**Project Title:**

**GARAGE MANAGEMENT SYSTEM**

College Name: **MAILAM ENGINEERING COLLEGE**

Code: **4216**

**Department Of Computer Science And Engineering**

**Team Leader:**

Name: VIJAYASREE G Reg No: 421622104158

**Team Members:**

|  |  |
| --- | --- |
| Name: SNEHA S | Reg No: 421622104135 |
| Name: SIVASAKTHI K R | Reg No: 421622104134 |
| Name: SUSHMITHA M | Reg No: 421622104138 |

Registered Email Id: vijayashree2005@gmail.com TrailheadUrl: https://ddl00000fey4ruab-dev-ed.develop.lightning.force.com/lightning/setup/SetupOneHome/home

# Project Name: GARAGE MANAGEMENT SYSTEM

**Project overview**:

A **Garage Management System (GMS)** is a modern, integrated solution tailored for automotive repair facilities. It is designed to streamline and optimize daily operations, from customer interactions to back-office management. By leveraging technology, GMS enhances service delivery, boosts operational efficiency, and fosters strong customer relationships.

**Key Benefits of GMS**

* **Efficiency**: Automates routine tasks, such as scheduling and invoicing, allowing staff to focus on delivering quality services.
* **Customer Satisfaction**: Maintains detailed records, enabling personalized communication and timely service reminders.
* **Cost Management**: Tracks inventory and expenses, reducing waste and avoiding stockouts.
* **Data-Driven Decisions**: Provides insightful analytics to identify trends, improve services, and enhance profitability.

**Objectives:**

The primary aim of a Garage Management System is to optimize the operations of automotive repair facilities while enhancing customer satisfaction and profitability. Below are the key objectives of implementing a GMS:

**Business Goals of a Garage Management System (GMS)**

The implementation of a **Garage Management System (GMS)** supports the overarching business goals of an automotive repair facility, enabling sustainable growth, improved efficiency, and enhanced customer relationships. Below are the primary business goals aligned with adopting a GMS:

**1. Increase Revenue**

* Boost sales through better service management and upselling opportunities.
* Retain customers by offering timely reminders for maintenance and promotions.
* Reduce revenue loss by minimizing billing errors and managing overdue payments.

**2. Optimize Operational Efficiency**

* Reduce turnaround time by automating routine tasks and streamlining workflows.
* Minimize waste in labor and resources through effective scheduling and inventory control.
* Improve technician productivity by assigning tasks efficiently.

**3. Enhance Customer Loyalty**

* Build trust with transparent billing and service processes.
* Provide excellent service by maintaining detailed vehicle histories and preferences.
* Deliver a seamless customer experience through digital communication and online booking.

**4. Control Costs**

* Prevent overstocking or understocking with precise inventory management.
* Reduce operational costs by minimizing human errors and inefficiencies.
* Improve profitability by identifying cost-effective services and products.

**5. Support Business Growth**

* Scale operations to manage a larger customer base or multiple locations seamlessly.
* Introduce new services or packages based on customer demand and market trends.
* Stay competitive by adopting cutting-edge technology and industry best practices.

**6. Foster Data-Driven Decision-Making**

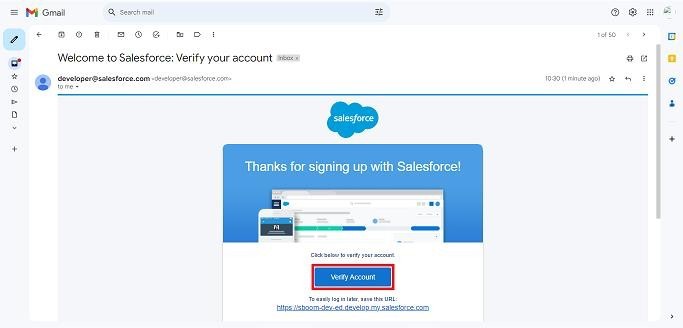
* Utilize reports and analytics to identify growth opportunities and performance gaps.
* Make informed decisions about investments, hiring, or marketing strategies.
* Stay updated with customer trends to innovate and adapt to changing market demands.

**Detailed steps to solution design**

**Step 1: Creating Developer Account and Activation** Creating a developer org in salesforce.

To Activate the account, click on the verify account. Give a password and answer a security question and click on change password.

Give a password and answer a security question and click on change password. Then you will redirect to your salesforce setup page.

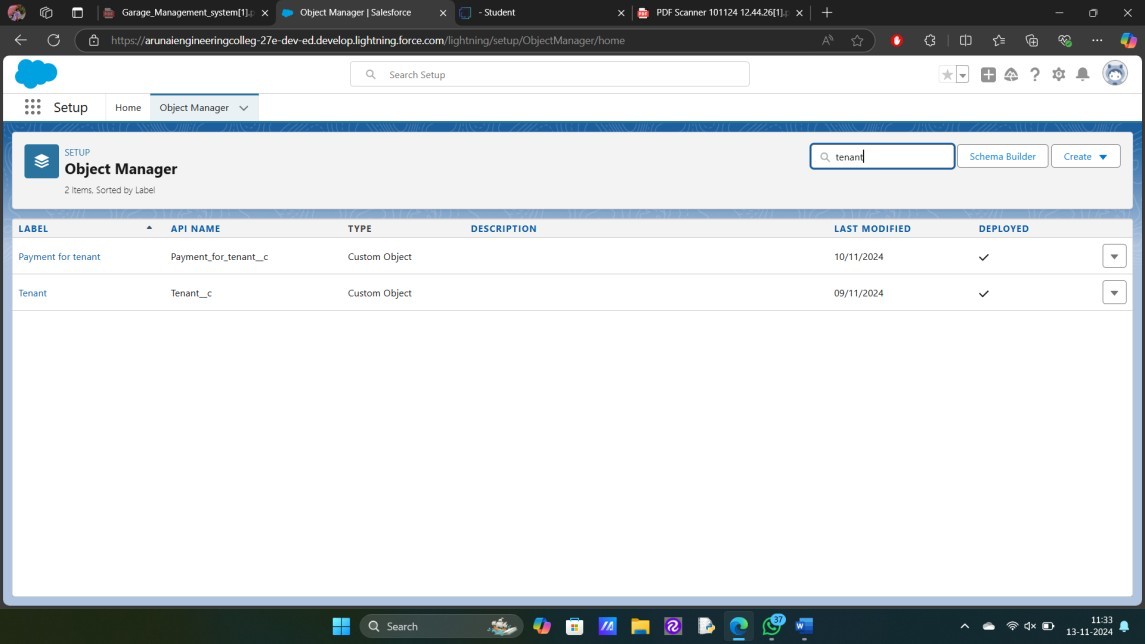




**Step 2: Create of objects**

Create Property Object, tenant object, lease object and payment object.

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

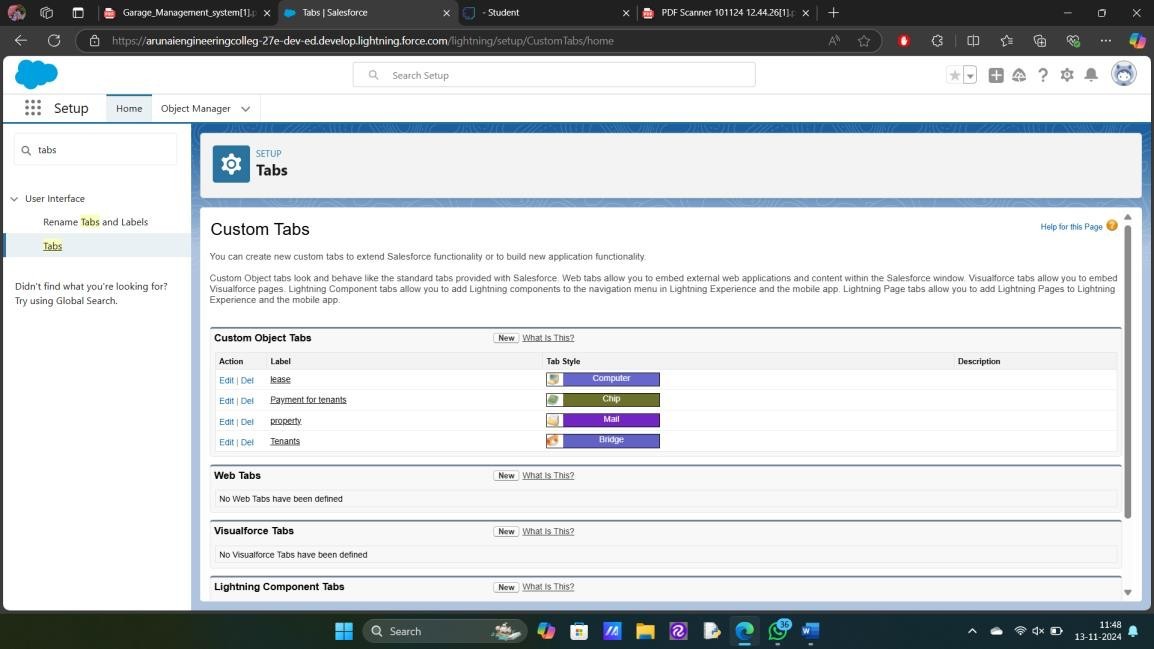


Enter the label name for the appropriate object. Then enter the Plural label name.

Enter Record Name Label and Format Record Name and Data Type.

**Step 3: Creating a Custom Tab**

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



**Step 4: Create a Lightning App**

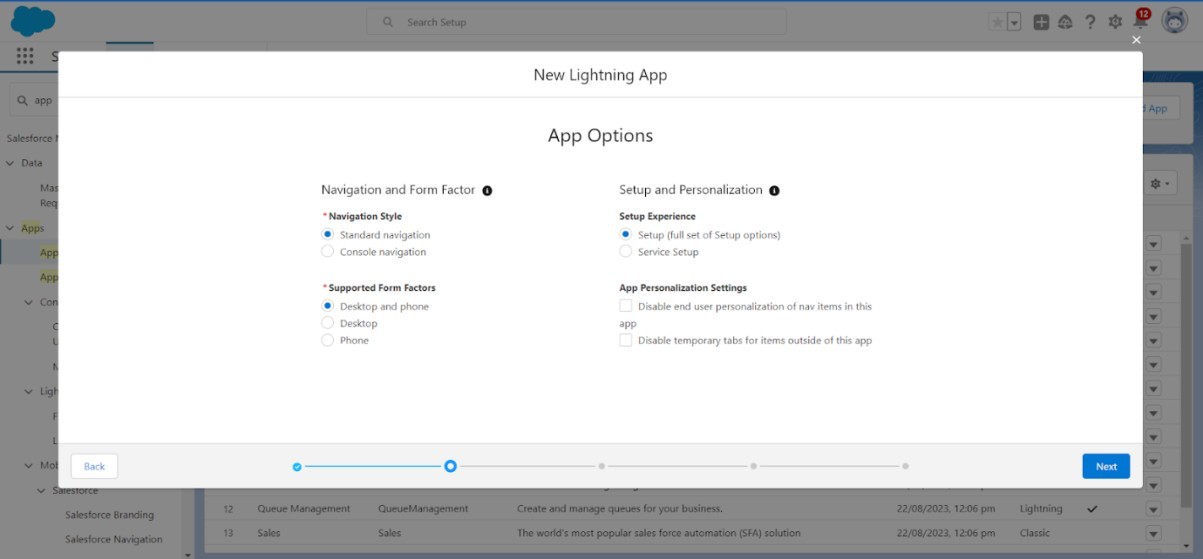
An app is a collection of items that work together to serve a particular function.

**To create a Tab:( Property)**

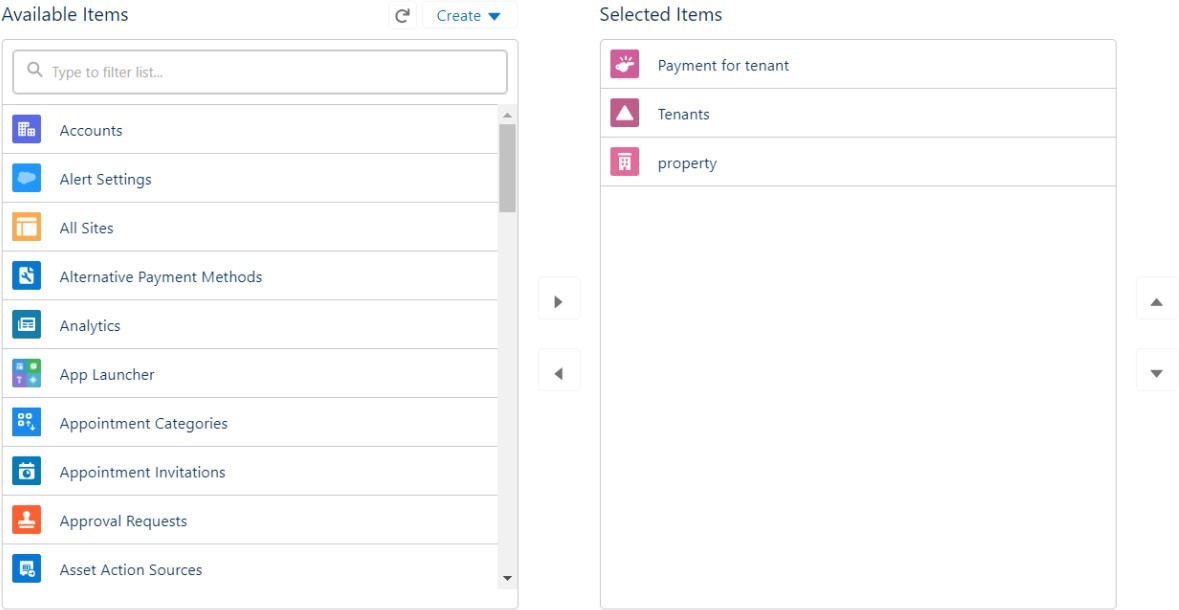
Goto set up page. Search tabs on quick find bar click new under custom tab.

The first step is to select object property and tab style.

Make sure that the Append tab to users' existing personal customizations is checked. Click save



Create a lightning app with an App Name of “garage management system”



**Step 5: Field Creation**

Creation of fields for the customer details , lookup ,checkbox , date ,currency,

Text ,picklist and formula field in service records object.



Go to setup then click on Object Manager to type object name(Customer Details) in search bar and click on the object.

Field Label :Gmail

Field Name : gets auto generated

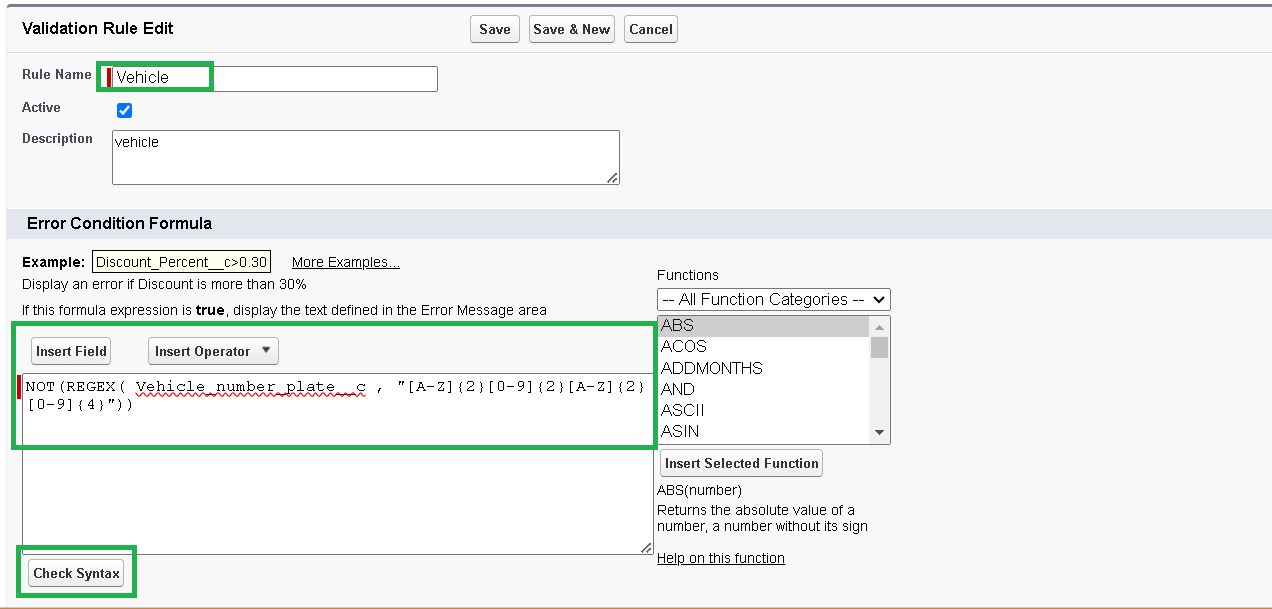
Click on required check box

Click on Next >> Next>> Save and new.

Field Label :Phone number

Field Name : gets auto generated

Click on Next >> Next >> Save and new.



**Step 6: Validation rules**

To create a validation rule to an Appointment object.

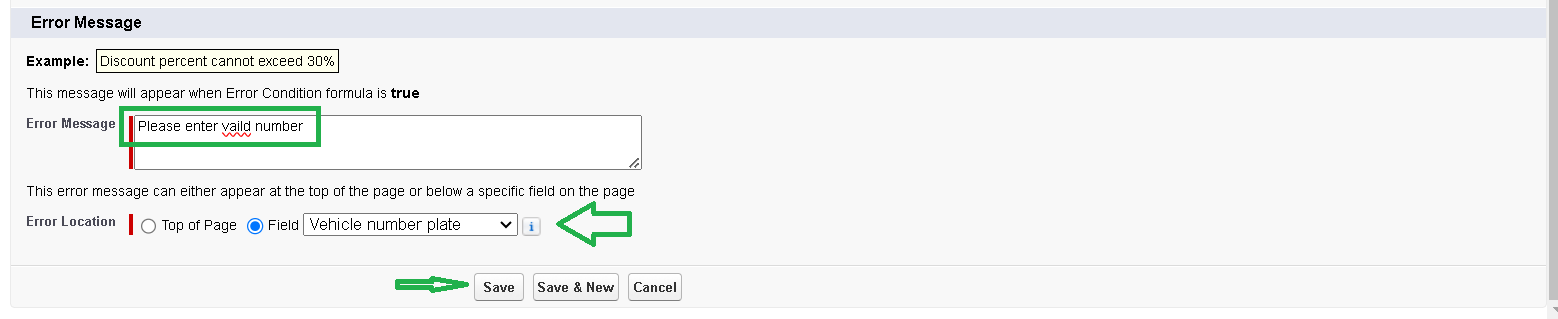
Go to the setup page click on object manager from drop down click edit for Lease object**.**

Click on the validation rule.click new

Rule name= Vehicle

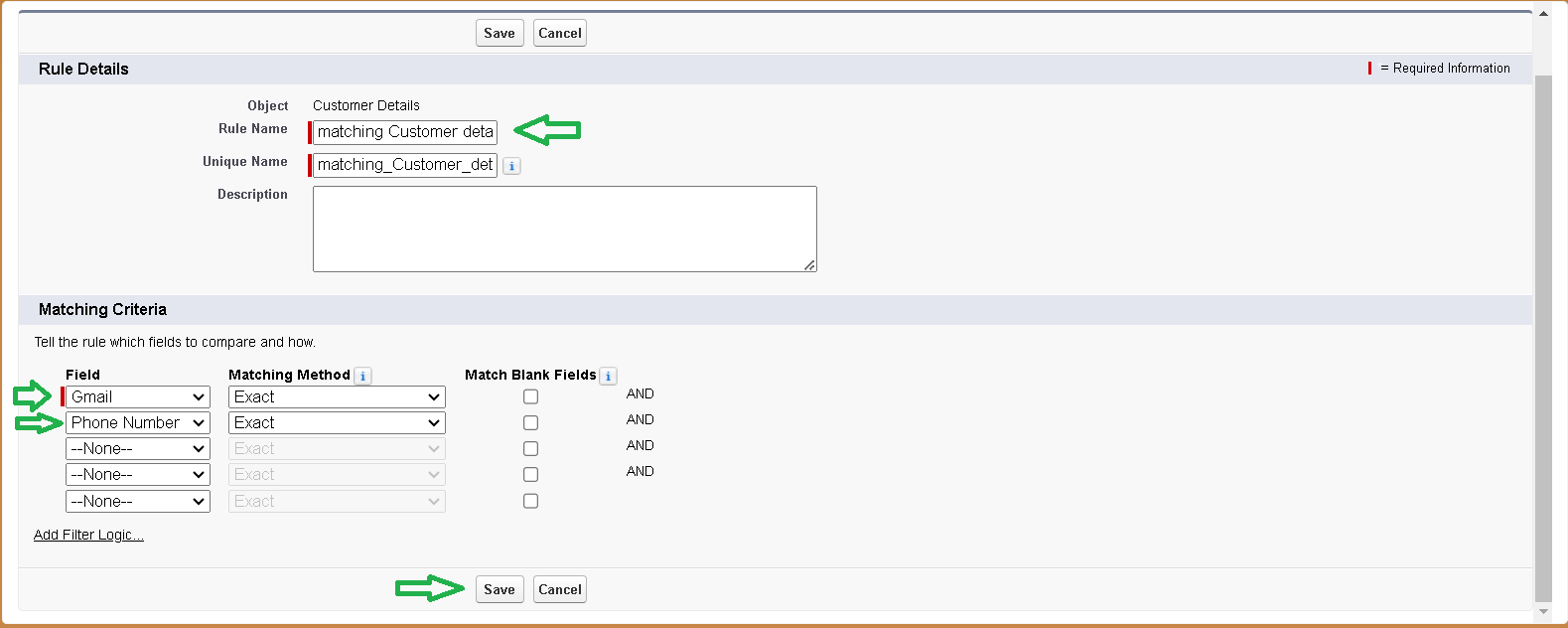
 •Insert the Error Condition Formula as :NOT(REGEX( Vehicle\_number\_plate\_\_c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))

Enter the Error Message as “Please enter vaild number”, select the Error location as Field and select the field as “Vehicle number plate”and click Save.

**Step 7: Duplicate rule**

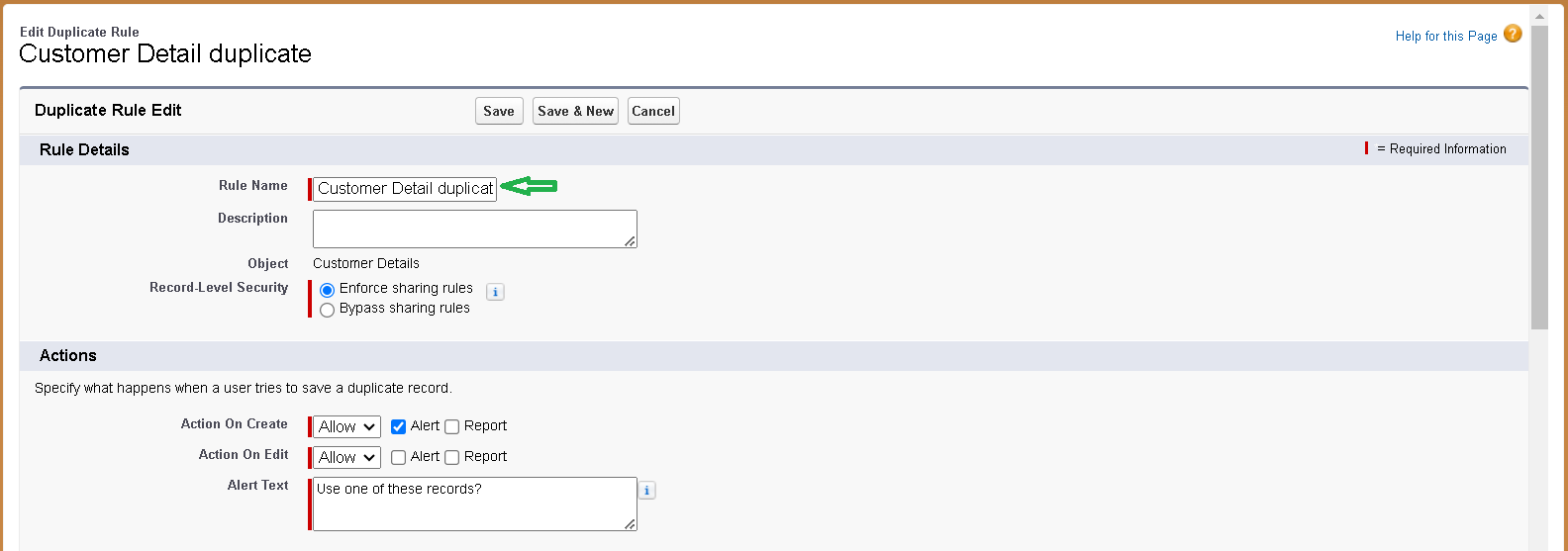
**1.To create a matching rule to an Customer details Object**

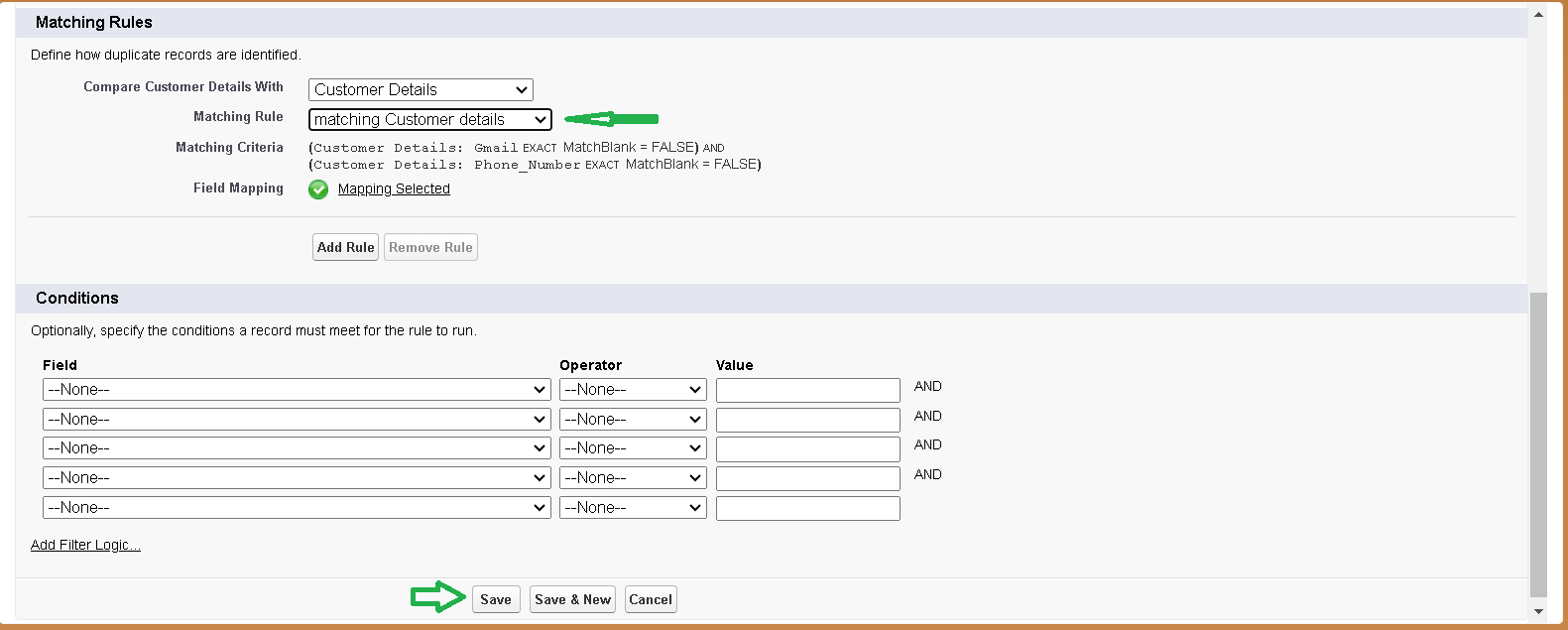
1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the object as customer details and click next.
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field                                               Matching Method
8. 1. Gmail                                                       Exact
9. 2. Phone Number                                         Exact
10. Click save.
11. After Saving Click on Activate.



**2.To create a Duplicate rule to an Customer details Object**

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >>  click on New Rule  >>  select customer details object.
3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.







### Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

**Types of profiles in salesforce**

1. **Standard profiles:**

By default salesforce provides below standard profiles.

* Contract Manager
* Read Only
* Marketing User
* Solutions Manager
* Standard User
* System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

1. **Custom Profiles:**

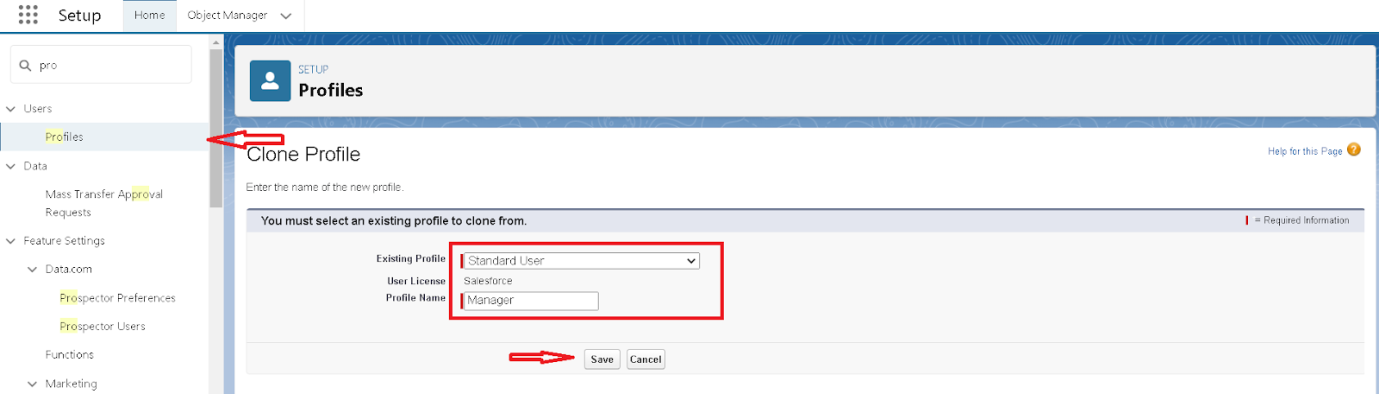
Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

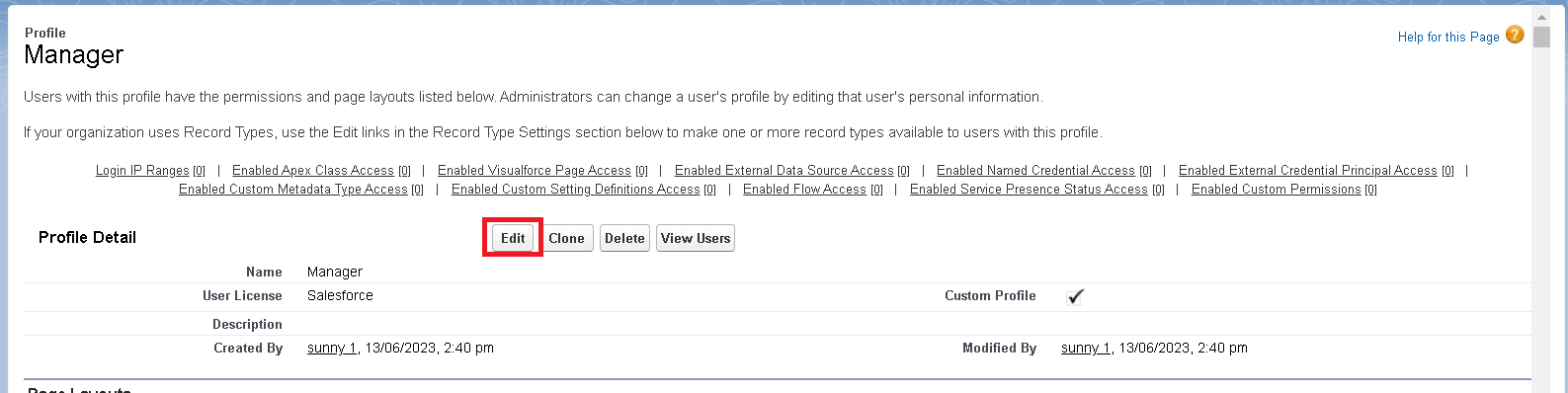
### Manager Profile

**To create a new profile:**

1. Go to setup >>  type profiles in quick find box  >>  click on profiles  >>  clone the desired profile (Standard User)  >>  enter profile name (Manager)  >>  Save.



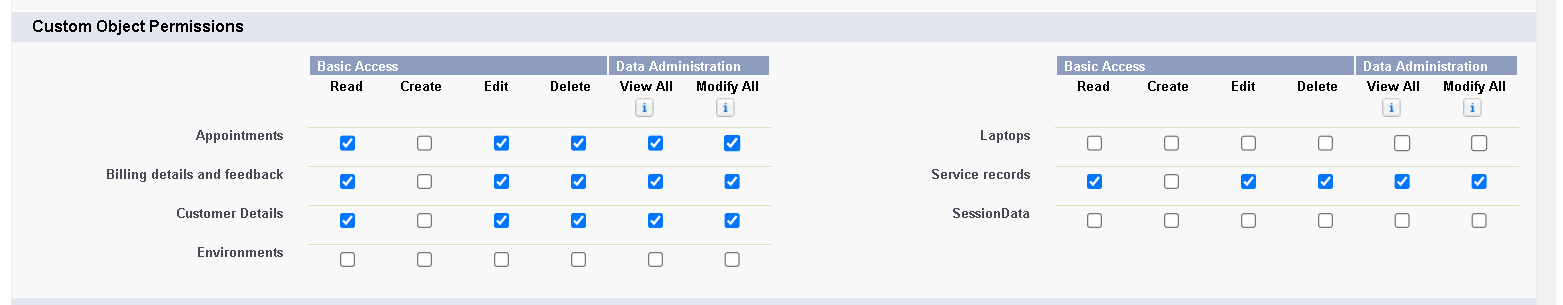
1. While still on the profile page, then click Edit.



1. Select the Custom App settings as default for the Garage management.



1. 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.



1. Changing the session times out after should be “ 8 hours of inactivity”.
2. Change the password policies as mentioned :
3. User passwords expire in should be “ never expires ”.
4. Minimum password length should be “ 8 ”, and click save.

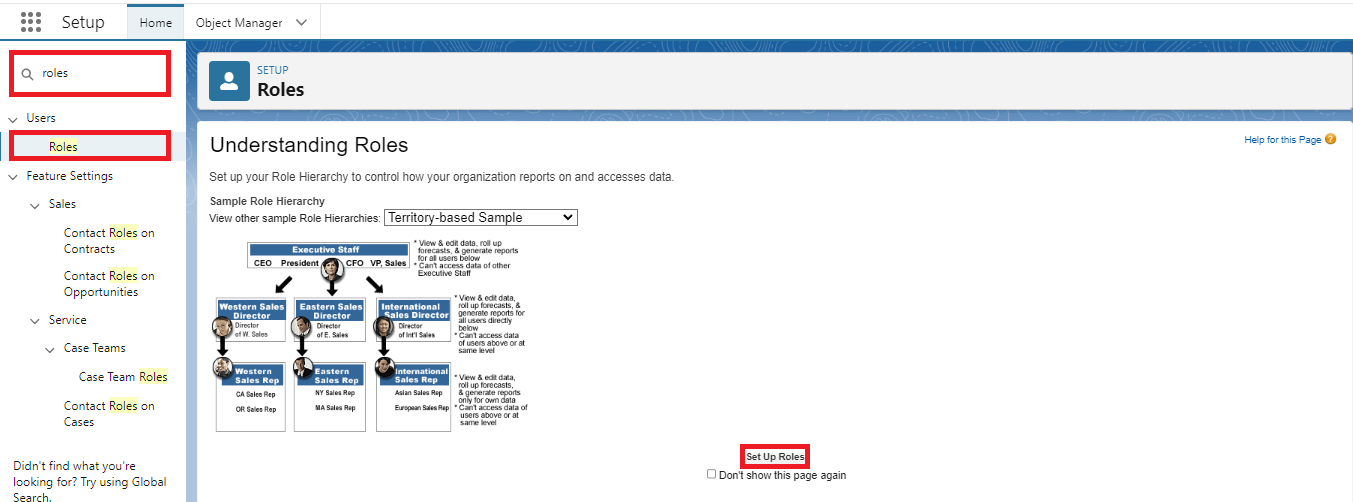
### Role & Role Hierarchy

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.

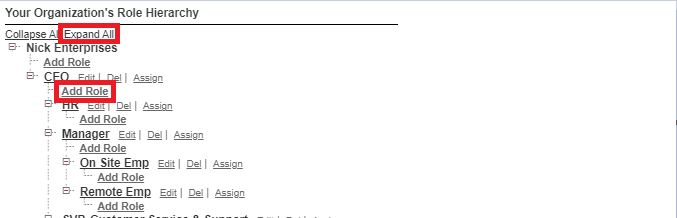
### Creating Manager role

Creating Manager Role:

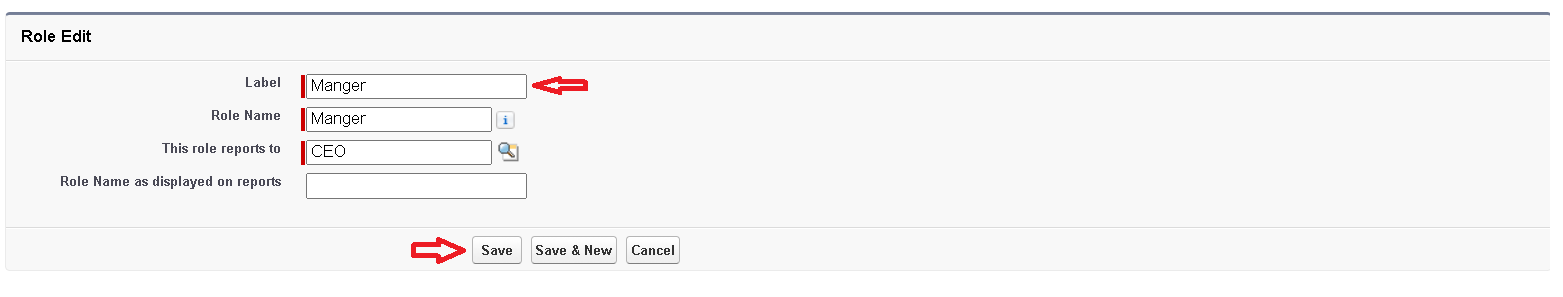
1. Go to quick find  >>  Search for Roles  >>  click on set up roles.



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



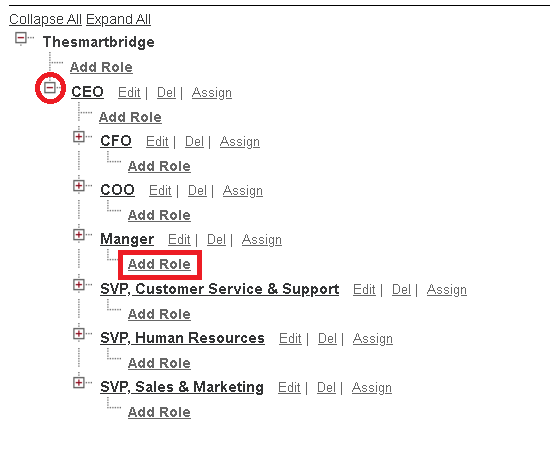
1. Give Label as “Manager” and Role name gets auto populated. Then click on Save.



### Creating another roles

Creating another two roles under manager

1. Go to quick find  >>  Search for Roles  >>  click on set up roles.
2. Click plus on CEO role, and click add role under manager.



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

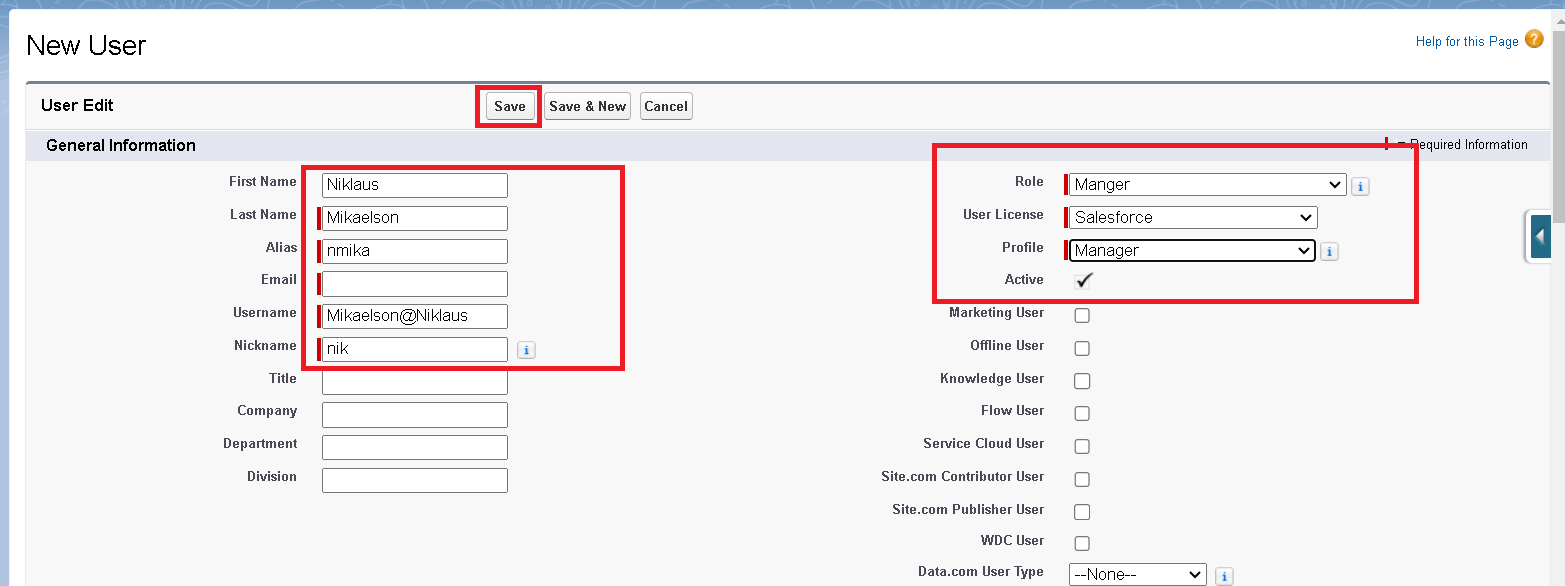
### Users

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.

### **Create User**

1. Go to setup  >>  type users in quick find box  >>  select users  >> click New user.
2. Fill in the fields
3. First Name : Niklaus
4. Last Name : Mikaelson
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Manager
10. User licence : Salesforce
11. Profiles : Manager



1. Save.

### Creating another users

1. Repeat the steps and create another user using
   1. Role                  : sales person
   2. User licence      : Salesforce Platform
   3. Profile               : sales person

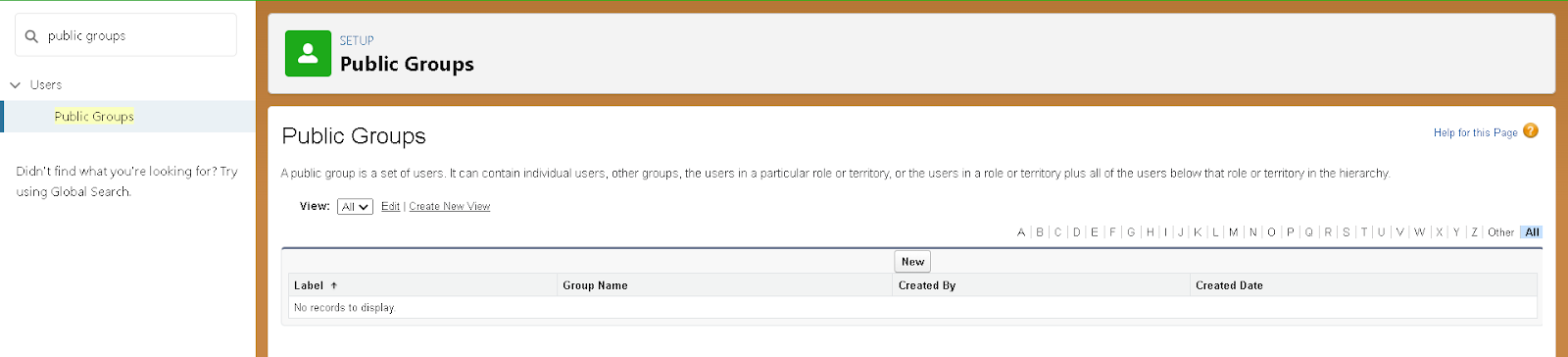
**Note : create atleast 3 users with these permissions.**

### Public groups

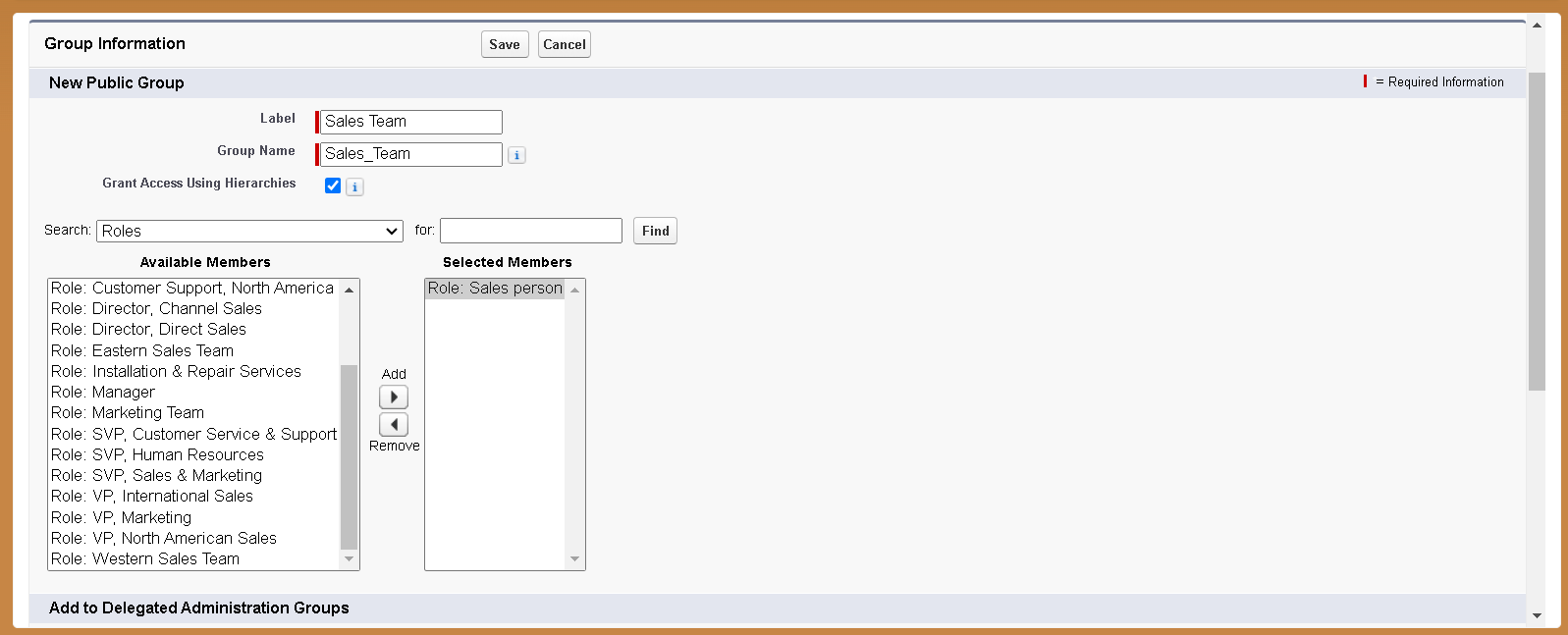
Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings. By creating and using public groups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have appropriate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

### Creating new public group

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



1. Give the Label as “sales team”.
2. Group name is autopopulated.
3. Search for Roles.
4. In Available Members select Sales person and click on add it will be moved to selected member.
5. Click on save.



### 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:

**Organization-Wide Default (OWD) Settings:**

These settings define the default level of access for all objects within your Salesforce org.

OWD settings include Private, Public Read-Only, Public Read/Write, and Controlled by Parent.

OWD settings can be configured for each standard and custom object.

**Role Hierarchy:**

Salesforce uses a role hierarchy to determine record access.

Users at higher levels in the hierarchy have greater access to records owned by or shared with users lower in the hierarchy.

The role hierarchy is often used in combination with OWD settings to grant different levels of access.

**Profiles and Permission Sets:**

Profiles and permission sets allow administrators to specify object-level and field-level permissions for users.

Profiles are typically used to grant general object and field access, while permission sets can be used to extend those permissions to specific users.

**Sharing Rules:**

Sharing rules are used to extend access to records for users who meet specific criteria.

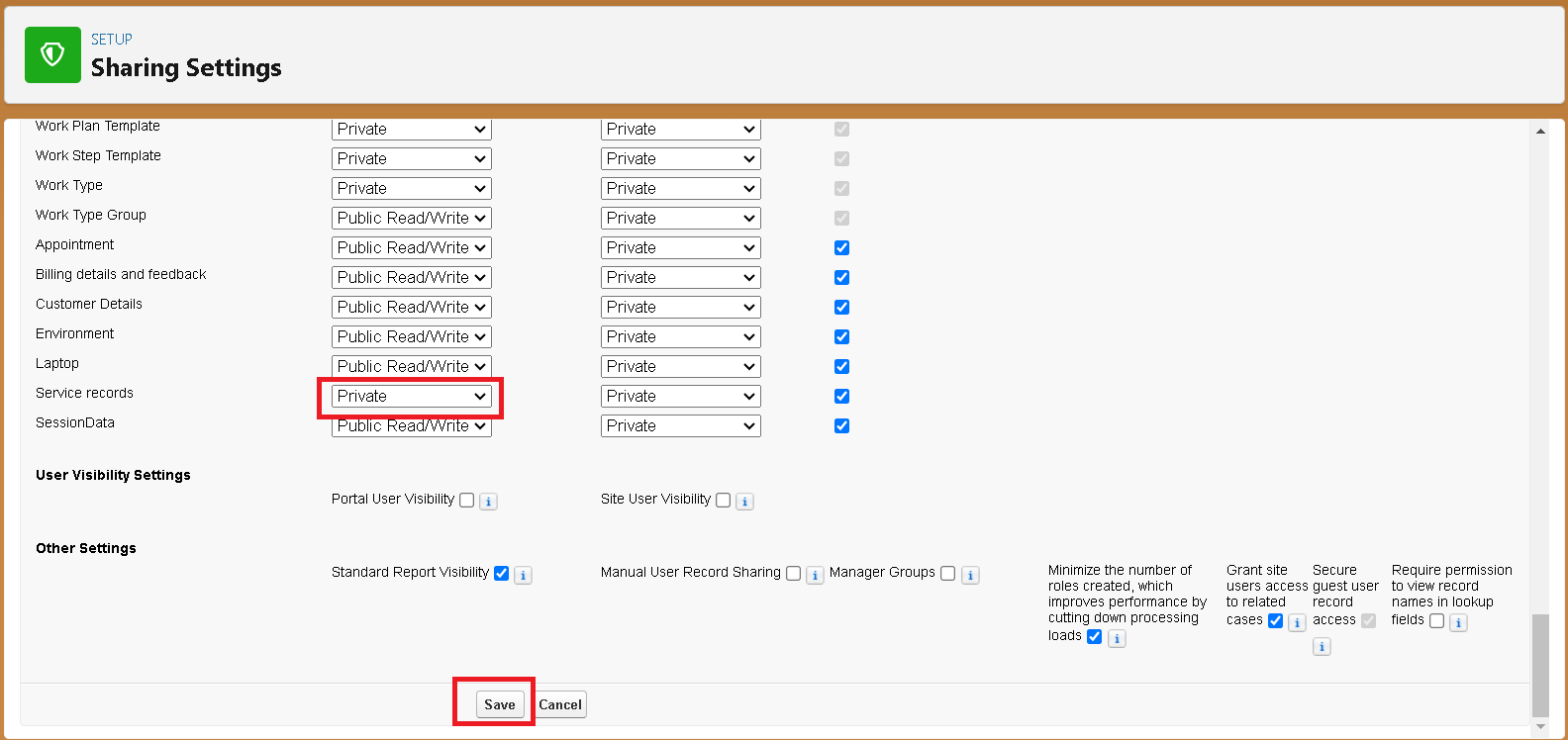
They can be used to grant read-only or read-write access to records owned by other users.

Manual Sharing:

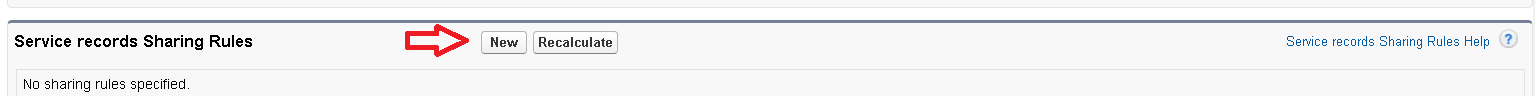
Administrators and record owners can manually share specific records with other users or groups.

### Creating sharing settings

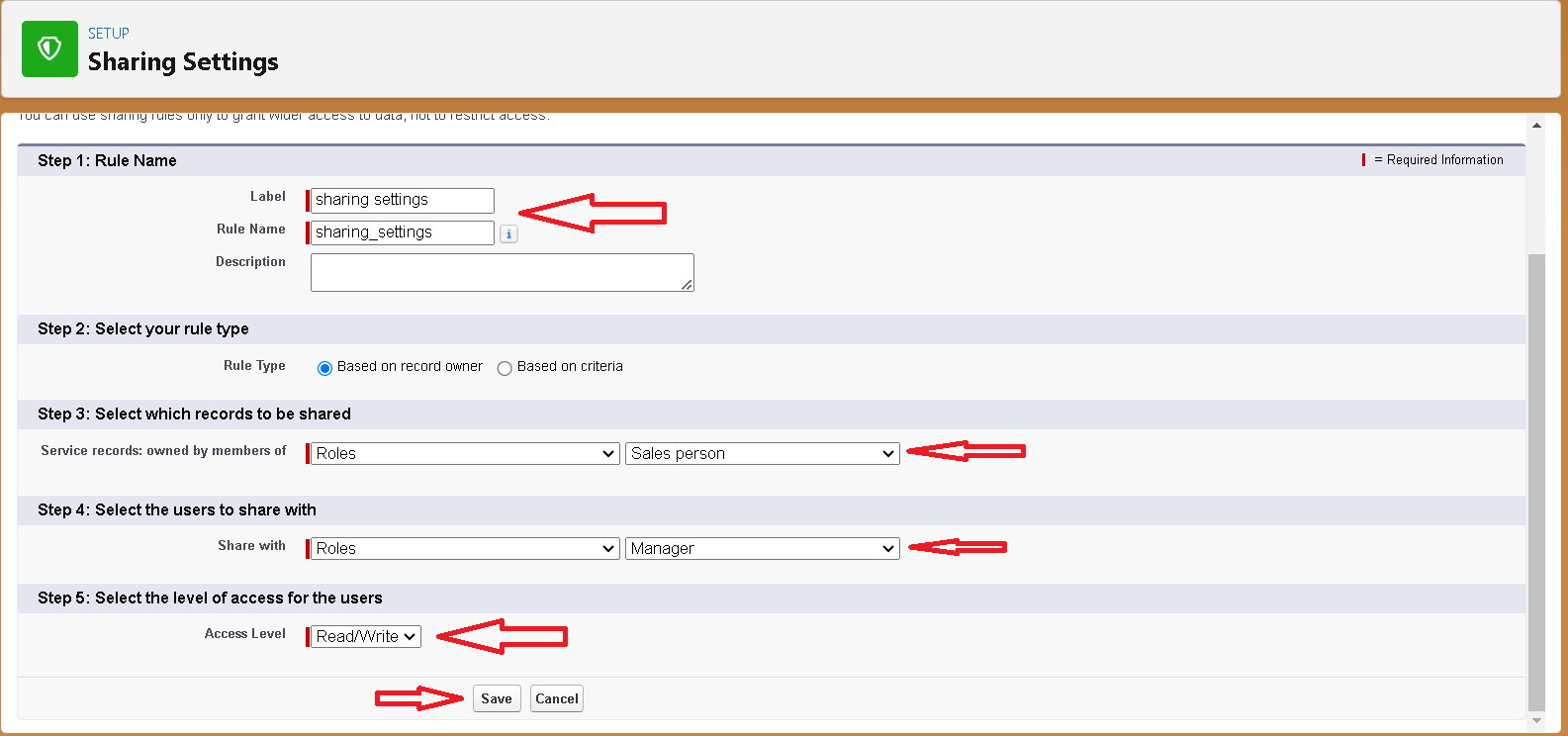
1. Go to setup  >>  type users in quick find box  >>  select Sharing Settings >>  click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.



1. Click on save and refresh.
2. Scroll down a bit, Click  new  on Service records sharing Rules.



1. Give the Label name as “ Sharing setting”
2. Rule name is auto populated.
3. In step 3 : Select which records to be shared, members of “ Roles ” >> “ Sales person”
4. In step 4: share with, select “ Roles ” >> “ Manager ”
5. In step 5 : Change the access level to “ Read / write ”.
6. Click on save.



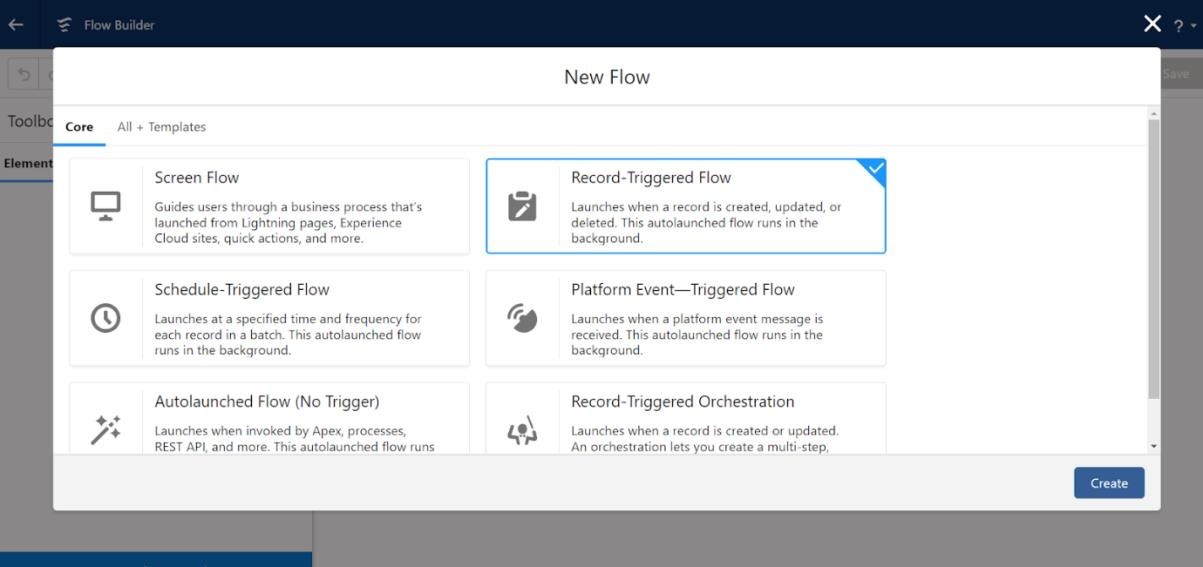
**Step 8: Flows**

In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data. Flow Builder is the declarative interface used to build individual flows.

**Create Flow for monthly payment**

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

Select the record Triggered flow. Click on create.

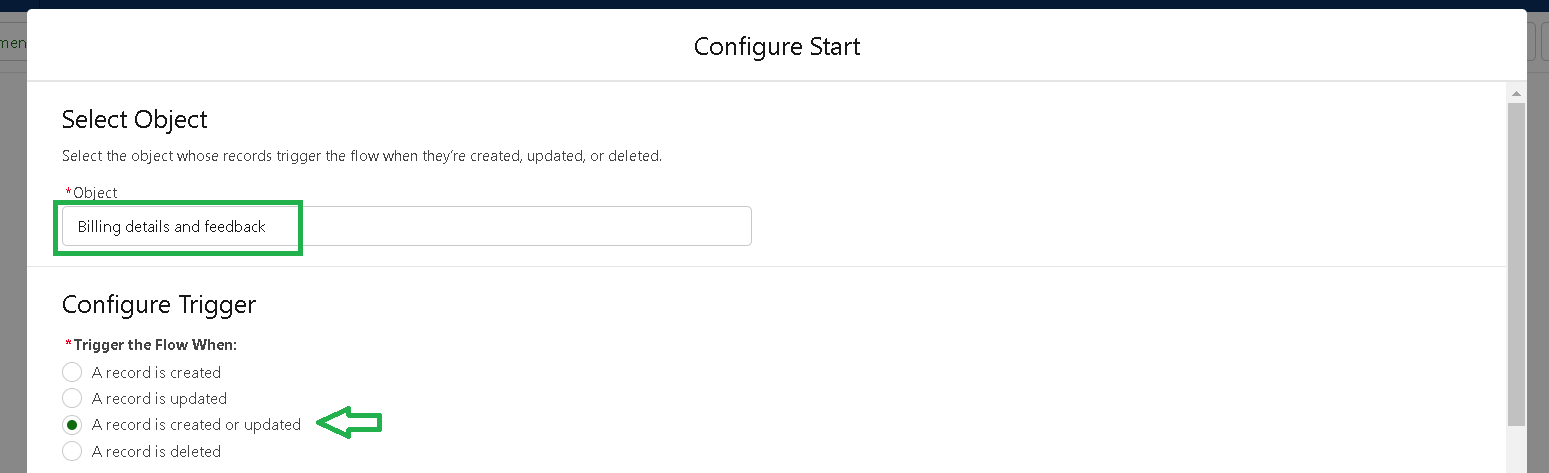


Click on: Every time a record is updated and meets the condition requirements Click on: Actions and related records is done

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.

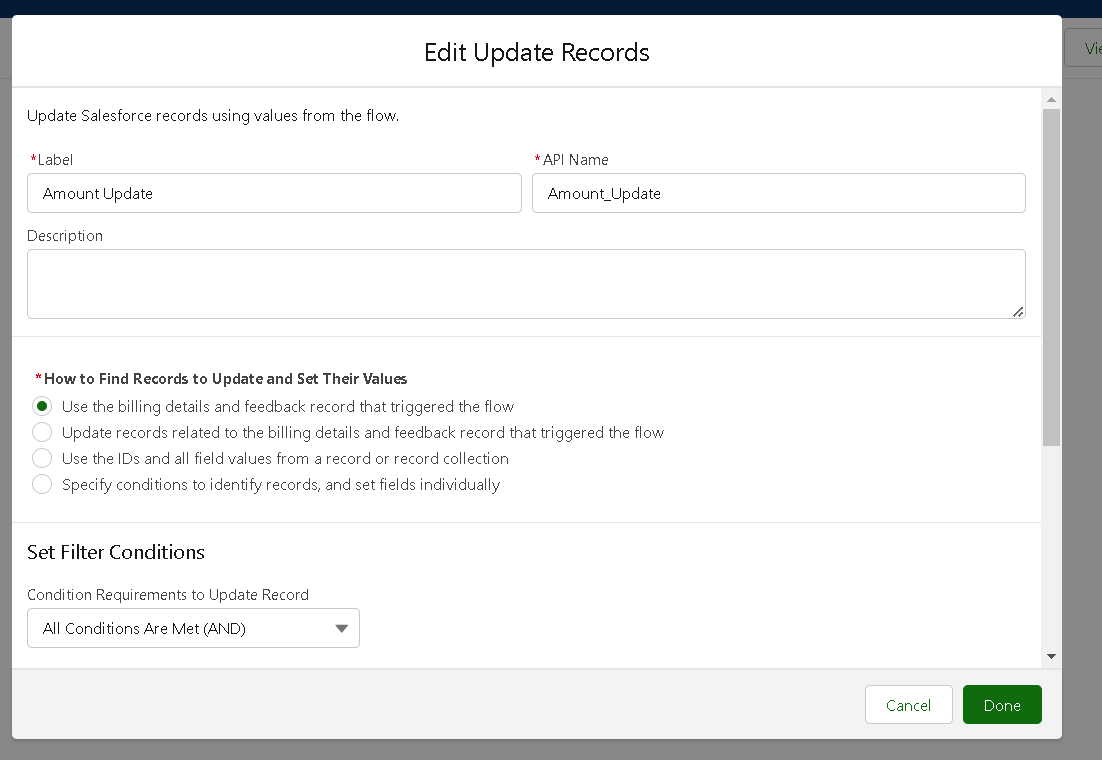


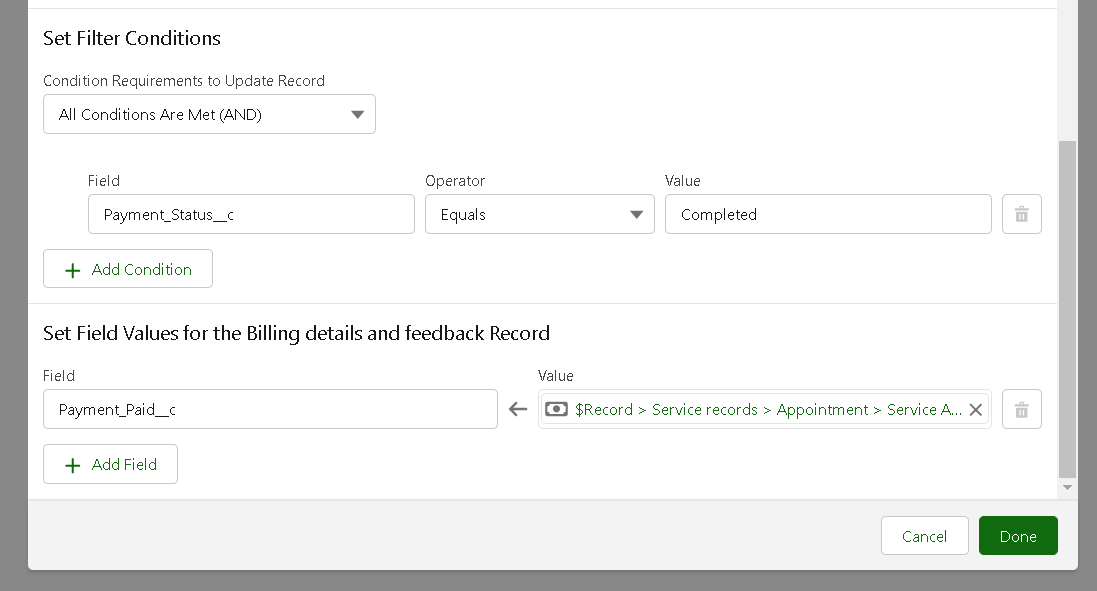


Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.

Give the Label Name : Amount Update

Api name : is auto populated





Set a filter condition : All Conditions are met(AND)

Field :Payment\_Status\_\_c

Operator : Equals

Value : Completed

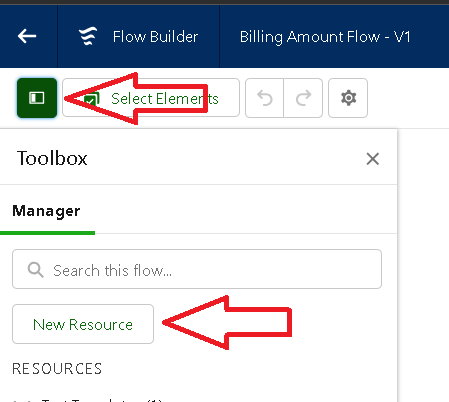
And Set Field Values for the Billing details and feedback Record

Field  :Payment\_Paid\_\_c

Value  : {!$Record.Service\_records\_\_r.Appointment\_\_r.Service\_Amount\_\_c}

Click On Done.

1. 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 template.

Enter the API name as “ alert”.

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

In body field  paste the syntax that given below.

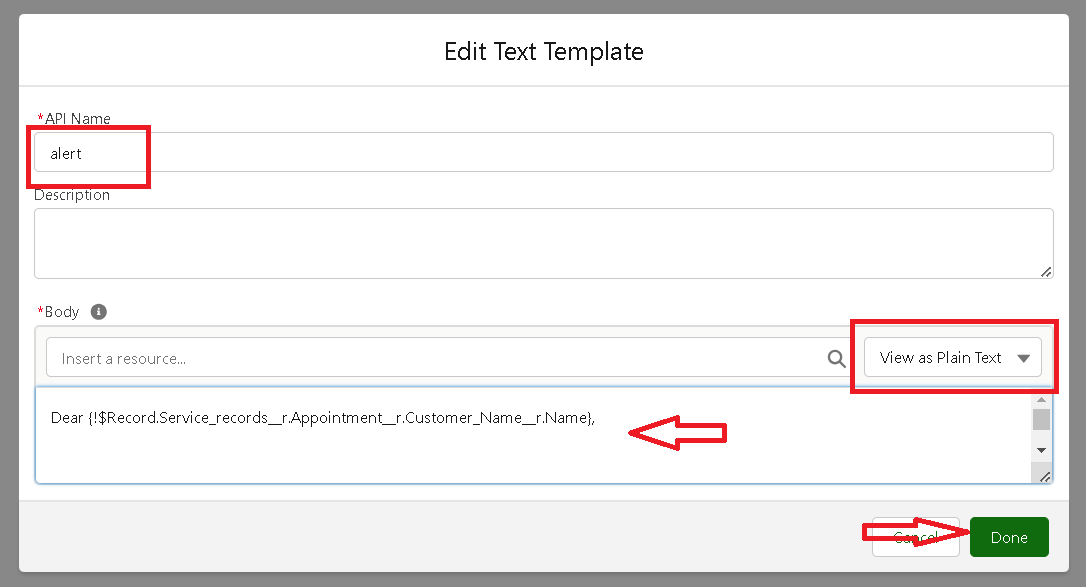
Dear {!$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Name},

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 isgreatly 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 .

Click done.



Now Click on Add Element , select Action.

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

Give the label name as “ Email Alert”

API name will be auto populated.

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

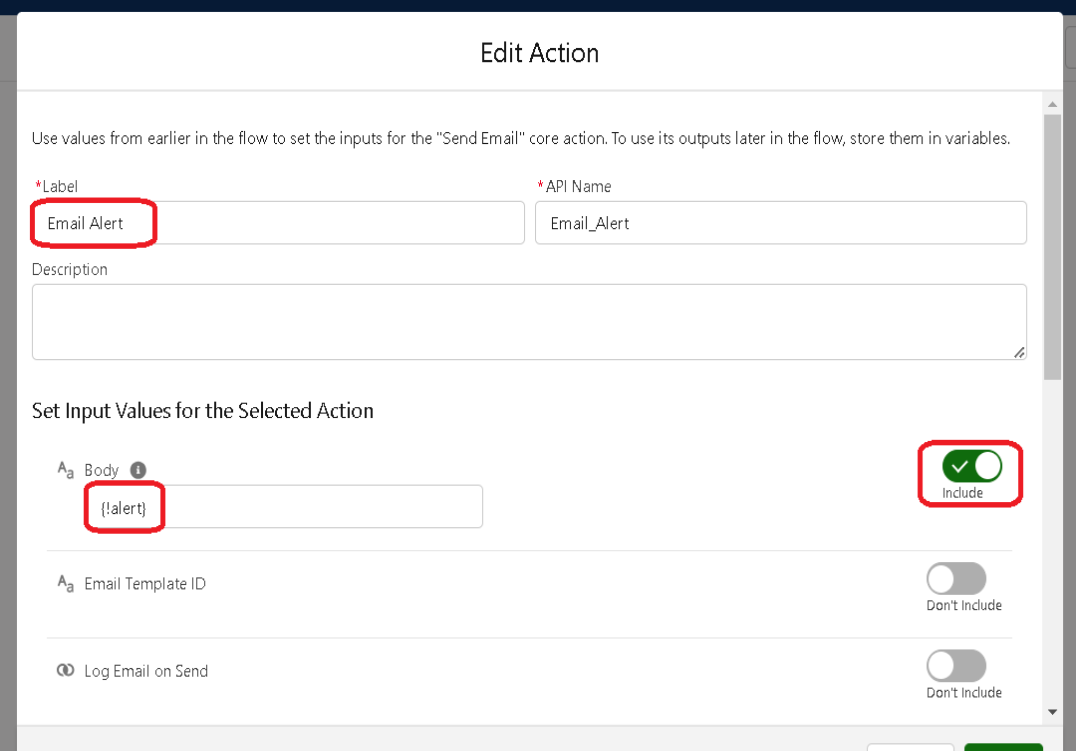
Select the text template that created , Body : {!alert}

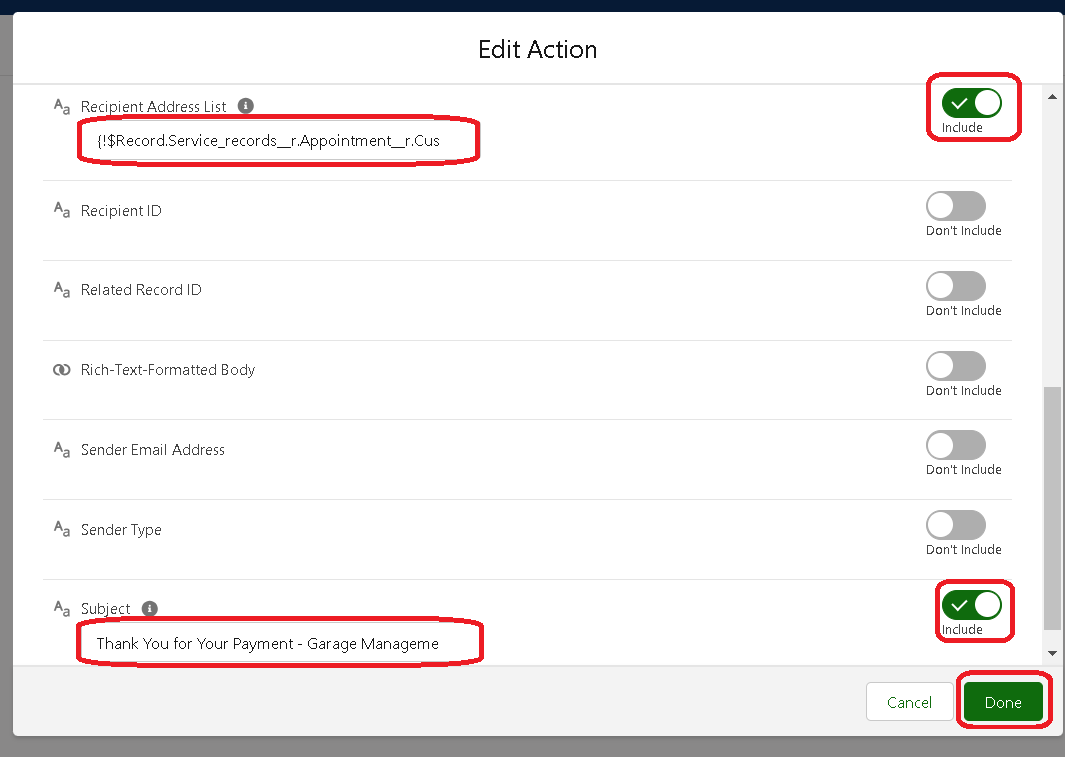
Include recipient address list select the email form the record.

RecipientAddressList:{!$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Gmail\_\_c}

Include subject as “ Thank You for Your Payment - Garage Management”.

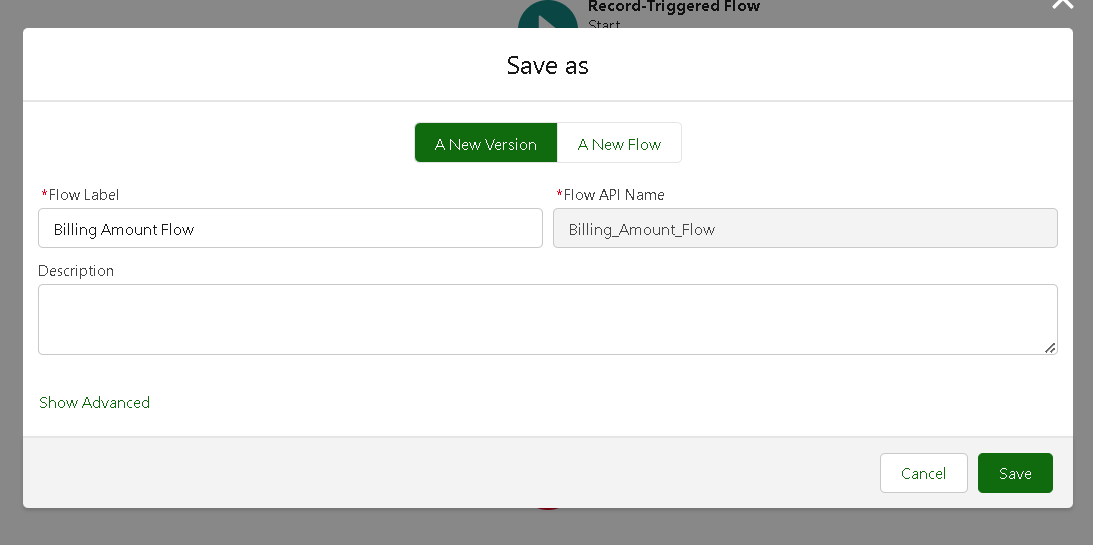
Click done.





Click on save. Give the Flow label , Flow Api name will be autopopulated.

And click save, and click on activate.



**Step 9: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.

A trigger is Apex code that executes before or after the following types of operations:

* insert
* update
* delete
* merge
* upsert
* undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

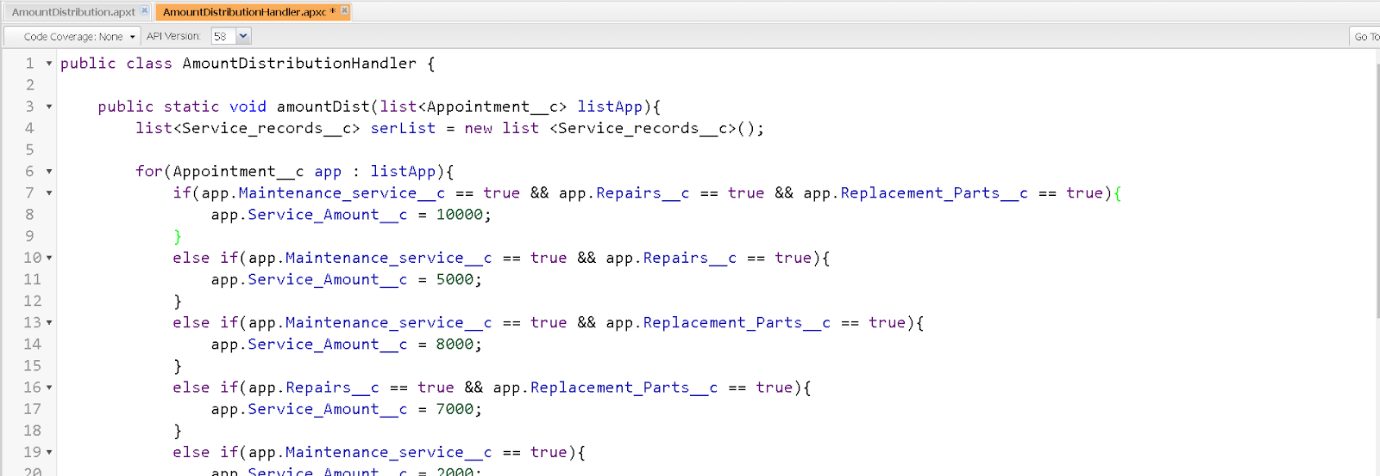
**Before Trigger:** This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

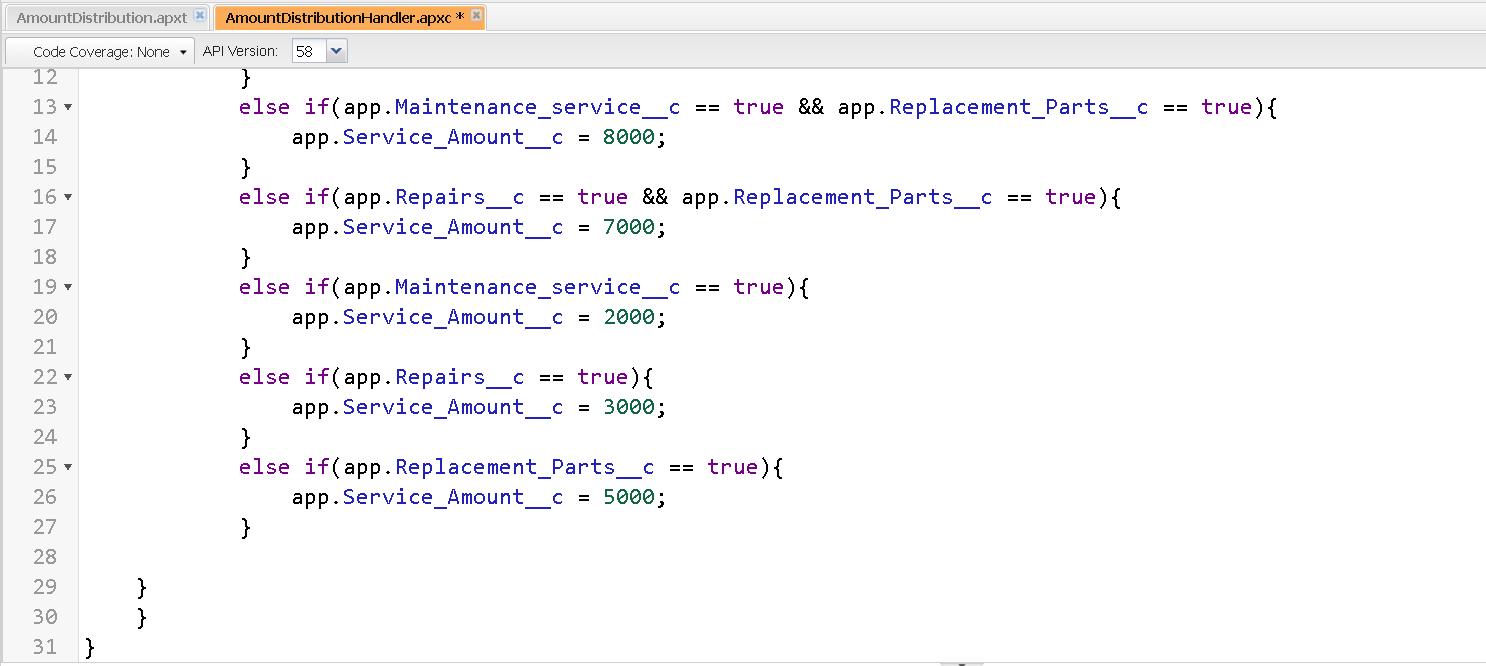
**After Trigger:**This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

**Apex Handler**

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler ”.





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;

            }

    }

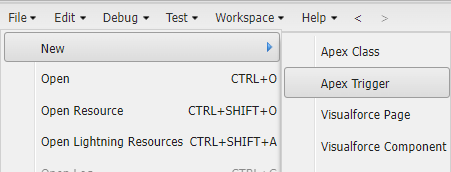
    }

}

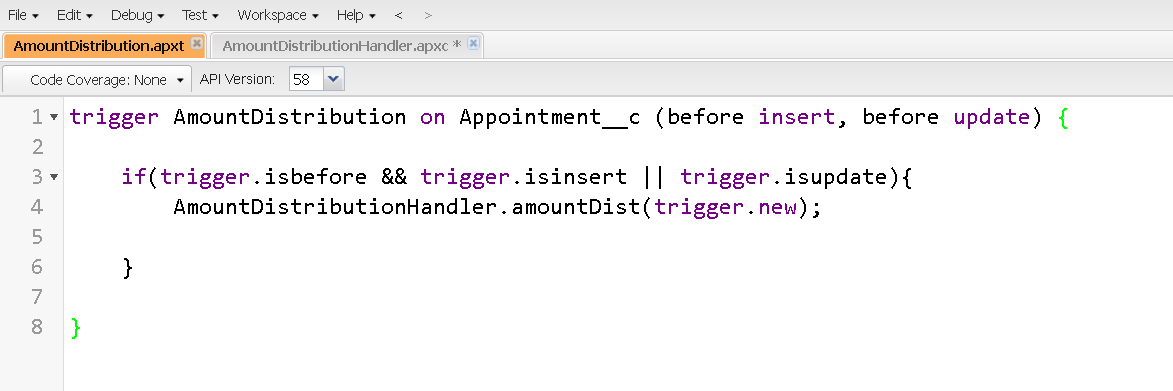
Trigger Handler :

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name  :AmountDistribution
6. sObject :Appointment\_\_c



1. Handler for the Appointment Object



Code:

trigger AmountDistribution on Appointment\_\_c (before insert, before update) {

    if(trigger.isbefore&&trigger.isinsert || trigger.isupdate){

        AmountDistributionHandler.amountDist(trigger.new);

    }

### Reports

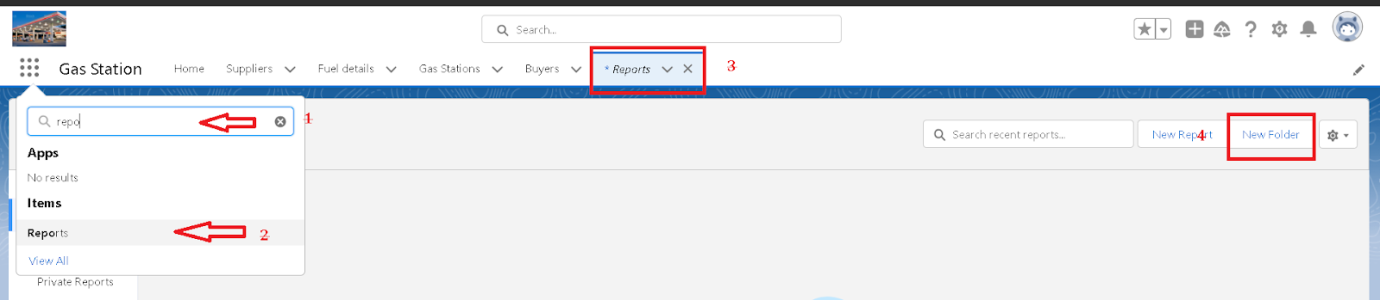
Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

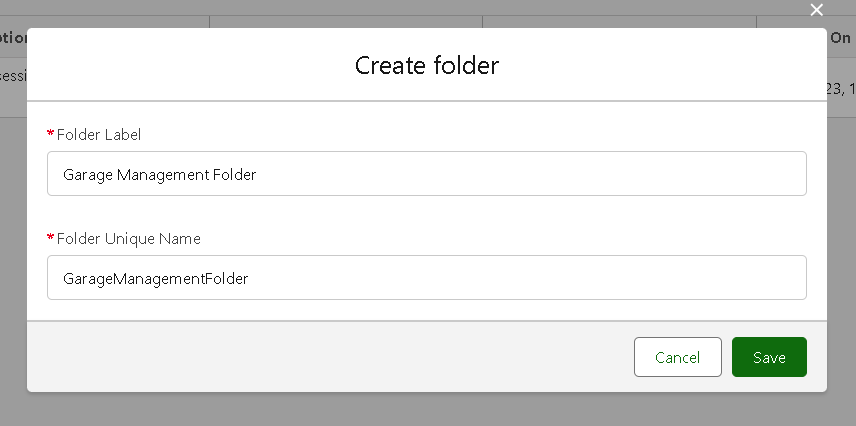
1. Tabular
2. Summary
3. Matrix
4. Joined Reports

### create a report folder

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.

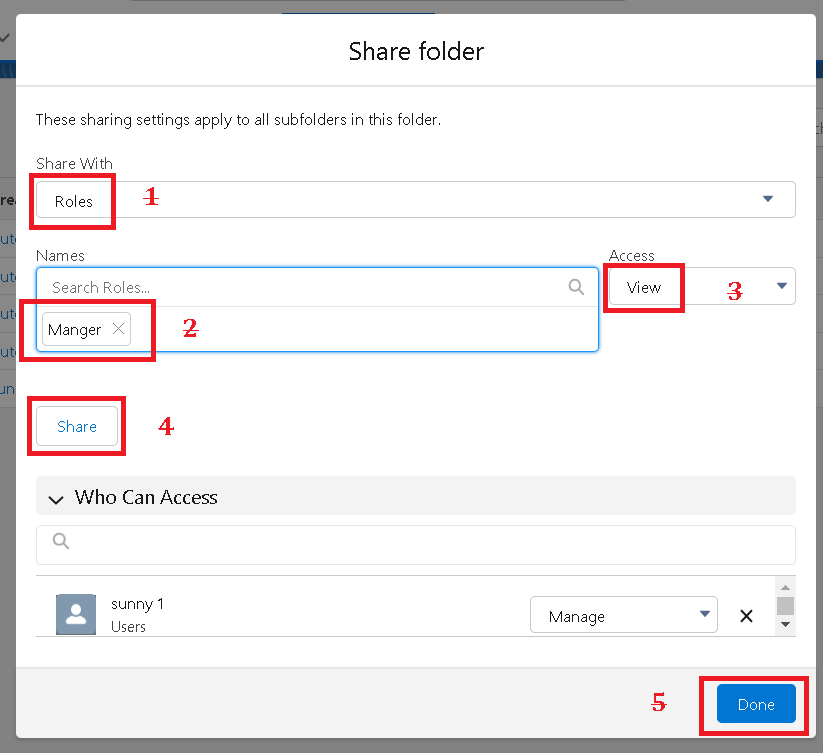


1. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
2. Click save.



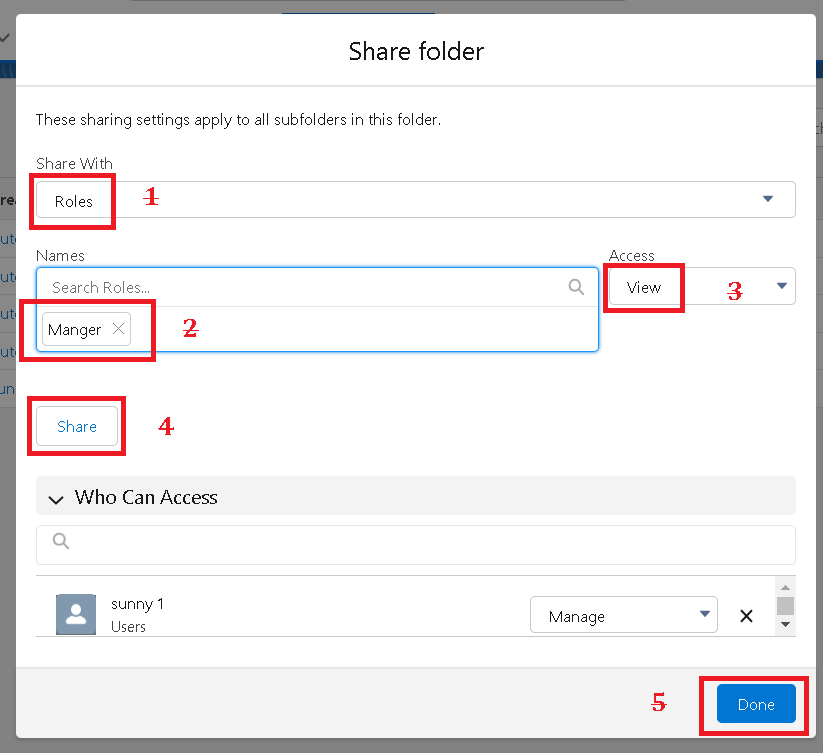
### Sharing a report folder

1. Go to the app  >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.



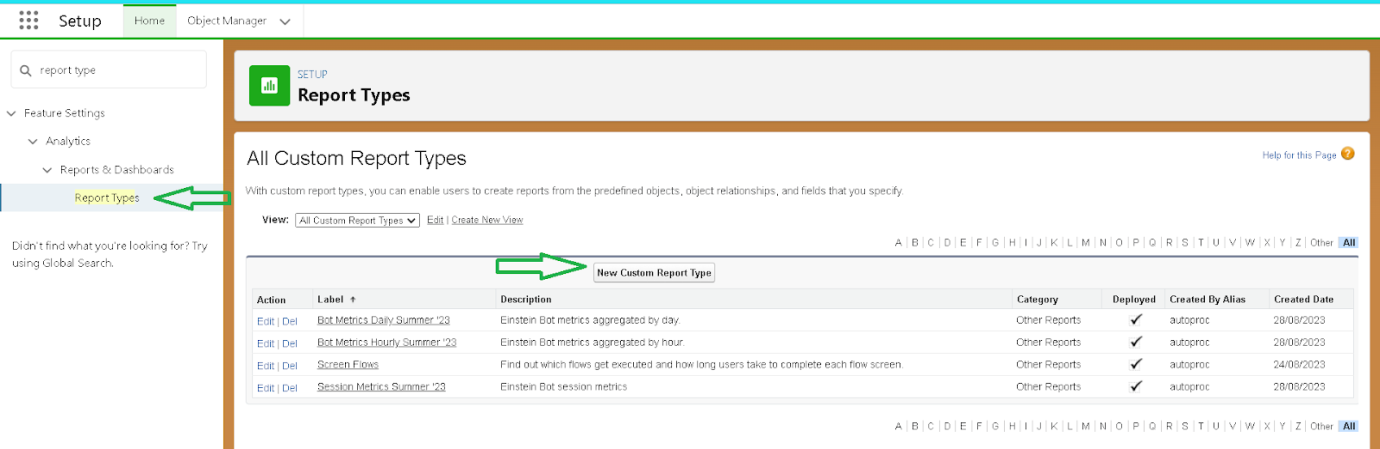
### Sharing a report folder

1. Go to the app  >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.

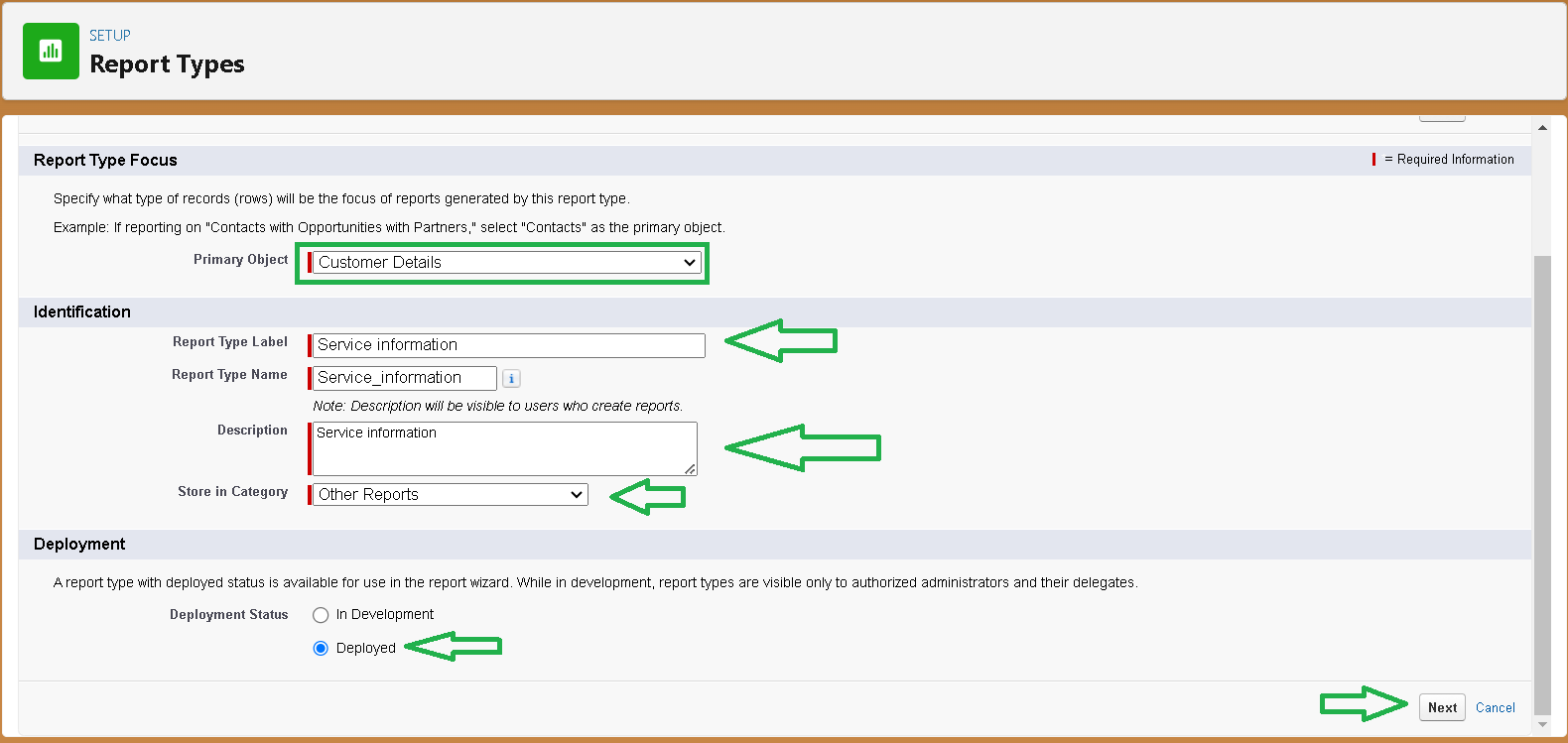


### Create a report type

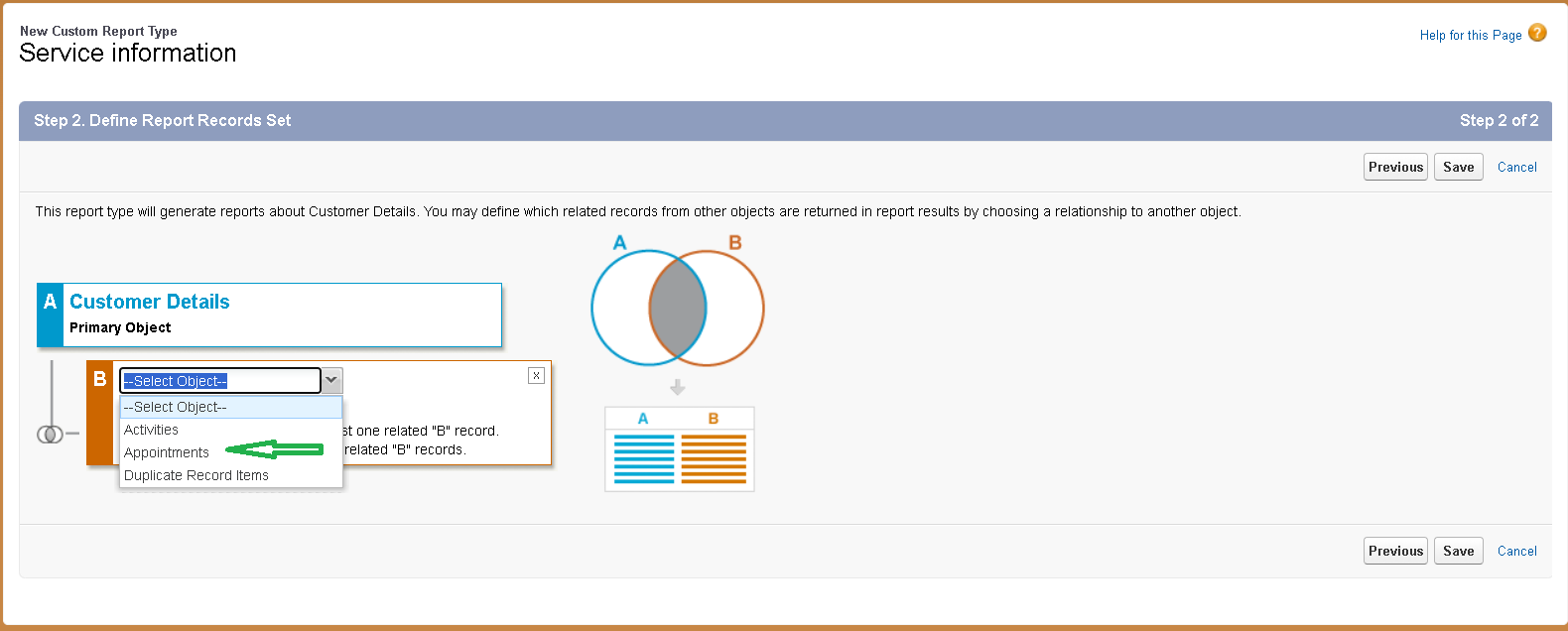
1. Go to setup  >>  type users in quick find box  >>  select Report Type >> click on Continue.
2. Click on new custom report type.

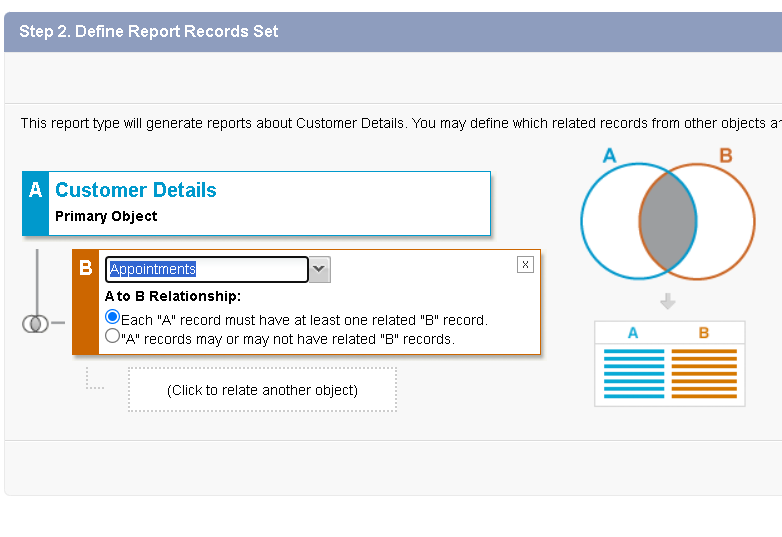


1. Select the Primary object as “ Customer details” .
2. Give the Report type Label as  “ Service information ”
3. Report type Name is autopopulated.
4. Keep the Description as same.
5. Select Store in Category as “ other Reports ”
6. Select the deployment status as “ Depolyed ”, click on Next.

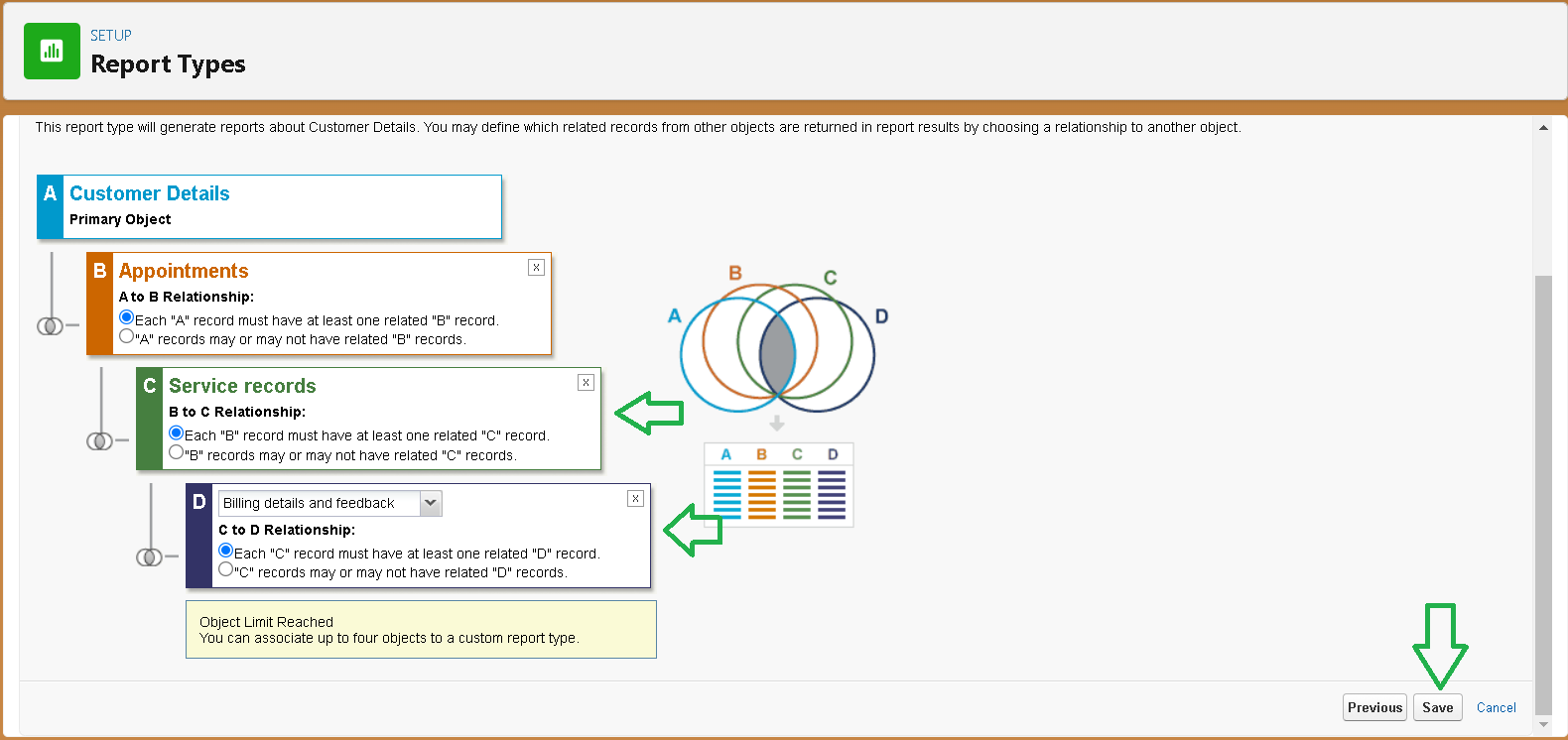


1. now , Click on Related object box.
2. Click on Select Object, choose Appointment Object as shown in fig.





1. Again Click to relate another object.
2. And select the related object as “ service records”.
3. Repeat the process and select the related object as “ Billing details and feedback”.
4. And click on save.

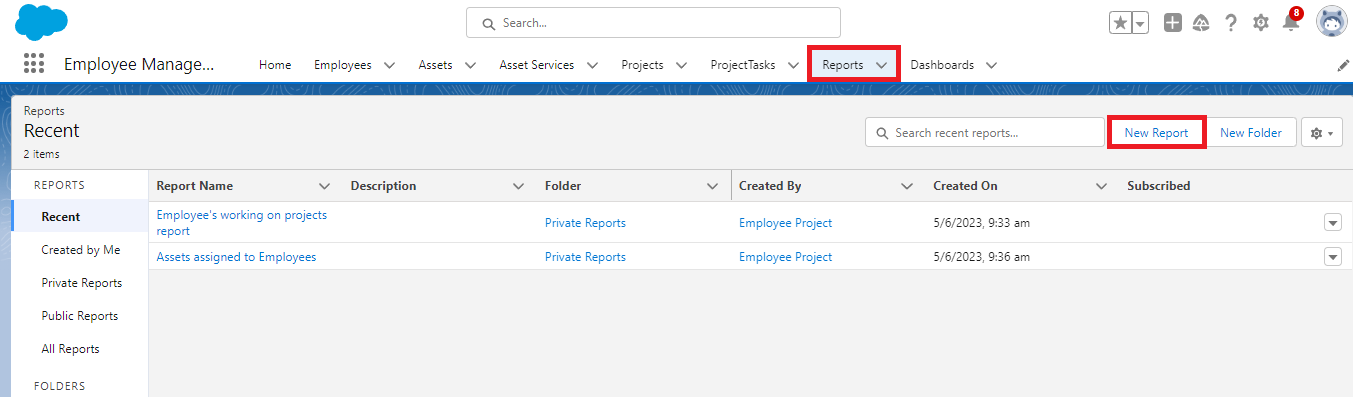


### Create report

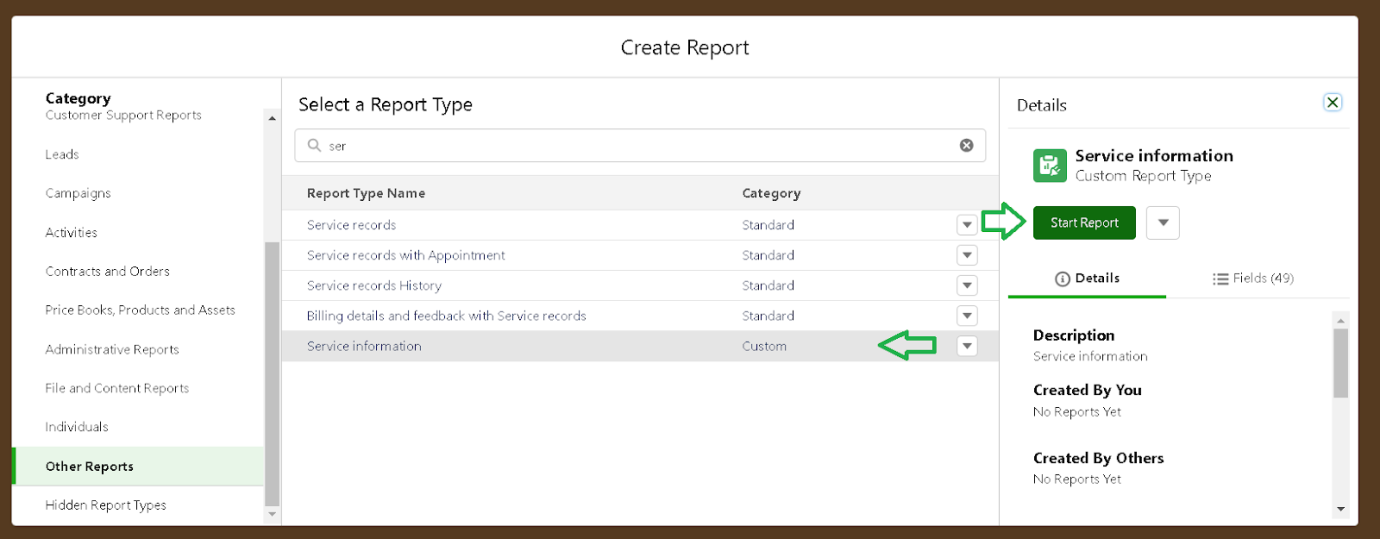
**Note : Before creating report, create latest “10” records in every object.**

**Try to fill every field in each record for better experience.**

1. Go to the app  >>  click on the reports tab
2. Click New Report.

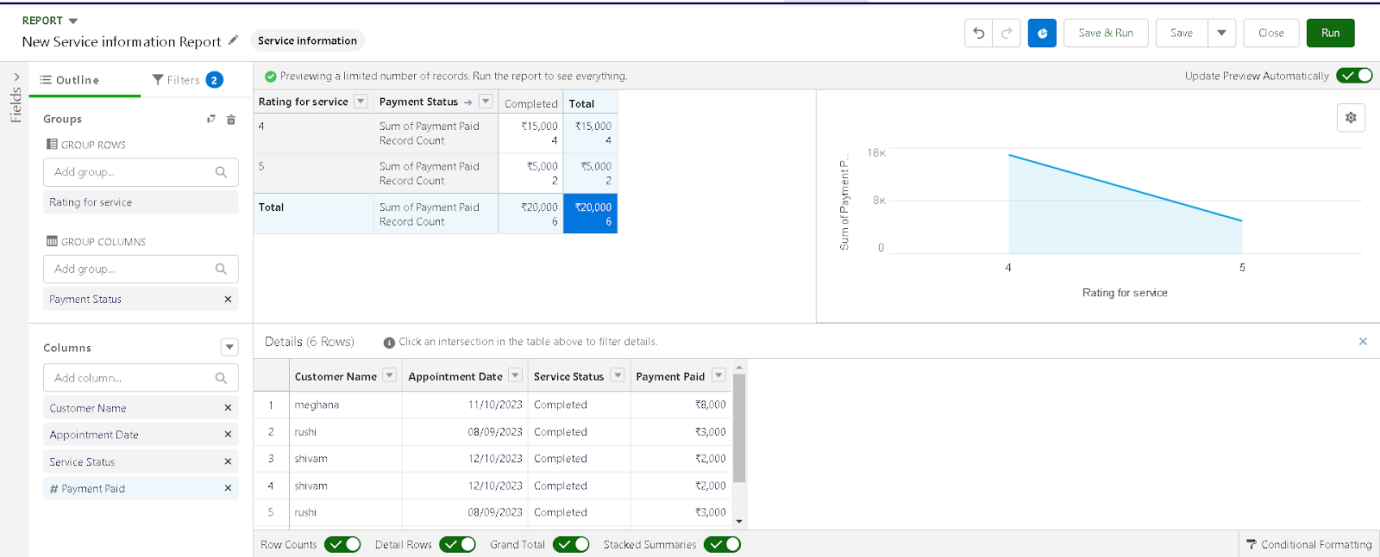


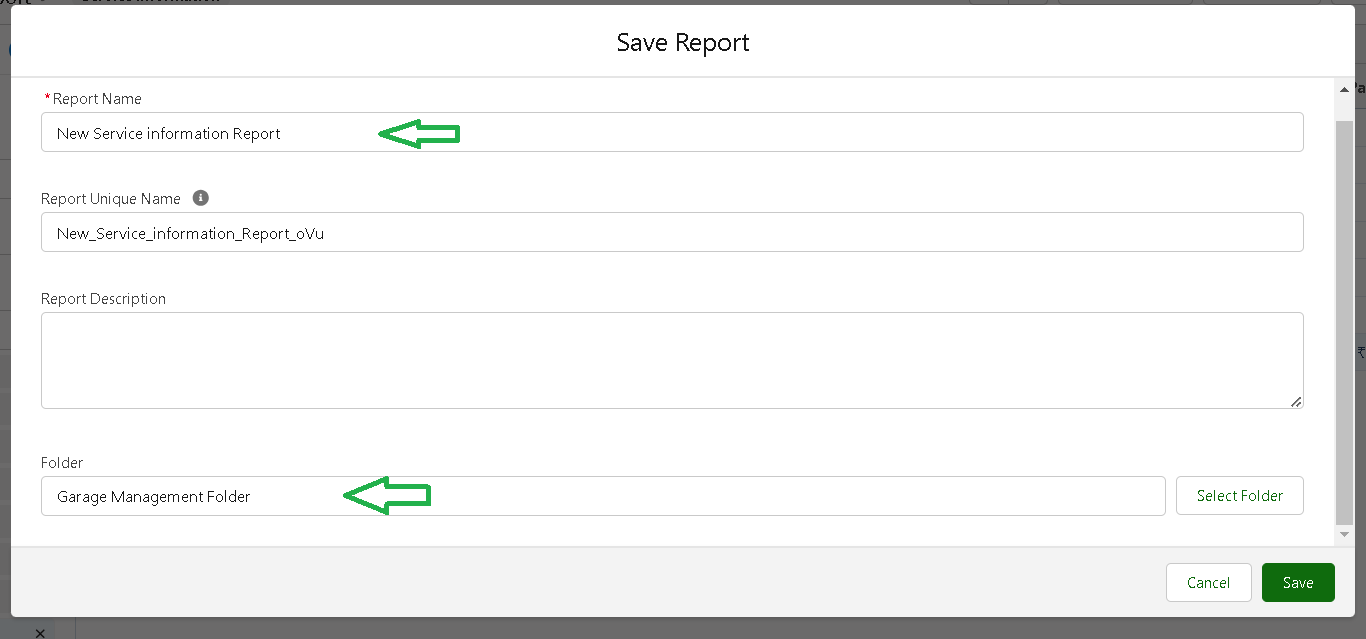
1. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.



1. Their outline pane is opened alredy, select the fields that mentioned below in column section.
   1. Customer name
   2. Appointment Date
   3. Service Status
   4. Payment paid
2. Remove the unnecessary fields.
3. Select the fields that mentioned below in GROUP ROWS section.
   1. Rating for Service
4. Select the fields that mentioned below in GROUP ROWS section.
   1. Payment Status
5. Click on Add Chart , Select the Line Chart.

1. Click on save, Give the report Name : New Service information Report
2. Report unique Name is auto populated.
3. Select the folder the created and Click on save.



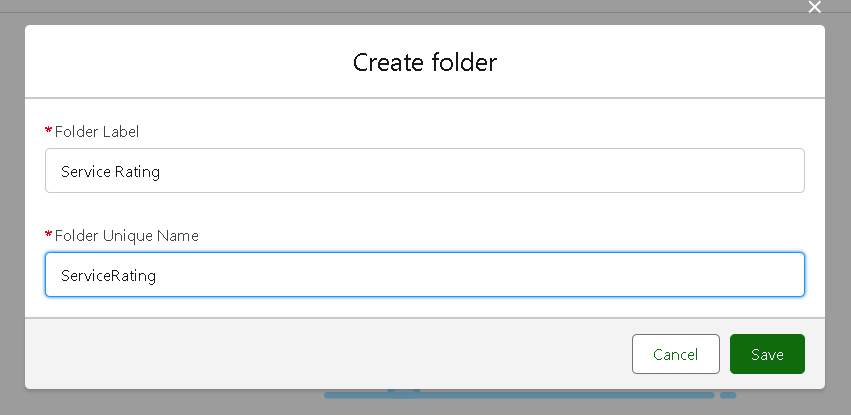


### Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you’ve gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

### Create a dashboard folder

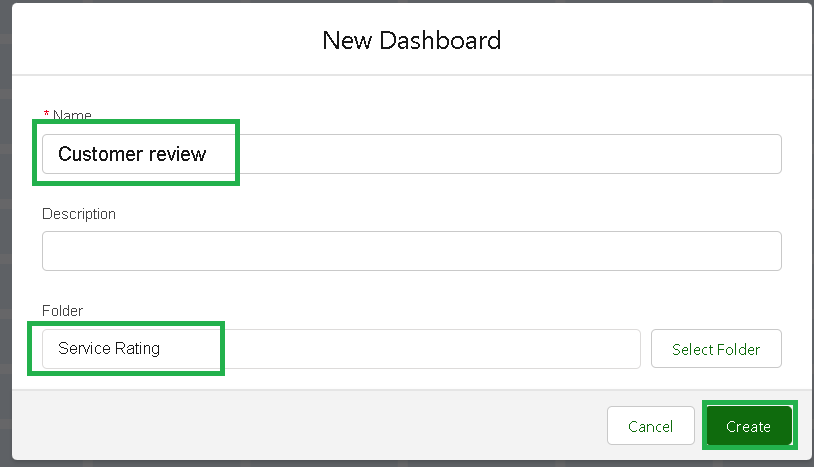
1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.



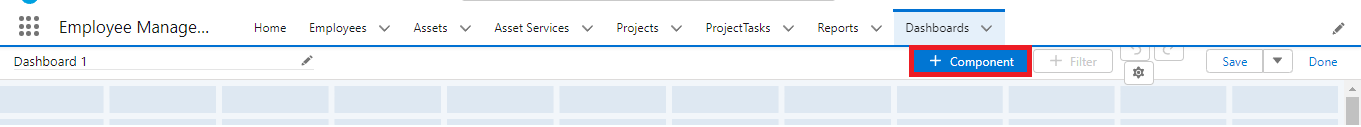
1. Follow the same steps, form milestone 15, and activity 2, and provide the sharing settings for the folder that just created.

### Creation Dashboard

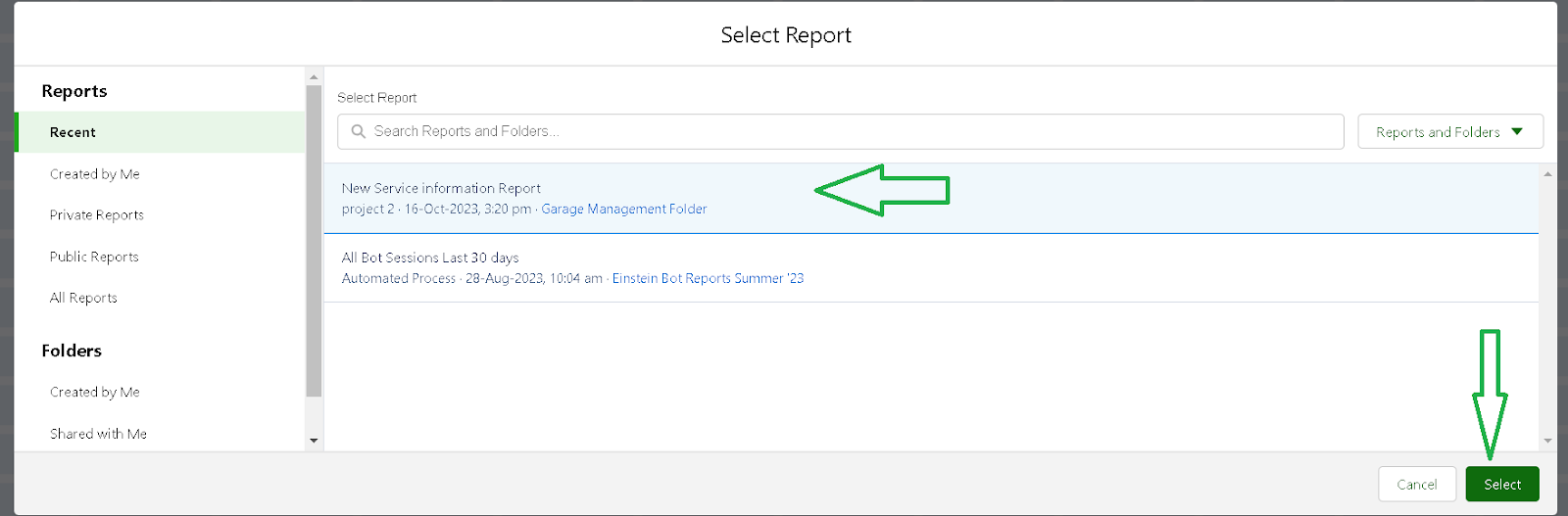
1. Go to the app >>  click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.



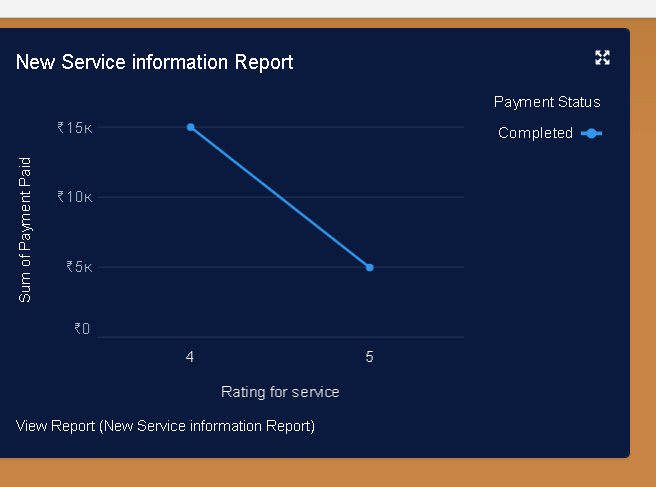
1. Select add component.



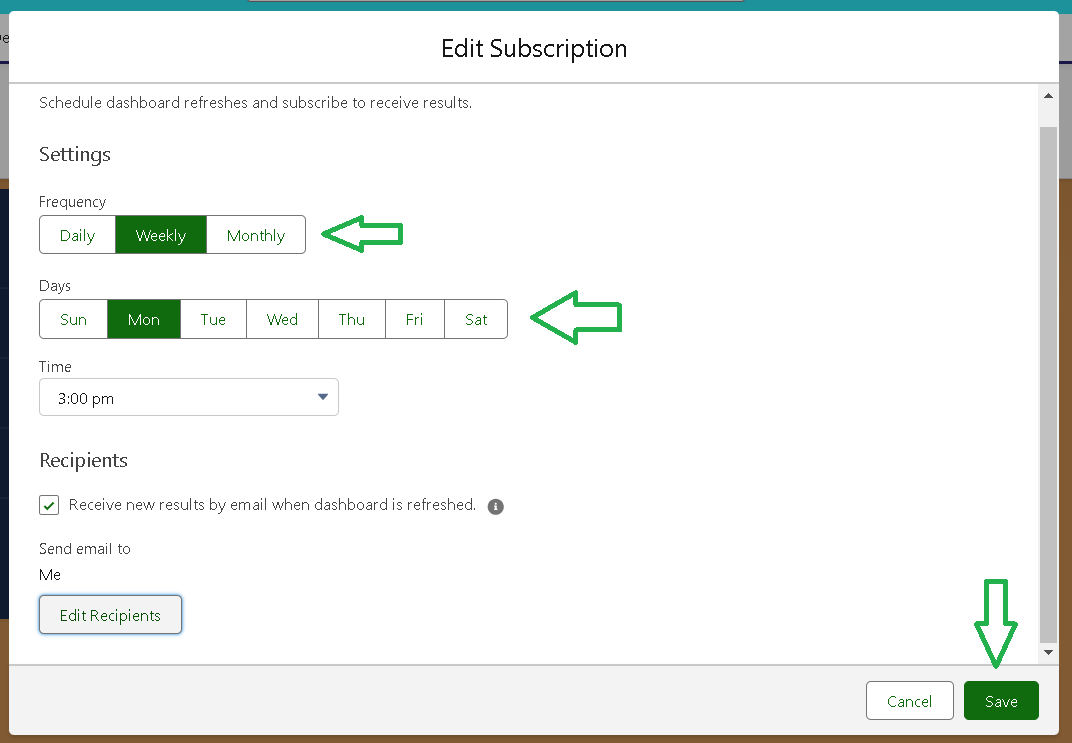
1. Select a Report and click on select.



1. Select the Line Chart. Change the theme.
2. Click Add then click on Save and then click on Done.
3. Preview is shown below.



1. After that Click on Subcribe on top right.
2. Set the Frequency as “ weekly ”.
3. Set a day as monday.
4. And Click on save.



**Conclusion:**

The implementation of a Garage Management System streamlines daily operations by automating critical tasks, improving efficiency, and enhancing customer satisfaction. With features such as appointment scheduling, inventory tracking, billing, and customer relationship management, the system ensures organized and reliable garage management.

This solution minimizes human errors, reduces paperwork, and optimizes time management for employees, allowing them to focus on delivering quality service. Additionally, the system's ability to generate reports and track performance metrics provides valuable insights for continuous improvement and strategic decision-making.