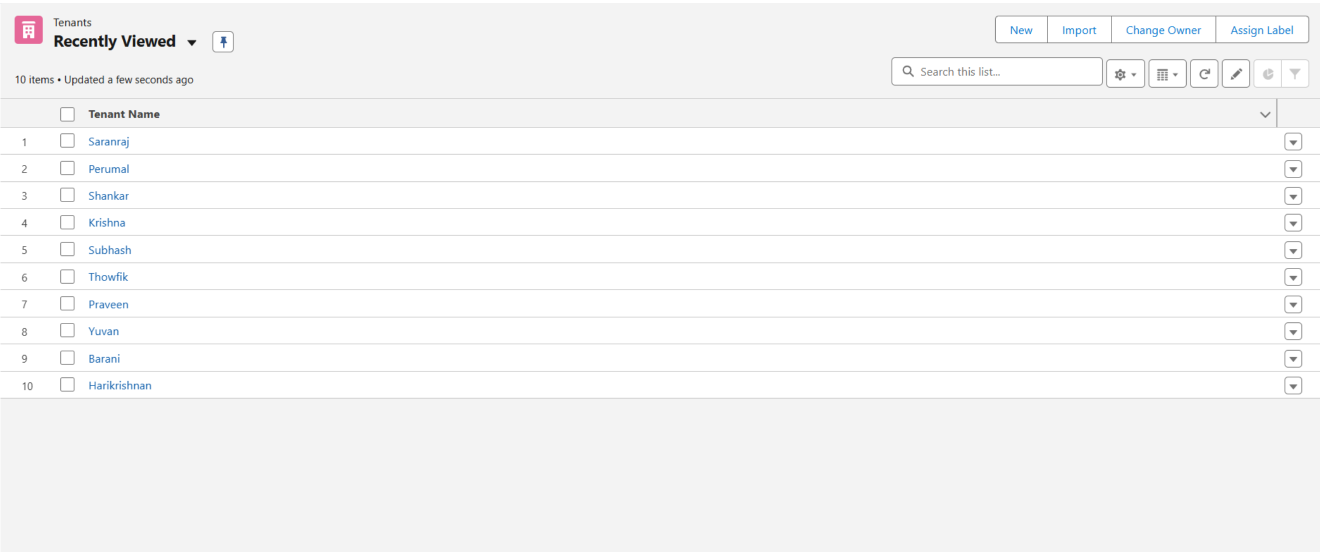
[Note]: Create at least 10 records in the tenant object.

Same procedure follows as:

Activity 2: Inserting Records in Lease Tracking Object



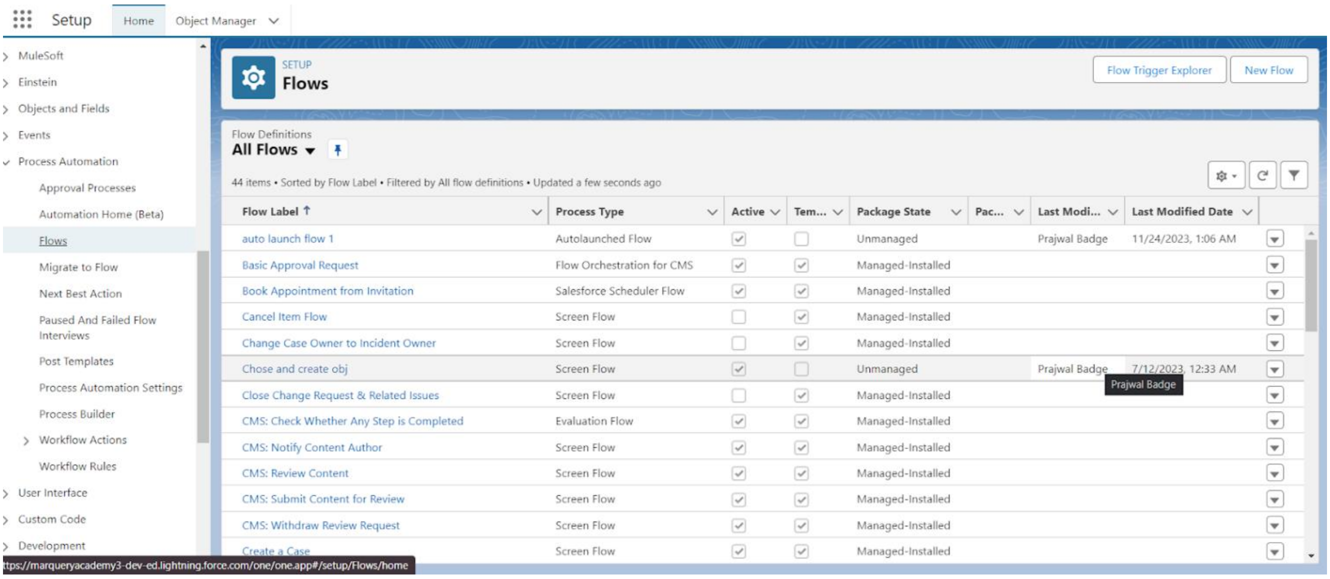
Activity 3 : Inserting Records in Tenants Issues object



<u>Session 6: Create Flows :</u>

Activity 1: Create a Record Triggered flow on tenant Object

1) To create a flow click on setup==> Flow ==> Click on New Flow==> Select Record Triggered Flow

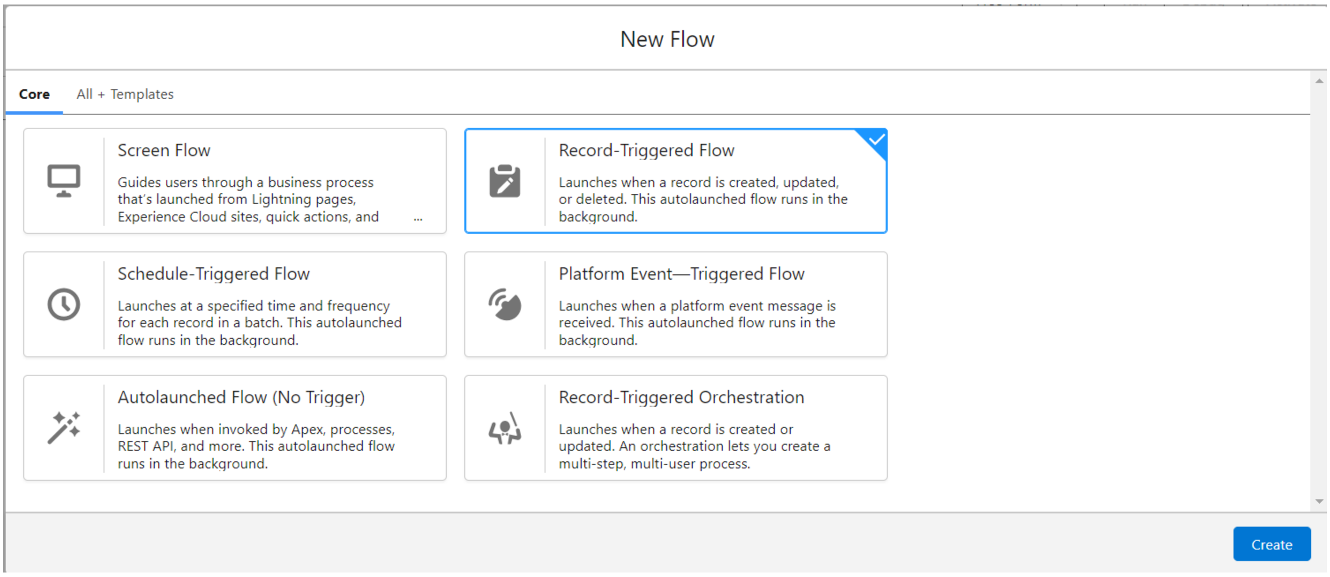


2) In Trigger the flow when Select A record is created

3) Select Condition required All Conditions are met (AND)

4) Select Field - GST_No__c , Operator - Is Null , Value - True.

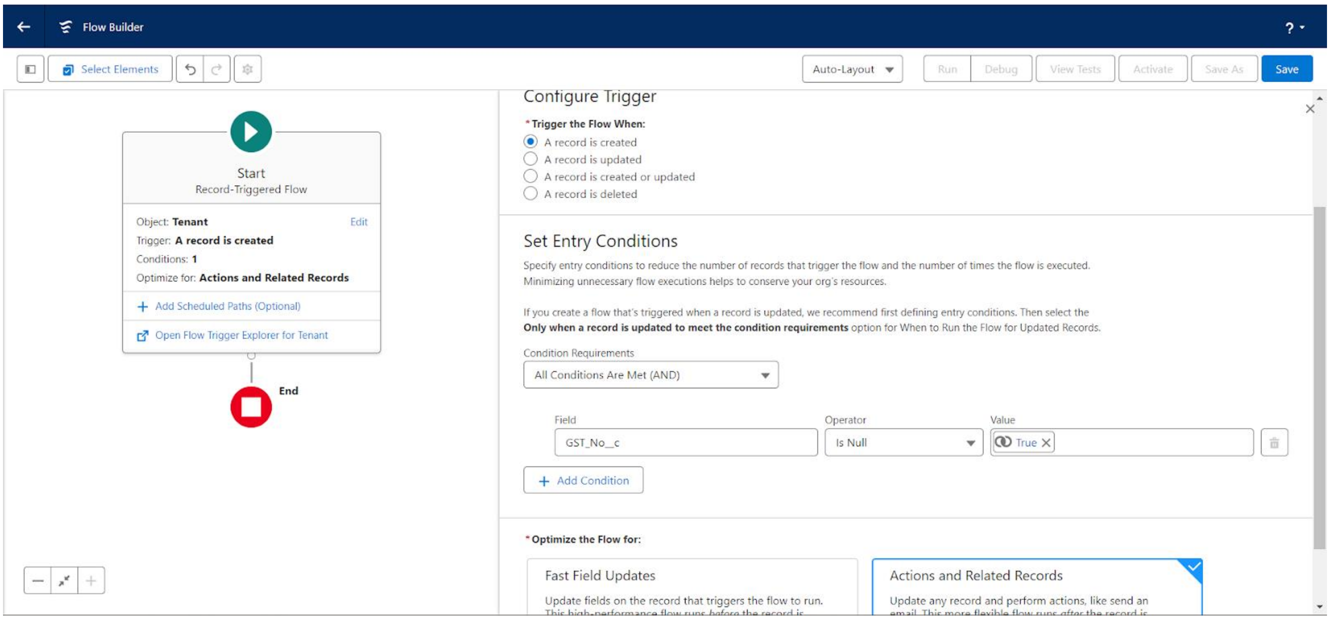
5) Optimize the Flow for: Actions and Related Records



6)Add Element and choose ACTION in the search bar Search Send Email.

7) Label Name - Send email for Gst no, Description - This email is to alert the tenant that he or she has not submitted the GST NO yet.

8) Include Body And Create a Resource - Text Template As below-

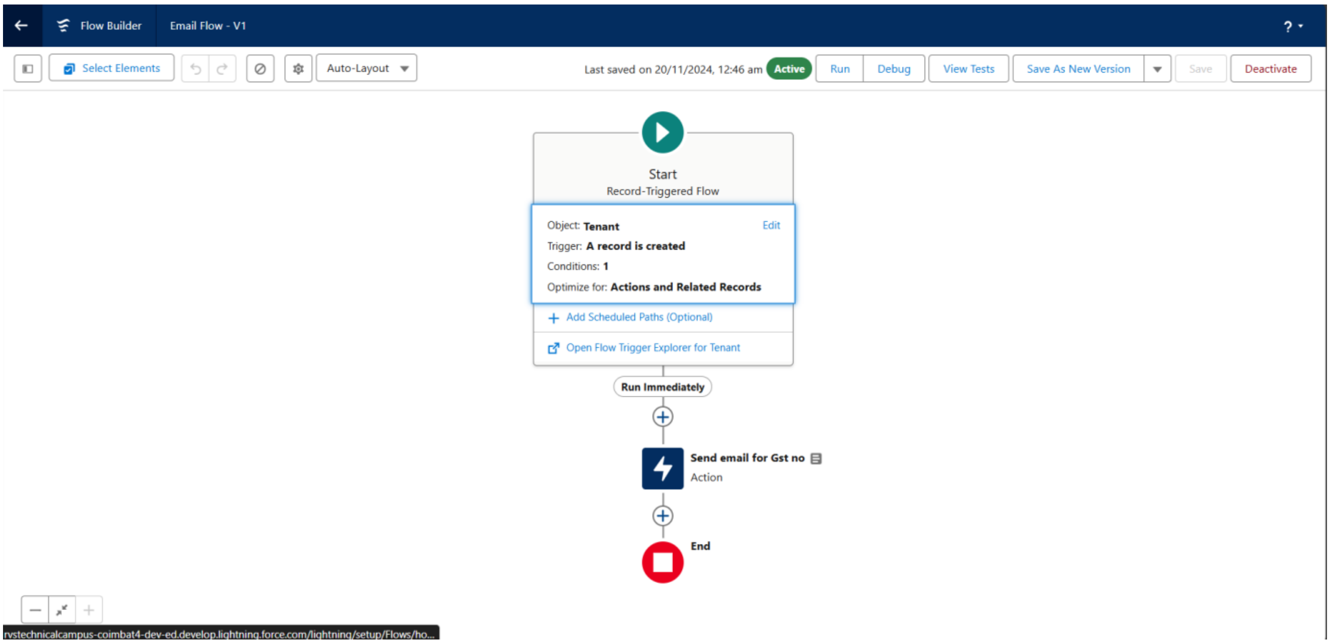


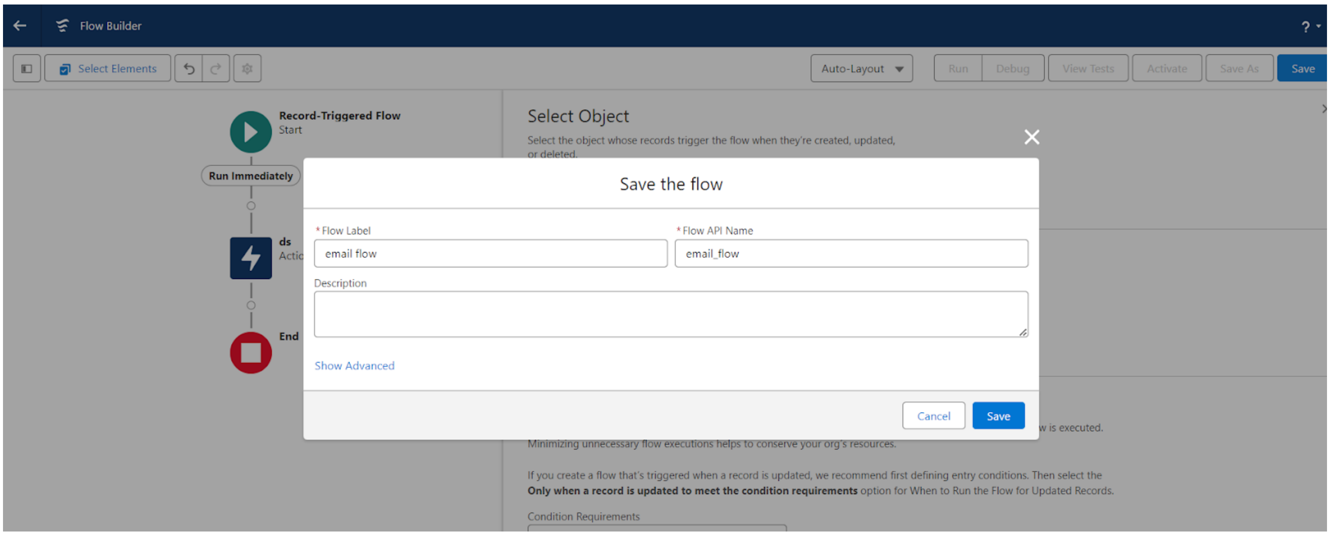
9) Include Recipient ID and from profile select tenant Email id.

10) Include Subject and enter - Regarding your GST Details

11) Click on Save and Name the Flow as Email Flow for tenant and Save.

12) Activate the flow.

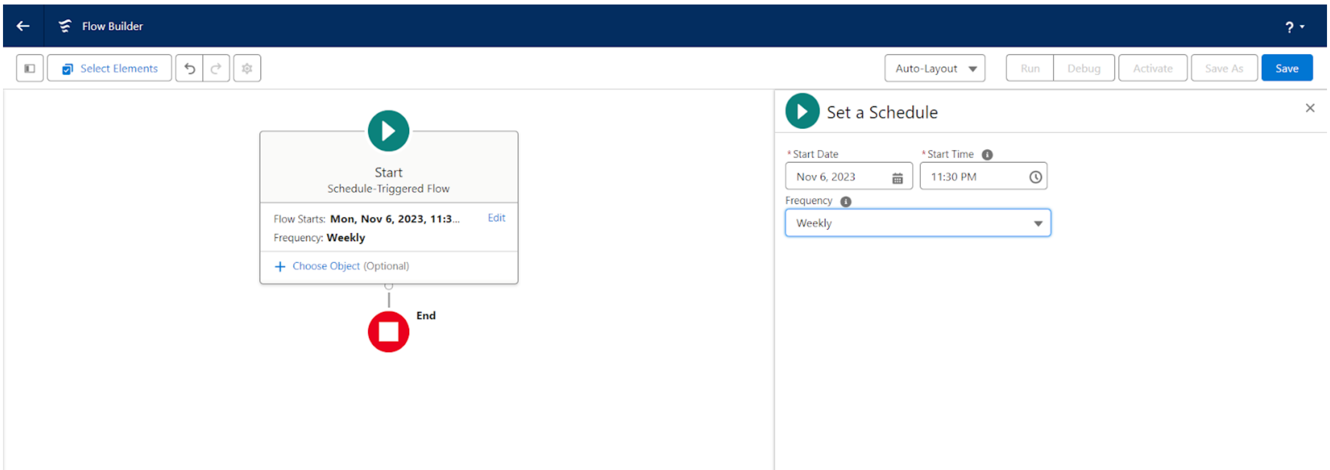




Activity 2: Create a Schedule Flow on Lease Management Object

1) To create a flow click on setup ==> Flow ==> Click on New Flow==> Select Schedule-Triggered Flow

2) Set the Schedule Date - Any Monday , Time - 11.30, Frequency - Weekly.

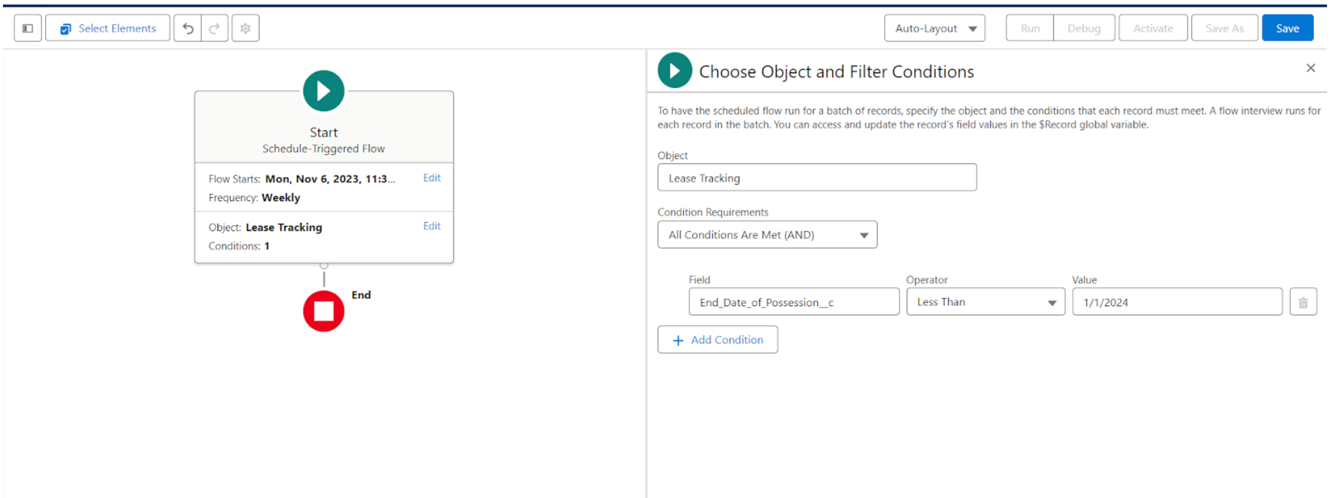


3) Choose Object as Lease Tracking , Condition Requirements - All Conditions are met(AND)

4) Enter field as End_Date_of_Possession__c

5) Operator - Less than

6) Value - 1/1/2024

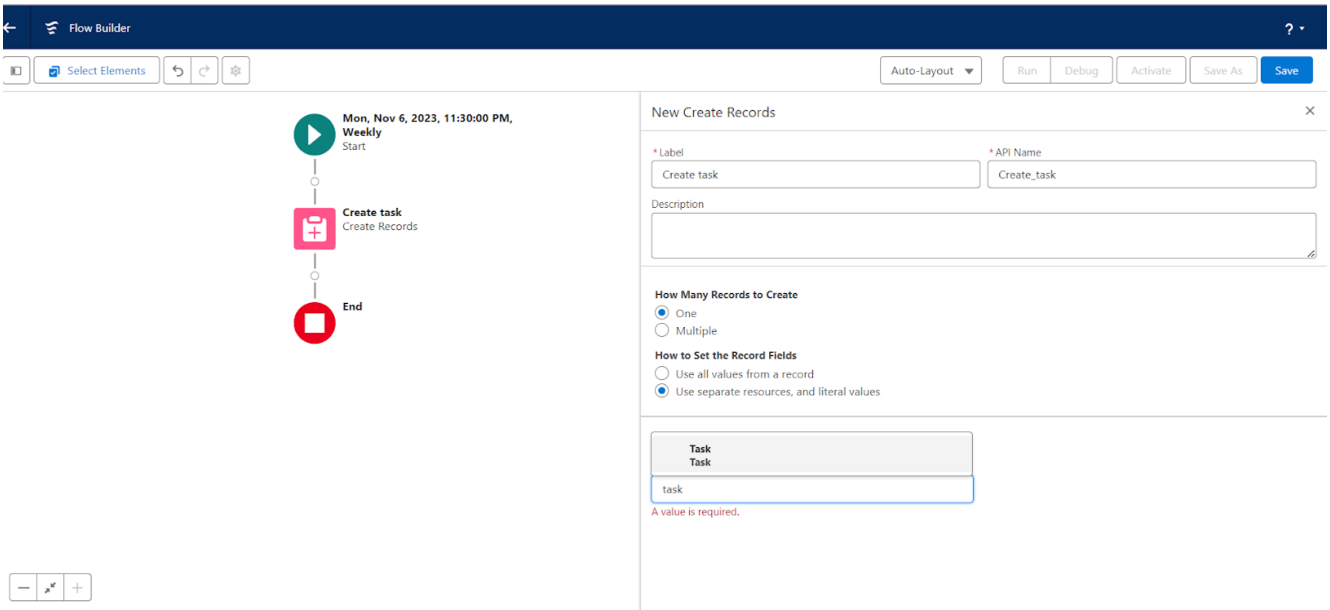


7) Create a Record give labe as Create task

8) How many Records to create - one

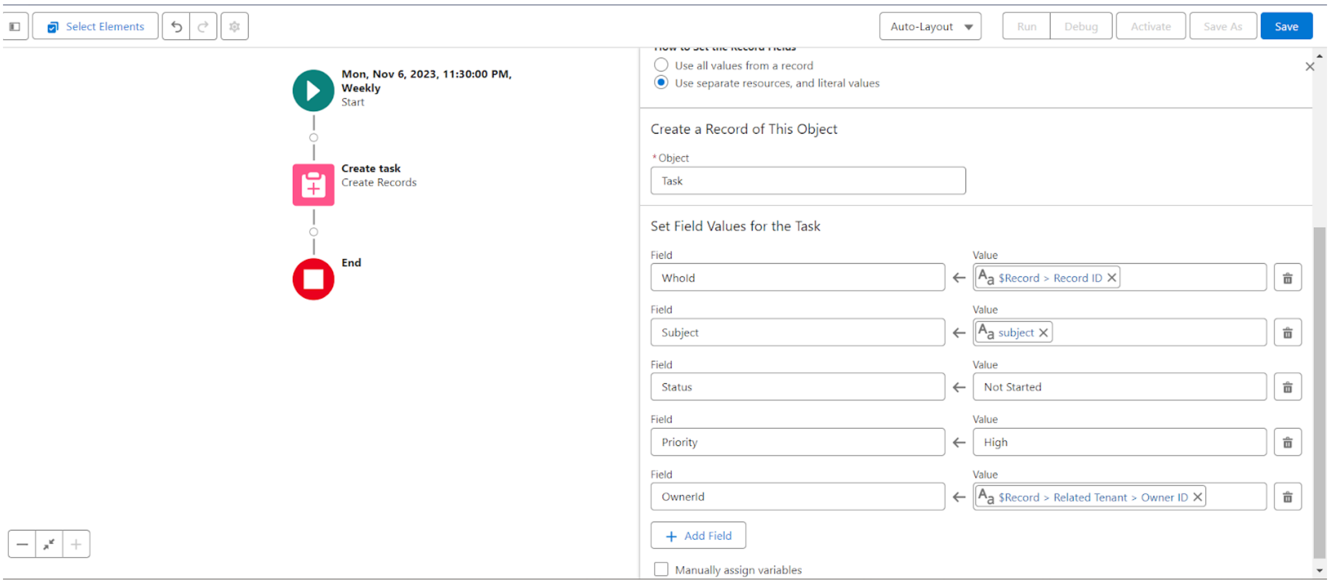
9) How to Set the Record Fields - Use Separate Values

10) object – Task



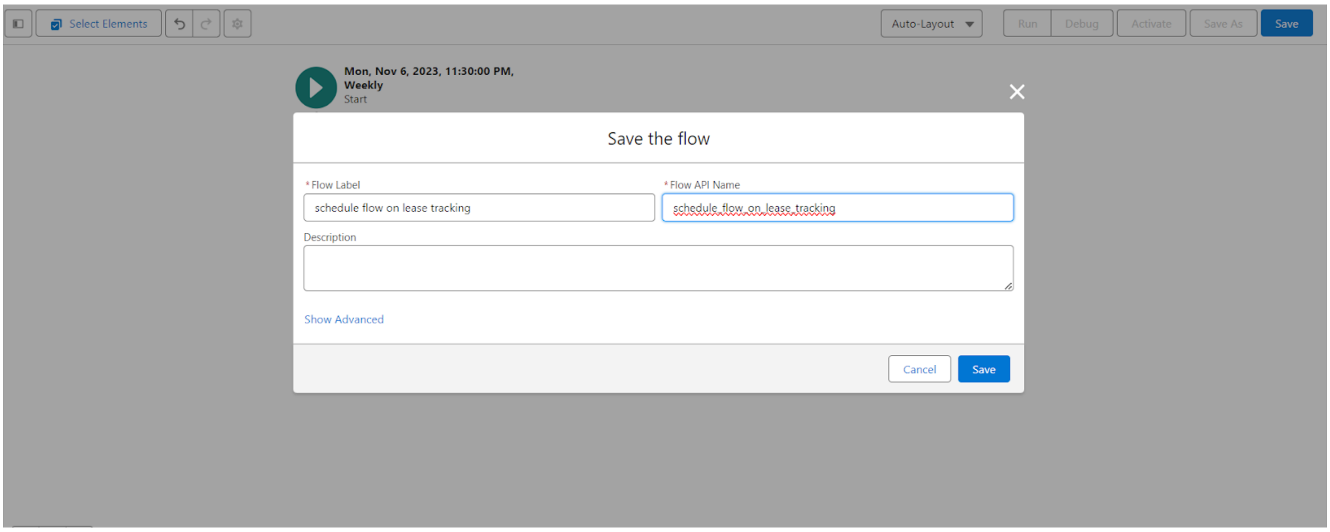
11) Select the field and map them as below : -

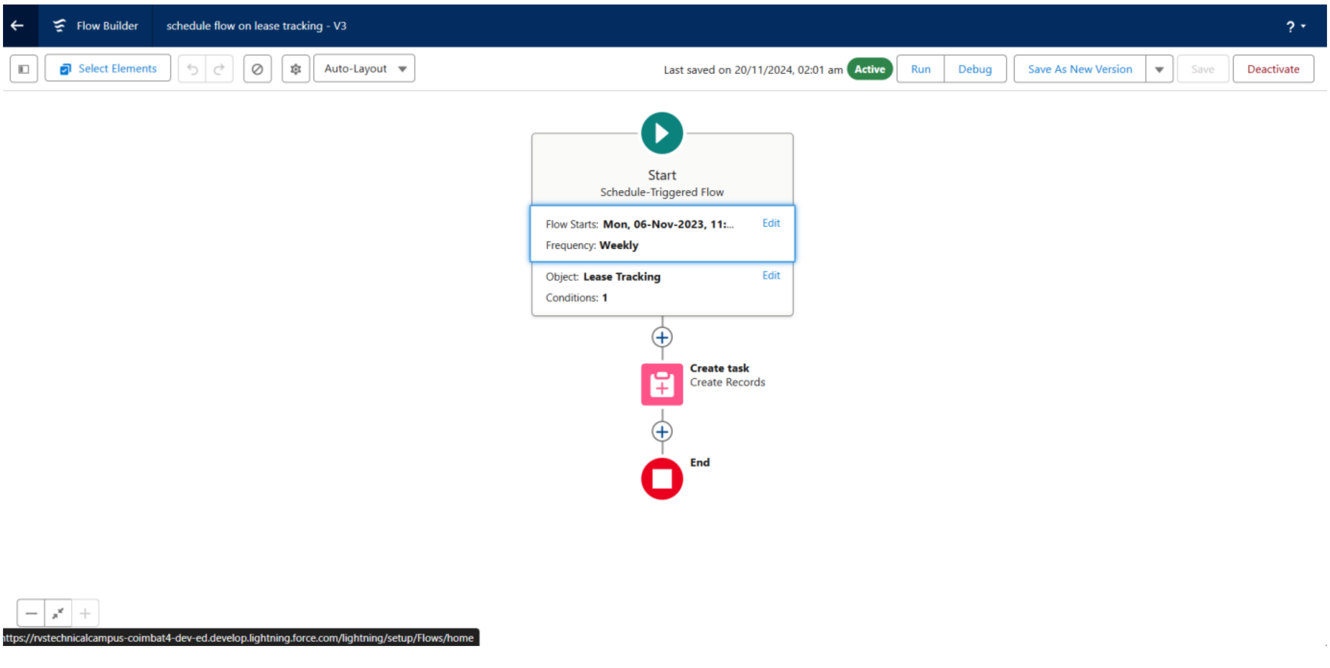
12) Subject of the Task - your possession is going to end soon. Please Contact with Manager to renew your Possession or to End the contract.



13) Save the flow and label it as ‘ schedule flow on lease tracking’

14) Activate the flow.

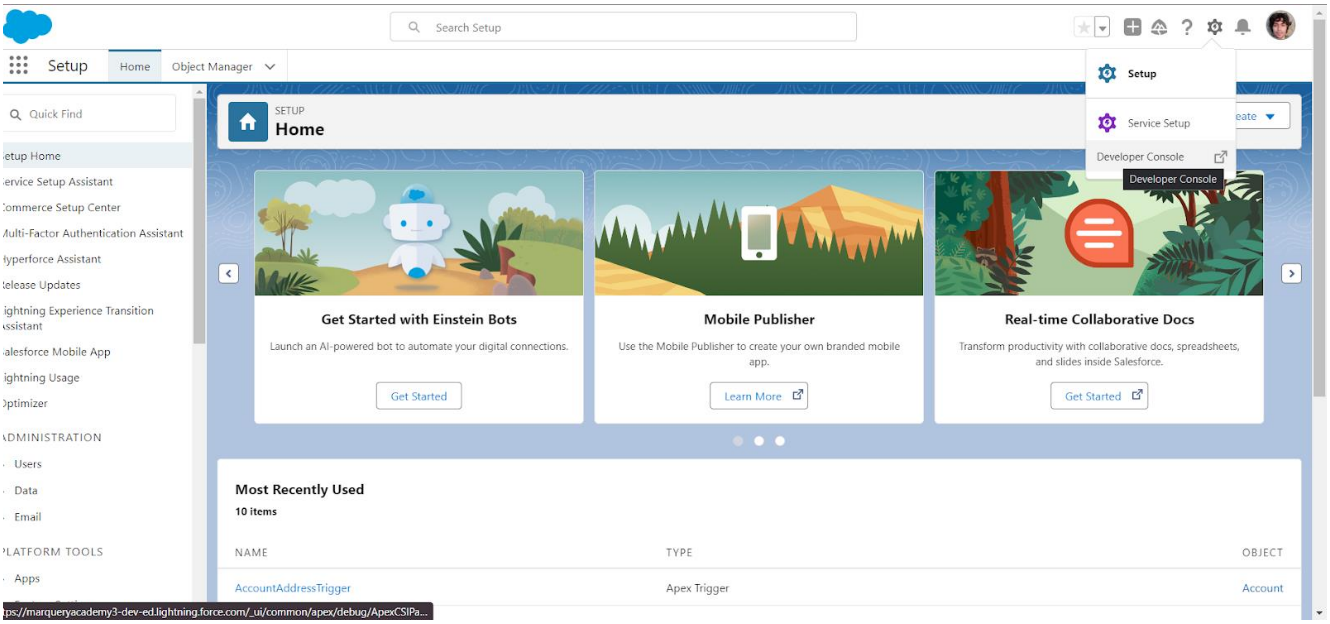




Session 7: Apex Triggers

Activity 1 : Write an Apex Trigger to send an email if the tenant has not paid 50 Percent of Total Rent.

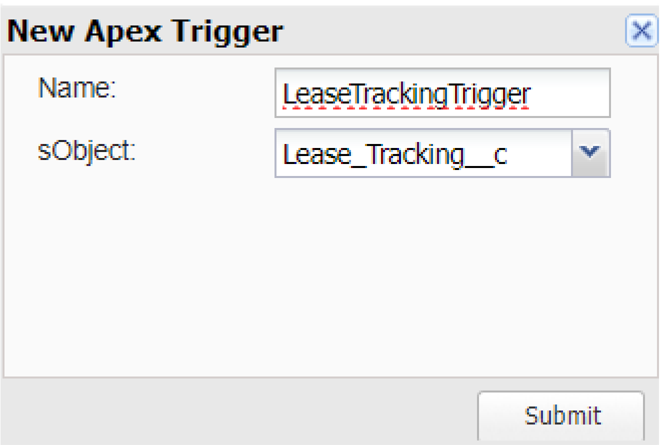
1) Click on the gear icon and click on the developer console.



2) Click on file select New Apex Trigger

3) Name- leasetrackingtrigger, Object —> Lease_Tracking__c

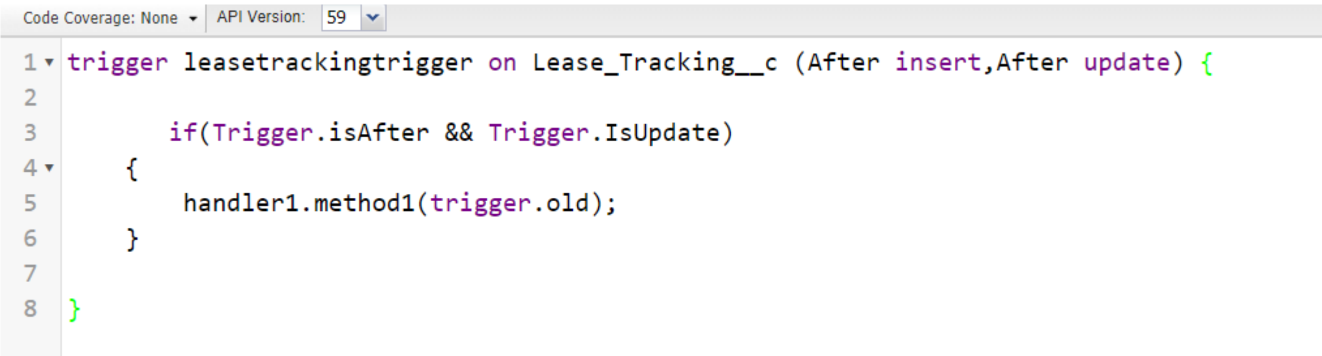
4) Use Event - After insert and After Update and Use Trigger Context Variables as IsAfter and IsUpdate.



Trigger : -

CODE SNIPPET :

```
trigger leasetrackingtrigger on Lease_Tracking__c (After insert,After update) {

       if(Trigger.isAfter && Trigger.IsUpdate)
    {
        LeaseTrackingTriggerHandler.method1(trigger.old);
    }

}
```



Code Coverage: None API Version: 59

```
trigger leasetrackingtrigger on Lease_Tracking__c (After insert,After update) {

        if(Trigger.isAfter && Trigger.IsUpdate)
    {
        handler1.method1(trigger.old);
    }

}
```

Trigger Handler : -

1) Create an apex class and Name it LeaseTrackingTriggerHandler

CODE SNIPPET : -

```
public class LeaseTrackingTriggerHandler {

     public static void method1(List<Lease_Tracking__c> lt1)
   {
       for(Lease_Tracking__c lt2: lt1 )
       {
          if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)
          {
             Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();

              List<String> ToADD = New List<String>{lt2.Email_id__c};

                    M.setToAddresses(ToADD);

                    M.setSubject('Regarding the Pending Rent');

                    M.setPlainTextBody('Hello, This is an Reminder for you to complete your
due rent by the end this month, your due rent thatneeds to be paid is'
+lt2.Amount_to_be_paid__c);
```

```
            List<Messaging.Email> AB = New List<Messaging.Email>{};
            AB.add(M);
            Messaging.sendEmail(AB);

        }
      }
   }
```

Code Coverage: None API Version: 59 Go

```
public class LeaseTrackingTriggerHandler {

    public static void method1(List<Lease_Tracking__c> lt1)
    {
        for(Lease_Tracking__c lt2: lt1 )
        {
            if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)
            {
                Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();

                 List<String> ToADD = New List<String>{lt2.Emai_id__c};

                      M.setToAddresses(ToADD);
                      M.setSubject('Regarding the Pending Rent');
                      M.setPlainTextBody('Hello, This is an Reminder for you to complete your due rent by the end this month, your due rent thatneeds to be paid i



                    List<Messaging.Email> AB = New List<Messaging.Email>{};
                    AB.add(M);
                    Messaging.sendEmail(AB);



            }
        }
    }
```

Activity 2: Write an Apex Trigger on Tenant Object to Show error if the pan card is invalid.

1.Click on the gear icon and click on the developer console.

2. Click on file select New Apex Trigger

3.Name- TenantTrigger, Object - Tenant

4.Use Events - Before insert and Trigger context Variable – IsBefore

Trigger : -

CODE SNIPPET : -

```
trigger TenantTrigger on Tenant__c (before insert) {

     if(Trigger.isBefore)
   {
       TenantTriggerhandler.method1(Trigger.New);
   }
}
```

Code Coverage: None API Version: 59

```
trigger TenantTrigger on Tenant__c (before insert) {

       if(Trigger.isBefore)
     {
        handler2.method1(Trigger.New);
     }

}
```

Trigger Handler : -

1) Create an apex class and Name it TenantTriggerhandler

CODE SNIPPET: -

```
public class TenantTriggerhandler {

      public static void method1(List<Tenant__c> te)
    {

       for(Tenant__c tenant : te)
       {
           if(tenant.Pan_Card_no__c.length() > 10)
           {
              tenant.addError('This Pan Card number is invalid, Please Enter Valid Pan Card number');
           }
       }
    }

}
```

```
public class TenantTriggerhandler {

      public static void method1(List<Tenant__c> te)
    {

       for(Tenant__c tenant : te)
       {
           if(tenant.Pan_Card_no__c.length() > 10)
           {
              tenant.addError('This Pan Card number is invalid, Please Enter Valid Pan Card number');
           }
       }
    }

}
```

Session 8: Asynchronous Apex

Activity 1: Schedule Apex

Delete the Tenant Records Monthly whose Status Of Possession is closed.

1) Create a class with name tenantschedulable

2) Give extension Schedulable to the class.

3) Open the Anonymous Window.

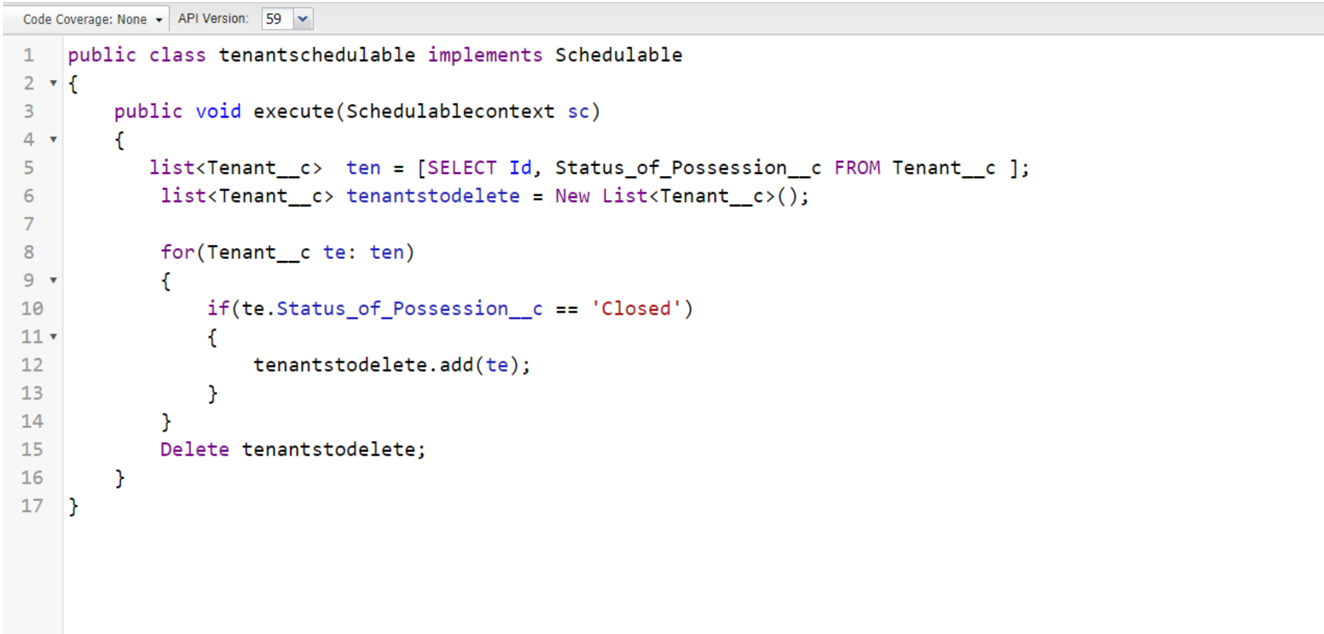
4) Schedule the class-

tenantschedulable a = new tenantschedulable();
string cron = '0 0 0 1 * ? * ';
system.schedule('Delete the records monthly', cron, a);

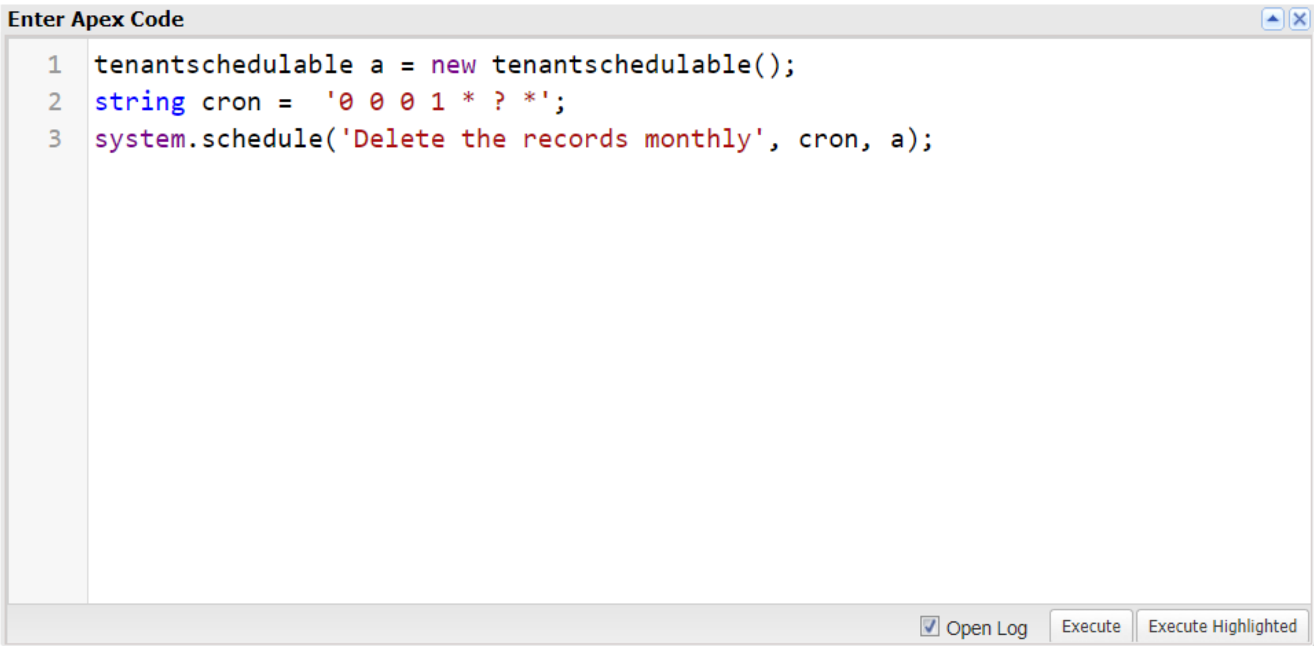
CODE SNIPPET : -

```
public class tenantschedulable implements Schedulable
{
public void execute(Schedulablecontext sc)
   {
      list<Tenant__c>  ten = [SELECT Id, Status_of_Possession__c FROM Tenant__c ];
       list<Tenant__c> tenantstodelete = New List<Tenant__c>();


       for(Tenant__c te: ten)
       {
          if(te.Status_of_Possession__c == 'Closed')
          {
```

```
        tenantstodelete.add(te);
          }
        }       Delete tenantstodelete;
    }
}
```




```
public class tenantschedulable implements Schedulable
{
    public void execute(Schedulablecontext sc)
    {
       list<Tenant__c>  ten = [SELECT Id, Status_of_Possession__c FROM Tenant__c ];
        list<Tenant__c> tenantstodelete = New List<Tenant__c>();

        for(Tenant__c te: ten)
        {
            if(te.Status_of_Possession__c == 'Closed')
            {
                tenantstodelete.add(te);
            }
        }
        Delete tenantstodelete;
    }
}
```



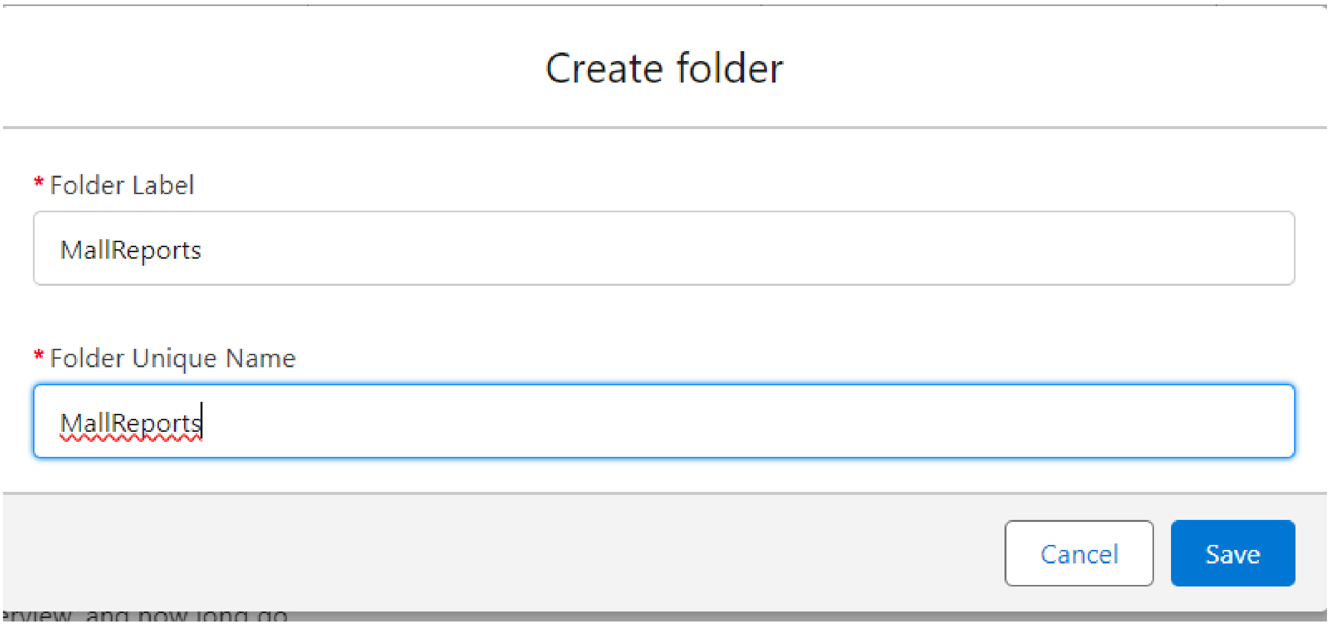

```
tenantschedulable a = new tenantschedulable();
string cron =  '0 0 0 1 * ? *';
system.schedule('Delete the records monthly', cron, a);
```

Session 9: Create Reports And Dashboards

Activity 1: Create a Report of lease Management Records

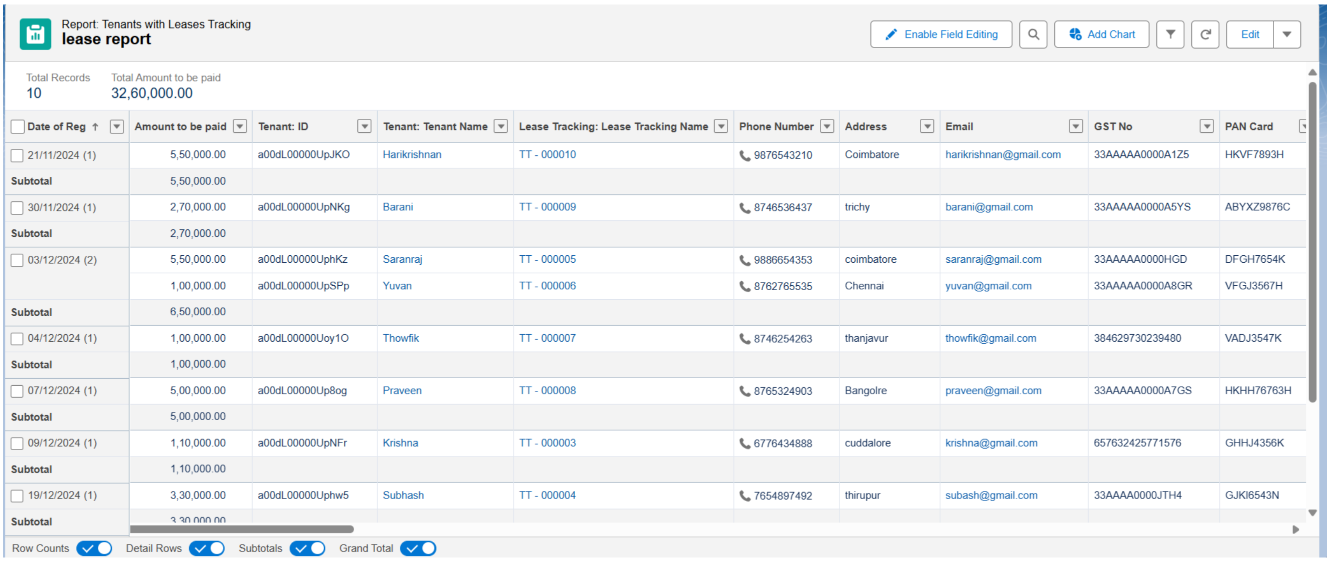
The Manager needs a report which shows the tenant and their joining date and their Remaining payment details and also group this by date of Registration, and make a bucket list of remaining amount as greater than 1000000 as red, less than 1000 and greater than 500000 as blue and less than equal to 500000 as yellow.

1) Create a new Folder and name it as MallReports



2) Click On new report ==> Select object Activities with LeaseTracking ==> Click on start report

3) Click on the Amount to be paid column and click on bucket this list and name it as Remaining amount



Report: Tenants with Leases Tracking
lease report

Enable Field Editing | Add Chart | Edit

Total Records 10 | Total Amount to be paid 32,60,000.00

Date of Reg ↑	Amount to be paid	Tenant: ID	Tenant: Tenant Name	Lease Tracking: Lease Tracking Name	Phone Number	Address	Email	GST No	PAN Card
21/11/2024 (1)	5,50,000.00	a00dL00000UpJKO	Harikrishnan	TT - 000010	9876543210	Coimbatore	harikrishnan@gmail.com	33AAAAA0000A1Z5	HKVF7893H
Subtotal	5,50,000.00								
30/11/2024 (1)	2,70,000.00	a00dL00000UpNKg	Barani	TT - 000009	8746536437	trichy	barani@gmail.com	33AAAAA0000A5YS	ABYXZ9876C
Subtotal	2,70,000.00								
03/12/2024 (2)	5,50,000.00	a00dL00000UphKz	Saranraj	TT - 000005	9886654353	coimbatore	saranraj@gmail.com	33AAAAA0000HGD	DFGH7654K
	1,00,000.00	a00dL00000UpSPp	Yuvan	TT - 000006	8762765535	Chennai	yuvan@gmail.com	33AAAAA0000A8GR	VFGJ3567H
Subtotal	6,50,000.00								
04/12/2024 (1)	1,00,000.00	a00dL00000Uoy1O	Thowfik	TT - 000007	8746254263	thanjavur	thowfik@gmail.com	384629730239480	VADJ3547K
Subtotal	1,00,000.00								
07/12/2024 (1)	5,00,000.00	a00dL00000Up8og	Praveen	TT - 000008	8765324903	Bangolre	praveen@gmail.com	33AAAAA0000A7GS	HKHH76763H
Subtotal	5,00,000.00								
09/12/2024 (1)	1,10,000.00	a00dL00000UpNFr	Krishna	TT - 000003	6776434888	cuddalore	krishna@gmail.com	657632425771576	GHHJ4356K
Subtotal	1,10,000.00								
19/12/2024 (1)	3,30,000.00	a00dL00000Uphw5	Subhash	TT - 000004	7654897492	thirupur	subash@gmail.com	33AAAA0000JTH4	GJKI6543N
Subtotal	3,30,000.00								

Row Counts | Detail Rows | Subtotals | Grand Total

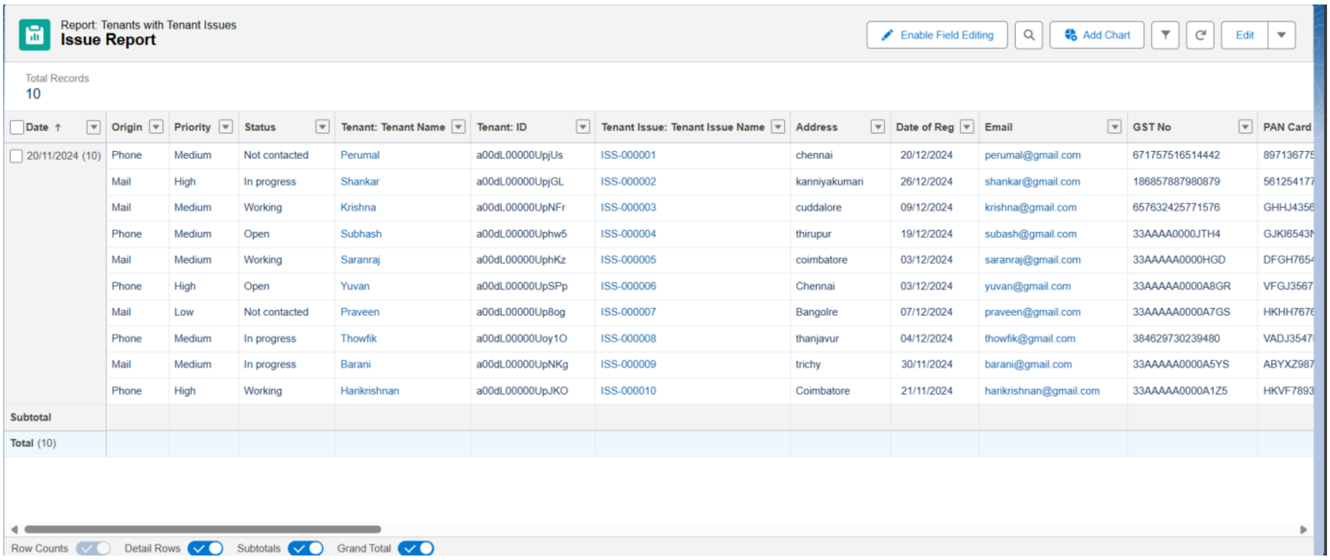
4) Save the report named as lease report and Save it in MallReports.

Activity 2: Create a Report on Tenant issue Records

Now the manager is asking for a report on issues which has not been contacted or Open yet and has high priority which are directly encountered by Phone and Mail and the date of issue is from last 7 days

1) Click On new report ==> Select object Activities with Tenant issue==> Click on start report

2) Click on save, enter Name - Issue Report



Report: Tenants with Tenant Issues
Issue Report

Enable Field Editing | Add Chart | Edit

Total Records 10

Date ↑	Origin	Priority	Status	Tenant: Tenant Name	Tenant: ID	Tenant Issue: Tenant Issue Name	Address	Date of Reg	Email	GST No	PAN Card
20/11/2024 (10)	Phone	Medium	Not contacted	Perumal	a00dL00000UpjUs	ISS-000001	chennai	20/12/2024	perumal@gmail.com	671757516514442	89713677
	Mail	High	In progress	Shankar	a00dL00000UpjGL	ISS-000002	kanniyakumari	26/12/2024	shankar@gmail.com	186857887980879	56125417
	Mail	Medium	Working	Krishna	a00dL00000UpNFr	ISS-000003	cuddalore	09/12/2024	krishna@gmail.com	657632425771576	GHHJ4356
	Phone	Medium	Open	Subhash	a00dL00000Uphw5	ISS-000004	thirupur	19/12/2024	subash@gmail.com	33AAAA0000JTH4	GJKI6543
	Mail	Medium	Working	Saranraj	a00dL00000UphKz	ISS-000005	coimbatore	03/12/2024	saranraj@gmail.com	33AAAAA0000HGD	DFGH765
	Phone	High	Open	Yuvan	a00dL00000UpSPp	ISS-000006	Chennai	03/12/2024	yuvan@gmail.com	33AAAAA0000A8GR	VFGJ3567
	Mail	Low	Not contacted	Praveen	a00dL00000Up8og	ISS-000007	Bangolre	07/12/2024	praveen@gmail.com	33AAAAA0000A7GS	HKHH767
	Phone	Medium	In progress	Thowfik	a00dL00000Uoy1O	ISS-000008	thanjavur	04/12/2024	thowfik@gmail.com	384629730239480	VADJ3547
	Mail	Medium	In progress	Barani	a00dL00000UpNKg	ISS-000009	trichy	30/11/2024	barani@gmail.com	33AAAAA0000A5YS	ABYXZ987
	Phone	High	Working	Harikrishnan	a00dL00000UpJKO	ISS-000010	Coimbatore	21/11/2024	harikrishnan@gmail.com	33AAAAA0000A1Z5	HKVF7893
Subtotal											
Total (10)											

Row Counts | Detail Rows | Subtotals | Grand Total

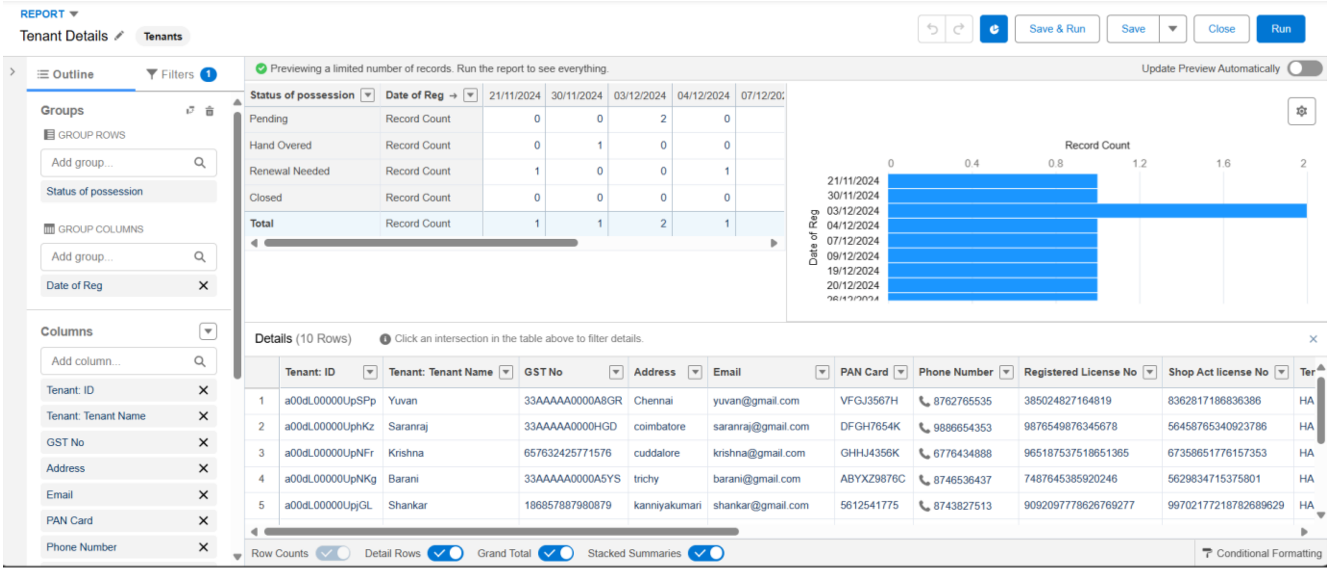
3) Choose the folder Mall Reports and save.

Activity 3: Create a Report on Tenant Records

Now, The Manager wants a Report which shows all the pending possessions and also shows the tenant's Pan Card no and GST NO and group date of reg by column and row by Status of Possession.

1) Click On new report ==> Select object Tenants ==> Click on start report

2) Choose the folder Mall Reports.

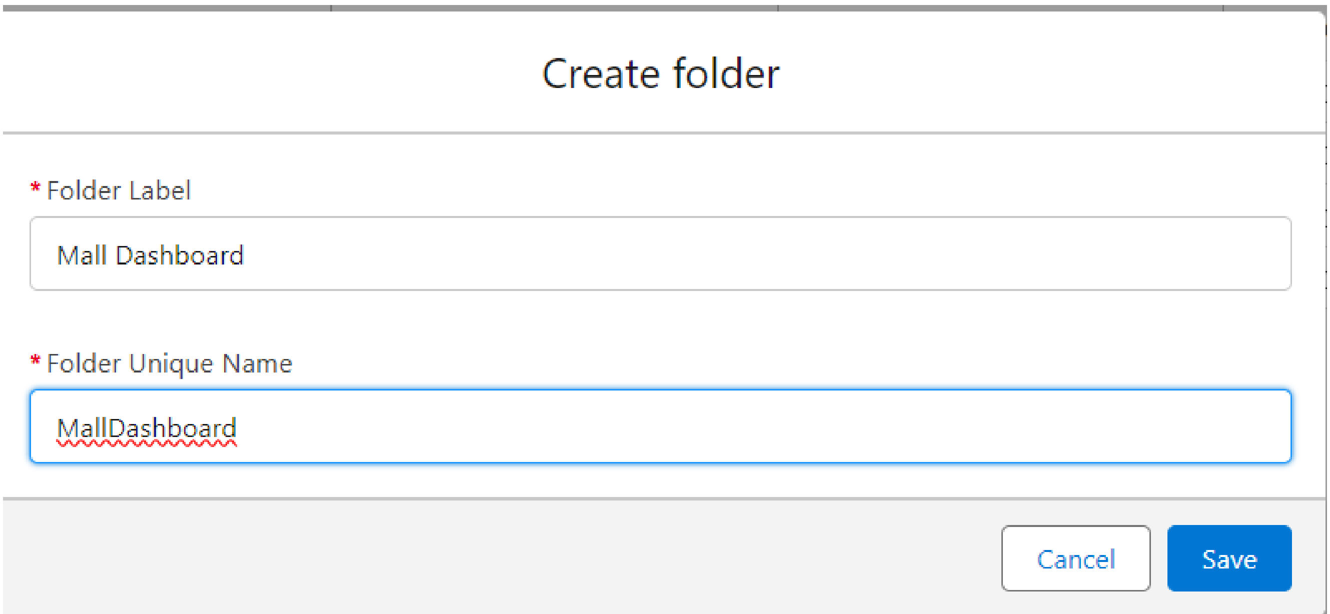


3) Save the report and Name it as Tenant Details.

Activity 4: Create a Dashboard

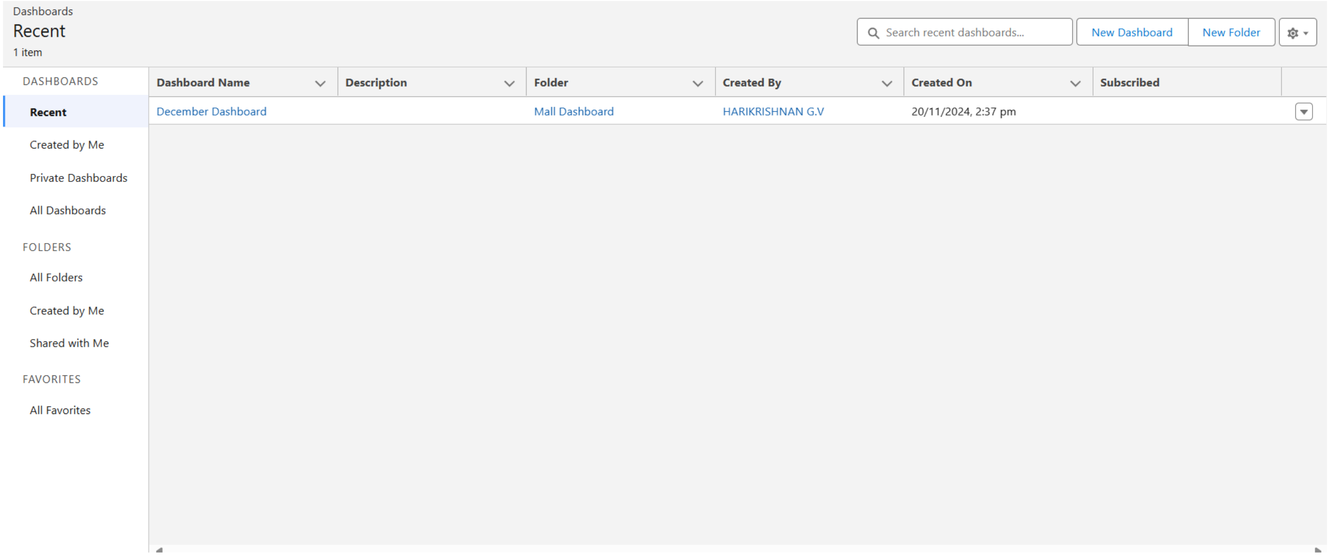
Very Good, You have created multiple reports but now for better convenience the owner wants a Dashboard which shows the data of these reports, So Create a Dashboard and follow the instruction below -

1) To Create a Dashboard first create a folder to store the dashboard and save it as Mall Dashboard.



2) Now click on New Dashboard

3) Enter Name - December Dashboard, select the Mall Dashboard folder and click on create.

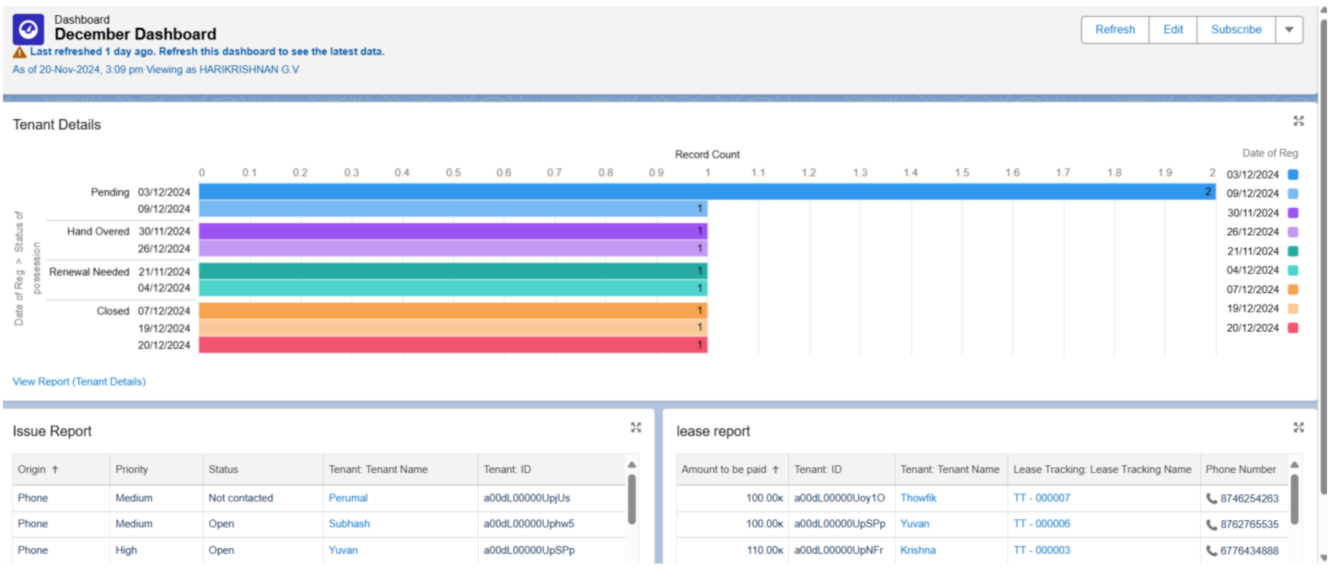


4) Click on + Component and Select Tenant Details, Display as Horizontal Bar Chart, Dimensions - Height * width = 10*12.

5) Click on + Component and Select Issue Report, Display as Lightning Table, Dimensions - Height * width = 8*6.

6) Click on + Component and Select lease report, Display as Lightning Table, Dimensions - Height * width = 8*6

7) Click on save and Done.



Conclusion

The CRM application implemented for managing mall-related tenant details, tenant issues, and lease tracking has achieved significant operational improvements. By automating key processes, streamlining communication, and providing actionable insights through analytics, the system has enhanced the efficiency of mall management and fostered a more positive and responsive environment for tenants. Through improved lease tracking, issue resolution, and tenant management, the application has positioned the mall for continued success in maintaining tenant satisfaction and operational excellence.

******************************END**********************************