PROJECT OF

SALESFORCE DEVELOPER

GARAGE MANAGEMENT SYSTEM

To optimize customer details, appointment, service records, and billing & feedback for automotive garages.

By

Team Leader:

Sivaraj E

Team Members:

Sri Rohithnivas S J

Suresh R

Roja K

Government Arts And Science College Anthiyur.638501

Email : sivarajsivaraj929@gmail.com

PROJECT OVERVIEW

The Garage Management System (GMS) project on Salesforce is a CRM solution aimed at transforming the way automotive garages manage customer relationships, vehicle service records, and day-to-day operations. With centralized customer profiles, service histories, and appointment scheduling, the system empowers garages to deliver more personalized and efficient customer service. Automated reminders and service notifications help enhance customer engagement, ensuring regular maintenance and timely service updates to boost customer satisfaction.

In addition to customer management, the GMS CRM includes comprehensive vehicle tracking and work order management, allowing technicians and advisors to access detailed service histories, track service progress, and assign tasks seamlessly. An integrated parts inventory module ensures that essential components are in stock, with automated reordering for low-stock items to prevent service delays. Technicians can use mobile access to view work orders and inventory on the go, optimizing workflow and reducing downtime.

Finally, the GMS CRM supports billing and invoicing, providing a streamlined system for generating service estimates, digital invoices, and online payment options. With built-in analytics and reporting, managers gain insights into customer trends, revenue, and operational efficiency, enabling data-driven decisions for business growth. The system is designed to scale with the garage, offering robust data security and compliance, making it a future-proof solution for modern garage management needs.

OBJECTIVES

- **Enhance Customer Retention and Loyalty:** By providing personalized service reminders, streamlined appointment scheduling, and post-service follow-ups, the GMS CRM aims to build long-term customer relationships and increase repeat visits.

- **Optimize Resource Allocation & Technician Efficiency:** Efficient scheduling, real-time work order tracking, and inventory management help maximize technician productivity, reduce wait times, and ensure that resources are allocated effectively for each service job.

- **Increase Revenue through Data-Driven Upselling:** With insights into customer preferences and service history, the CRM enables garages to offer tailored service recommendations and upsell additional maintenance packages, driving revenue growth.

- **Reduce Operational Costs through Automation:** By automating routine tasks like appointment reminders, parts reordering, and invoicing, the GMS CRM minimizes manual workload, reduces errors, and cuts down operational costs.

- **Improve Business Insights and Strategic Planning:** Advanced reporting and analytics provide managers with valuable insights into service trends, technician performance, and financial metrics, supporting informed decision-making and strategic business growth.

KEY FEATURES

- **Customer details:**
  - Store and manage customer details including contact information, service history, and preferences.
  - Access comprehensive service records for customer insights and tailored service.
- **Appointment:**
  - Automated reminders and notifications to reduce no-shows.
  - Integrated calendar to optimize working time.
- **Service records:**
  - Track detailed service information including service type, parts used, and associated costs.
  - Link service records to specific customer appointments for easy tracking and reference.
- **Billing details:**

- Generate and manage billing details, linking them to service records.
- Track payment status and send timely payment reminders.
- Automated invoice generation and email notifications for a streamlined payment process.

- **Feedback:**
  - Collect and manage customer feedback post-service.
  - Track ratings and comments to enhance service quality.
  - Analyze feedback trends for continuous improvement.

DETAILED PROCESS

Activity – 1

Salesforce account Creation & Activation

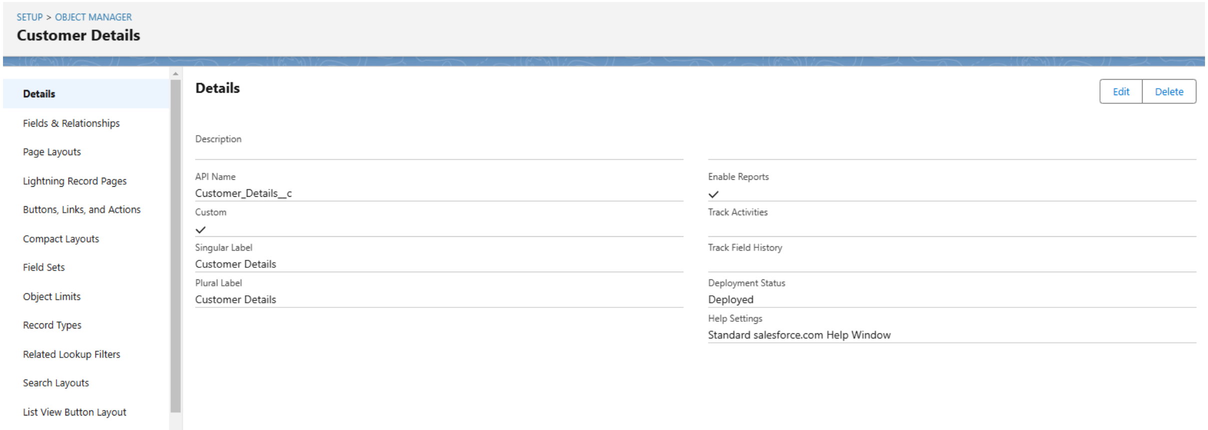
- Create a Developer Account by providing all the required details and Activate the account.
- Go to `https://developer.salesforce.com/signup.`

Activity – 2

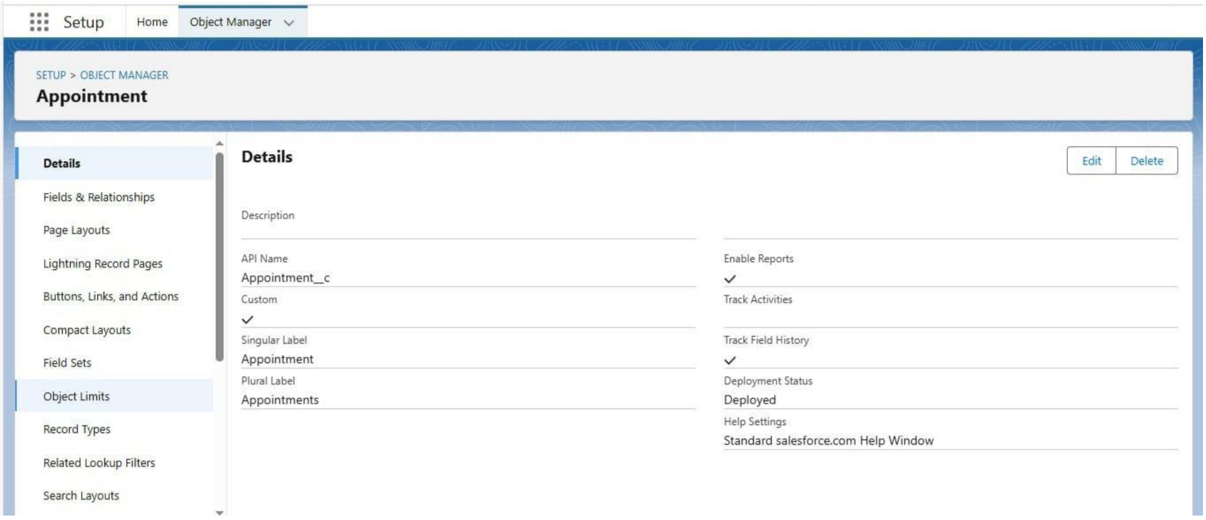
Object Creation

- Create all the required objects with the given label name and format.
- **Customer details Object**
- `Enter the label name >> Customer Details`
- `Plural label name >>  Customer Details`
- `Enter Record Name Label and Format`
- `Record Name >>  Customer Name`

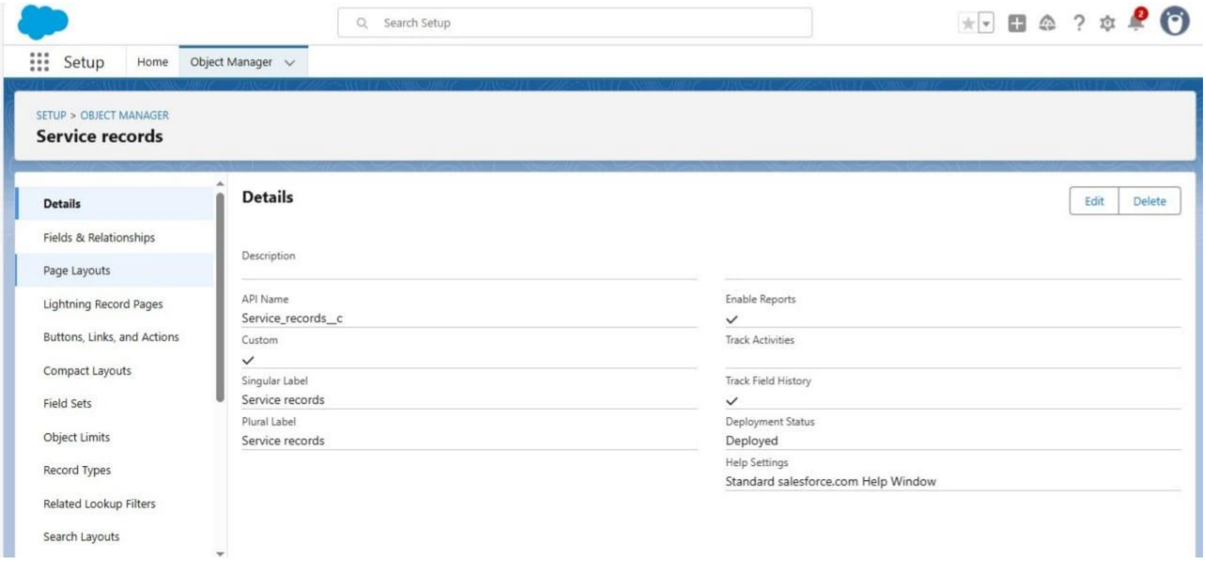
- Data Type >> Text
-



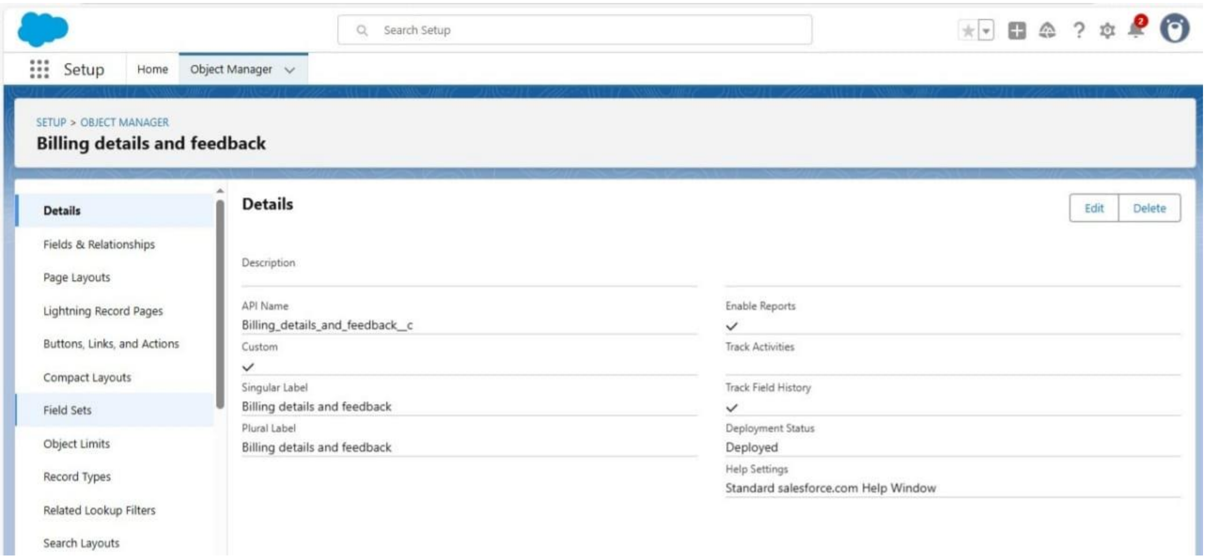
- **Appointment Object**
- Enter the label name >> Appointment
- Plural label name >> Appointments
- Enter Record Name Label and Format
- Record Name >> Appointment Name
- Data Type >> Auto Number
- Display Format >> app-{000}
- Starting number >> 1



- **Service records Object**
- Enter the label name >> Service records
- Plural label name >> Service records
- Enter Record Name Label and Format
- Record Name >>Service records Name
- Data Type >> Auto Number
- Display Format >> ser-{000}
- Starting number >> 1



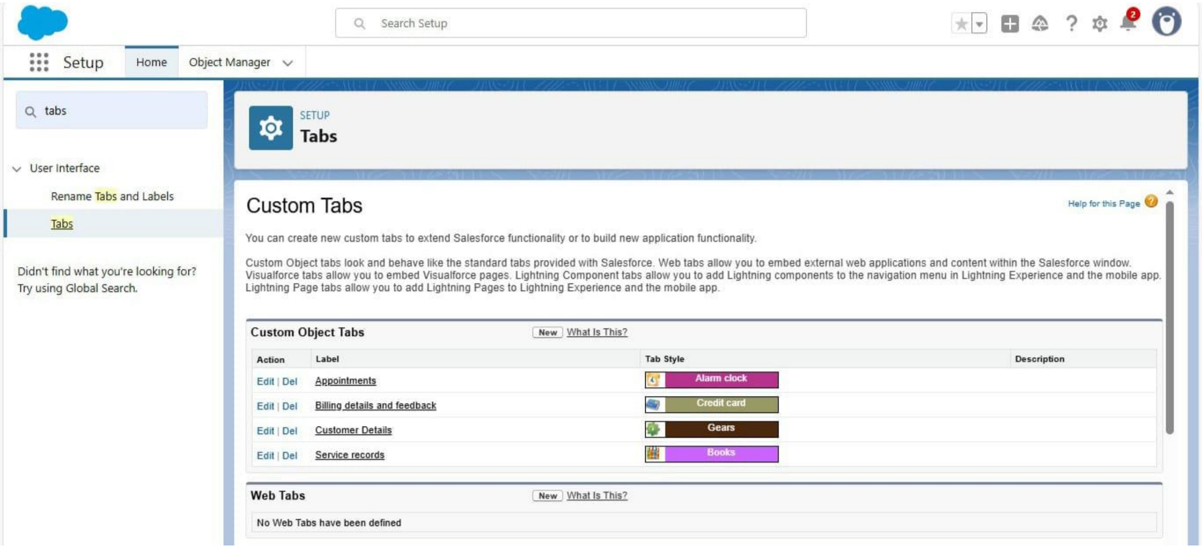
- **Billing details & Feedback Object**
- Enter the label name >> Billing details and feedback
- Plural label name >> Billing details and feedback
- Enter Record Name Label and Format
  - Record Name >> Billing details and feedback Name
  - Data Type >> Auto Number
  - Display Format >> bill-{000}
  - Starting number >> 1



Activity – 3

Custom Tabs Creation

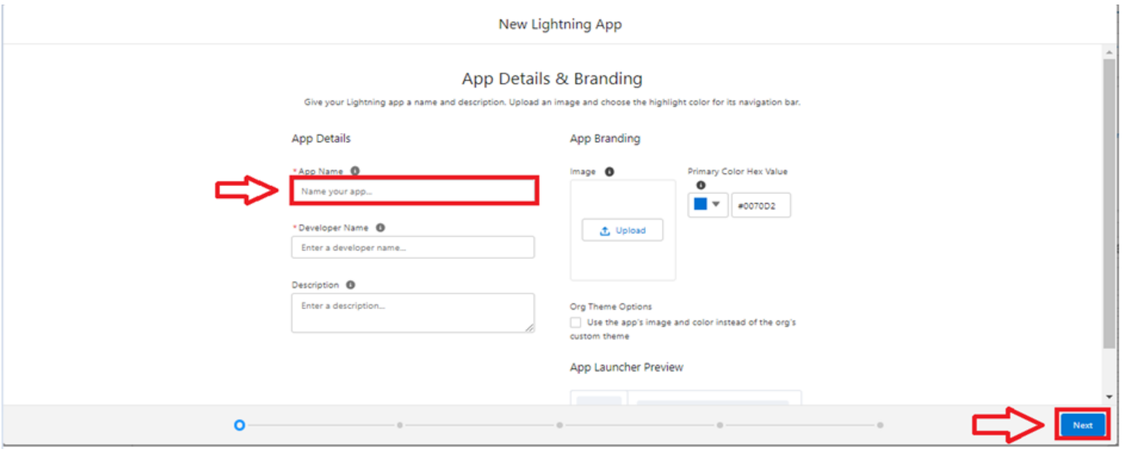
- A tab is like a user interface that is used to build records for objects and to view the records in the objects.
- Create these Custom tabs for every Object by following below steps:
- Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
- Make sure that the Append tab to users' existing personal customizations is checked.



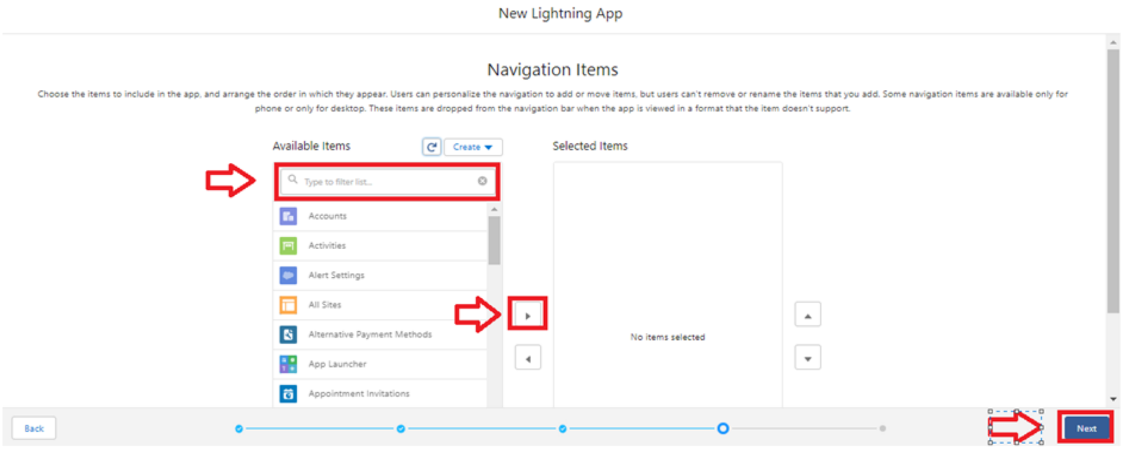
Activity – 4

Create a Lighting App

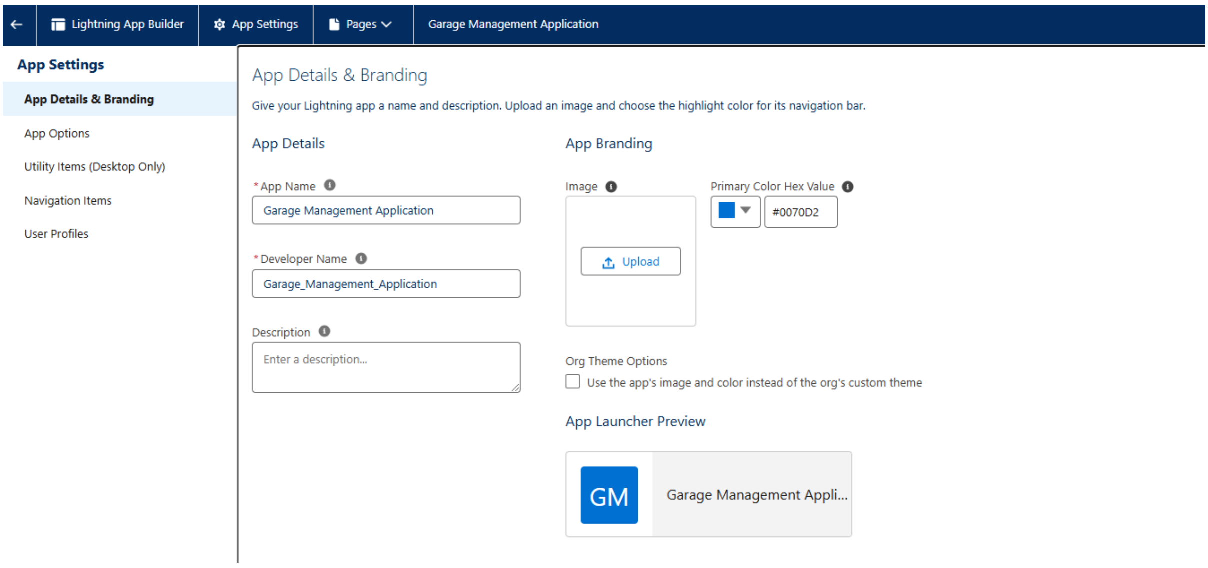
1.Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.



2. To Add Navigation Items:



3. 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.

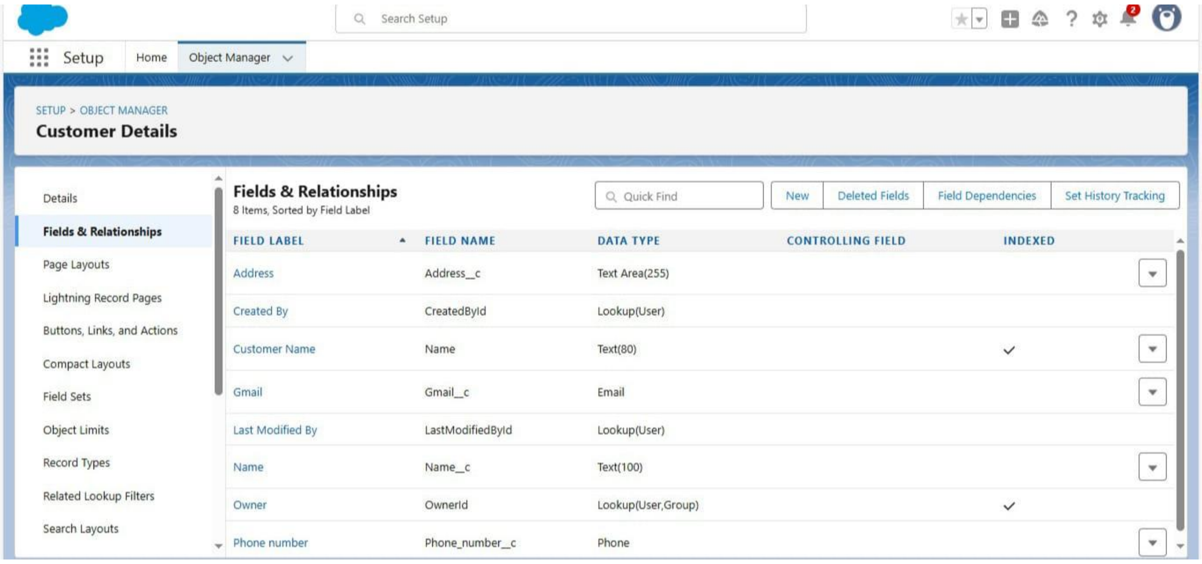


Activity – 5

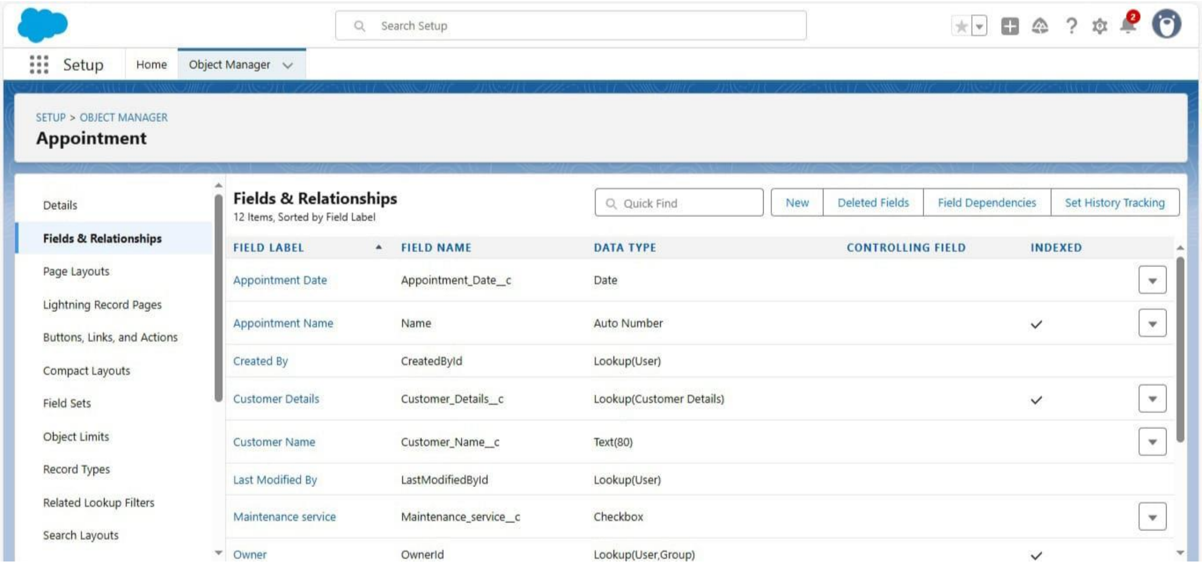
Field Creation for every object:

Create fields for each and every object as per shown in below images.

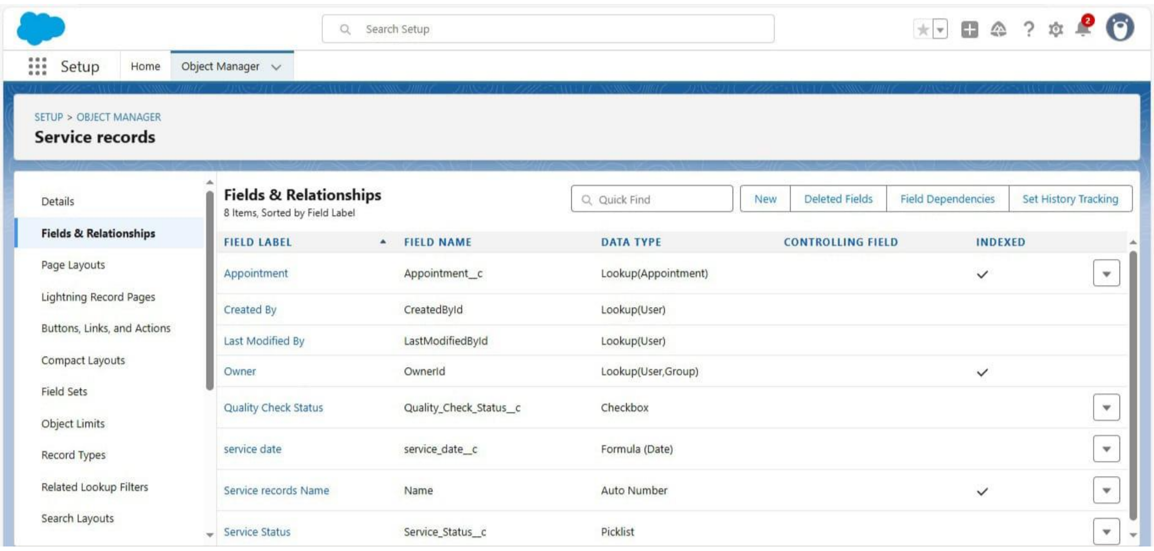
- **Customer details object**



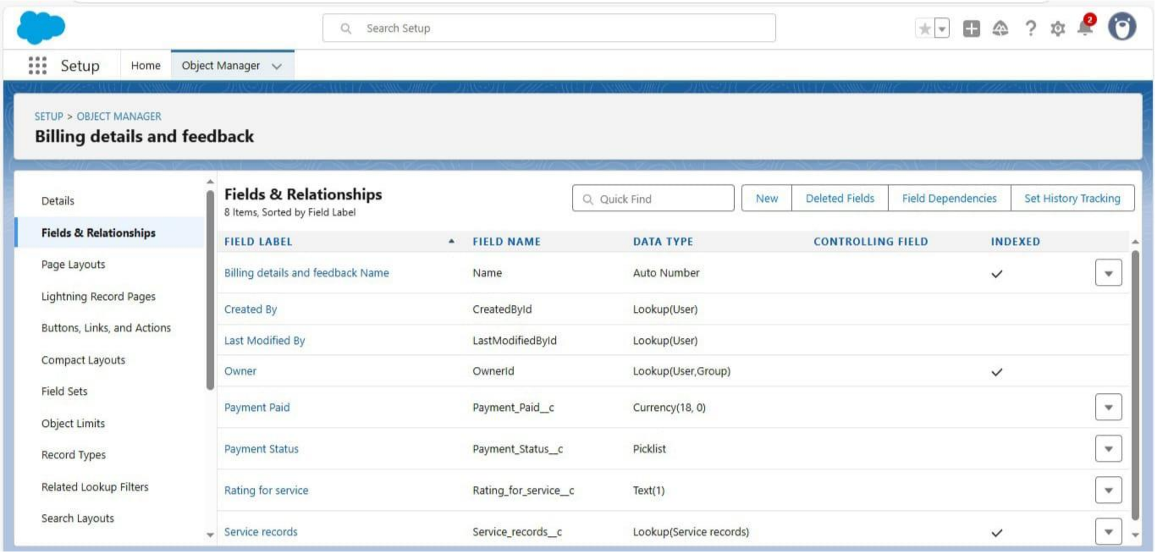
- **Appointment object**



- **Service records object**



- ## Billing details & Feedback object

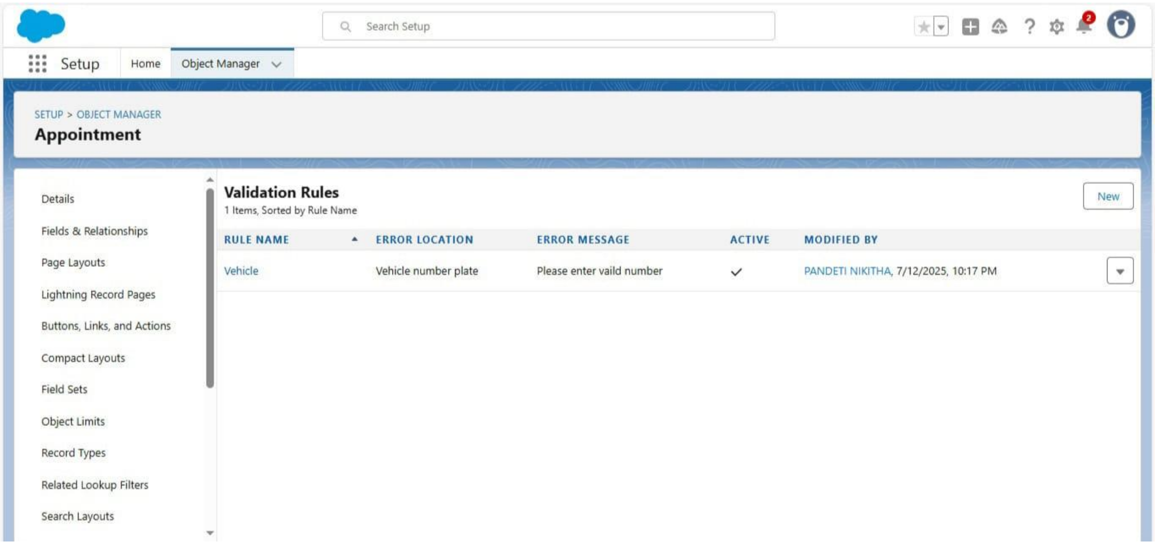


Activity – 6

Validation Rules

- ## For Appointment

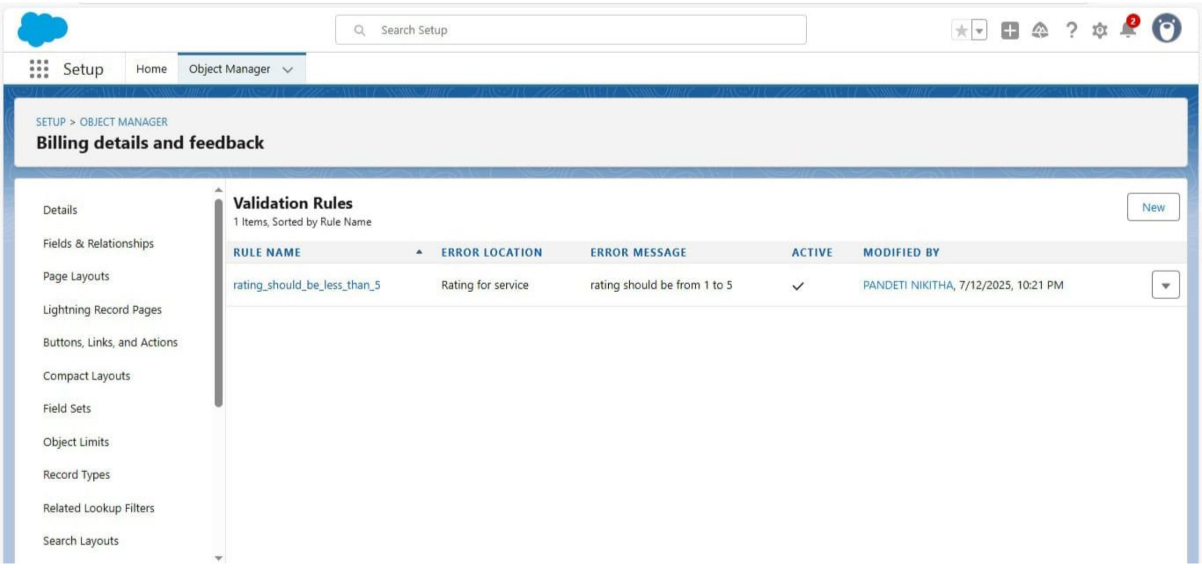
- Enter the Rule name as " Vehicle ".
- Insert the Error Condition Formula as : -
- NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))



- ## For Billing details & Feedback

- Enter the Rule name as " rating_should_be_less_than_5".
- Insert the Error Condition Formula as : -

- NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))
- Enter the Error Message as "rating should be from 1 to 5", select the Error location as Field and select the field as "Rating for Service".



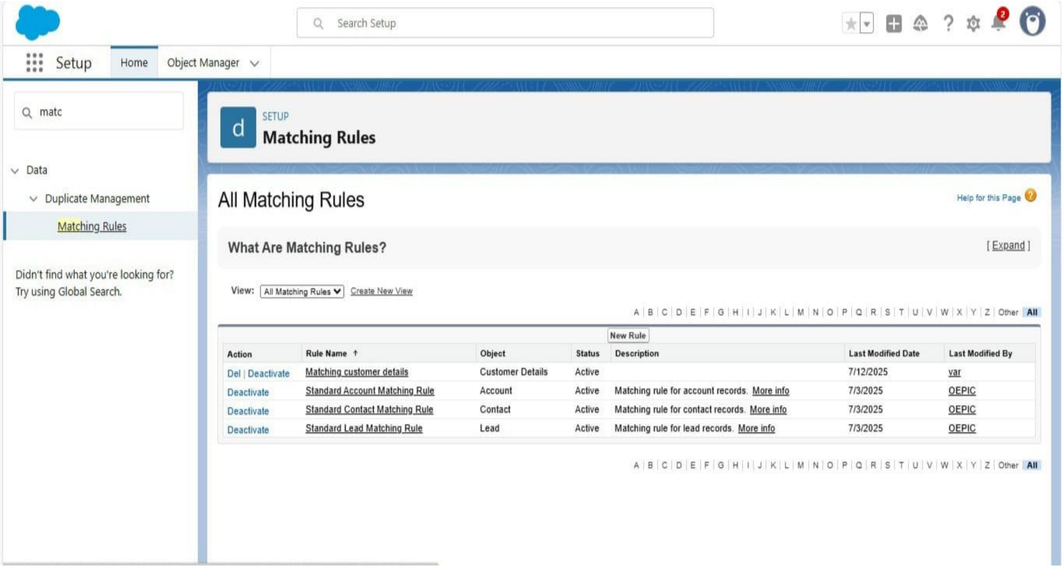
Activity – 7

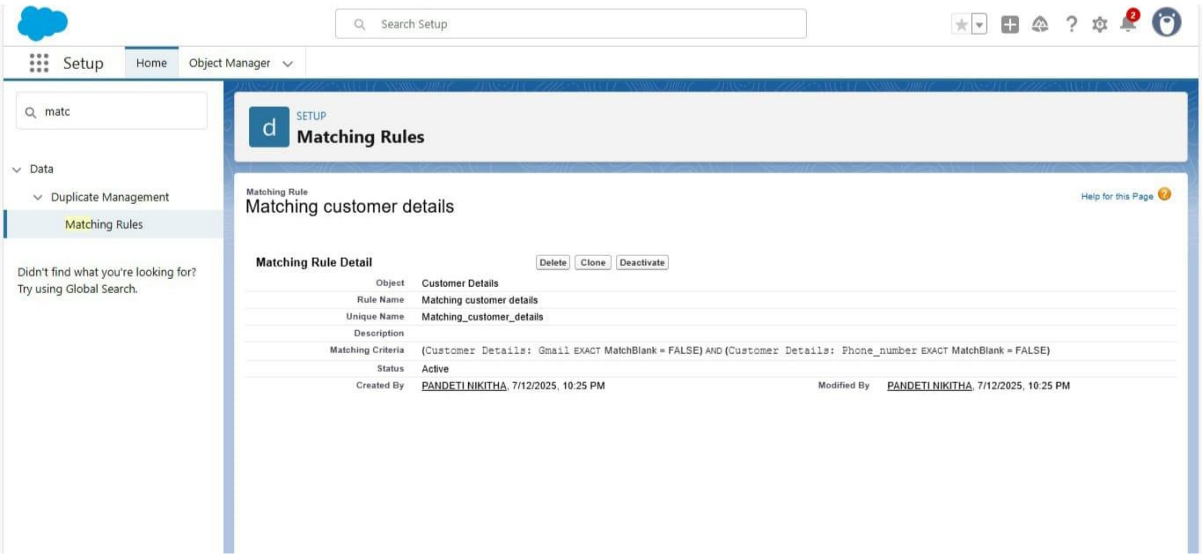
Duplication Rules

- **Matching Rule:**
- 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

- Create a matching rule for the Customer details Object with the given criteria.



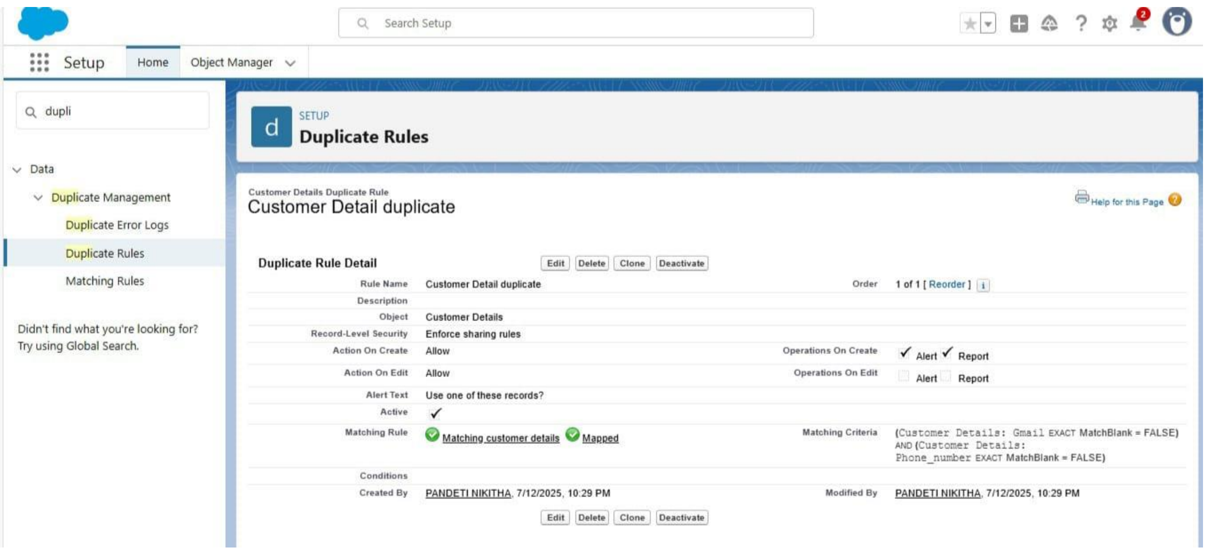


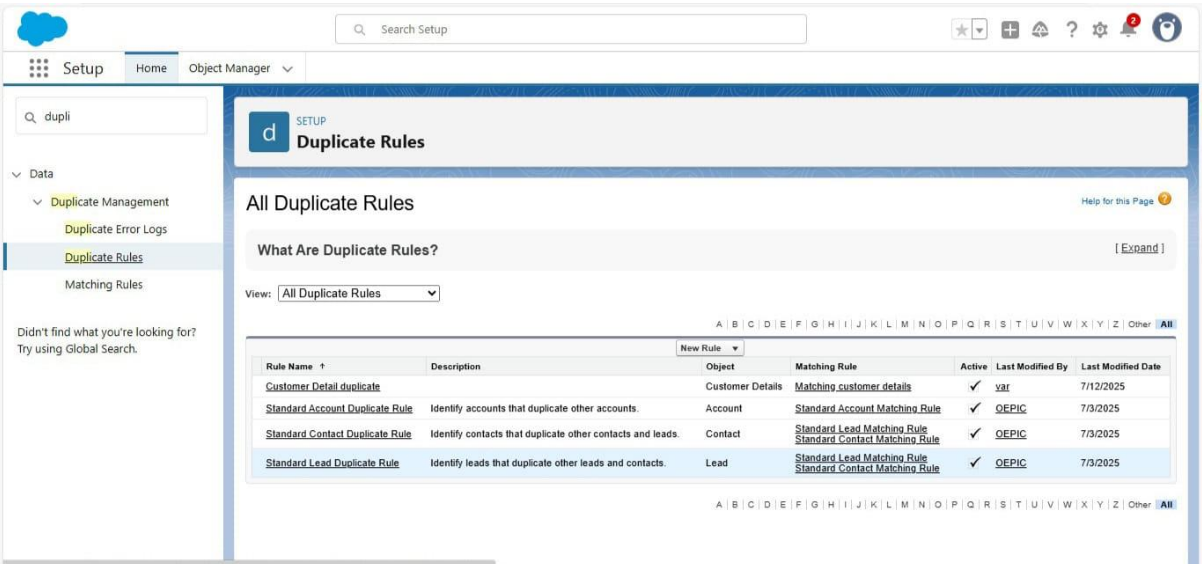
-Duplicate Rules:

Create a duplicate rule for the same customer details object which allows the duplication in input fields.

Steps:

- Give the Rule name as : Customer Detail duplicate.
- Select the matching rule : Matching customer details.

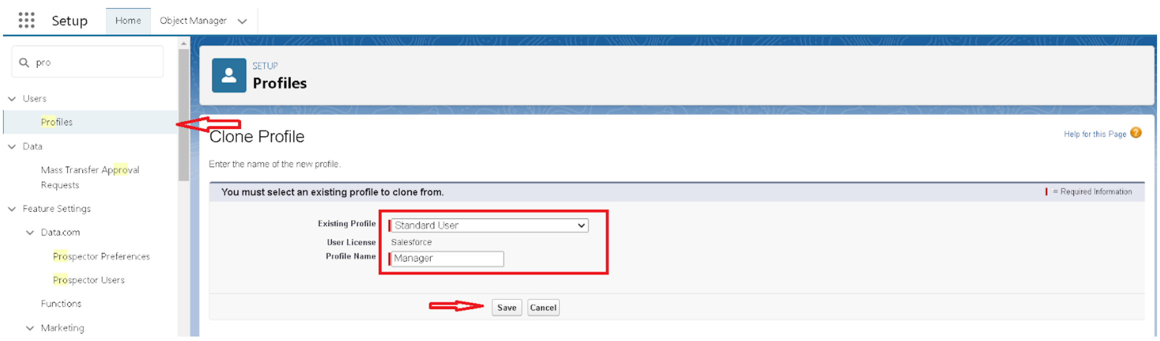




Activity – 8

Creating Profiles

- ### Manager Profile

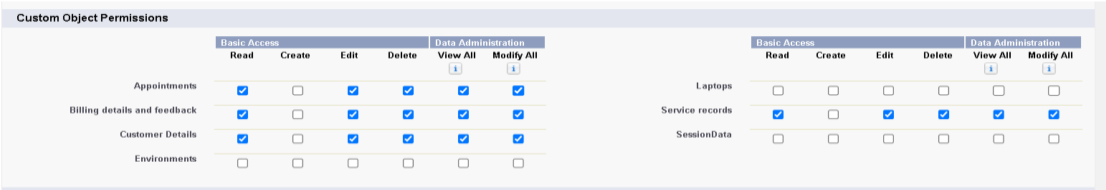


-

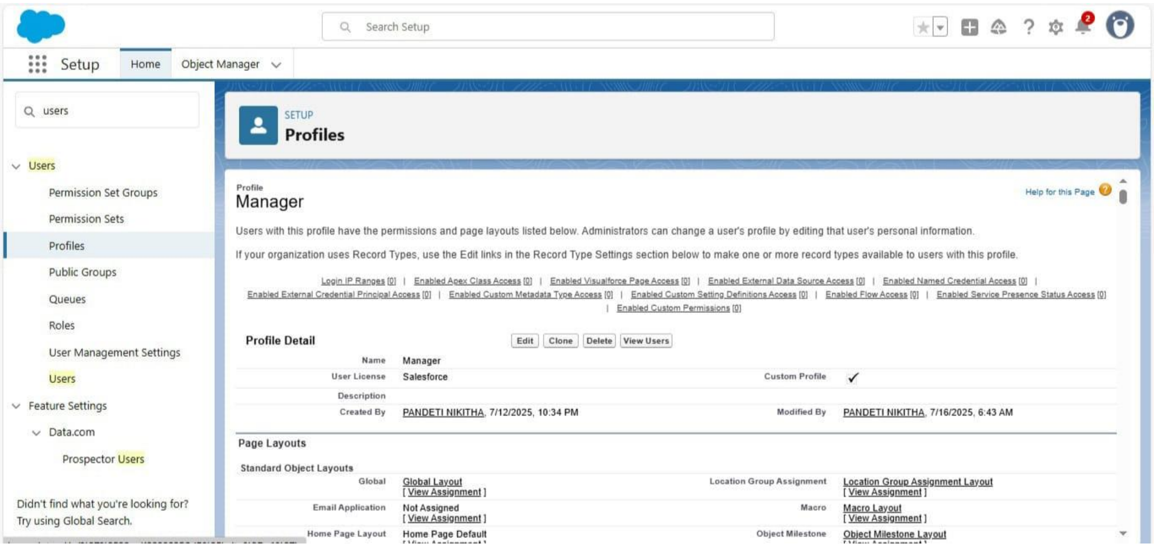
`Select the Custom App settings as default for the Garage management.`

- 

- 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.

- 

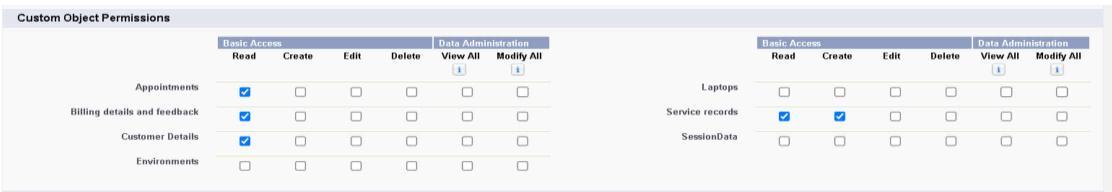
- Changing the session times out after should be " 8 hours of inactivity”.
- Change the password policies as mentioned :
- User passwords expire in should be " never expires ”.
- Minimum password length should be " 8 ”.

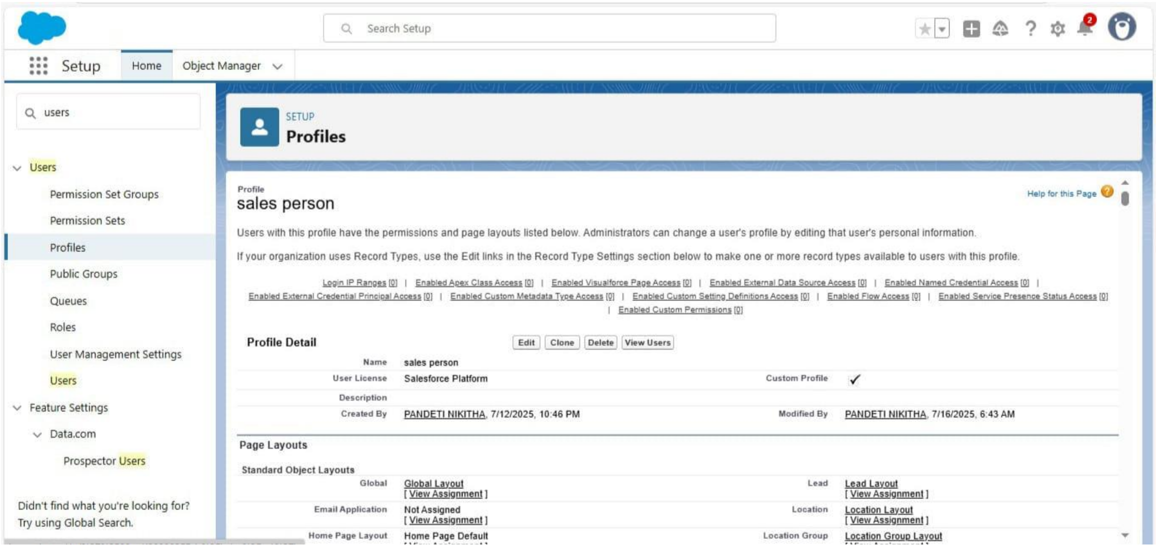
- 

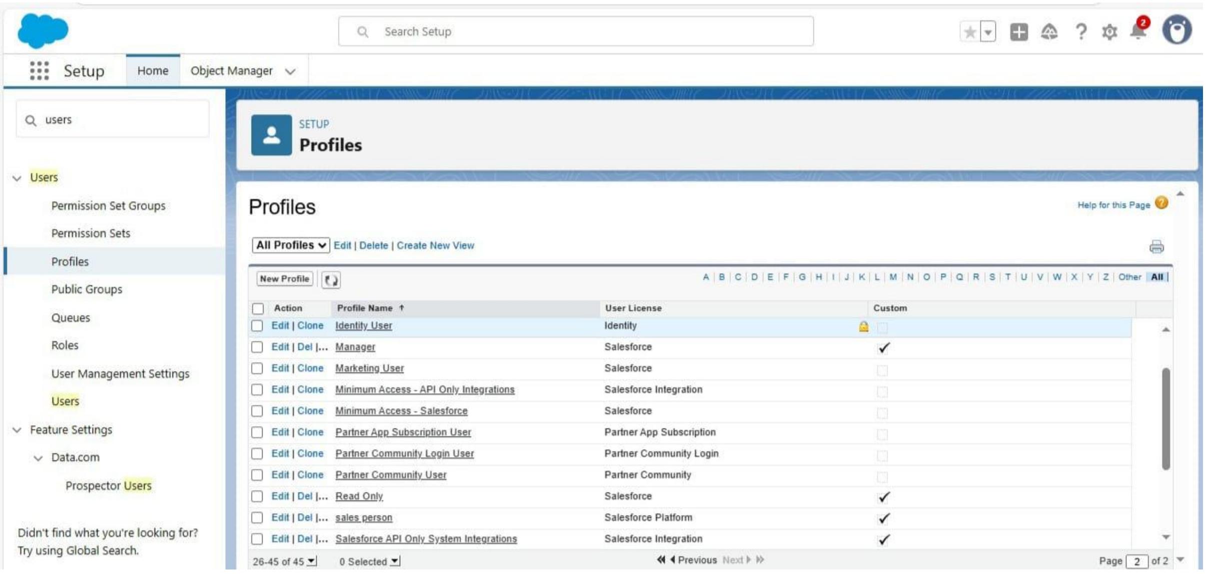
- **Sales Person Profile**

- 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.

- 





Activity – 9

Roles & Role Hierarchy

Manager Role:

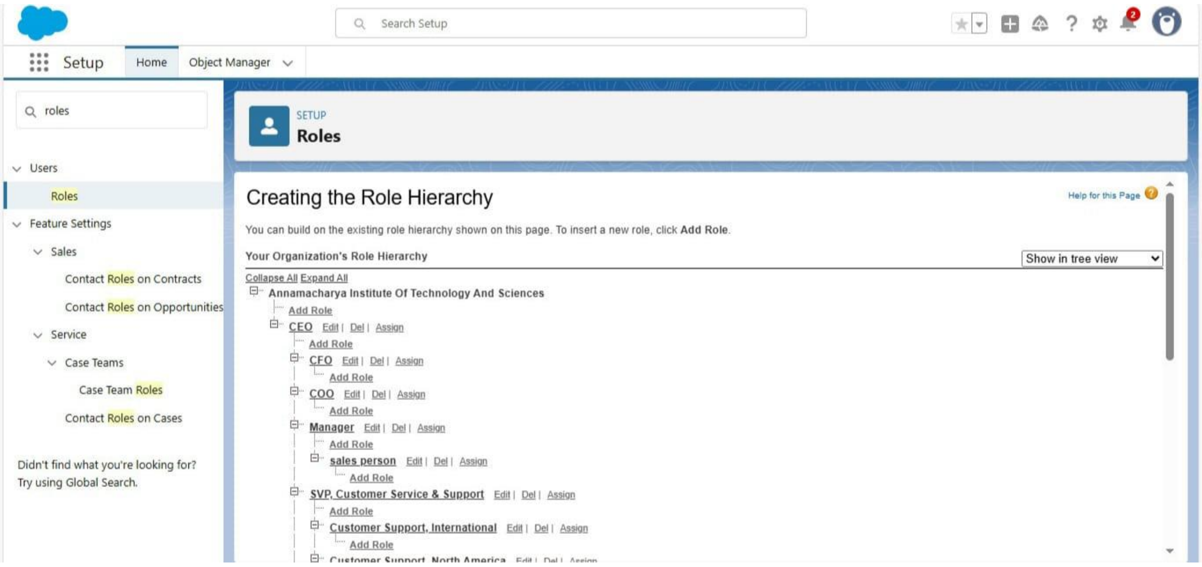
-Click on Expand All and click on add role under whom this role works.

-Give Label as "Manager" and Role name gets auto populated.

Another Role:

-Click plus on CEO role, and click add role under manager.

-Give Label as "sales person" and Role name gets auto populated.



Activity – 10

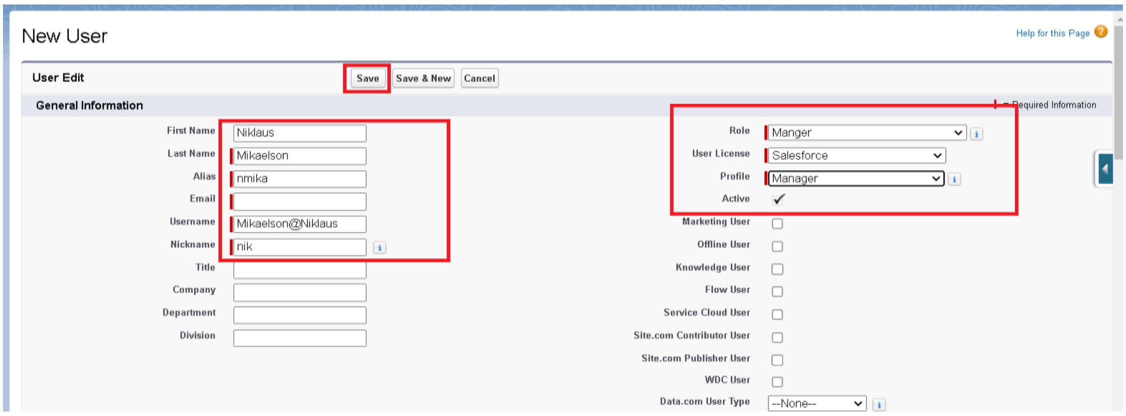
Users

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.

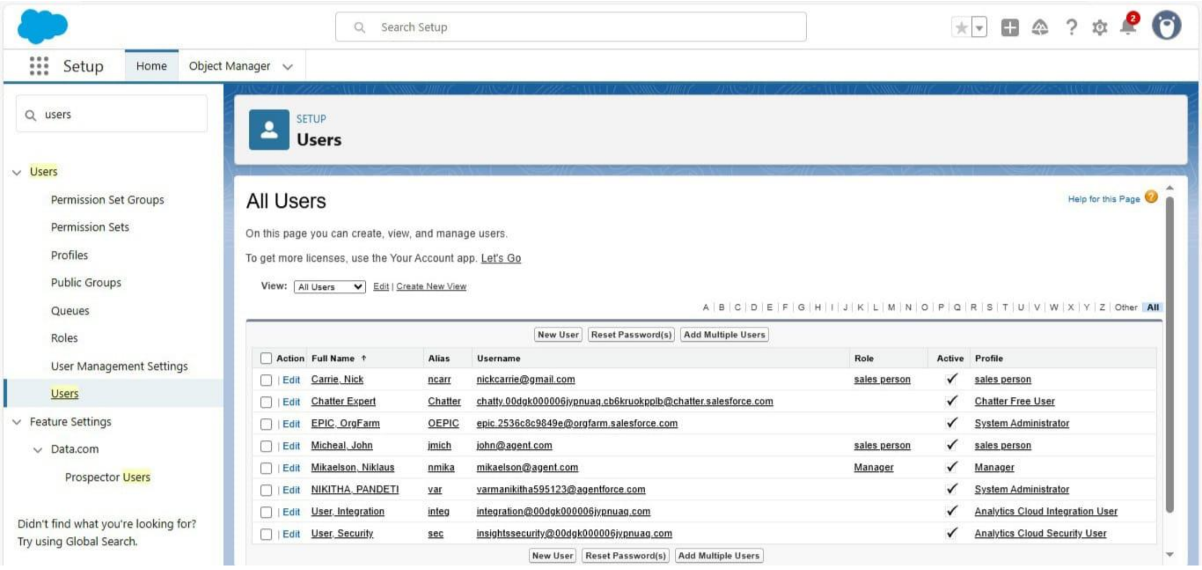
Create Users:

Fill in the fields :

1. First Name : Niklaus
2. Last Name : Mikaelson
3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role : Manager
8. User licence : Salesforce
9. Profiles : Manager



Create Another Users:



Activity – 11

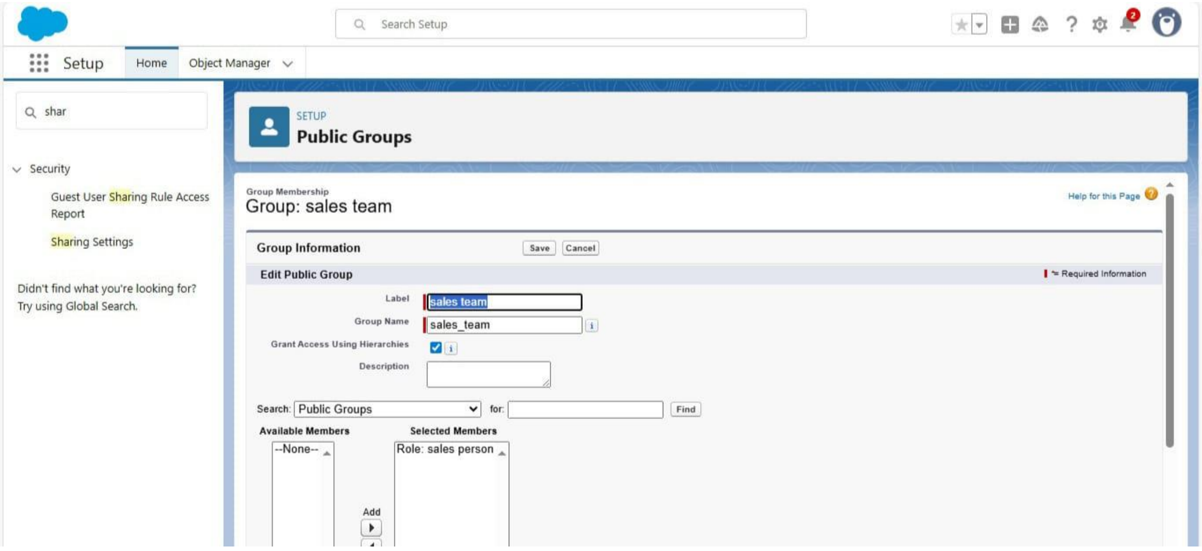
Public Groups

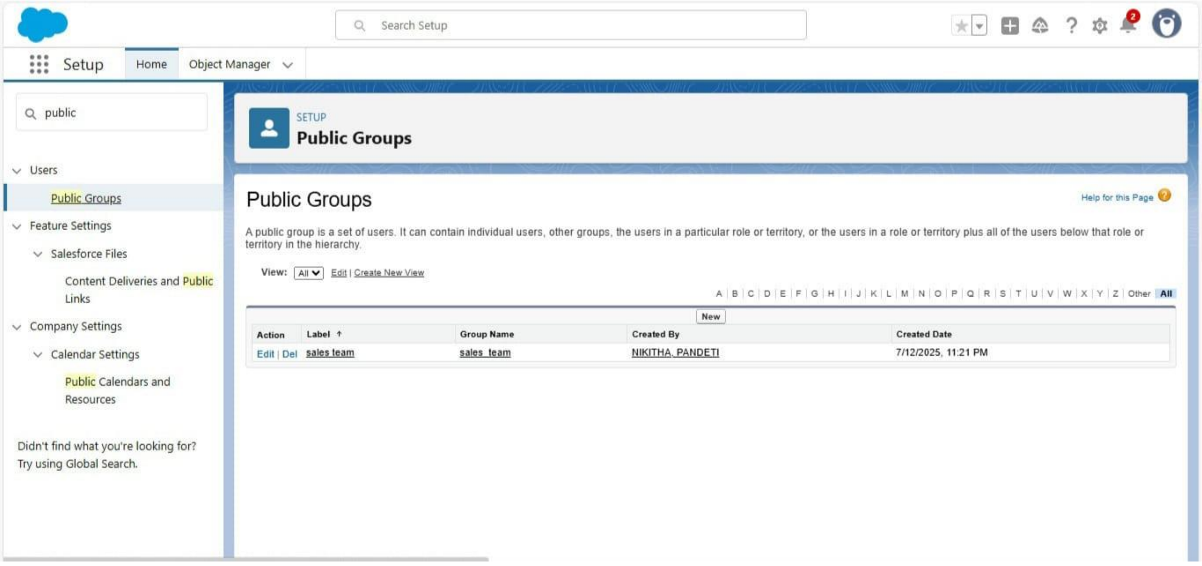
- **Creating New Public Groups**

  a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings.

Steps:

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.



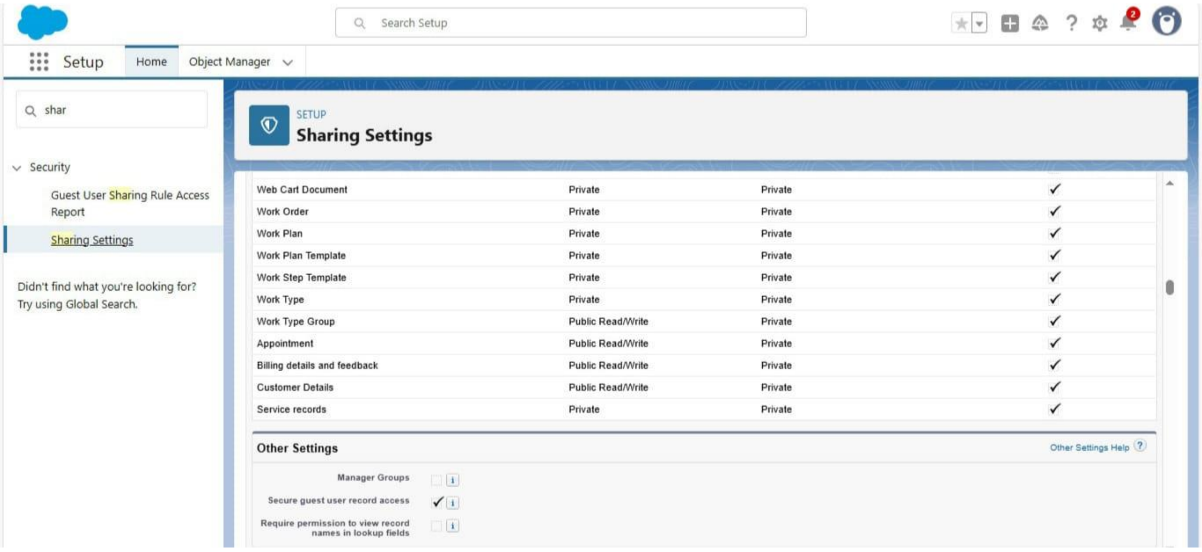


Activity – 12

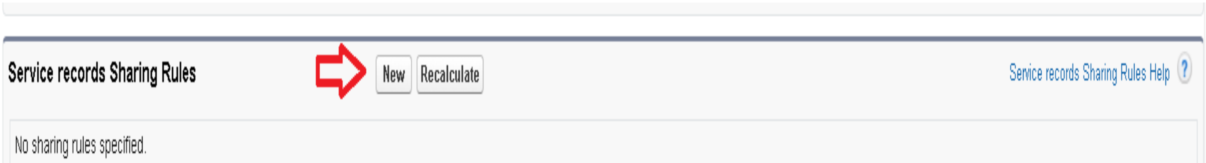
Sharing Settings: control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy.

Steps:

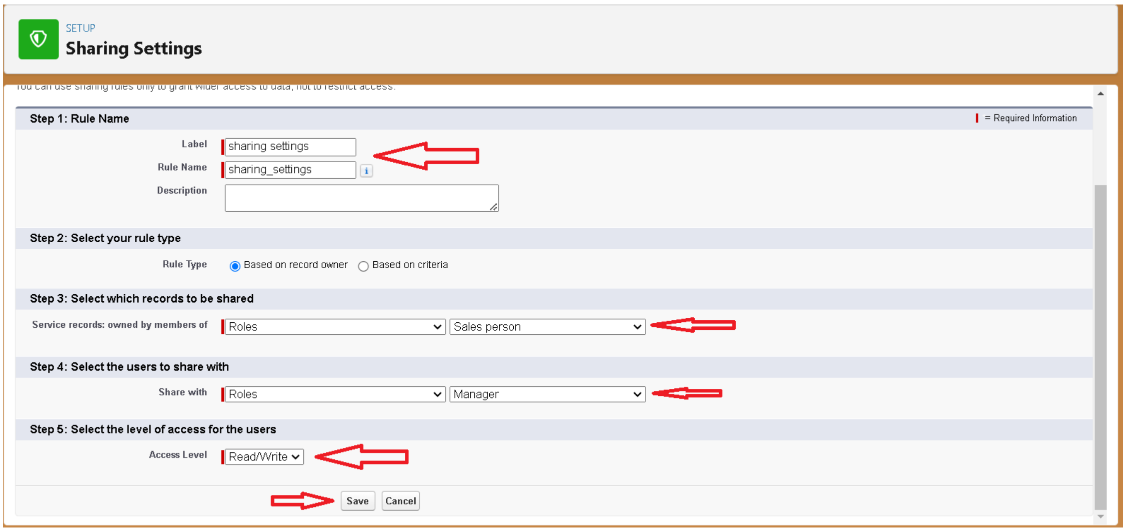
```
-Change the OWD setting of the Service records Object to private as shown in fig.
```



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



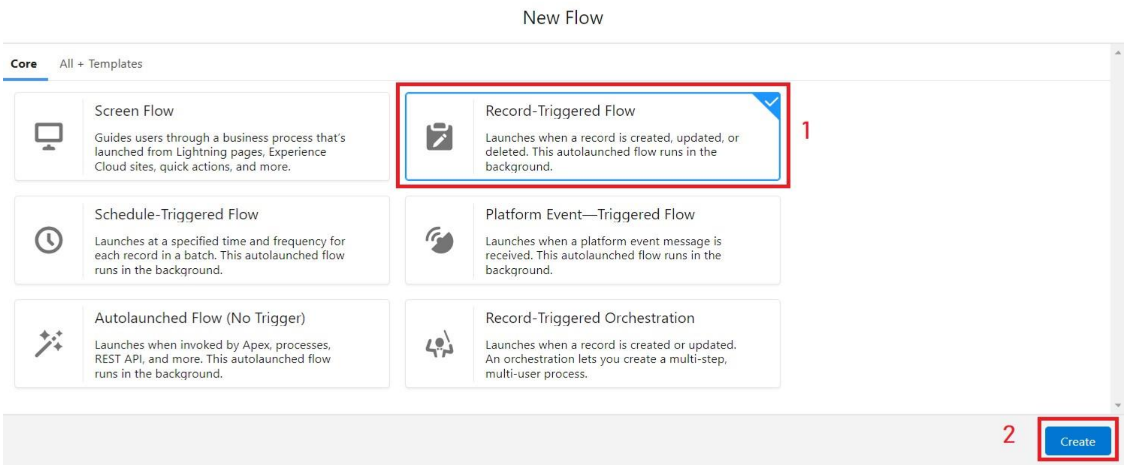
1. Give the Label name as " Sharing setting"
2. Rule name is auto populated.
3. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
4. In step 4: share with, select " Roles " >> " Manager "
5. In step 5 : Change the access level to " Read / write ".
6. Click on save.



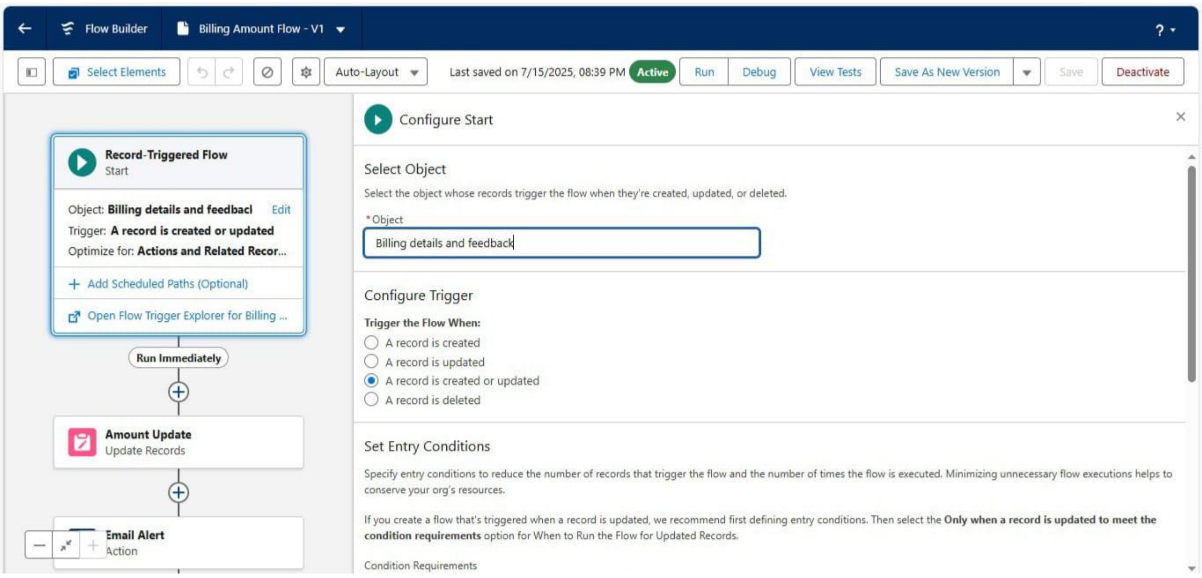
Activity – 13

Flows: a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps.

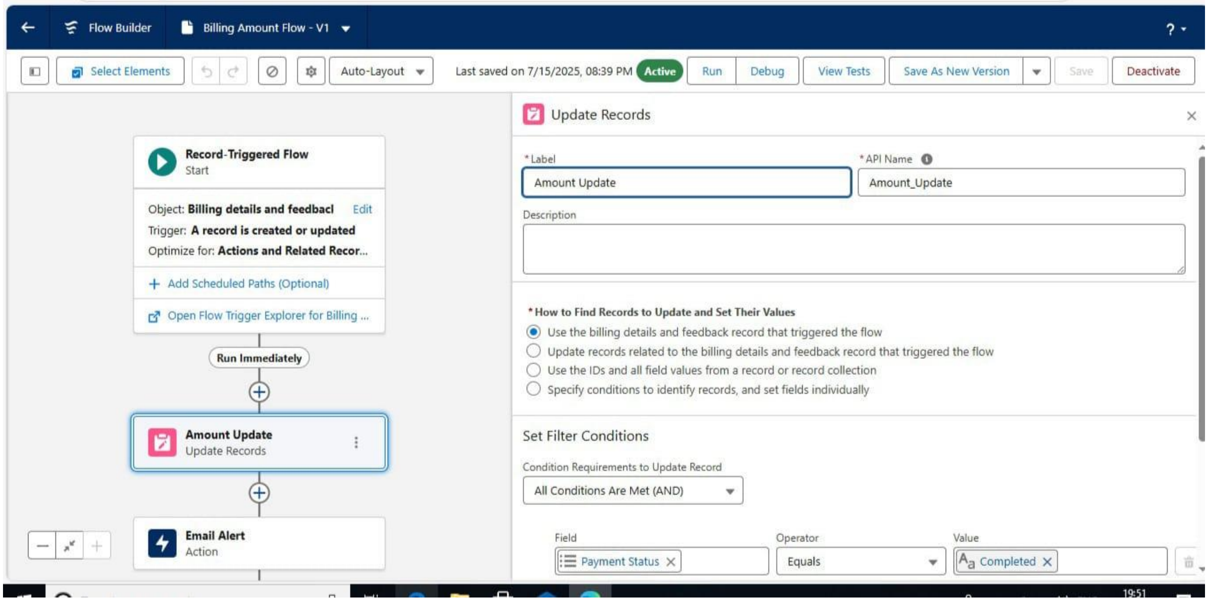
- **Record-Trigger Flow** Select the Record-triggered flow and Click on Create.



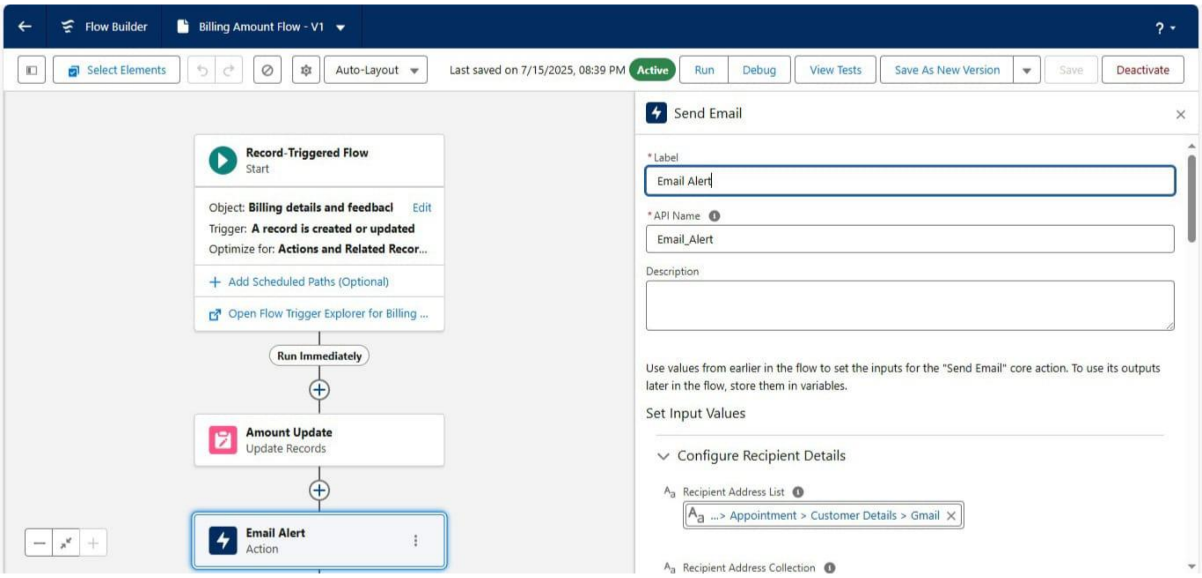
-
- Select the Object as "Billing details and feedback"in the Drop down list.
- Select the Trigger Flow when: "A record is Created or Updated".
- Select the Optimize the flow for: "Actions and Related Records" and Click on Done.



- **Amount Update Flow**
  - Field : Payment_Paid__c
  - Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}

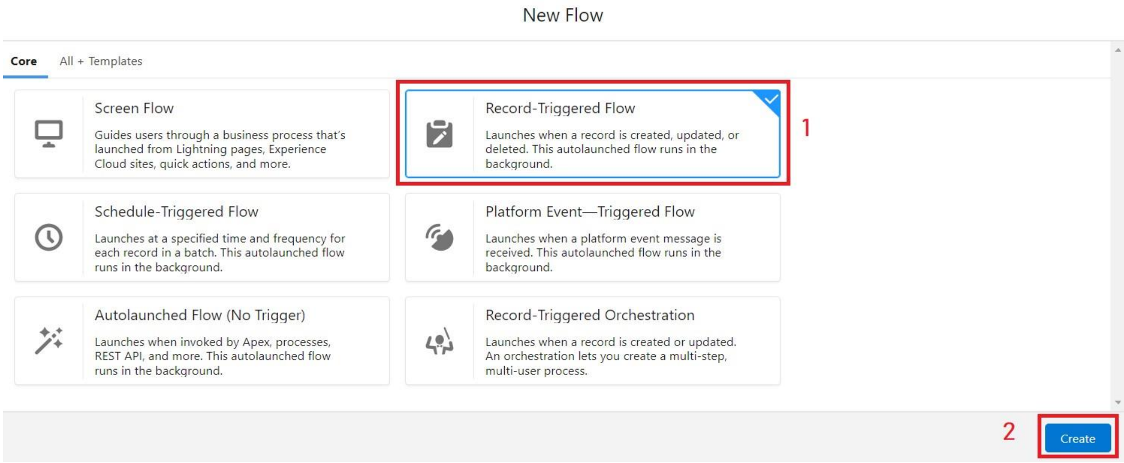


- ## Email Alert Flow
  - `RecipientAddressList: {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}`



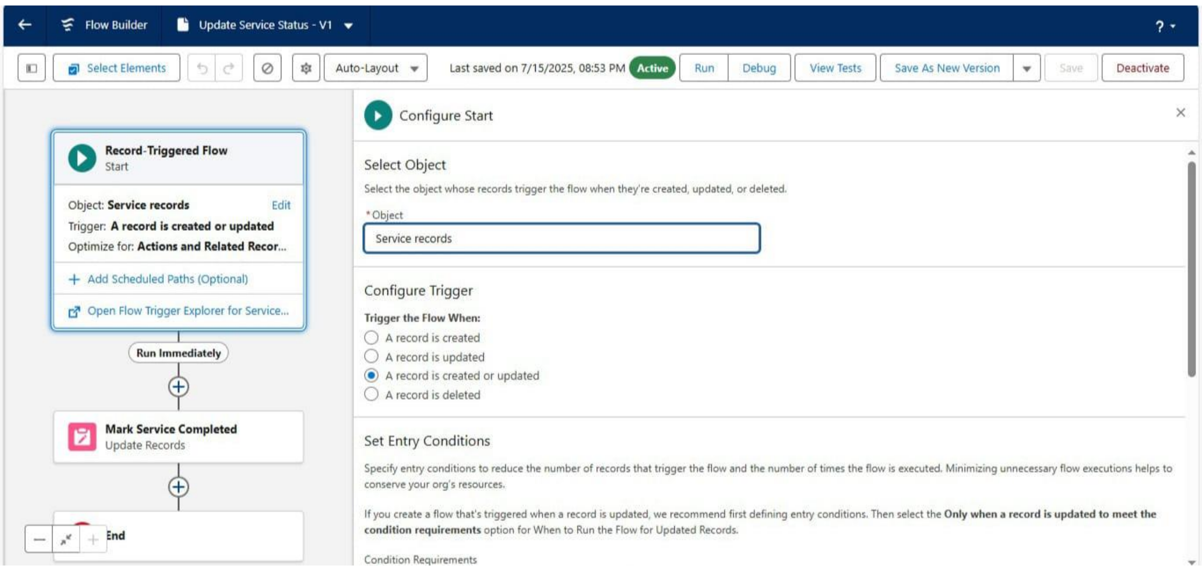
Create Another Flow:

1. Select the Record-triggered flow and Click on Create.



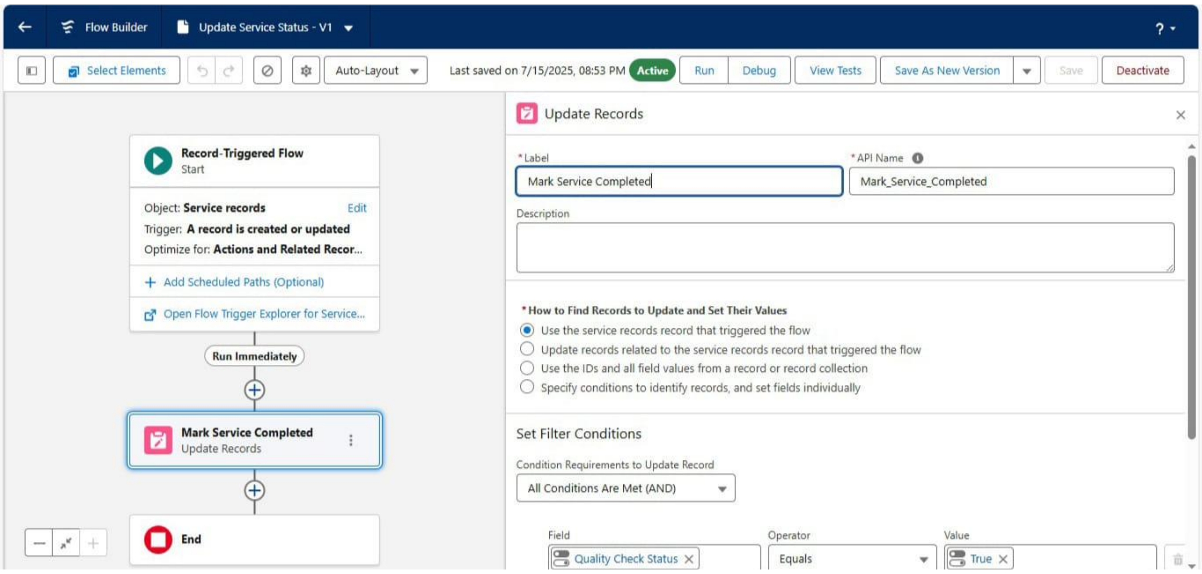
2.

3. Select the Object as " Service records"in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimise the flow for: "Actions and Related Records"



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

1. Field : Quality_Check_Status__c
2. Operator : Equals
3. Value : True
4. And Set Field Values for the Billing details and feedback Record
5. Field : Service_Status__c
6. Value : Completed



Activity – 14

Triggers

1. 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;
            }

    }
    }
}
```

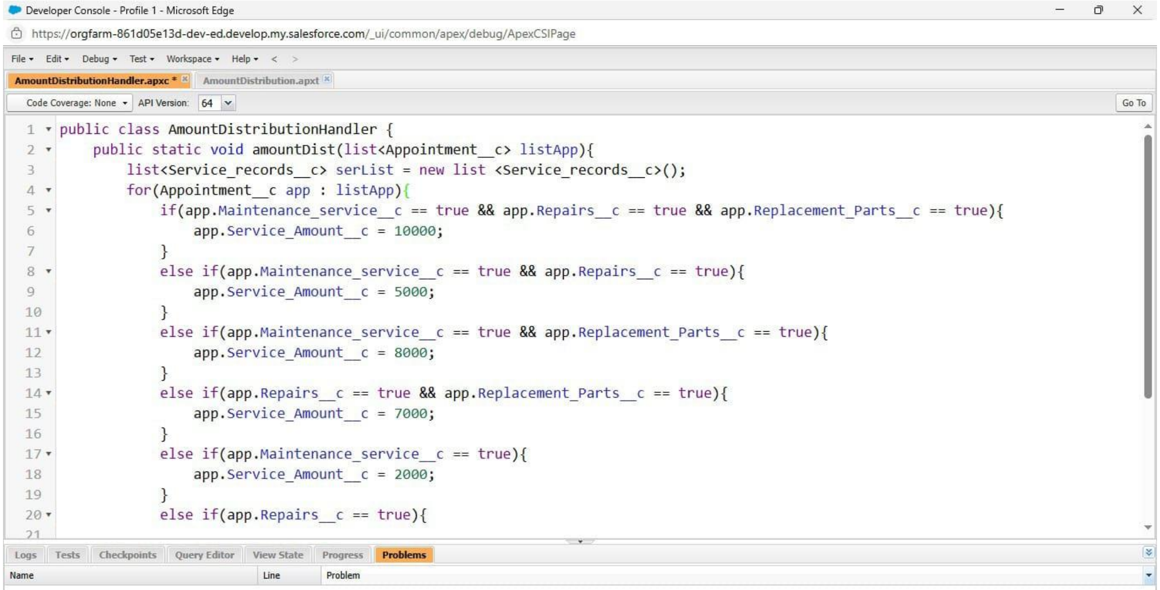
Trigger Handler :

1. Name : AmountDistribution
2. sObject : Appointment__c

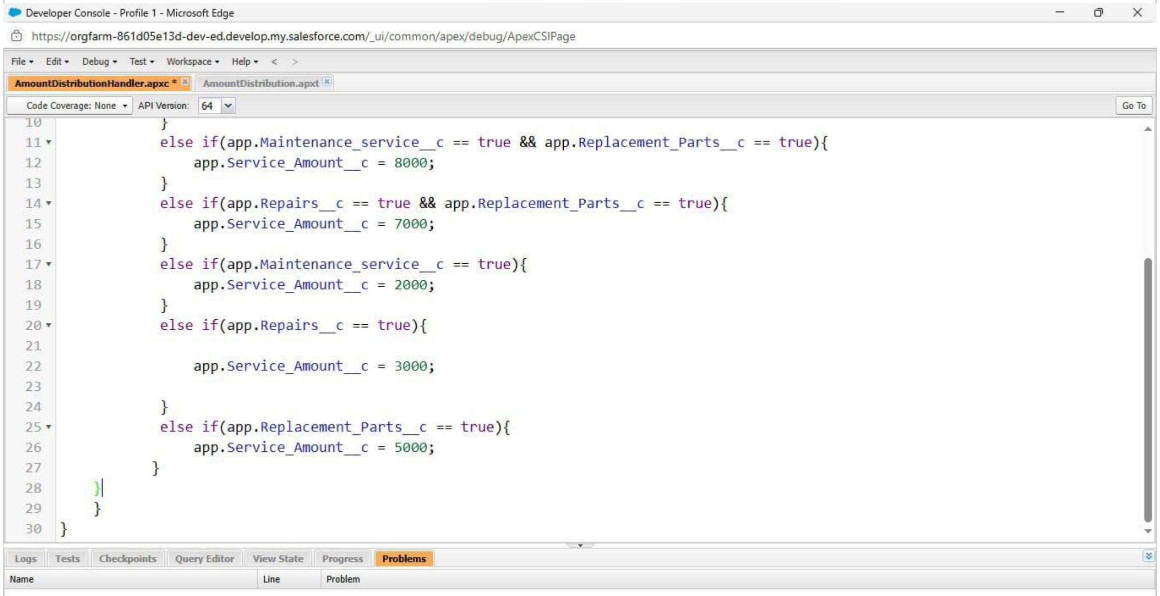
Code:

```
trigger AmountDistribution on Appointment__c (before insert, before
update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }


}
```

- **AmountDistributionHandler.apxc (Class)**

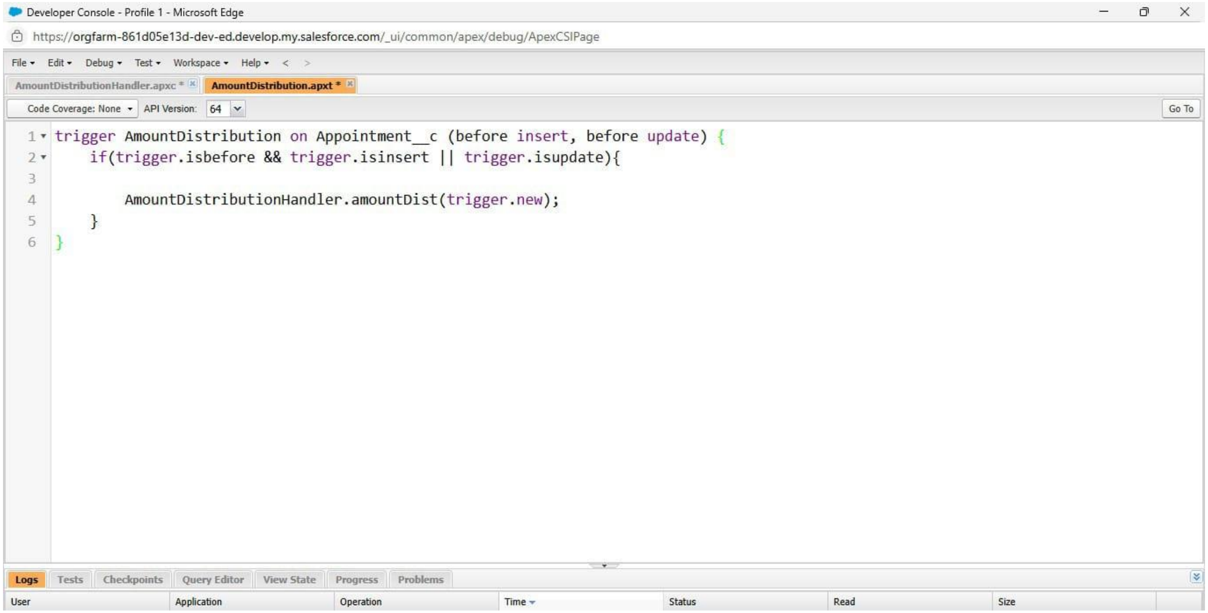



```
public class AmountDistributionHandler {
    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();
        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true && 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){
```




```
            }
            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;
            }
        }
    }
}
```

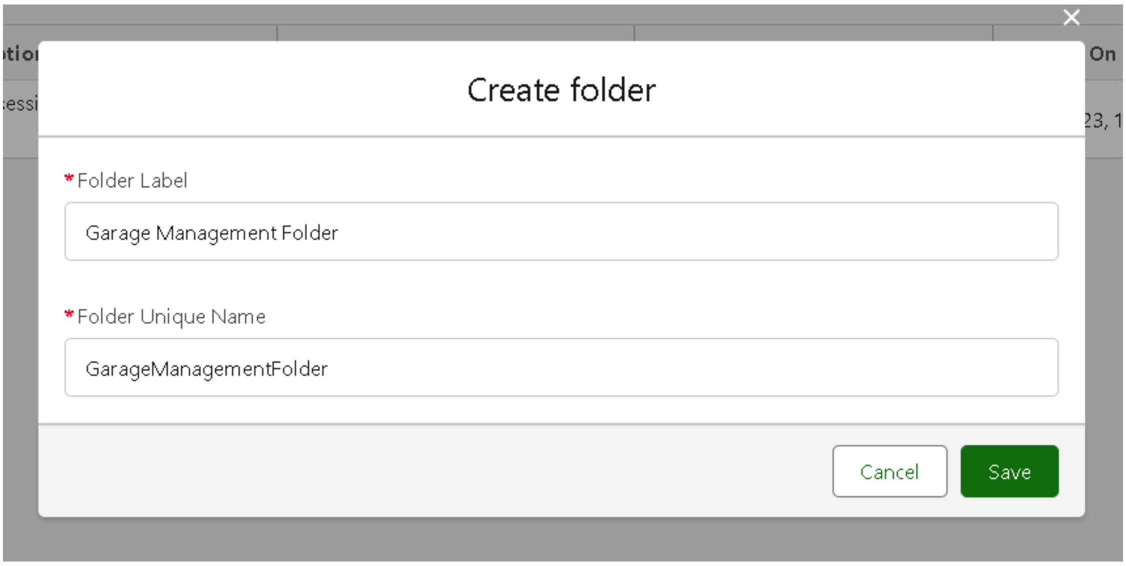
- **AmountDistribution.apxt (Trigger)**



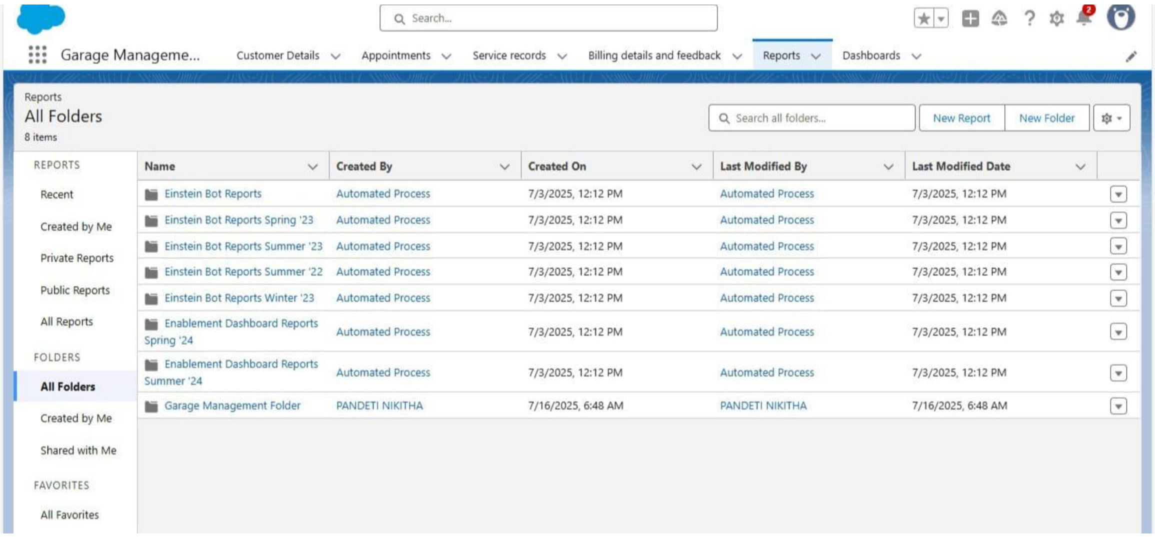
Activity – 15

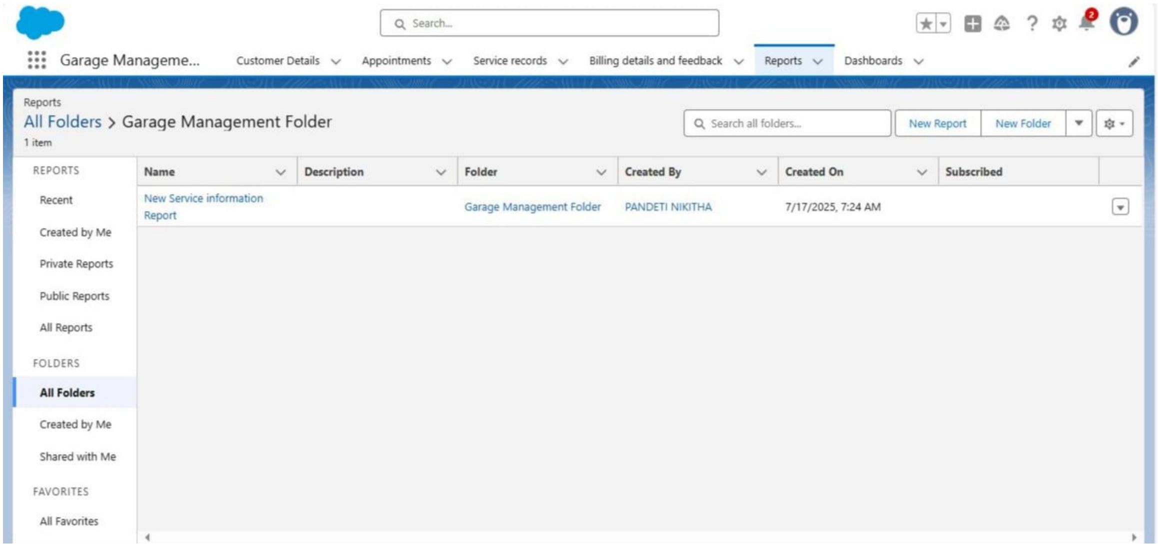
Reports

- ### Report Folder Creation
- Give the Folder label as "Garage Management Folder", Folder unique name will be auto populated.



-

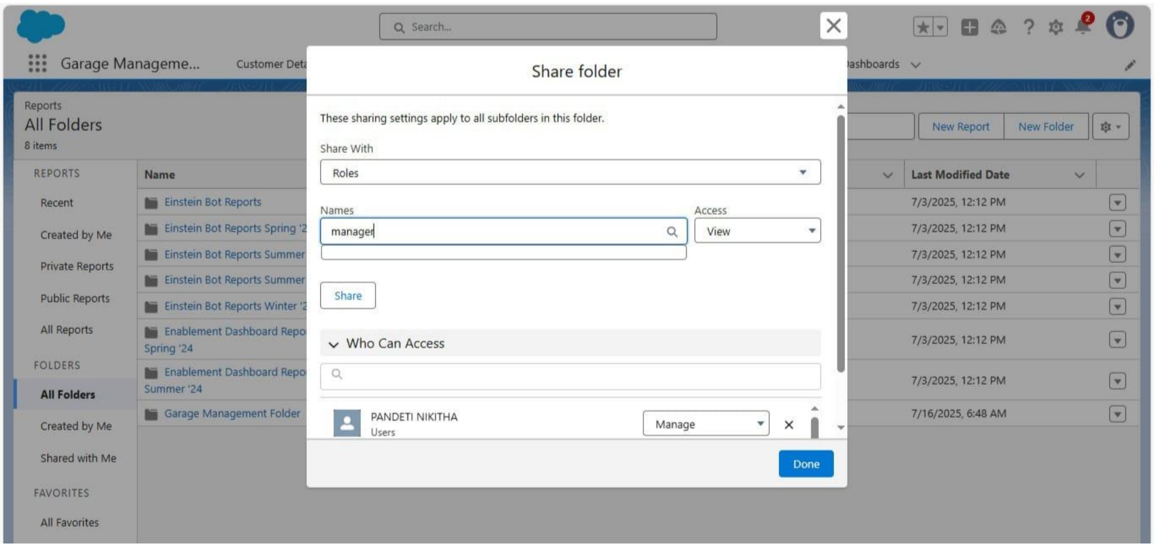




-

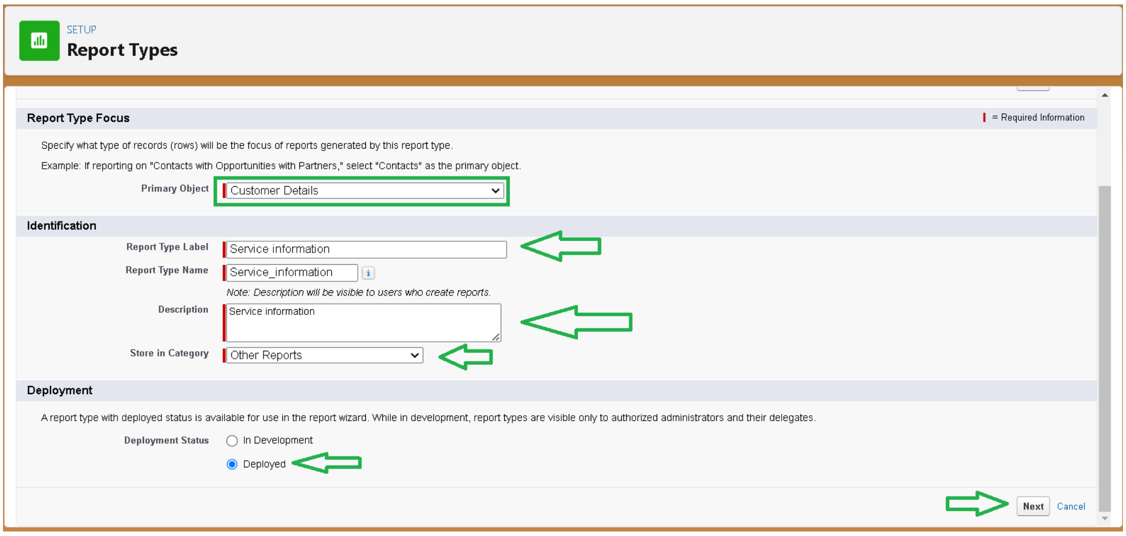
- Sharing a folder:

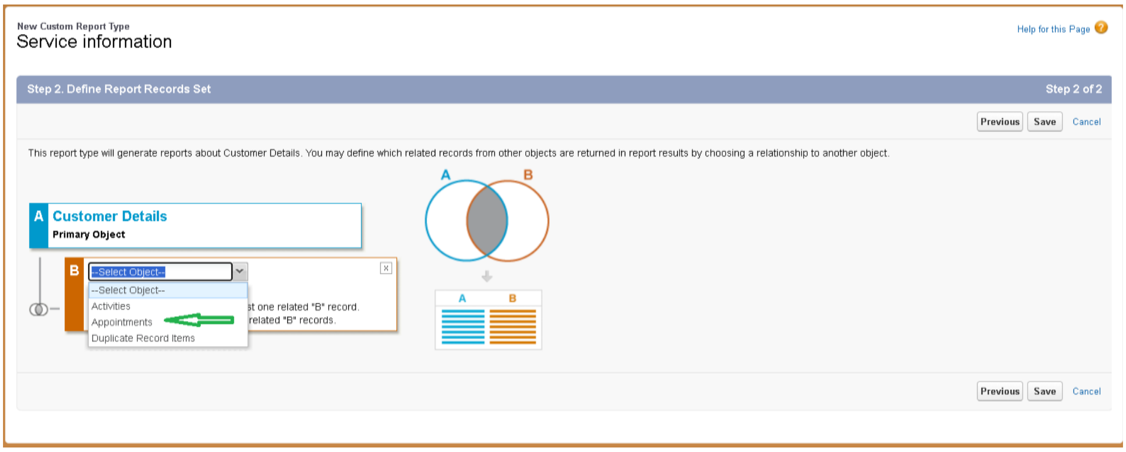
- Select the share with as "roles", in name field search for "manager", give "view" as access for that role.

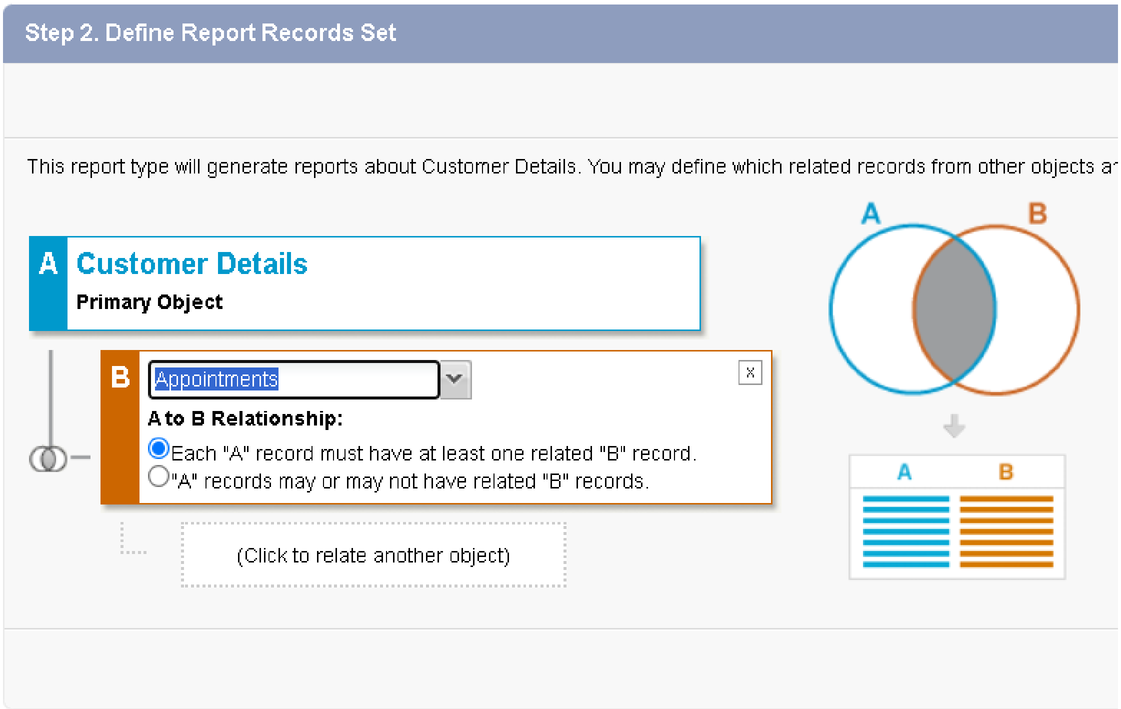


-

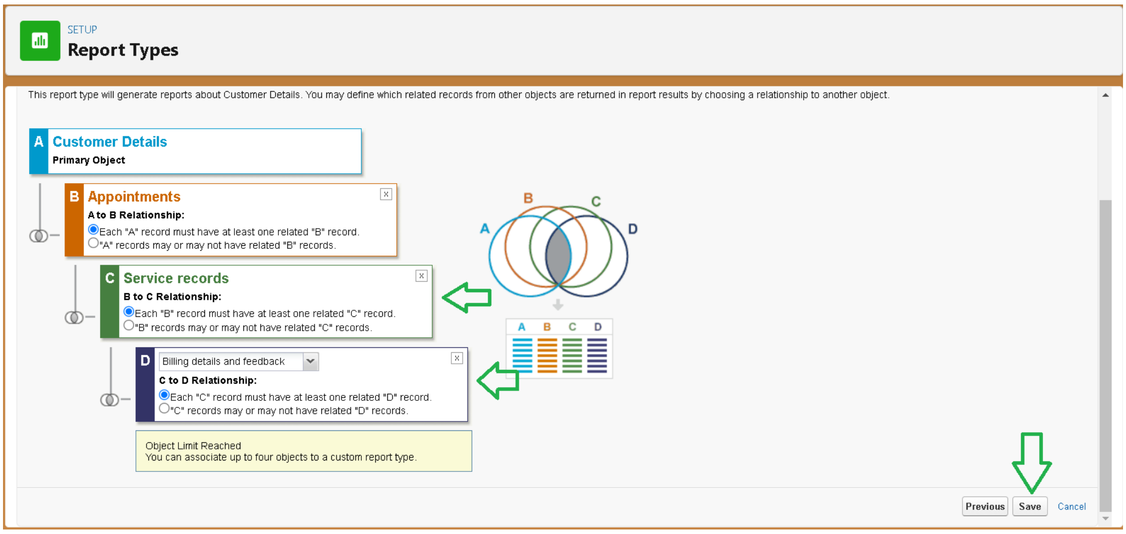
- Report Types:

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.
7. 

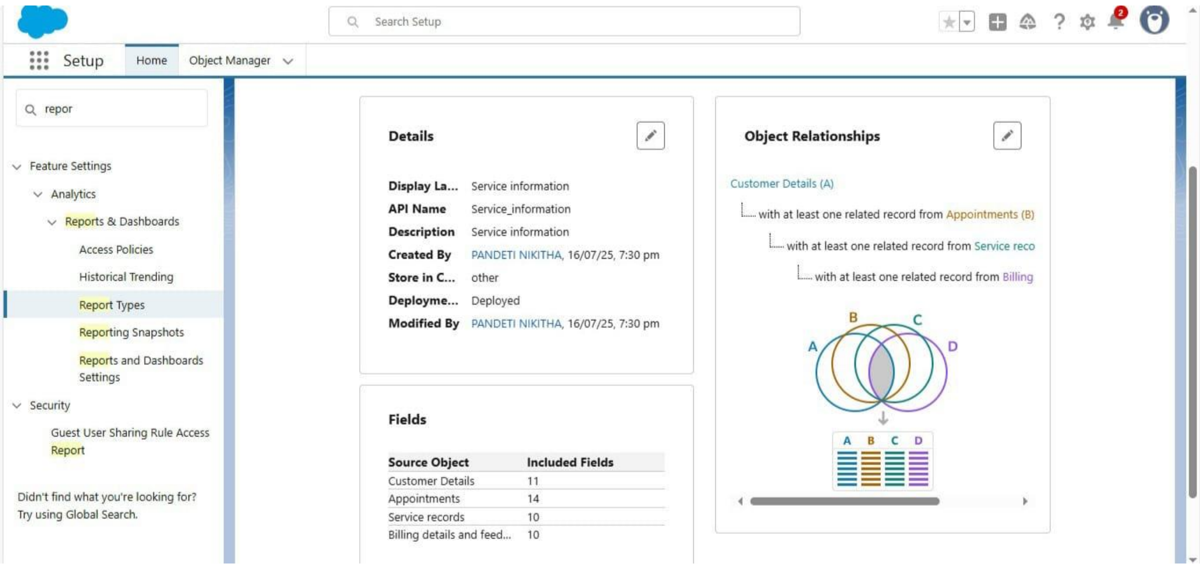
8. now , Click on Related object box.
9. Click on Select Object, choose Appointment Object as shown in fig.
10. 



11.
12. Again Click to relate another object.
13. And select the related object as “ service records”.
14. Repeat the process and select the related object as “ Billing details and feedback”.

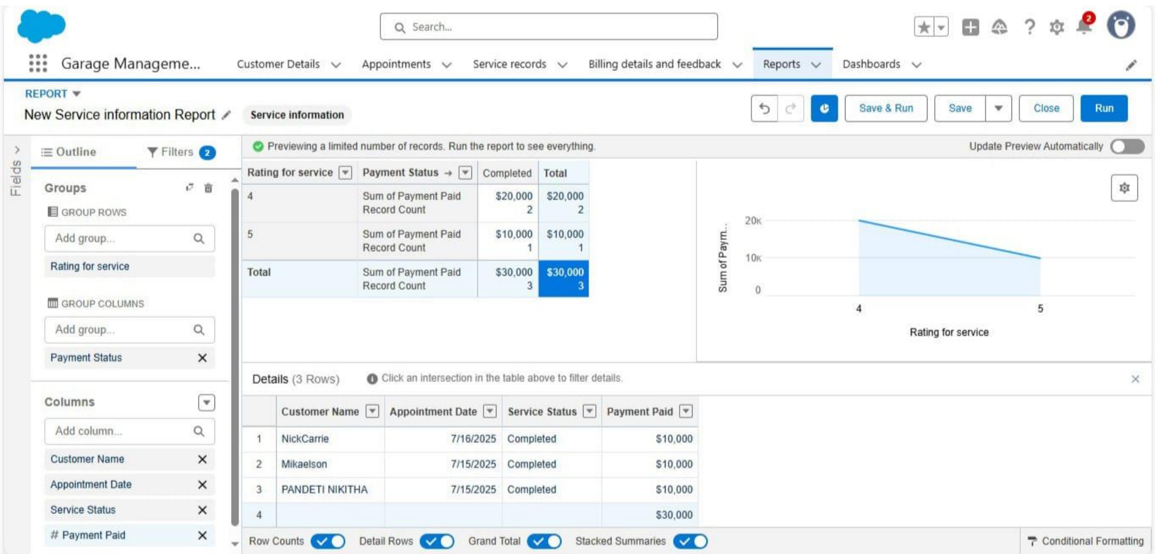


15.



- **New Service Information Report**
- In column section.
    - Customer name
    - Appointment Date
    - Service Status
    - Payment paid
- In GROUP ROWS section.
    - Rating for Service
- In GROUP COLUMN section.
    - Payment Status
- Click on Add Chart , Select the Line Chart.

-

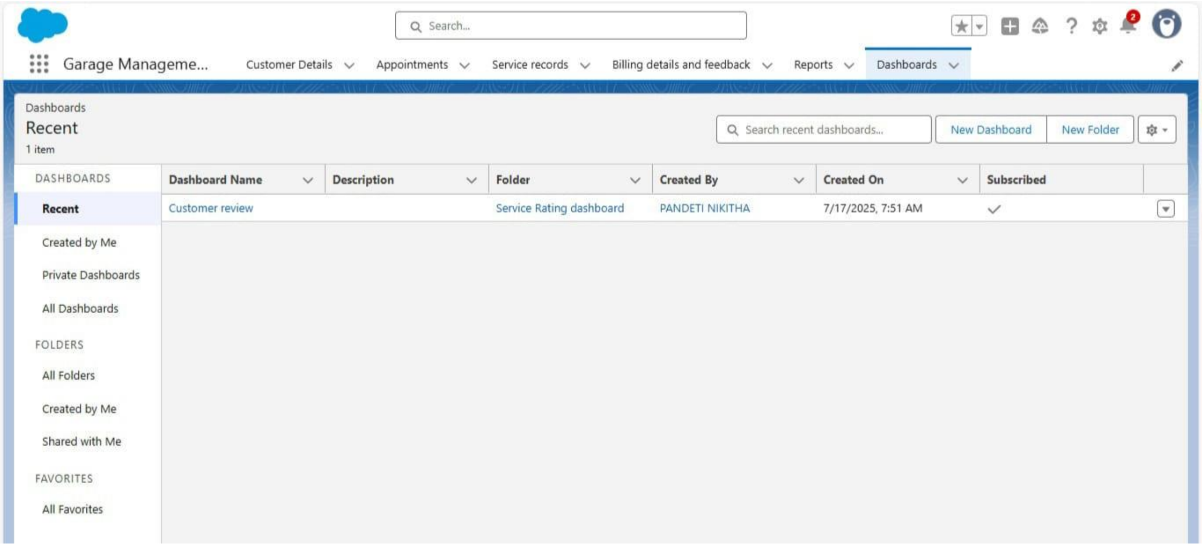


Activity – 16

DashBoard: Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports.

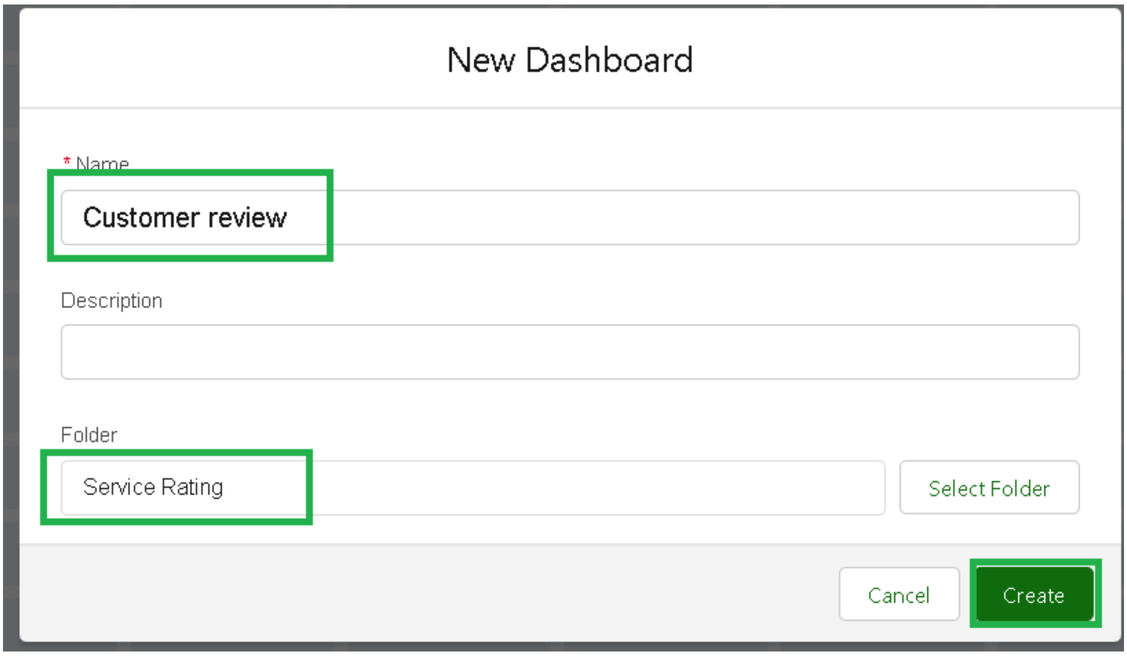
Create DashBoard Folder:

- The folder label as " Service Rating dashboard".

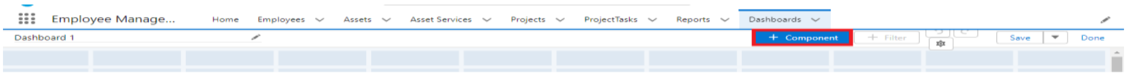


Create DashBoard:

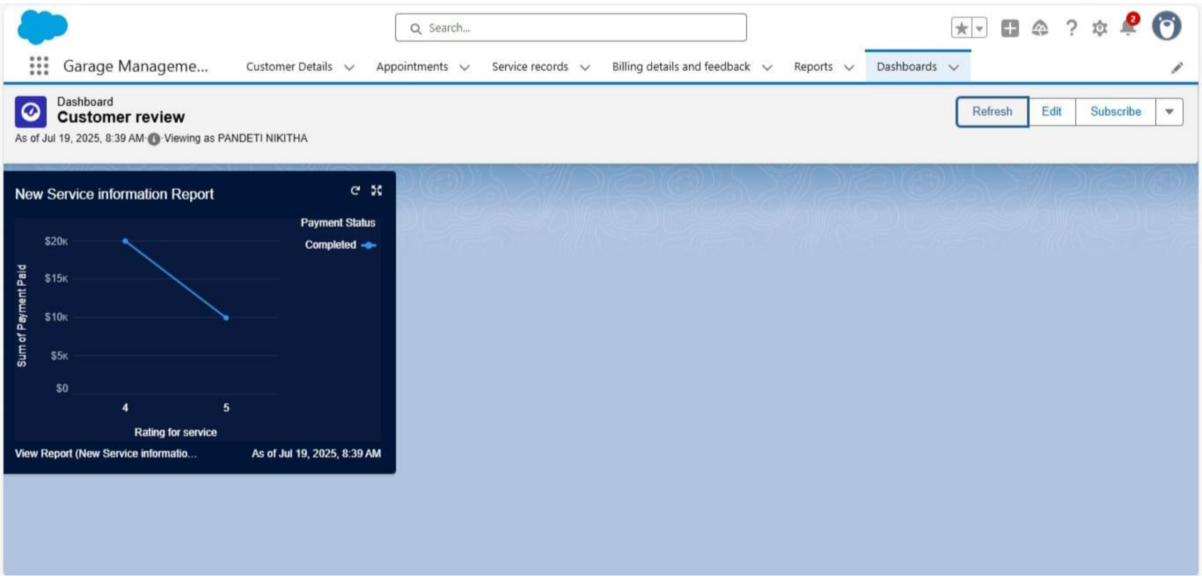
1. Give a Name and select the folder that created, and click on create.



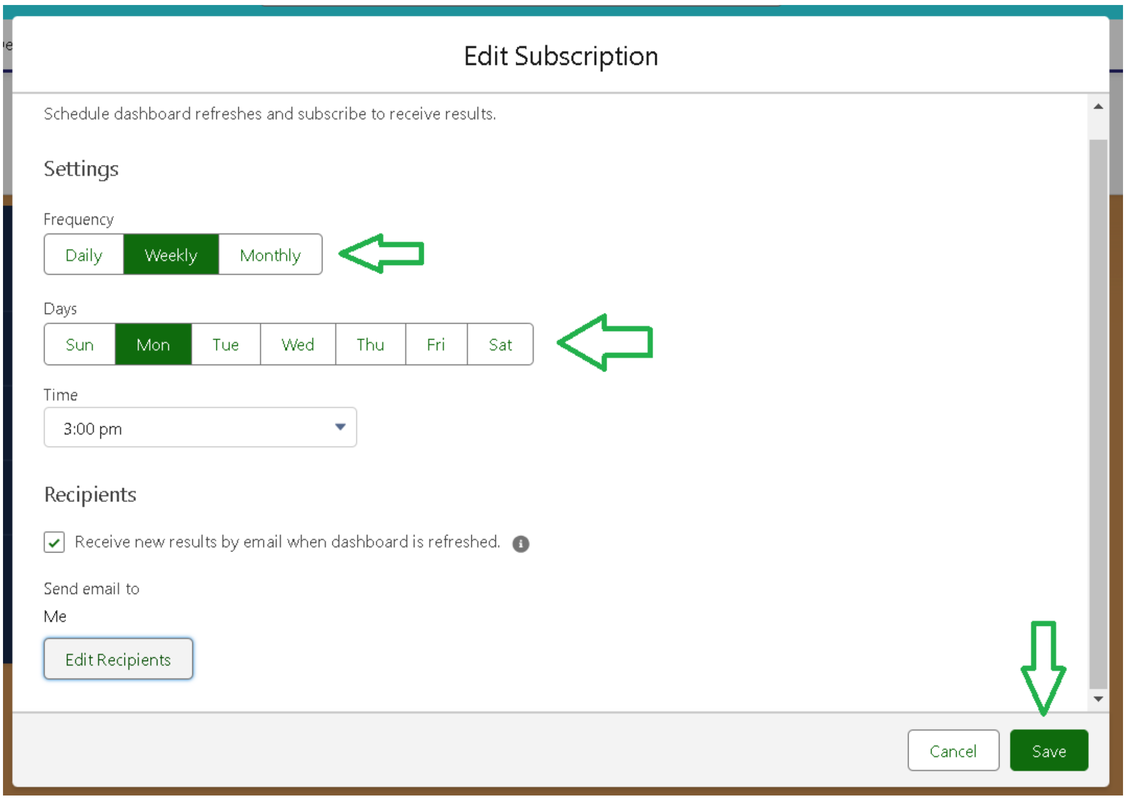
3. Select add component.



4. Select a Report and click on select.
5. Select the Line Chart. Change the theme.



6. Set the Frequency as " weekly ".
7. Set a day as monday.



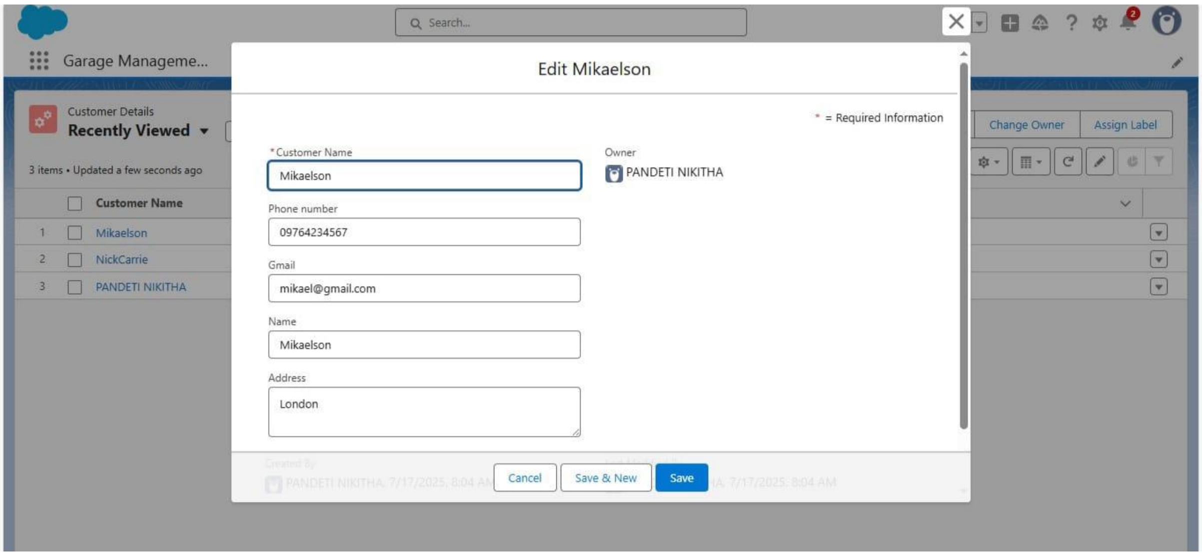
Activity – 17

User Adoption:-

-Creating Records

- Contact Details:

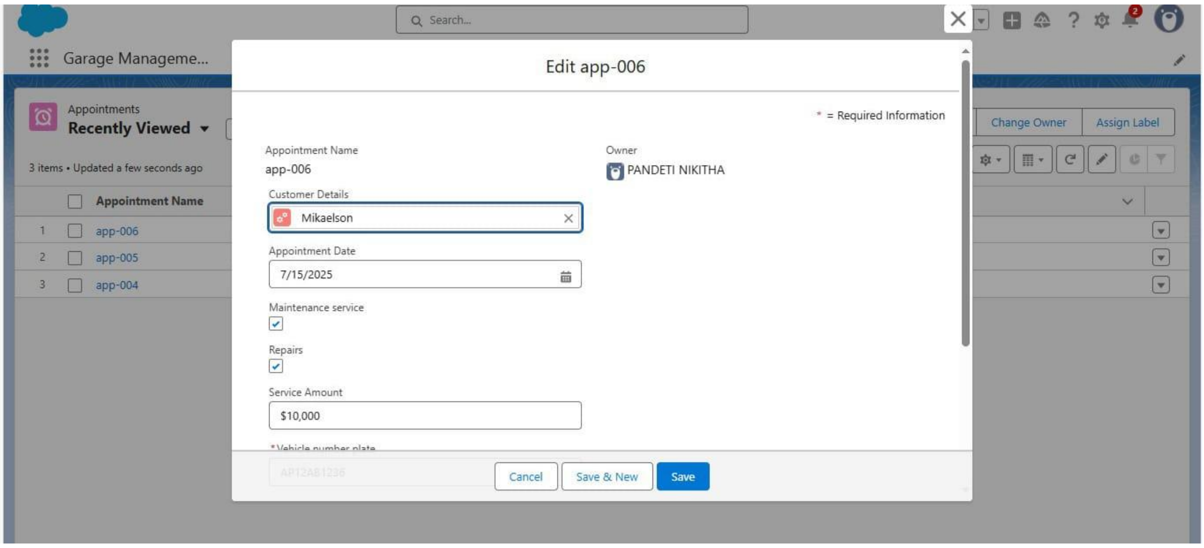
1. Click on the app launcher located at the left side of the screen.
2. Search for " **Garage Management**" and click on it.
3. Click on the " **Consumer details** tab".
4. Click on new and fill the details as shown below figs, and click save.



- Appointment Details:

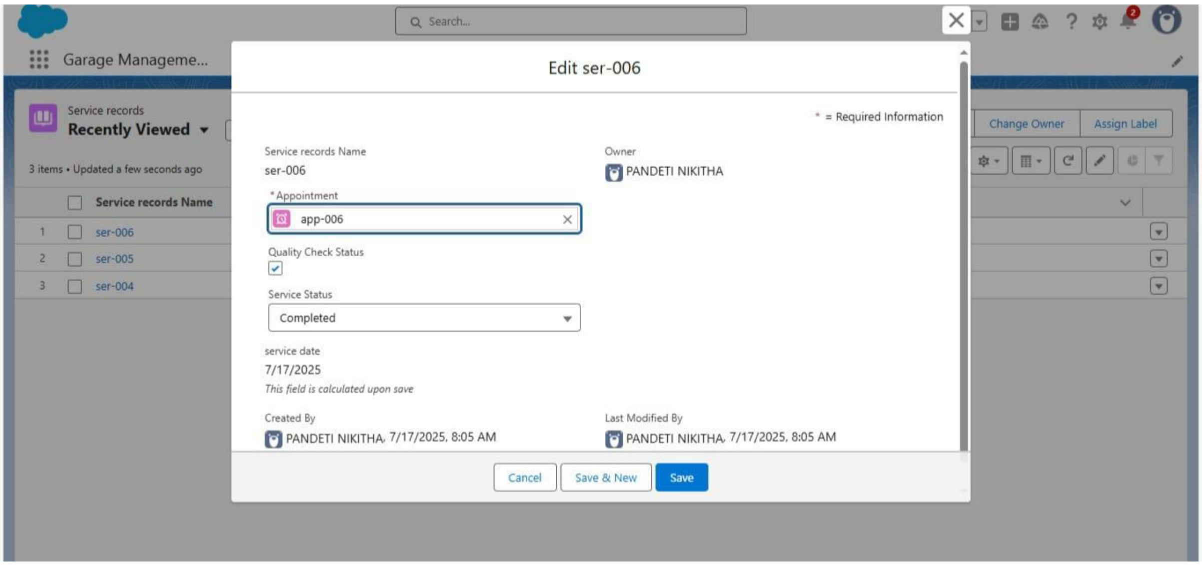
1. Click on the "**Appointment** tab".
2. Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
3. Match the validation while entering the vehicle number plate.

4. Select the services you need.
5. Click on save to see the Service Amount.



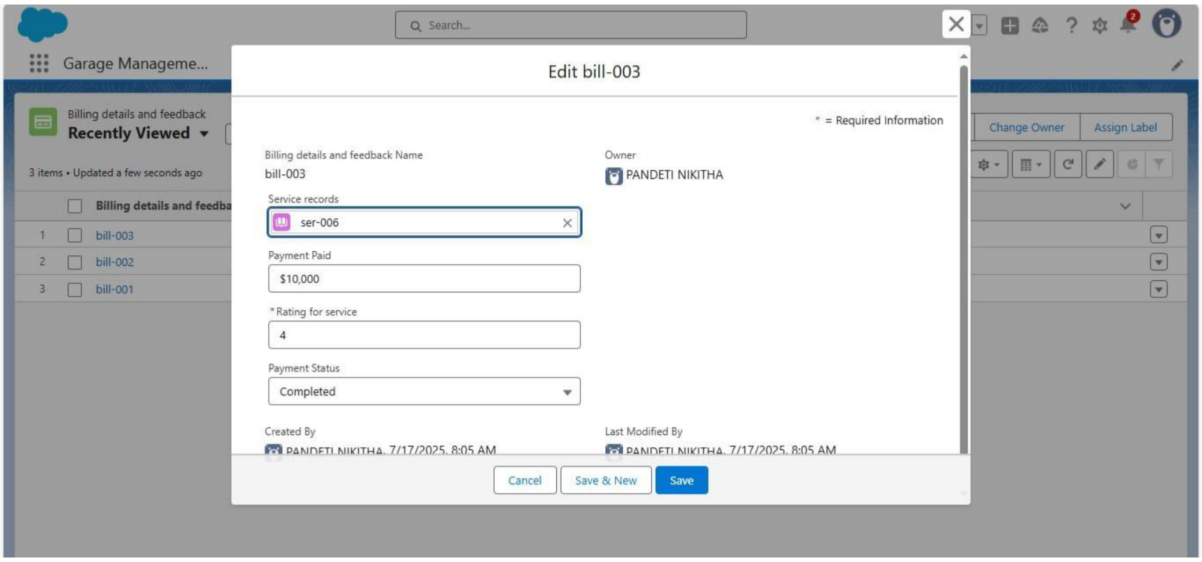
- Service Record Details:

1. Click on the "**Service record** tab".
2. Enter the Appointment, and started is selected as default.
3. Click on save.

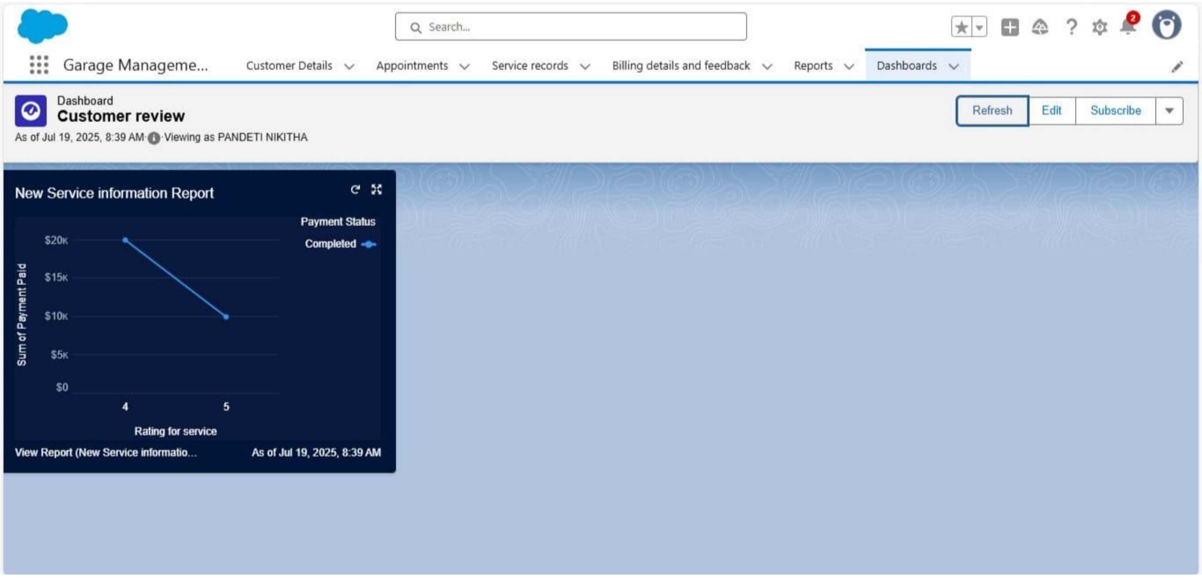


- Billing&Feedback Details:

1. Open the record and click on Quality check status as true.
2. Click on save.



Final Output:



CONCEPTS UTILIZED

(FOR OUR PROJECT IMPLEMENTATION)

- **What is SALESFORCE?**

  Salesforce is like a digital hub that helps businesses stay connected with their customers. Imagine having a central place where a company can keep track of every conversation, email, and sale, so no detail slips through the cracks. It's built to bring together sales, customer support, and marketing efforts, so everyone is on the same page.

With Salesforce, businesses can customize tools to fit their needs, automate repetitive tasks, and use built-in AI to get insights on what customers want. It’s essentially a smart assistant for companies to build stronger, more personalized customer relationships.

- **Objects**

  Objects are database tables that permit you to store data that is specific to an organization.

  Types:-

  1. Standard Objects
     - Provided by salesforce
  2. Custom Objects
     - Created by users as per their requirements and flexibilities.

- **Tabs**

  Tabs are navigation links that give users quick access to different objects, features, or data, like accounts, contacts, or dashboards, within the platform.

- **Lightning App**

  A Lightning App in Salesforce is a tool that lets users build easy-to-use, interactive apps to streamline work and improve how they interact with data.

- **Fields**

Fields in Salesforce are like labelled blanks in a form where you fill in specific details like a person's name, phone number, or Gmail, so that all relevant information is organized and easy to find.

- **Validation & Duplication Rules**
  - Validation rules in Salesforce are conditions that ensure data entered meets specific criteria before being saved, helping maintain data quality and accuracy.
  - Duplication rules in Salesforce are settings that help prevent or manage duplicate records by identifying and blocking or alerting users about potential duplicates during data entry.
- **Profiles**

  Profiles define what a user can see and do within the platform, acting like "permission sets" for different roles.
- **Role & Role Hierarchy**

  Roles define a user's level of access to data and functions, creating a hierarchy that helps organizations manage who can see and edit information, ensuring everyone has the right permissions while still promoting collaboration across teams.
- **Users**

  Users are people who can log in to the platform to manage customer information, track sales, and work with their teammates, helping them do their jobs better.

- **Public Groups**

  Public groups in Salesforce are like team huddles that bring users together, making it easier to share information and collaborate on projects seamlessly.

- **Sharing Settings**

  Sharing settings determine how and with whom you share data within your organization, ensuring the right people have access to the right information to collaborate effectively.

- **Flows**

  Flows are powerful tools that allow users to automate complex business processes by visually designing step-by-step workflows that guide users or update records without needing to write code.

- **Apex Triggers**

  APEX Triggers are custom pieces of code that automatically execute before or after specific events occur in the database, helping businesses automate tasks and enforce rules without needing manual input.

- **Reports**

  Reports are tools that help businesses visualize and analyse their data, making it easier to understand trends and track performance at a glance.

- **Dashboard**

  Dashboards are visual displays that summarize key metrics and data, helping businesses quickly see their performance and make informed decisions at a glance.

TEST AND VALIDATION

- **Approach to Testing:**

  For the Garage Management System, our testing strategy ensures everything works smoothly and meets user needs. We'll combine both automated and manual testing methods, including:

  1. Functional Testing
  2. Integration Testing
  3. User Acceptance Testing (UAT)
  4. Performance Testing

- **Unit Testing** **(Apex Classes & Triggers):**

  Unit testing will focus on individual components, like Apex classes and triggers, to ensure they function correctly. Key elements include:

  1. **Positive Tests:** To confirm expected functionality under normal conditions.
  2. **Negative Tests:** To check how the system handles errors or unexpected inputs.
  3. **Bulk Tests:** To ensure the system can handle large amounts of data without issues.

CONCLUSION

In conclusion, the Salesforce Garage Management System is a powerful tool that makes running a service centre easier and more efficient. It combines features like online appointment scheduling, real-time updates on car repairs, and smart inventory management, all in one place. This means customers get faster service and clear communication, which helps keep them happy. For the staff, it reduces confusion and makes their jobs smoother.

Plus, the system gives managers valuable insights into performance, allowing them to spot trends and improve the business. Overall, this system helps garages provide excellent service and build strong relationships with their customers.

Thank You