GARAGE MANAGEMENT SYSTEM

By

Rusum Nagamani

nagamanirusum@gmail.com

ABSTRACT

The Garage Management System (GMS) is an all-in-one solution designed to streamline operations for automotive repair shops and garages. With its intuitive user interface and robust features, GMS enables businesses to deliver exceptional service, enhance productivity, and foster long-term customer loyalty.

Key Technologies

Salesforce CRM: Centralizes customer data, service history, and communication, allowing personalized interactions and streamlined management.

Salesforce Service Cloud: Manages service cases, work orders, and repair tracking. Automates customer support and communication for timely updates.

Salesforce Analytics and Reporting: Generates insightful reports and dashboards, helping businesses make data-driven decisions on performance, revenue, and customer trends.

Salesforce Integration Tools (APIs): Facilitates integration with third-party systems, such as accounting software, parts inventory, and payment gateways, ensuring seamless operations.

Benefits

- Improved Operational Efficiency
- Enhanced Customer Satisfaction
- Real-Time Access and Updates
- Comprehensive Reporting
- Scalability and Customization

By incorporating Salesforce's technologies into GMS, garages can boost productivity, improve customer satisfaction.

INDEX PAGE

PROCESS OF GARAGE MANAGEMENT SYSTEM

1.OBJECT CREATION

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:

Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.

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.

- CREATION OF “CUSTOMER DETAILS OBJECT”
    1. To create an object:
    2. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
    3. Enter the label name >> Customer Details
    4. Plural label name >> Customer Details
    5. Enter Record Name Label and Format
    6. Record Name >> Customer Name
    7. Data Type >> Text
    8. Click on Allow reports and Track Field History,
    9. Allow search >> Save

- **CREATION OF “APPIONTMENT” OBJECT:**

  To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Appointment
3. Plural label name >> Appointments
4. Enter Record Name Label and Format
5. Record Name >> Appointment Name
6. Data Type >> Auto Number
7. Display Format >> app-{000}
8. Starting number >> 1
9. Click on Allow reports and Track Field History,
10. Allow search >> Save.

- **CREATION OF “APPIONTMENT” OBJECT**

To create an object:

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

Enter the label name >> Service records

Plural label name >> Service records

Enter Record Name Label and Format

Record Name >>Service recordsName

Data Type >> Auto Number

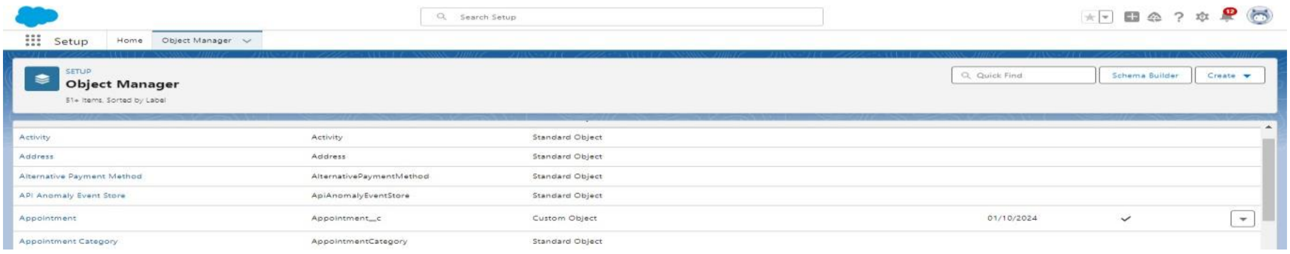
Display Format >> ser-{000}

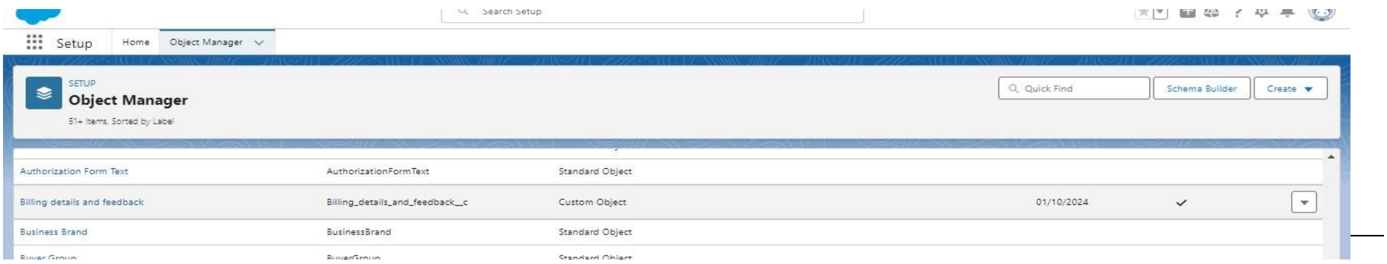
Starting number >> 1

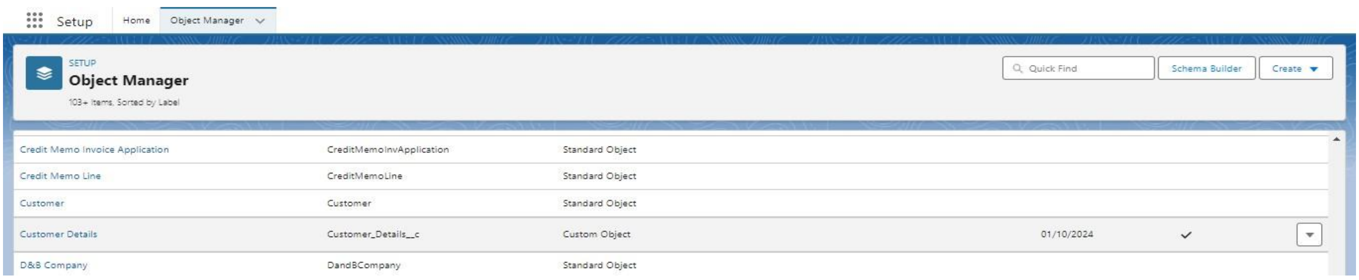
Click on Allow reports and Track Field History,

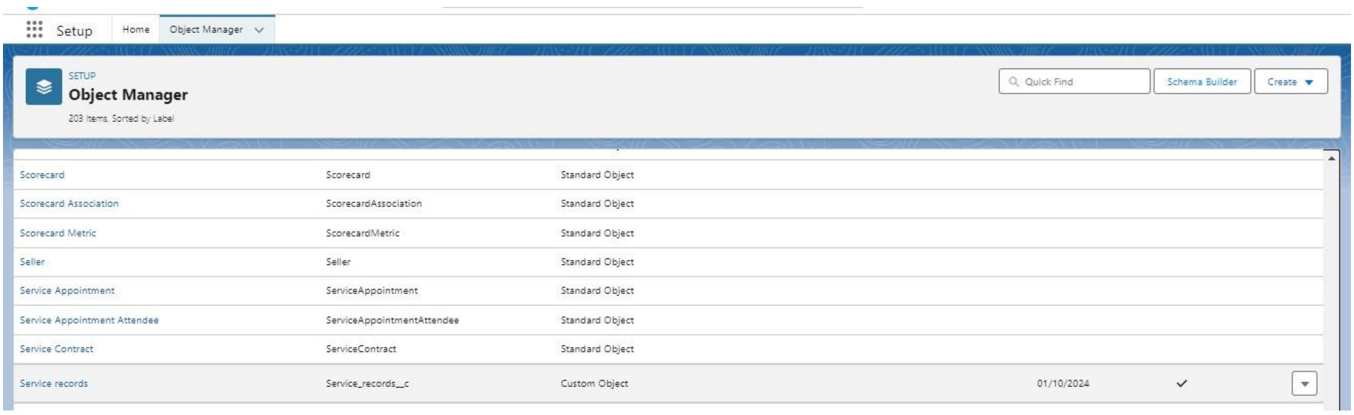
Allow search >> Save.

RESULTING PAGE OF AFTER CREATING OBJECTIVES:








Setup
Home
Object Manager
SETUP
Object Manager
203 Items, Sorted by Label
Quick Find
Schema Builder
Create
Scorecard
Scorecard
Standard Object
Scorecard Association
ScorecardAssociation
Standard Object
Scorecard Metric
ScorecardMetric
Standard Object
Seller
Seller
Standard Object
Service Appointment
ServiceAppointment
Standard Object
Service Appointment Attendee
ServiceAppointmentAttendee
Standard Object
Service Contract
ServiceContract
Standard Object
Service records
Service_records__c
Custom Object
01/10/2024

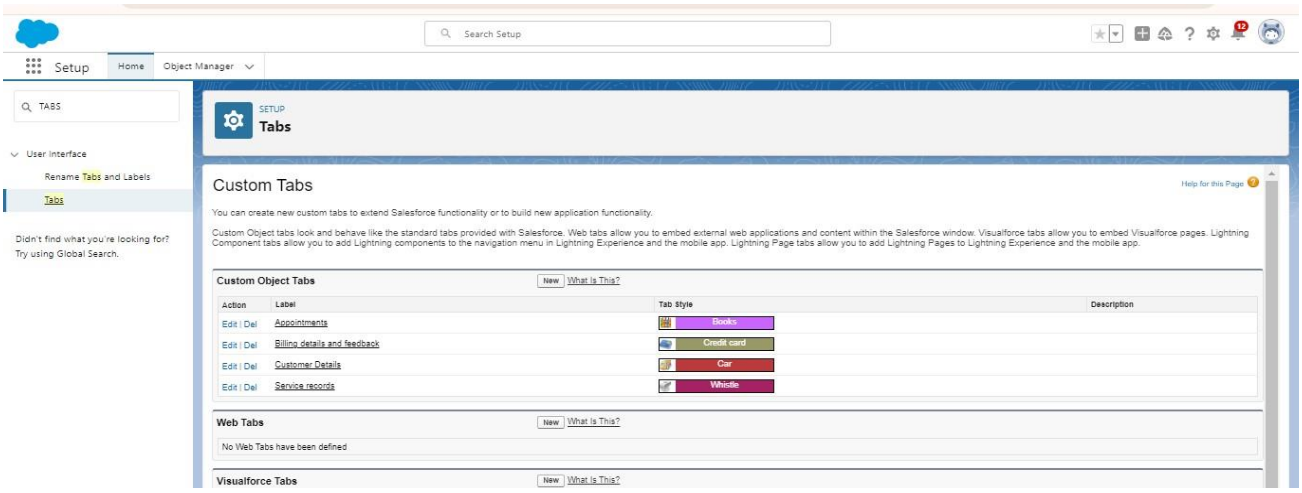
2.TABS:

- To create a Tab:(Customer Details)

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

- Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
- Make sure that the Append tab to users' existing personal customizations is checked.
- Click save.
- Now create the Tabs for the remaining Objects, they are " Appointments, Service records,Billing details and feedback".

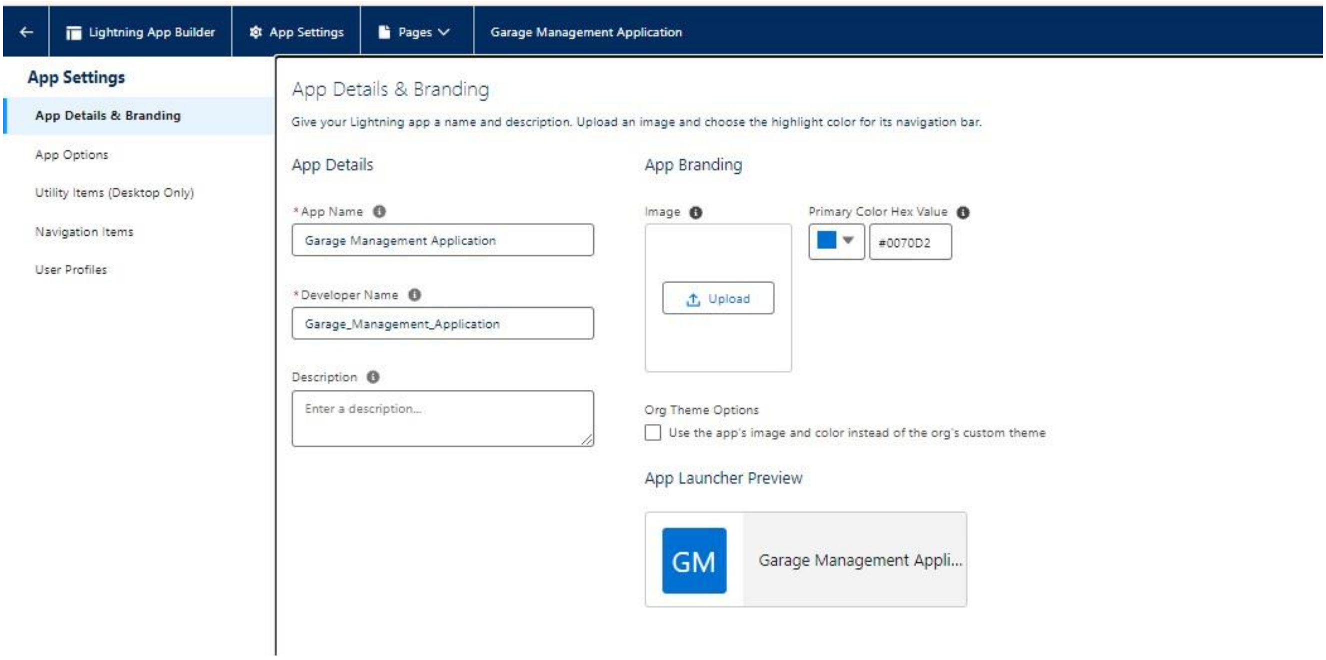
RESULTING PAGE OF AFTER CREATING TABS:



3. The Lightning App:

- To create a lightning app page:
- Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App
- Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.
- To Add Navigation Items:
- Select the items (Customer Details,Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.
- To Add User Profiles:
- Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

RESULTING PAGE OF LIGHTINING APPP

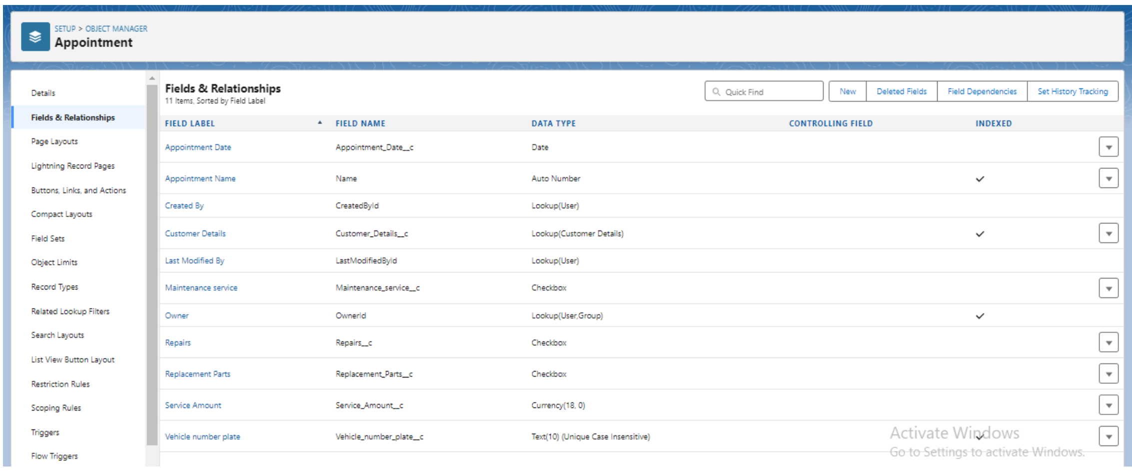


4.FIELDS:

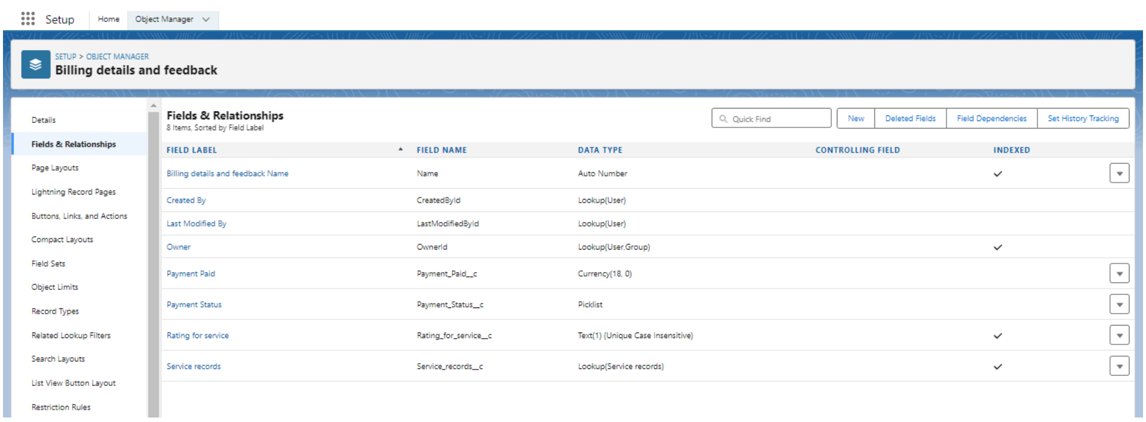
- Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select Data type as a “Email” and Click on Next
- Fill the Above as following:
- Field Label : Gmail
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

RESULTING PAGE OF EACH OBJECT:

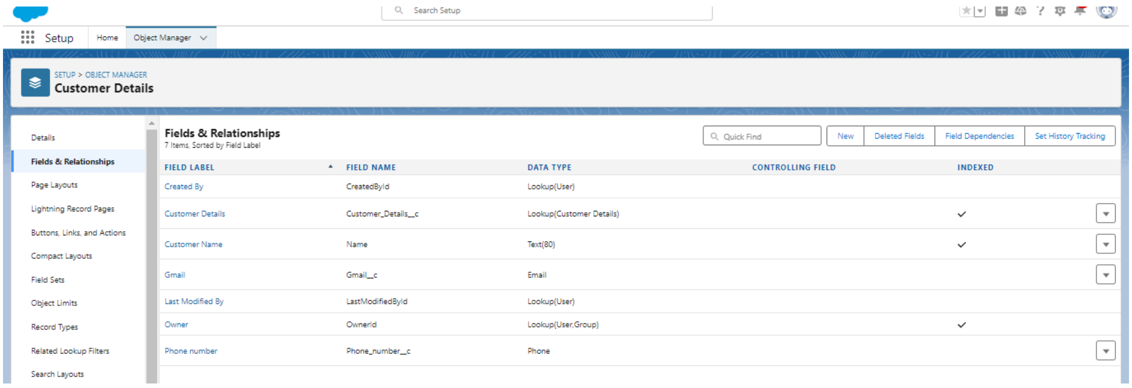
- APPIONTMENT OBJECT



- BILLING DETAILS AND FEED BACK OBJECT:

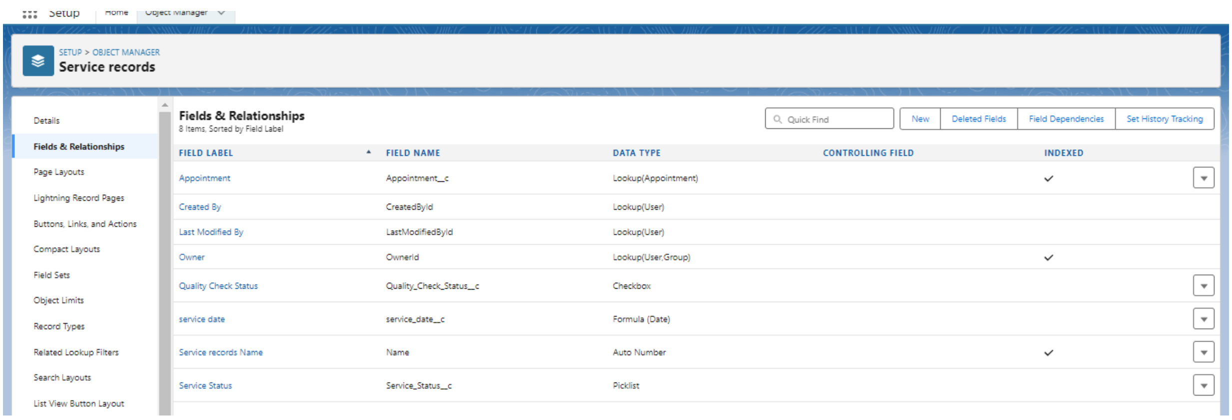


- CUSTOMER DETAILS OBJECT:



FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓
Customer Name	Name	Text(80)		✓
Gmail	Gmail__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone number	Phone_number__c	Phone		

- SERVICE RECORDS OBJECT:



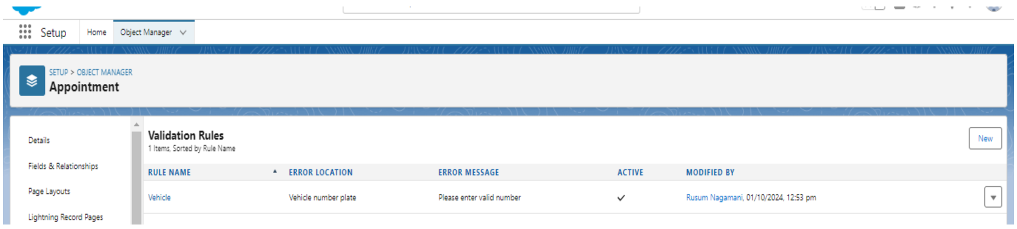
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment__c	Lookup(Appointment)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Quality Check Status	Quality_Check_Status__c	Checkbox		
service date	service_date__c	Formula (Date)		
Service records Name	Name	Auto Number		✓
Service Status	Service_Status__c	Picklist		

5.VALIDATION RULE:

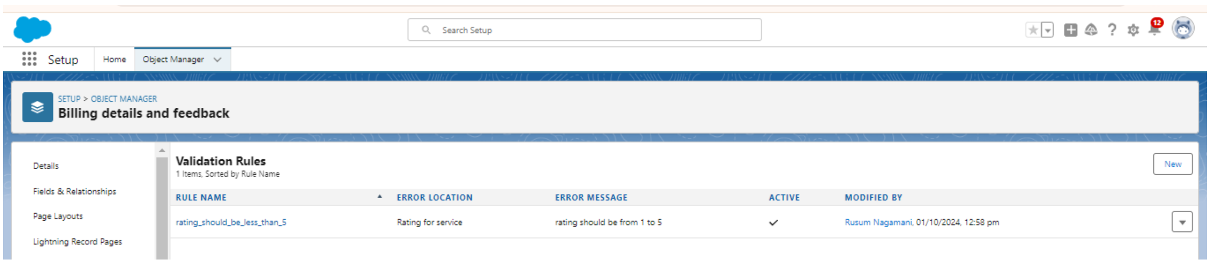
- Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
- Click on the validation rule >> click New.
- Enter the Rule name as “ ”.
- Insert the Error Condition Formula
- Enter the Error Message as “ ”, select the Error location as Field and select the field as “ ”, and click Save.

RESULTING OF VALIADATION RULE IN EACH OBJECT

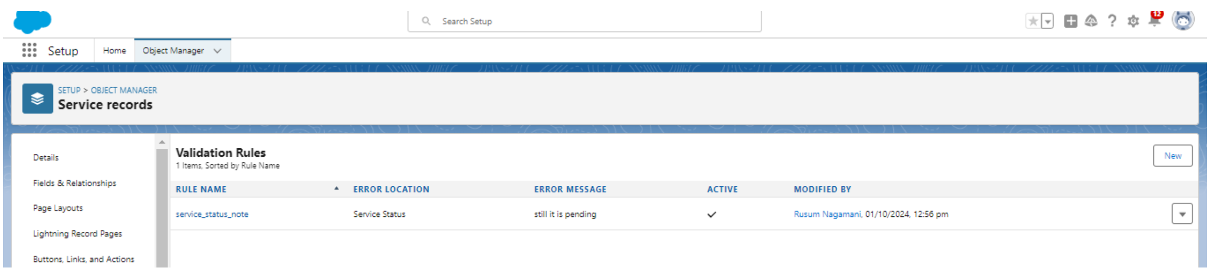
- APPIONTMENT OBJECT



- BILLING DETAILLS AND FEEDBACK OBJECT:



- SERVICE RECORDS OBJECT:



6.DUPLICATE RULE:

Go to quick find box in setup and search for matching Rule.

Click on matching rule >> click on New Rule.

Select the object as Customer details and click Next.

Give the Rule name : Matching customer details

Unique name : is auto populated

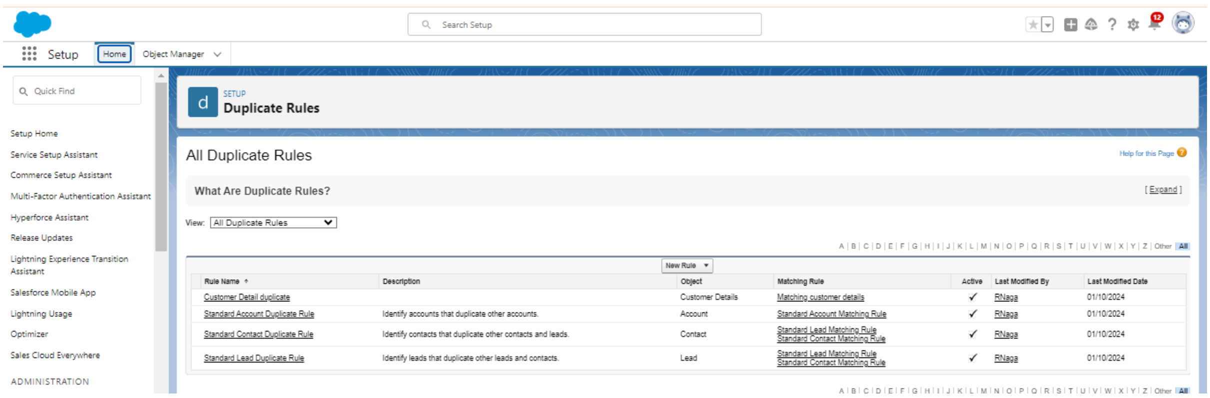
Define the matching criteria as

Field	Matching Method
1. Gmail	Exact
2. Phone Number	Exact

Click save.

After Saving Click on Activate.

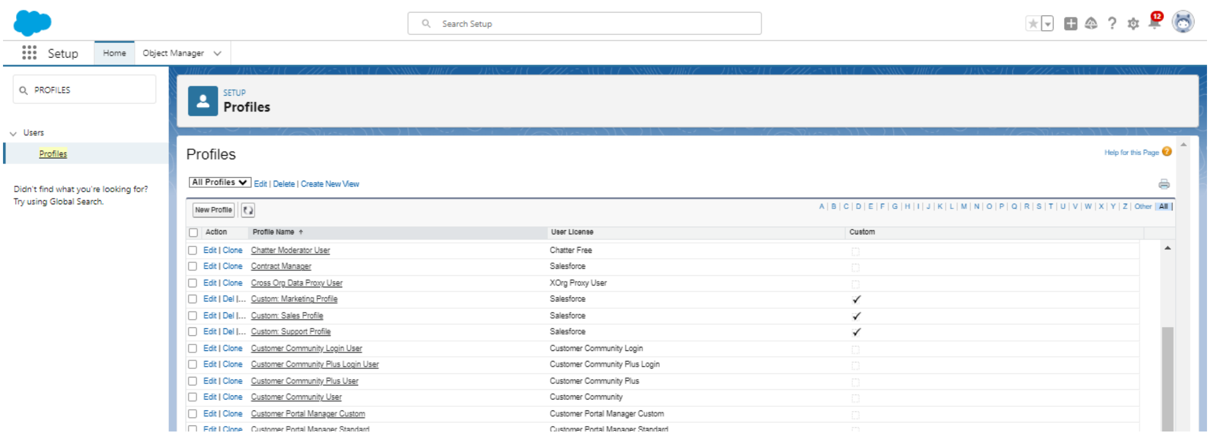
RESULT PAGE OF DUPLICATE RULES:



7.PROFILES:

- Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.
- While still on the profile page, then click Edit.
- Select the Custom App settings as default for the Garage management.
- Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
- Changing the session times out after should be “ 8 hours of inactivity”.
- Change the password policies as mentioned :
- User passwords expire in should be “ never expires ”.
- Minimum password length should be “ 8 ”, and click save

RESULT PAGE OF PROFILES:



8.Role & Role Hierarchy

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating Manager Role:

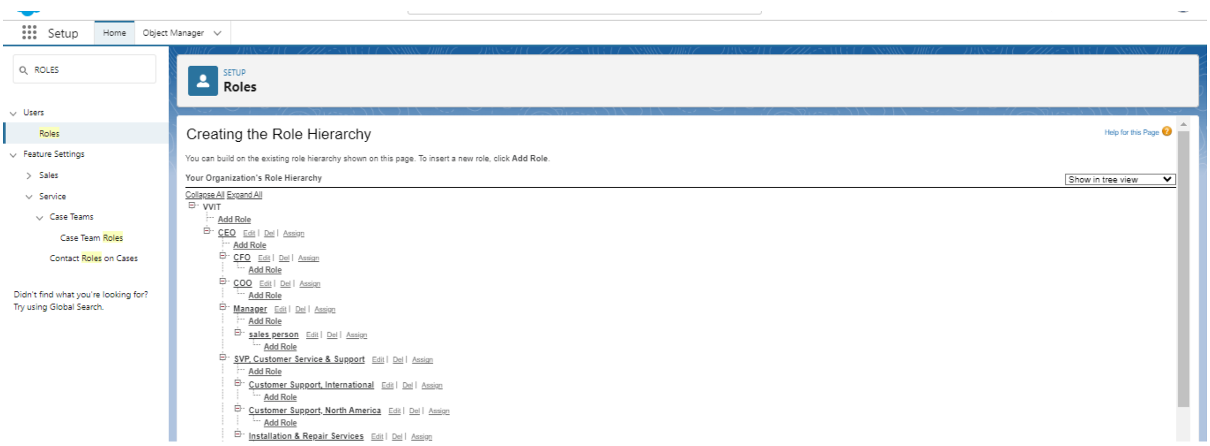
Go to quick find >> Search for Roles >> click on set up roles.

Click on Expand All and click on add role under whom this role works

Give Label as “Manager” and Role name gets auto populated. Then click on Save.

Give the sales person follwed by manager in its internal with same factors with the name”sales person”

RESULT OF ROLES AND HEIARCHY:



9. Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Go to setup >> type users in quick find box >> select users >> click New user.

Fill in the fields

First Name : Niklaus

Last Name : Mikaelson

Alias : Give a Alias Name

Email id : Give your Personal Email id

Username : Username should be in this form: text@text.text

Nick Name : Give a Nickname

Role : Manager

User licence : Salesforce

Profiles : Manager

Save

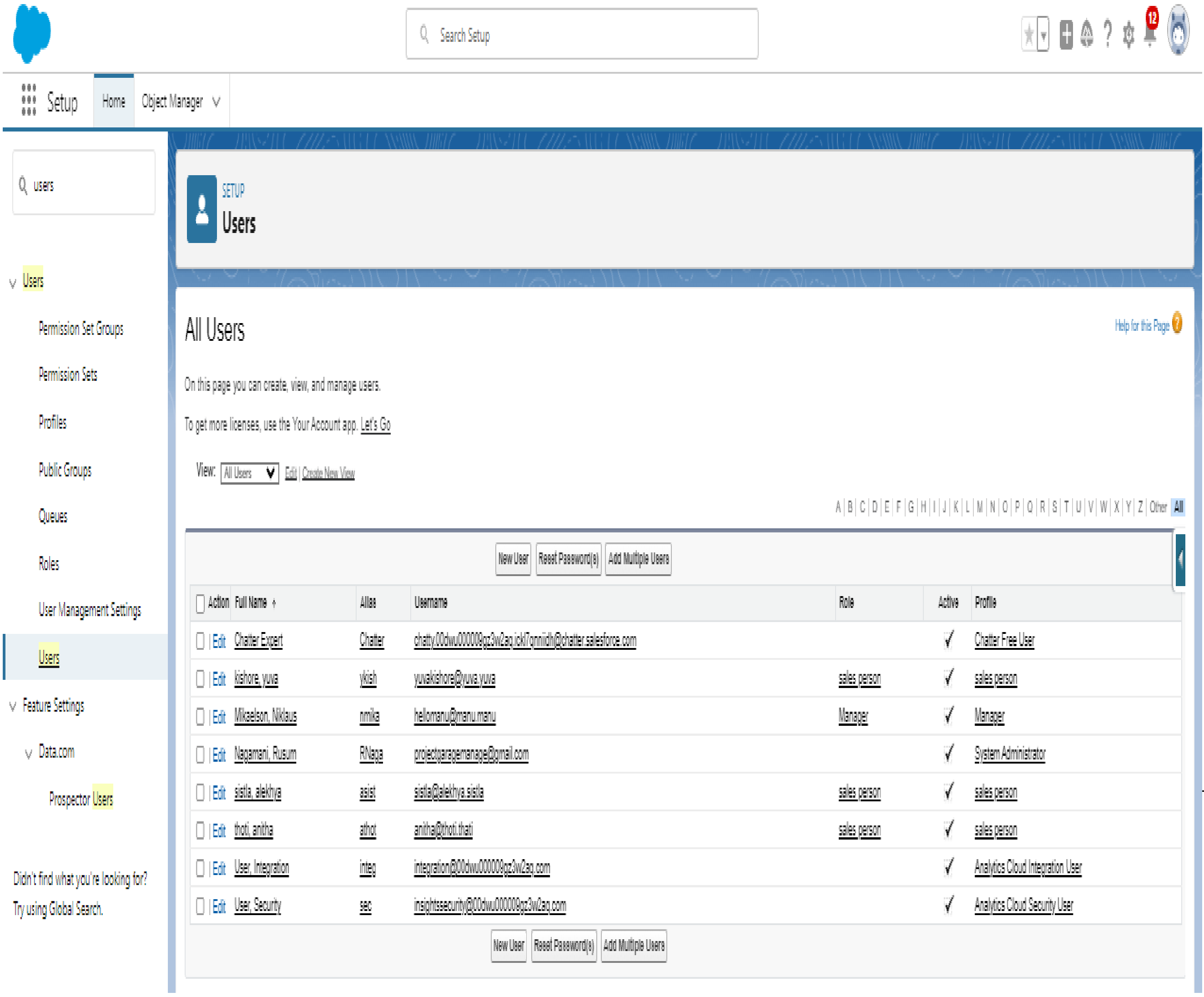
Repeat the steps and create another user using

Role : sales person

User licence : Salesforce Platform

Profile : sales person

RESULT PAGE OF USERS:



10.PUBLIC GROUPS:

Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings. By creating and using public groups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have appropriate access to the resources they need.

Creating New Public Group

Go to setup >> type users in quick find box >> select public groups >> click New

Give the Label as “sales team”.

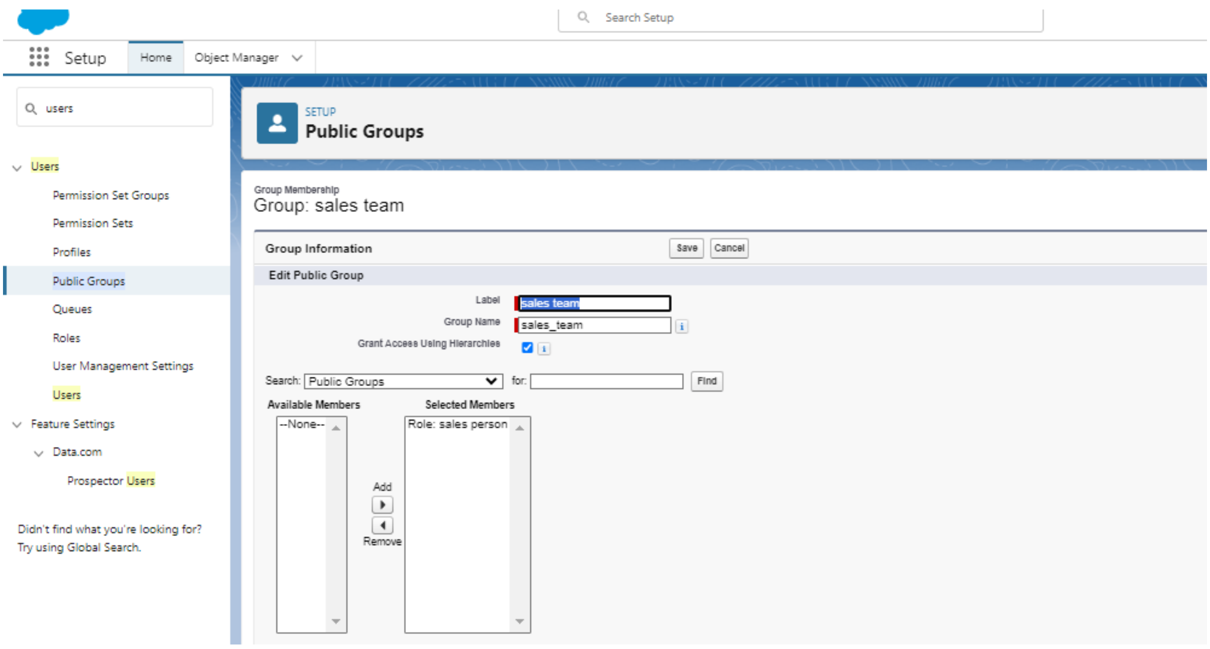
Group name is autopopulated.

Search for Roles.

In Available Members select Sales person and click on add it will be moved to selected member.

Click on save.

RESULT OF PUBLIC GROUPS:



11. Sharing Setting

Salesforce allows you to configure sharing settings to control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy. Salesforce provides a variety of tools and mechanisms to define and enforce sharing rules, such as:

Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.

Change the OWD setting of the Service records Object to private .

Click on save and refresh.

Scroll down a bit, Click new on Service records sharing Rules.

Give the Label name as “ Sharing setting”

Rule name is auto populated.

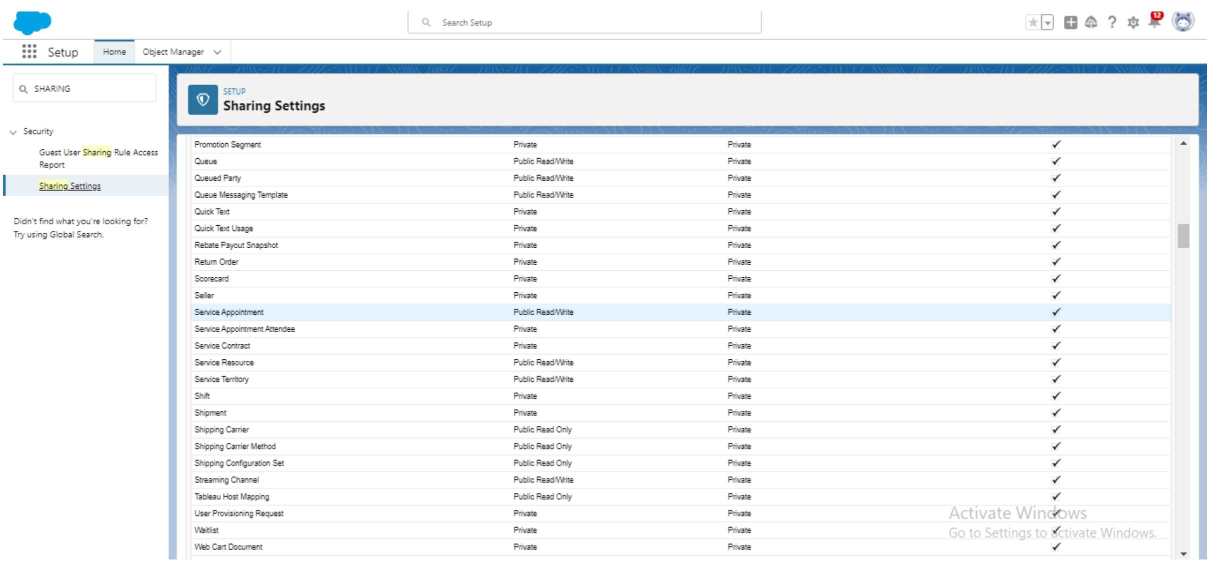
In step 3 : Select which records to be shared, members of “ Roles ” >>“ Sales person”

In step 4: share with, select “ Roles ” >> “ Manager ”

In step 5 : Change the access level to “ Read / write ”.

Click on save.

RESULT PAGE:



12.FLOWS

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

Select the Record-triggered flow and Click on Create.

Select the Object as “Billing details and feedback”in the Drop down

list.

Select the Trigger Flow when: “A record is Created or Updated”.

Select the Optimize the flow for: “Actions and Related Records” and Click on Done.

Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.

Give the Label Name : Amount Update

Api name : is auto populated

Set a filter condition : All Conditions are met(AN

Field : Payment_Status__c

Operator : Equals Value : Completed

And Set Field Values for the Billing details and feedback Record

FielD Payment_Paid__c Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}

Click On Done.

Before creating another Element. Create a New Resource form Toolbox form top left.

Click on the New Resource, And select Variable.

Select the resource type as text templateEnter the API name as “ alert”.

Change the view as Rich Text ? View to Plain Text.

In body field paste the syntax that given below.

Dear
{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!$Record.Payment_Paid__c}

Thank you for Coming .

24. Now Click on Add Element , select Action.

25. Their action bar will be opened in that search for " send email " and click on it.

26. Give the label name as " Email Alert"

27. API name will be auto populated.

28. Enable the body in set input values for the selected action.

29. Select the text template that created , Body : {!alert}

30. Include recipient address list select the email form the record.

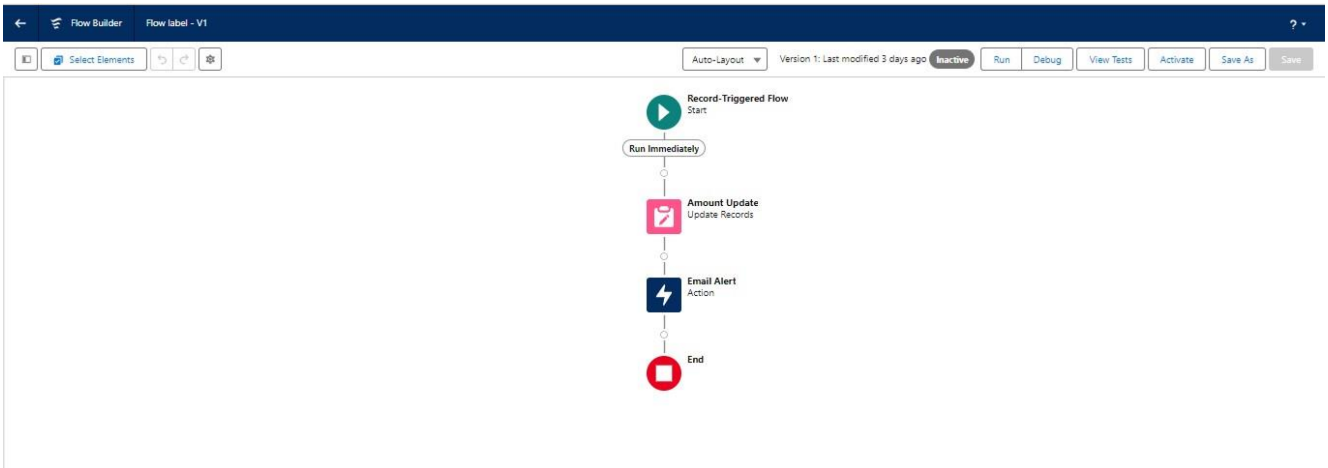
31. RecipientAddressList:
{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}

32. Include subject as " Thank You for Your Payment - Garage Management".

33. Click done.

Click on save. Give the Flow label , Flow Api name will be autopopulated.And click save, and click on activate

RESULT OF FLOW:



13.APEX TRIGGER:

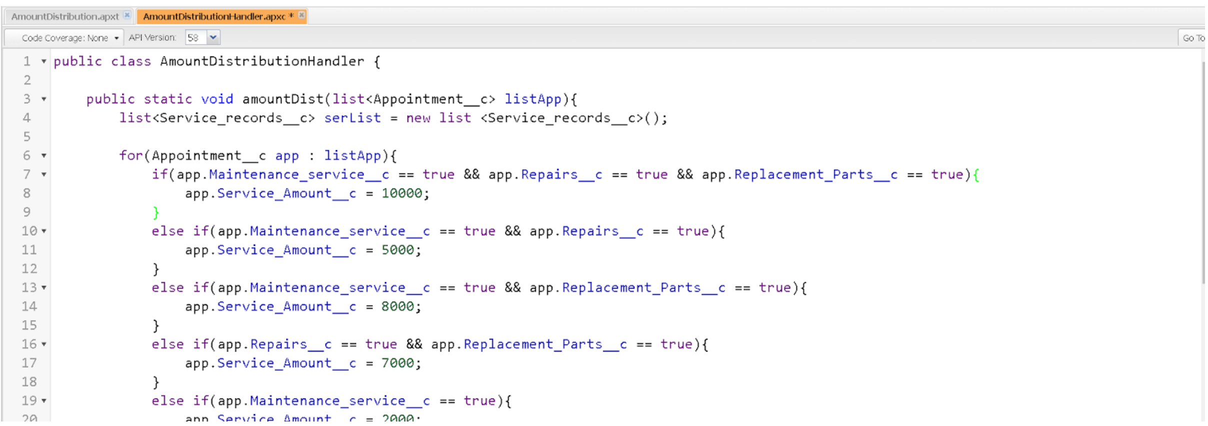
Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

Login to the respective trailhead account and navigate to the gear icon in the top right corner.

Click on the Developer console. Now you will see a new console window.

In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.

Name the class as “AmountDistributionHandler ”.



AmountDistribution.apxt | AmountDistributionHandler.apxc *

Code Coverage: None | API Version: 58

```
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
            }

        }
    }
}
```

CODE:

```
public class AmountDistributionHandler {


   public static void amountDist(list<Appointment__c> listApp){

      list<Service_records__c> serList = new list <Service_records__c>();


      for(Appointment__c app : listApp){

         if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){

            app.Service_Amount__c = 10000;

         }

         else if(app.Maintenance_service__c == true && app.Repairs__c == true){

            app.Service_Amount__c = 5000;

         }

         else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){

            app.Service_Amount__c = 8000;

         }

         else if(app.Repairs__c == true && app.Replacement_Parts__c == true){

            app.Service_Amount__c = 7000;

         }

         else if(app.Maintenance_service__c == true){
```

```
            app.Service_Amount__c = 2000;
        }
        else if(app.Repairs__c == true){
            app.Service_Amount__c = 3000;
        }
        else if(app.Replacement_Parts__c == true){
            app.Service_Amount__c = 5000;
        }

    }
    }
}
```

Trigger Handler :

How to create a new trigger :

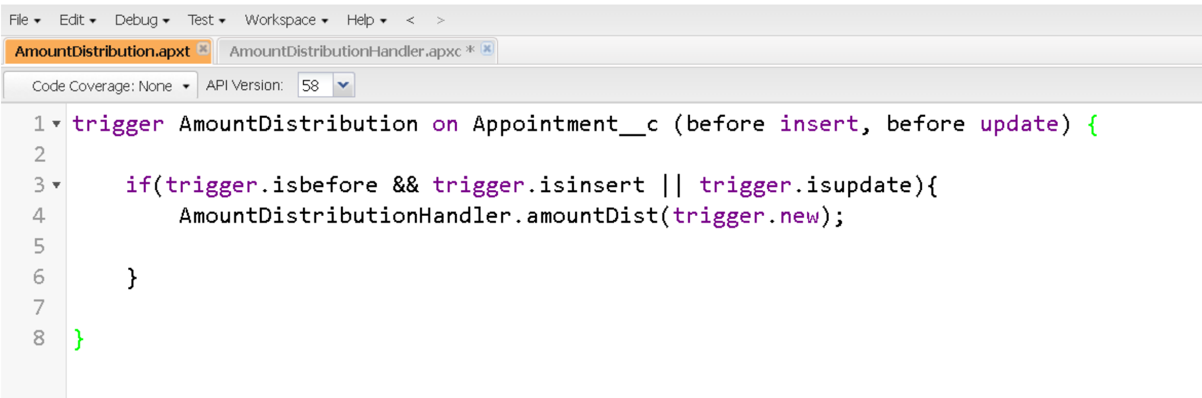
While still in the trailhead account, navigate to the gear icon in the top right corner.

Click on developer console and you will be navigated to a new console window.

Click on File menu in the tool bar, and click on new? Trigger.

Enter the trigger name and the object to be triggered.

Name : AmountDistribution
sObject : Appointment__c



CODE:

```
trigger AmountDistribution on Appointment__c (before insert, before update)
 {
 if(trigger.isbefore && trigger.isinsert || trigger.isupdate){

        AmountDistributionHandler.amountDist(trigger.new);
 }
}
```

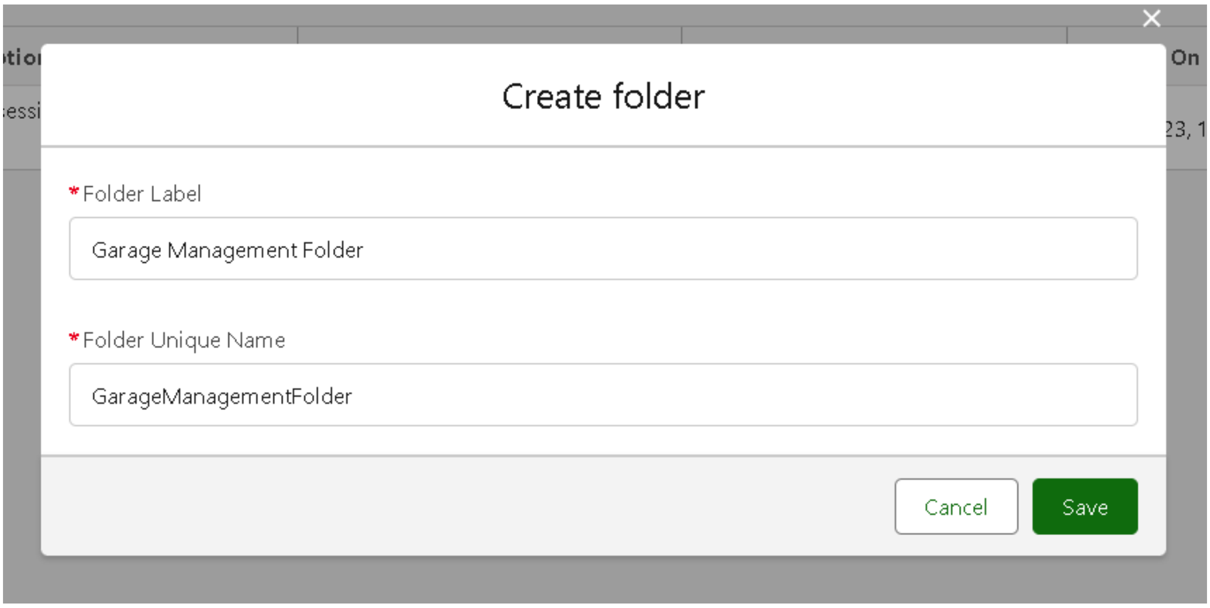
14.CREATE REPORT:

Click on the app launcher and search for reports.

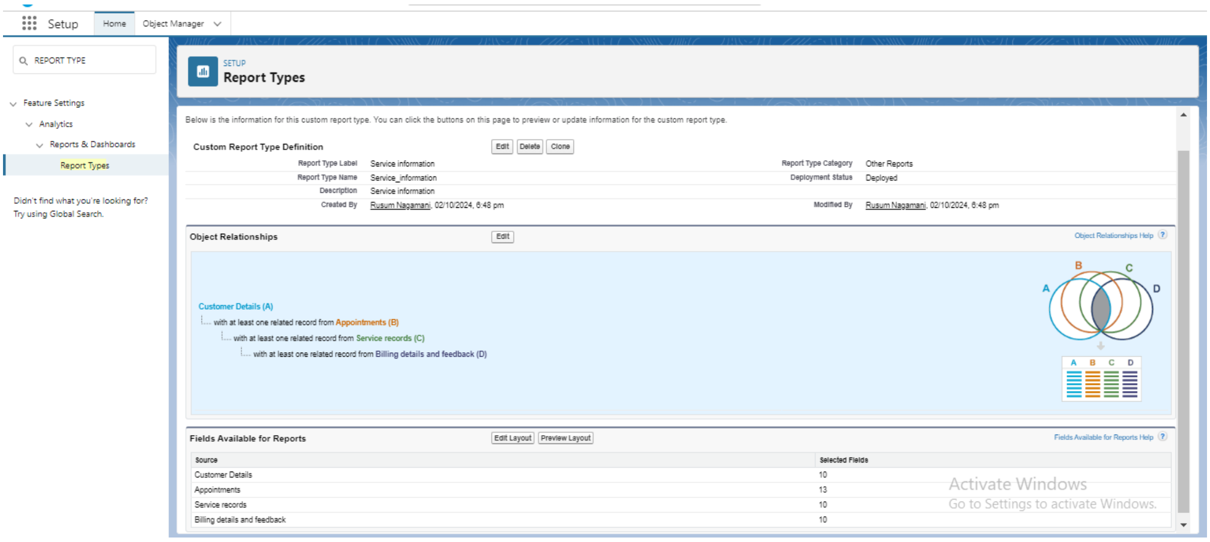
Click on the report tab, click on new folder

Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.

Click save.



RESULT OF SHARING REPORTS:



15.DASHBOARDS:

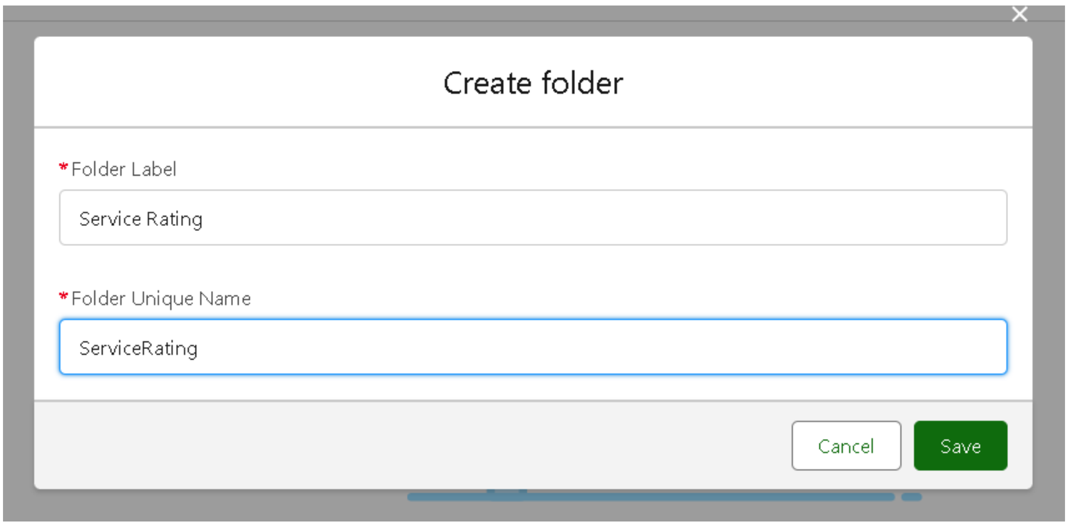
Click on the app launcher and search for dashboard.

Click on dashboard tab.

Click new folder, give the folder label as " Service Ratingg dashboard".

Folder unique name will be auto populated.

Click save



Go to the app >> click on the Dashboards tabs.

Give a Name and select the folder that created, and click on create.

Select add component.

Select the Line Chart. Change the theme.

Click Add then click on Save and then click on Done.

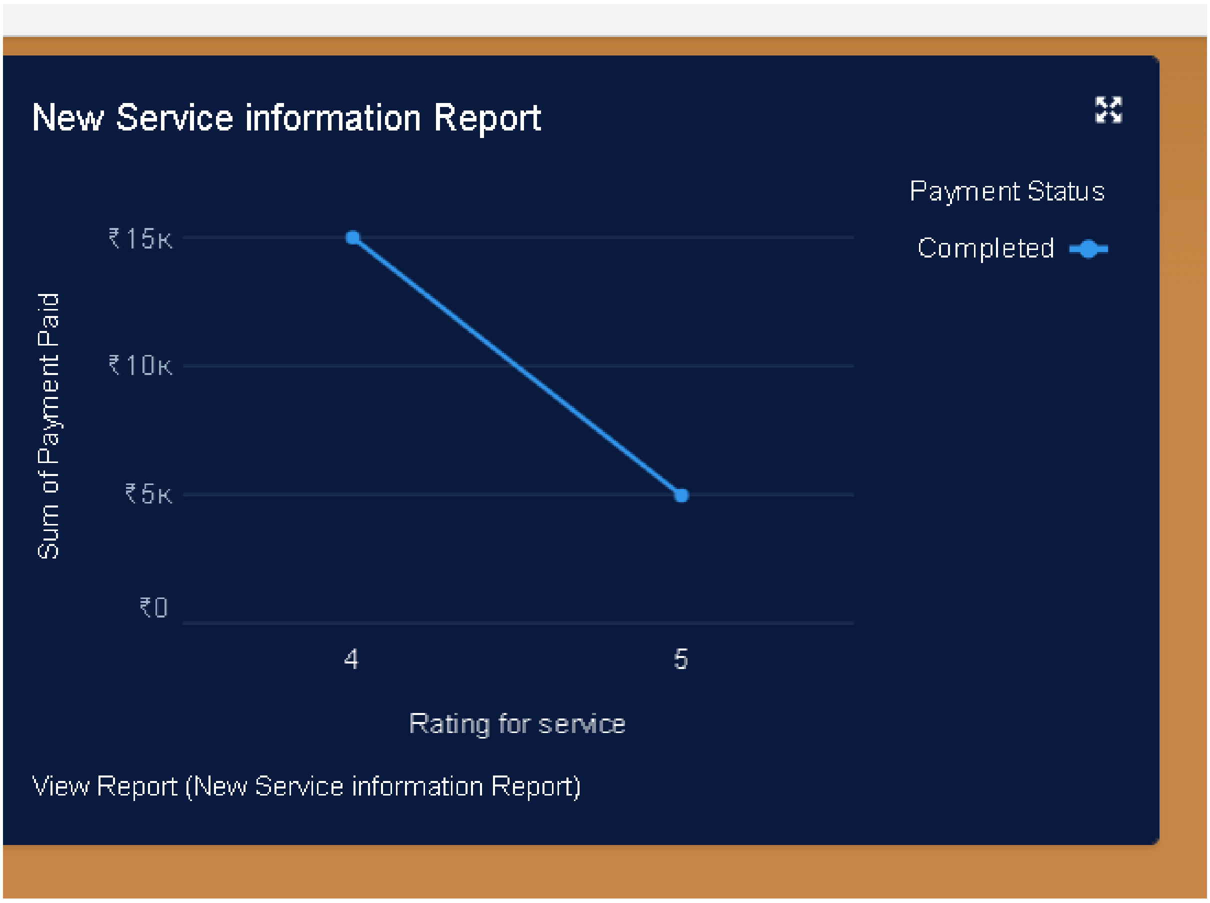
Preview is shown below.

After that Click on Subcribe on top right.

Set the Frequency as " weekly ".

Set a day as monday.

And Click on save.



THANKYOU