**A CRM Application to Manage the Mall**

Sam Daniel Raj A(89BFCB8D6EFF0B682634AF208CB85E00)

Siva Guru K(3CD83A96F9B33C3877C4785FEC7344E2)

Mahesh Kumar M(2EE7B1D7B296B20815A47A8EFDCC2538)

Pranesh S(02BC6B3AE85C1E5AEA343D290F7BE373)

**Salesoforce**

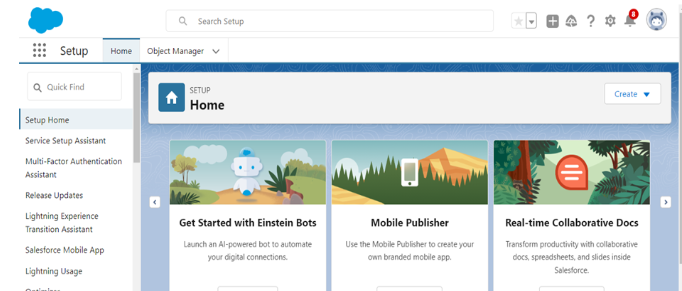
Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don’t know where you should start on your learning journey? If you’ve answered yes to any of these questions, then you’re in the right place. This module is for you. Welcome to Salesforce! Salesforce is game-changing technology,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

**Why are we using Salesforce Platform for this project**

We're using Salesforce for the Mall Management App because it lets us build the app quickly without a lot of complicated coding. It's flexible, so we can easily customize it to fit the specific needs of managing a mall. Salesforce keeps everything organized in one place, making it easy to handle tenant info, leases, and maintenance issues. It's also secure, ensuring that sensitive data is protected. With automation features, tasks can be done automatically, saving time. The user-friendly interface makes it easy for everyone to use, and the analytics tools help us make smart decisions based on data. Plus, Salesforce has a helpful community for support and additional resources, making it a great overall choice for our project.

**Creating Developer Account**

**https://developer.salesforce.com/signup**



**Create Custom Objects**

To store the data as per business requirement.

**Create Tenant Object**

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

1.Enter the label name:  Tenant

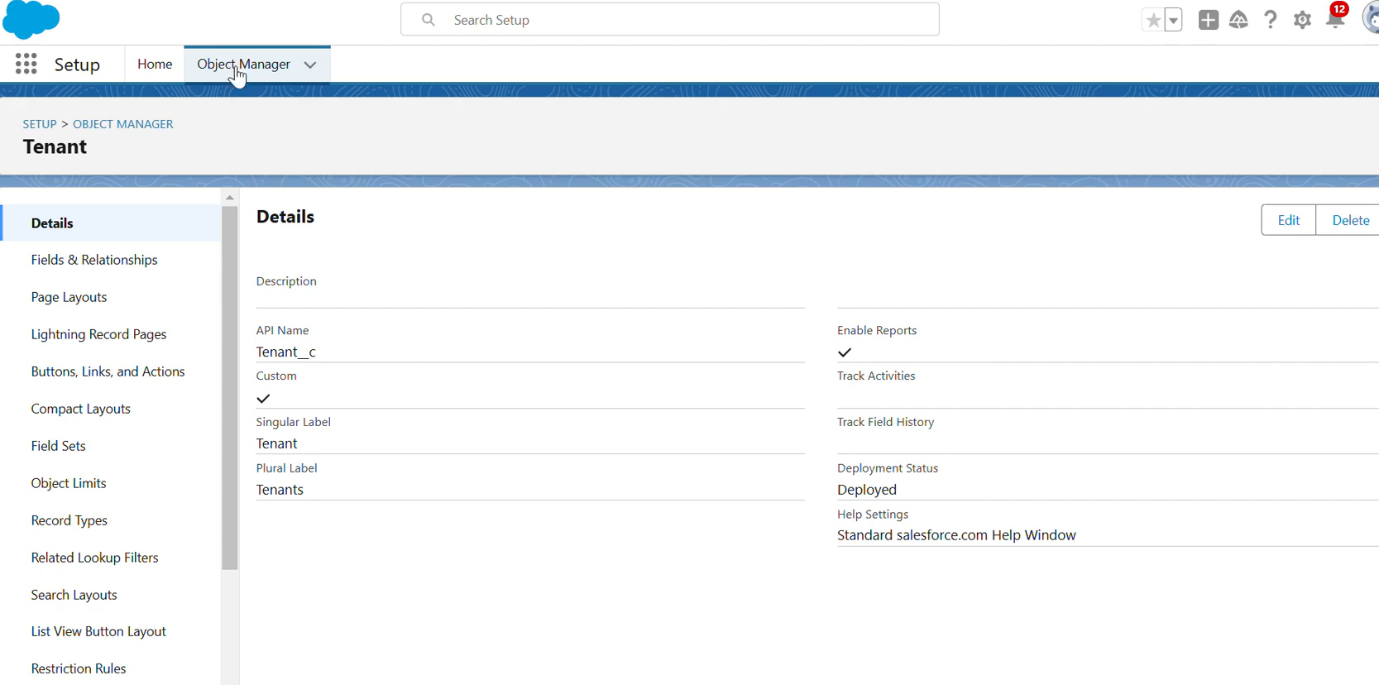
2.Plural label name: Tenants

3.Enter Record Name Label and Format

* Record Name : Tenant Name
* Data Type: Text

4.Click on Allow reports.

5.Allow search and Save.



**Create Lease Tracking Object**

1.Enter the label name ==>Lease Tracking

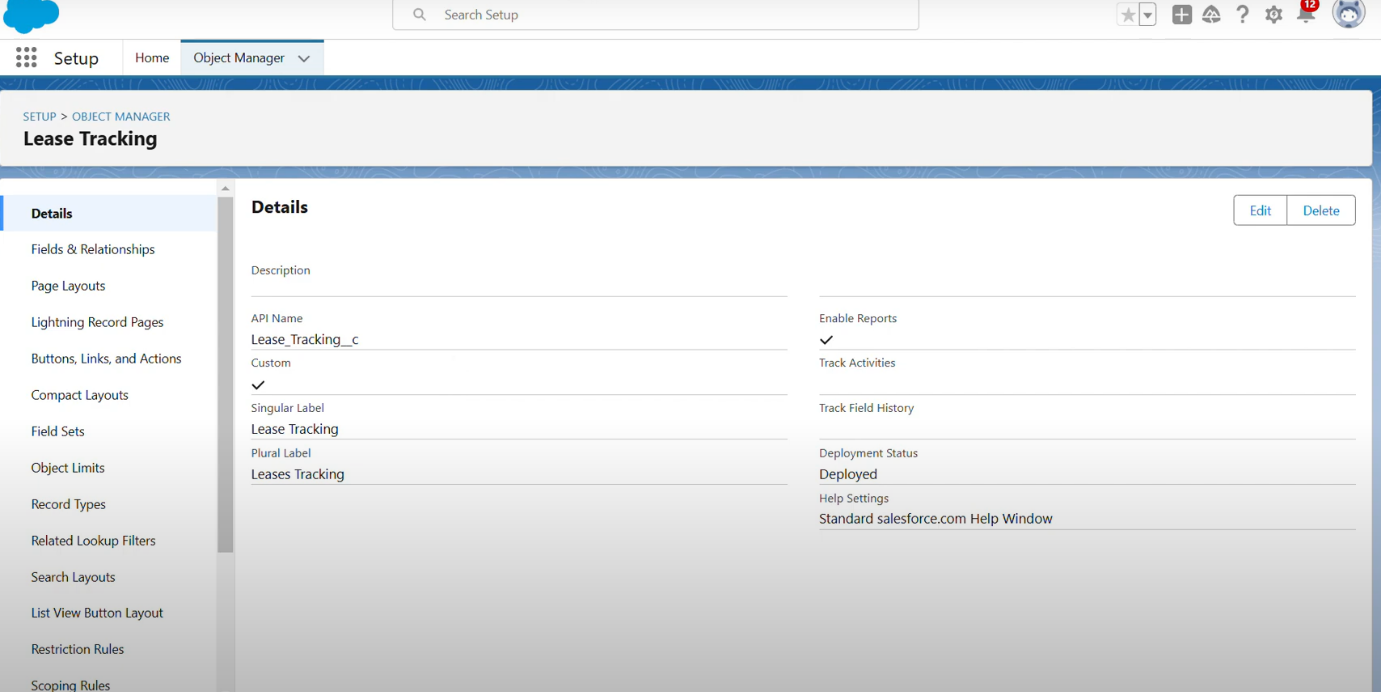
2.Plural label name ==> Leases Tracking

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.



**Create Tenant Issues Object**

1.Enter the label name ==> Tenant Issue

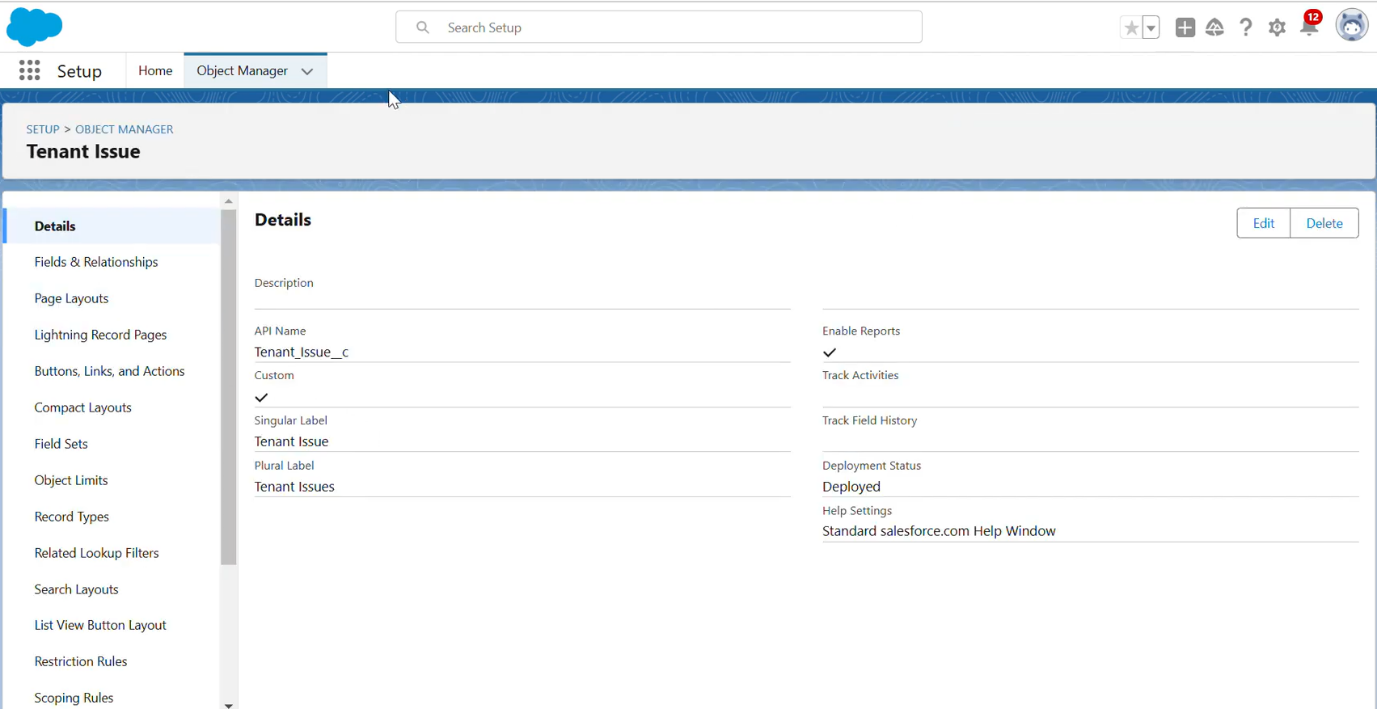
2.Plural label name ==> Tenant Issues

3,Enter Record Name Label and Format

* Record Name :  Issues
* Data Type : Auto number

4.Click on Allow reports.

5.Allow search and Save.

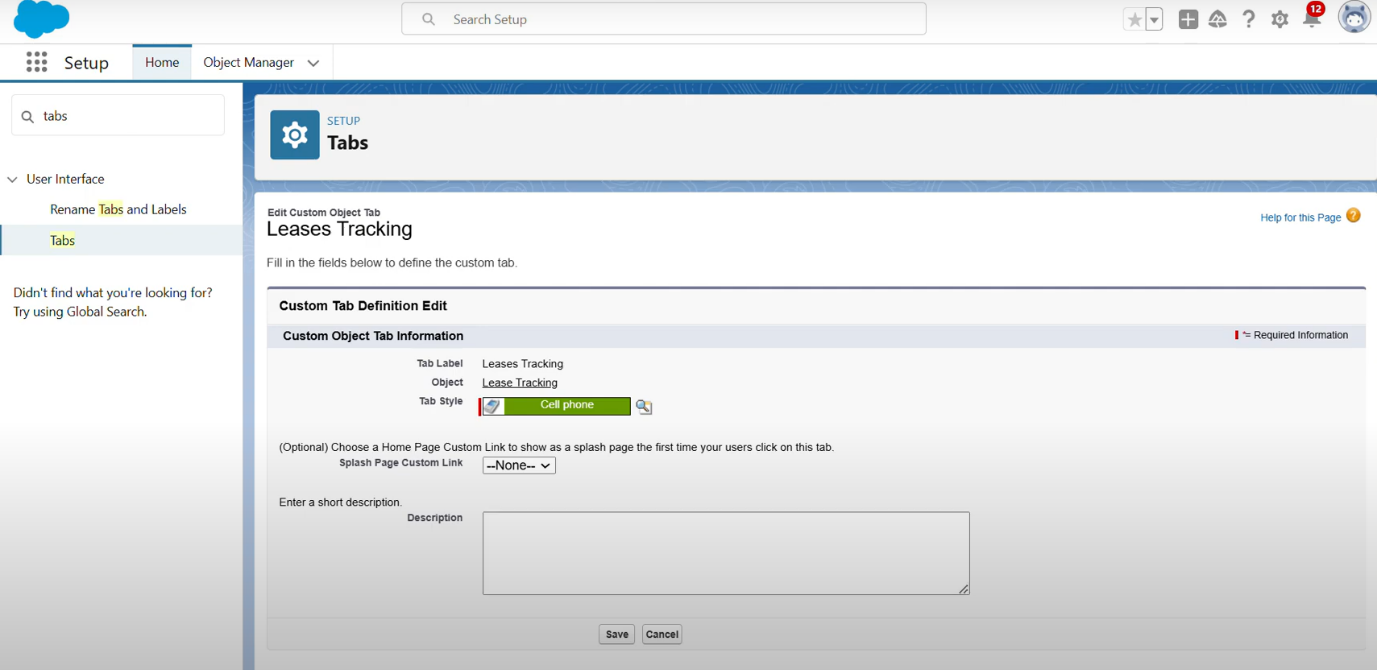


**Tabs**

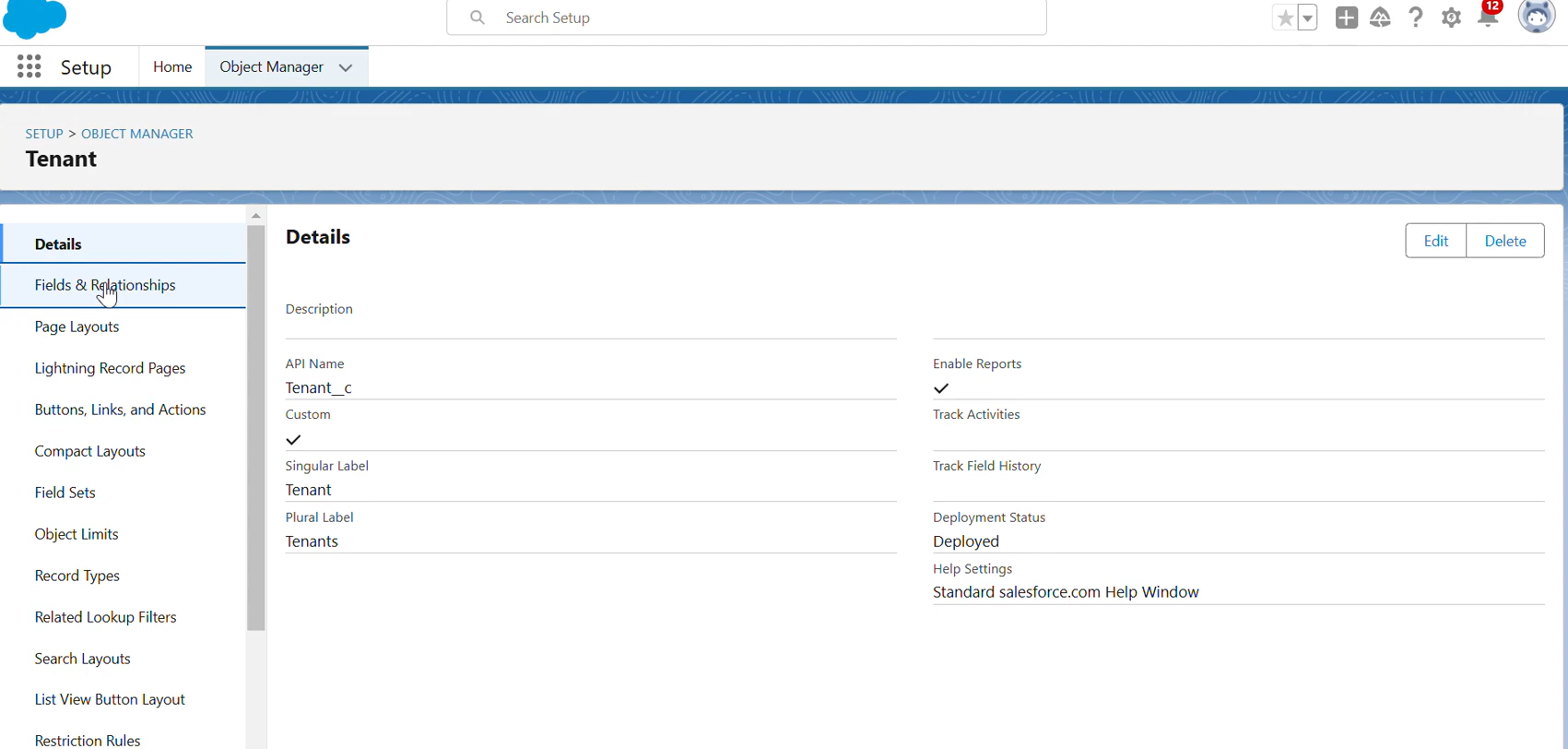
What is tab and Types of tabs :

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

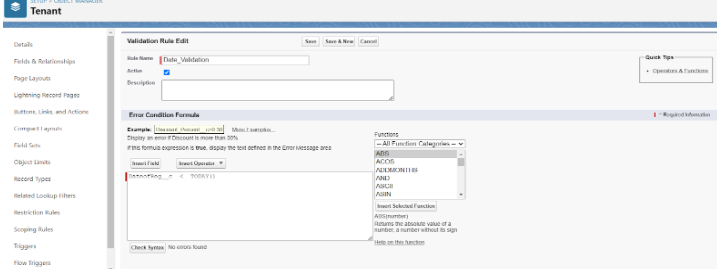
**Create a custom tab for tenant object**

****

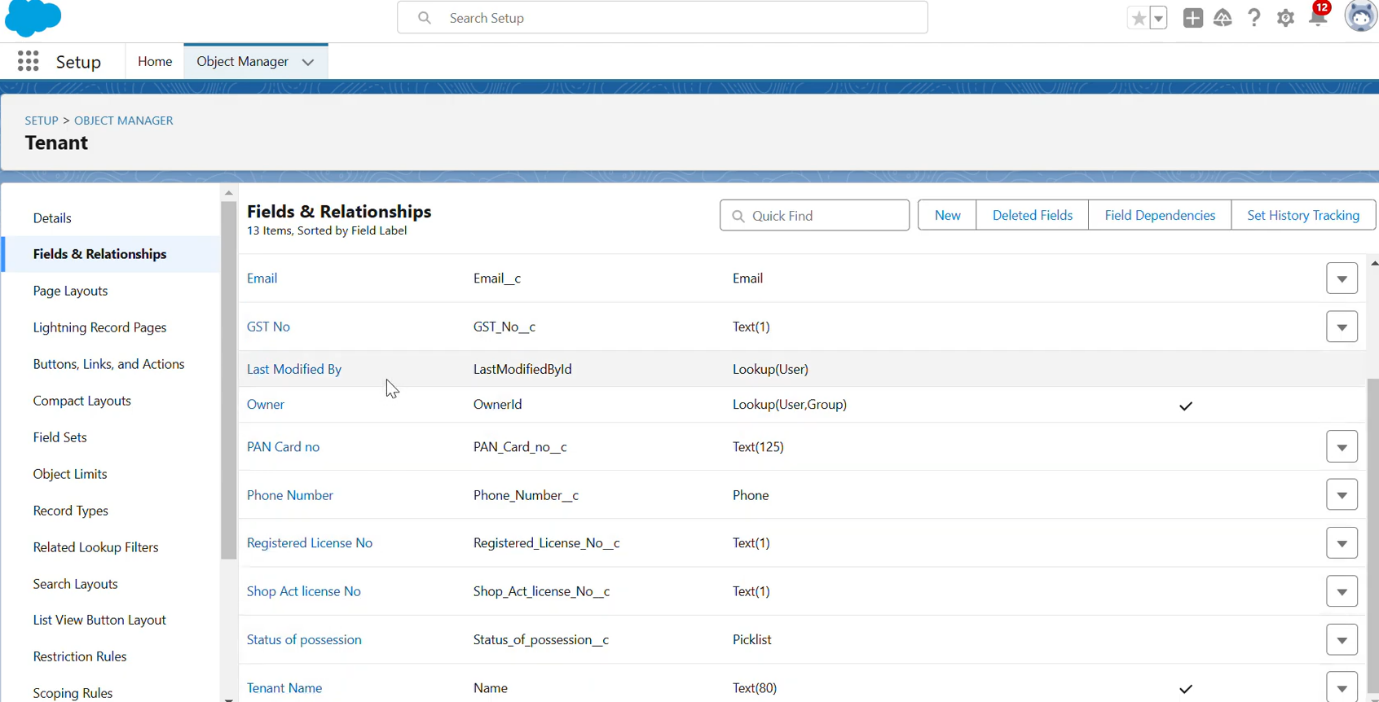
**Create Fields and Relationships**



**Create fields on Lease Tracking Object**

****

**Create fields on Tenant Issues**

****

**Create the Lightning App**

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

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

There are two types of Salesforce Applications:

* Standard Apps
* Custom Apps

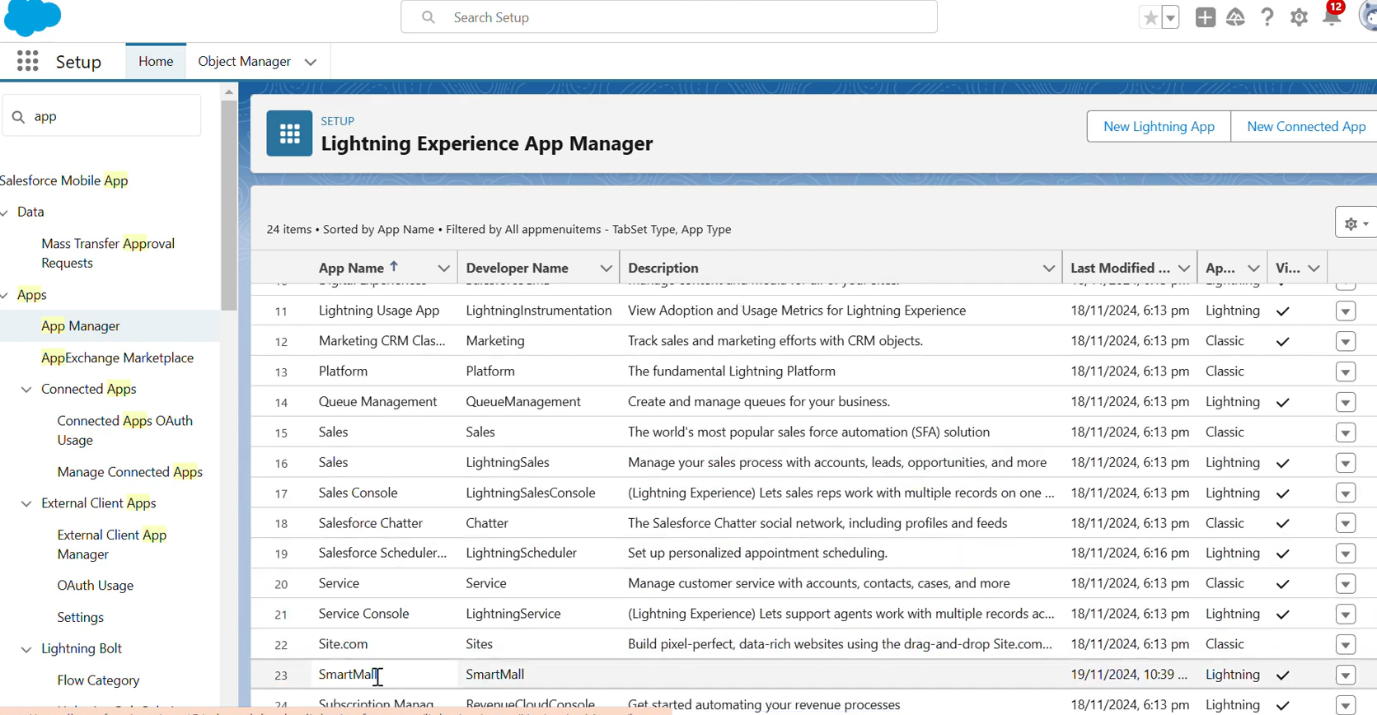
Standard Apps:

Standard apps come with every occurrence of Salesforce as default. Community, Call Center, Content, Sales, Marketing, Salesforce Chatter, Site.com, and App Launcher are included in these apps. The description, logo, and label of a standard app cannot be altered.

Custom Apps:

Custom apps are created according to the needs of a company. They can be made by putting custom and standard tabs together. Logos for custom apps can be changed.

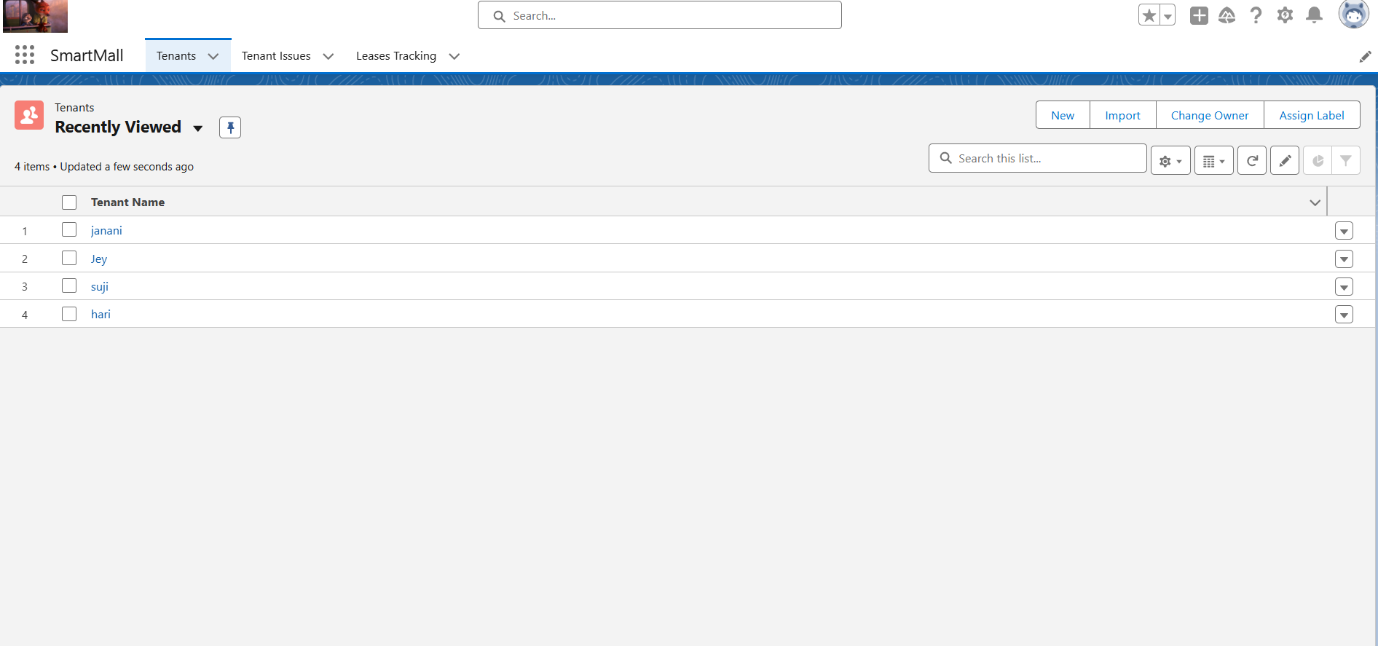
**Steps to create custom app in salesforce**

****

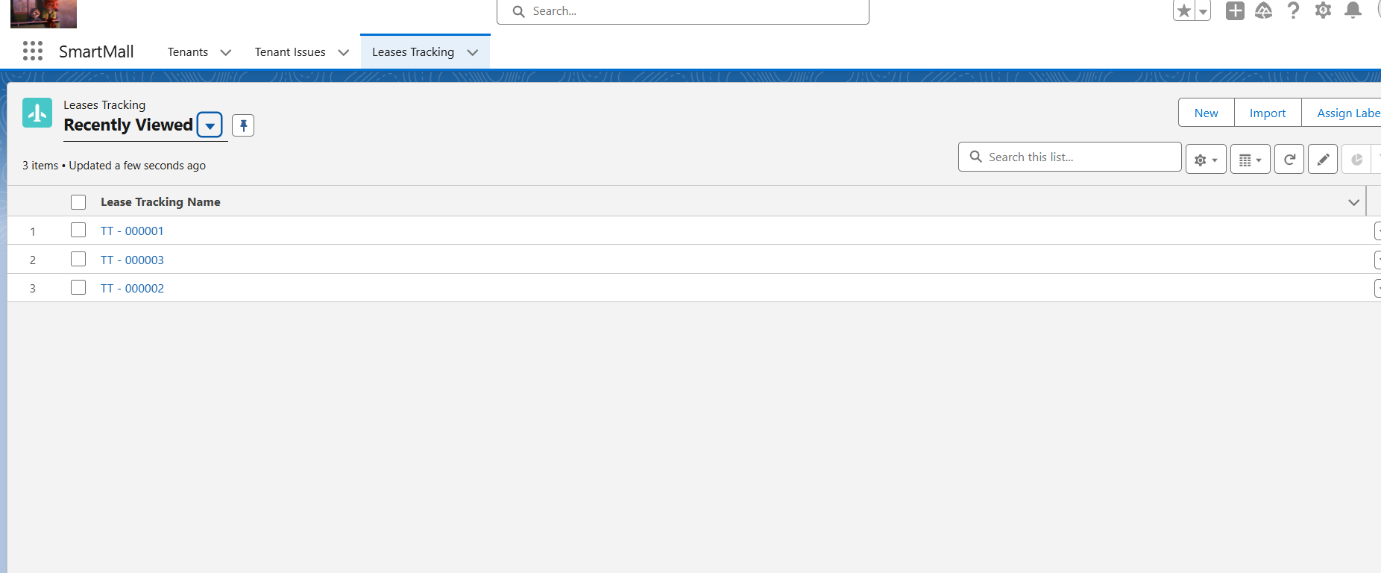
**Record Insertion**

Creating records in Salesforce is a fundamental and essential activity that serves multiple purposes, contributing to the effective management of data, streamlined processes, and overall success of an organization.

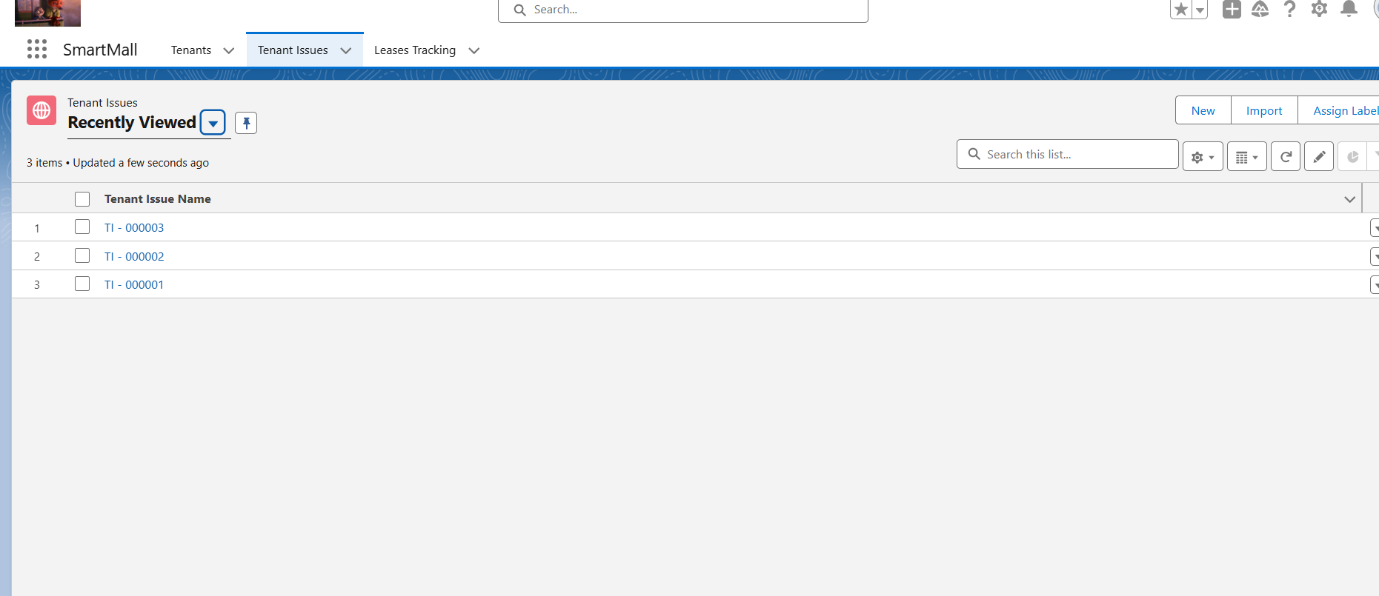
**Inserting Records in Tenant Object**



**Inserting Records in Tenants Issues object**



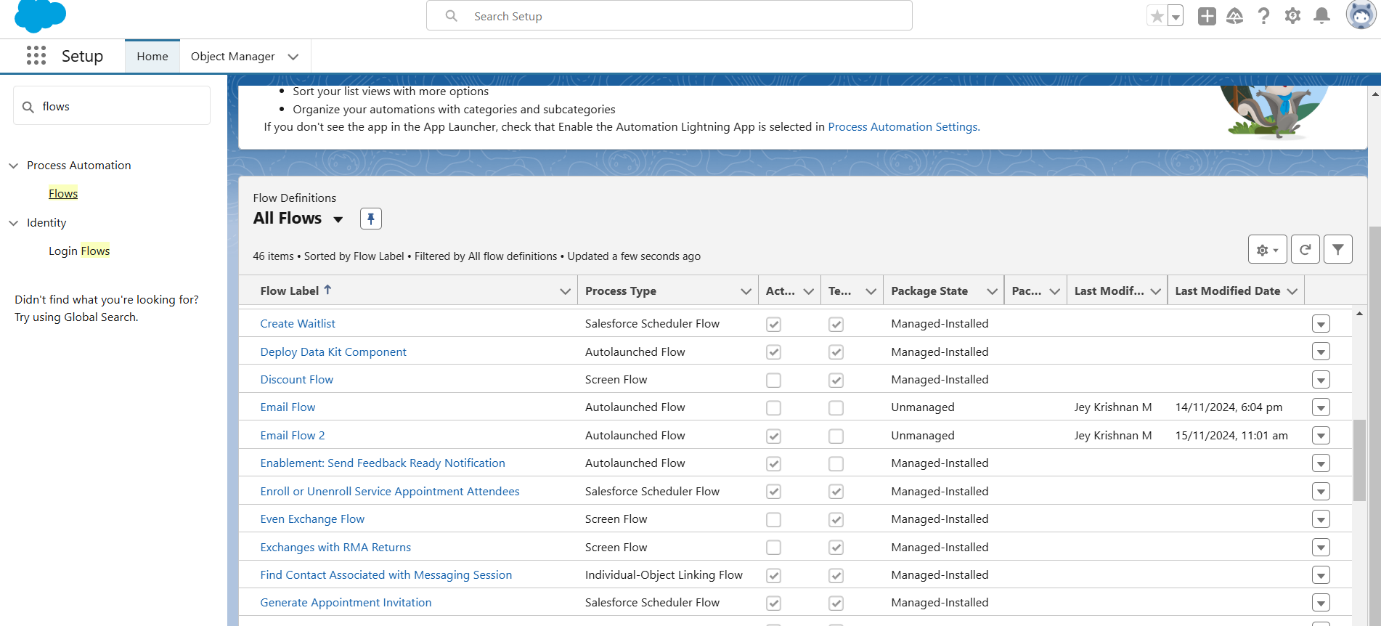
**Inserting Records in Lease Tracking Object**



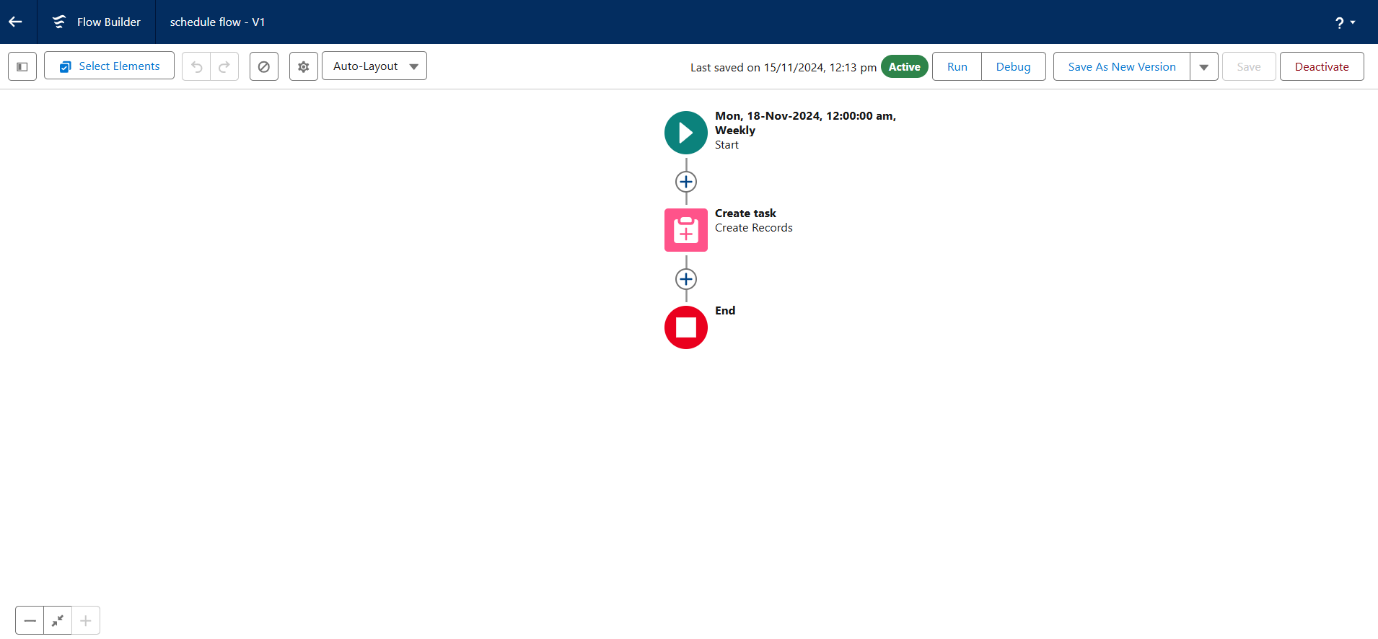
**Create Flows**

In Salesforce, a Flow is a powerful tool that allows users to automate complex business processes by orchestrating and automating sequences of tasks, data manipulations, and user interactions. Flows are designed through a visual interface, making them accessible to users with varying technical expertise.

**Create a Record Triggered flow on tenant Object**

****

**Create a Schedule Flow on Lease Management Object**



**Apex Triggers**

**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);

    }

}

**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);

           }

       }

   }

**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);

    }

}

**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');

            }

        }

    }

}

**Asynchronous Apex**

Asynchronous Apex in Salesforce refers to a programming paradigm where code execution is detached from the immediate context and occurs independently, typically in the background. This approach is designed to handle long-running processes, heavy computations, or tasks that should not block user interactions.

**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;

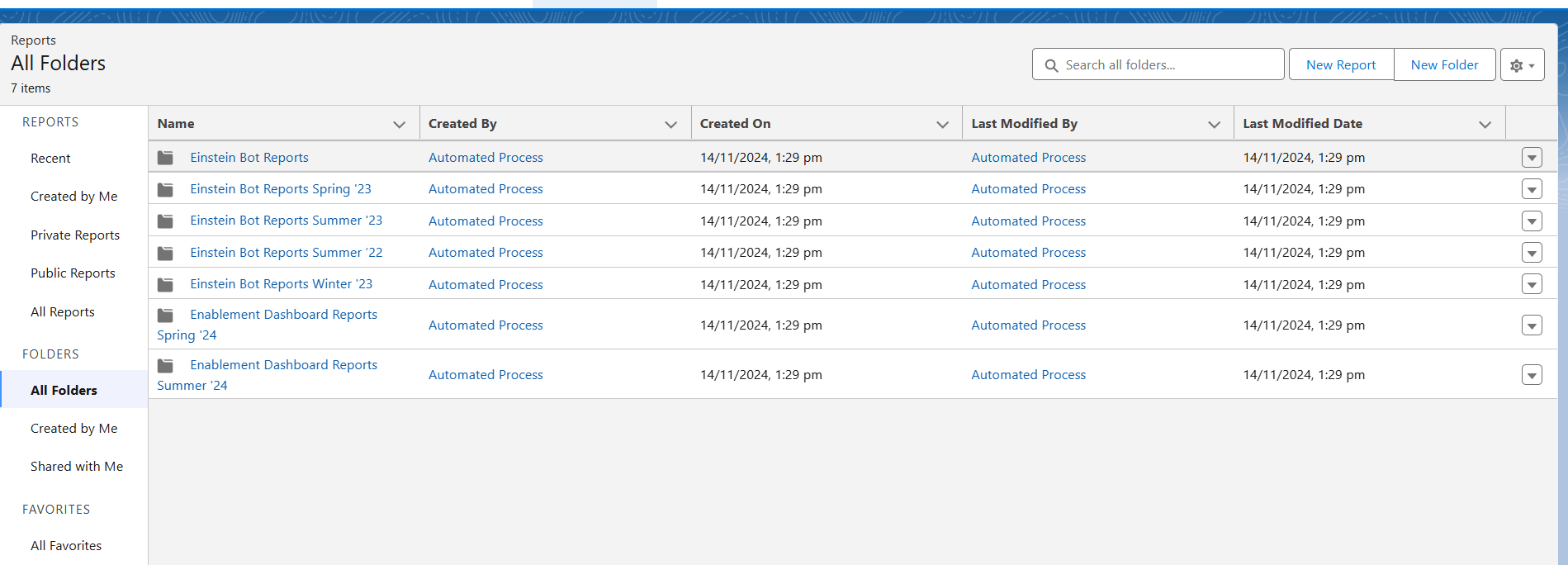
    }

}

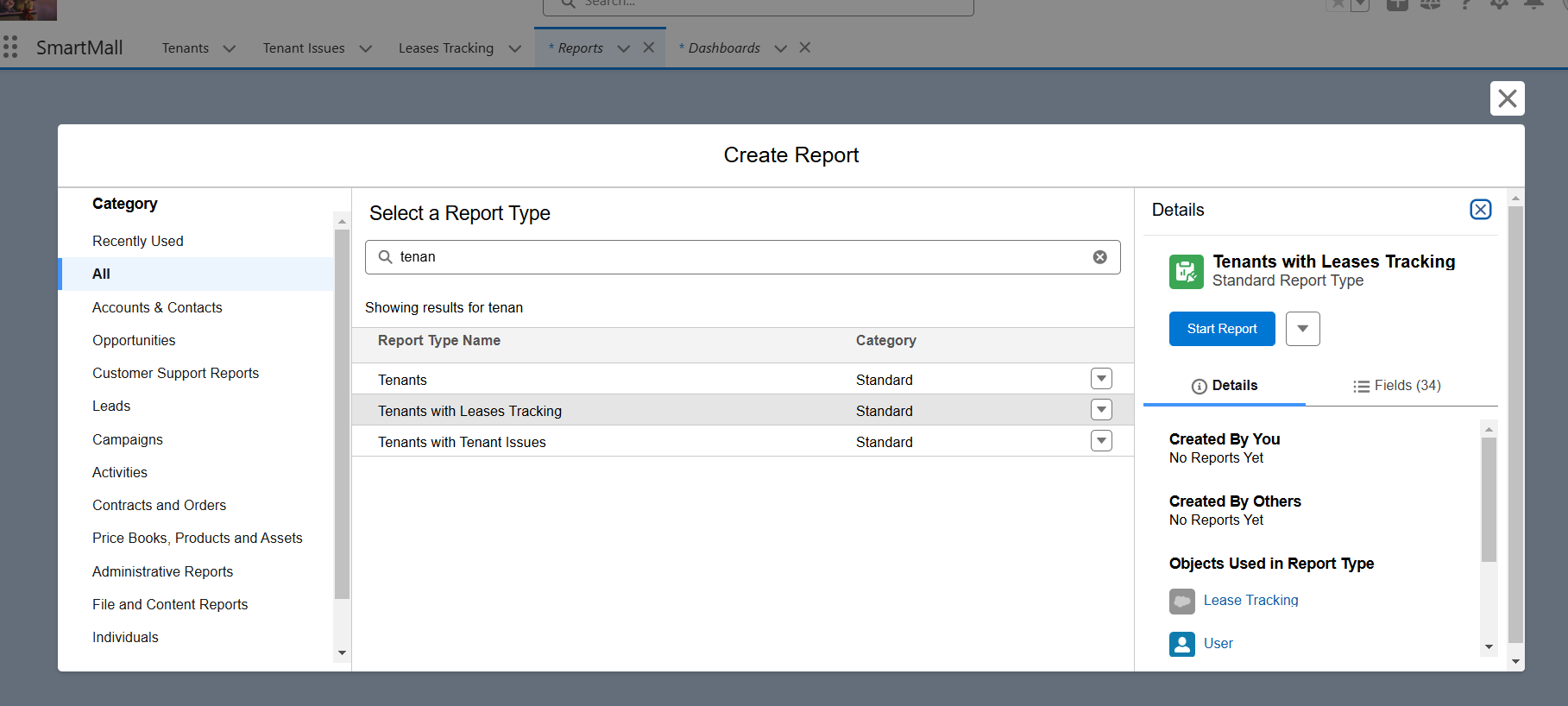
**Create Reports And Dashboards**

Salesforce Reports and Dashboards are powerful tools that empower users to visualize and analyze data within the Salesforce platform. They play a crucial role in providing insights, monitoring performance, and making informed business decisions.

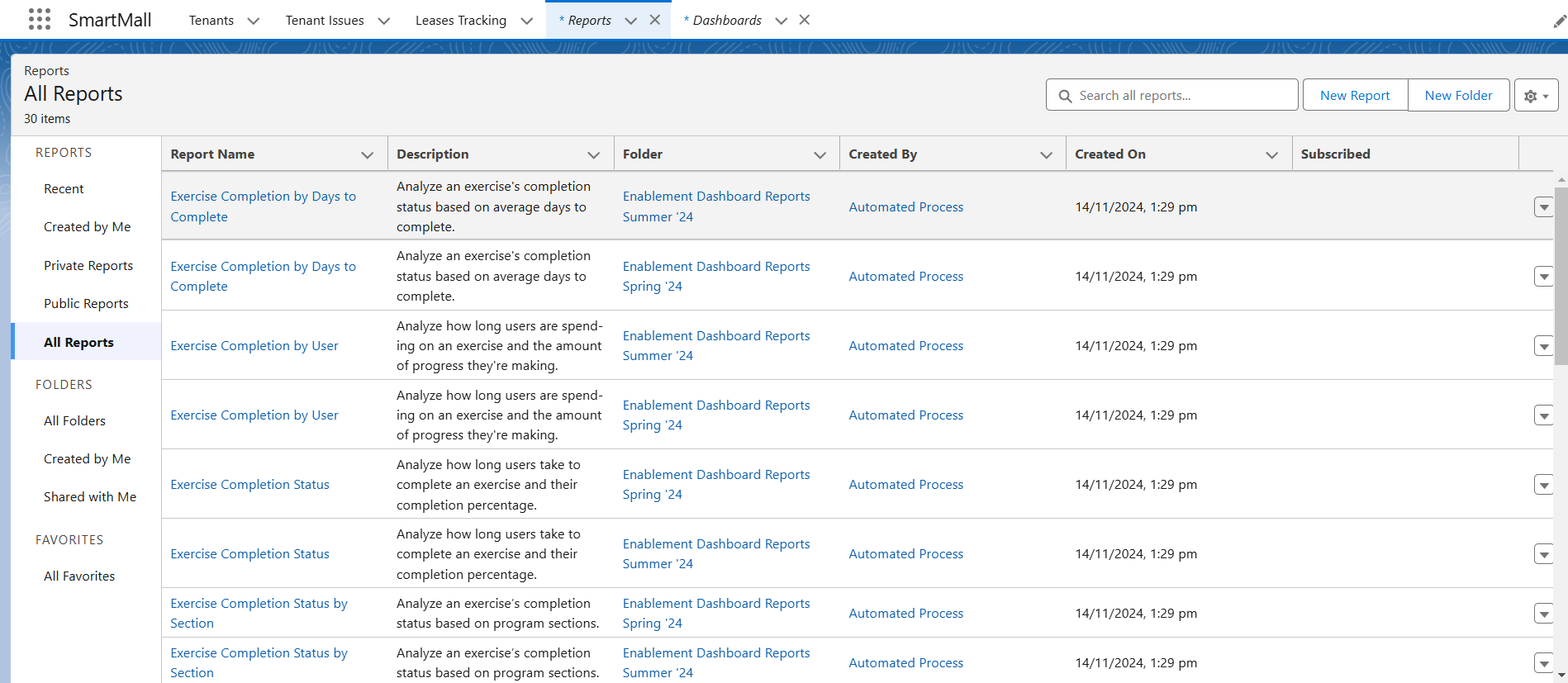
**Create a Report of lease Management Records**

****

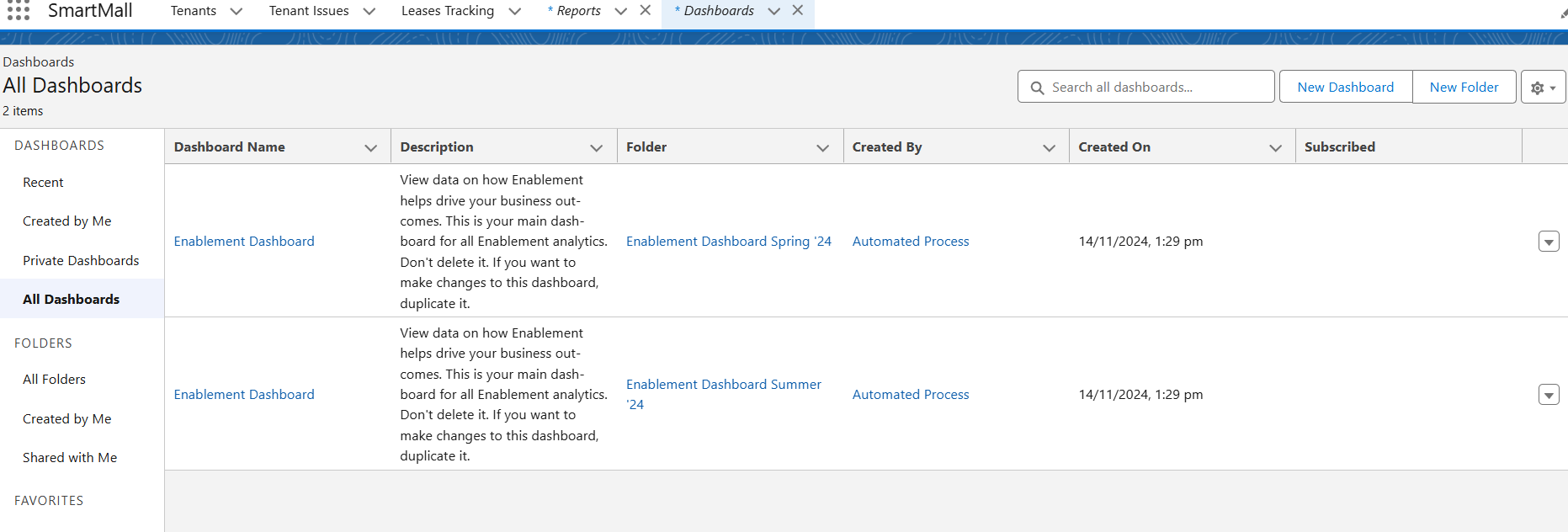
**Create a Report on Tenant issue Records**

****

**Create a Report on Tenant Records**

****

**Create a Dashboard**

****