

GARAGE MANAGEMENT SYSTEM

COLLEGE NAME : AVP COLLEGE OF ARTS AND SCIENCE

COLLEGE CODE : bruj

Team ID : NM2025TMID25414

Team Members : 4

Team Leader : VAISHNAVI.K

Email ID:vaishnavikathiravan1@gmail.com Team
member : ABINAYA.R

Email ID : abinayaravi701@gmail.com

Team member:SABITHA.C

Email ID: sabithachellamuthu@gmail.com

Team member : THARANIKA.R.S

Email ID: tharanikashanmugam50@gmail.com

INTRODUCTION

Project Overview

1.INTRODUCTION

1.1ProjectGarageManagement System


The Garage Management System (GMS) is a software tool designed to improve the efficiency of automotive repair shops. It manages customer details, service records, billing, and job tracking in a structured way. By reducing manual work and errors, it saves time and increases productivity. With its user-friendly interface, GMS ensures better service quality and customer satisfaction.

DEVELOPMENT PHASE

Creating Developer Account:

Creating a developer org in Salesforce By using this URL


<https://developer.salesforce.com/signup>



Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs



Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud


| | |
|---------------------------------------------|-------------------------------------------------|
| First name | Last name |
| <input type="text" value="VAISHNAVI"/> | <input type="text" value="K"/> |
| Job title | Work email |
| <input type="text" value="SOFTWARE"/> | <input type="text" value="vaishnavikathirava"/> |
| Company | Country/Region |
| <input type="text" value="AVP COLLEGE OF"/> | <input type="text" value="India"/> |

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

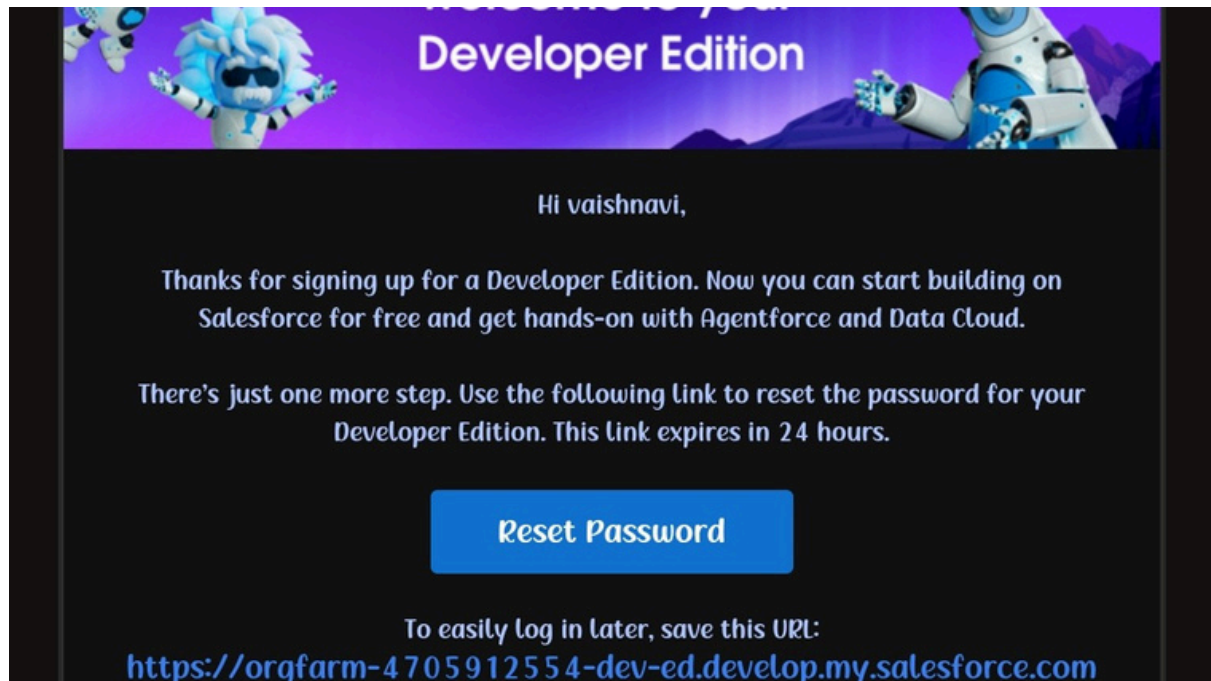
We value your privacy. To learn more, visit our [Privacy Statement](#).

☐ I'm not a robot

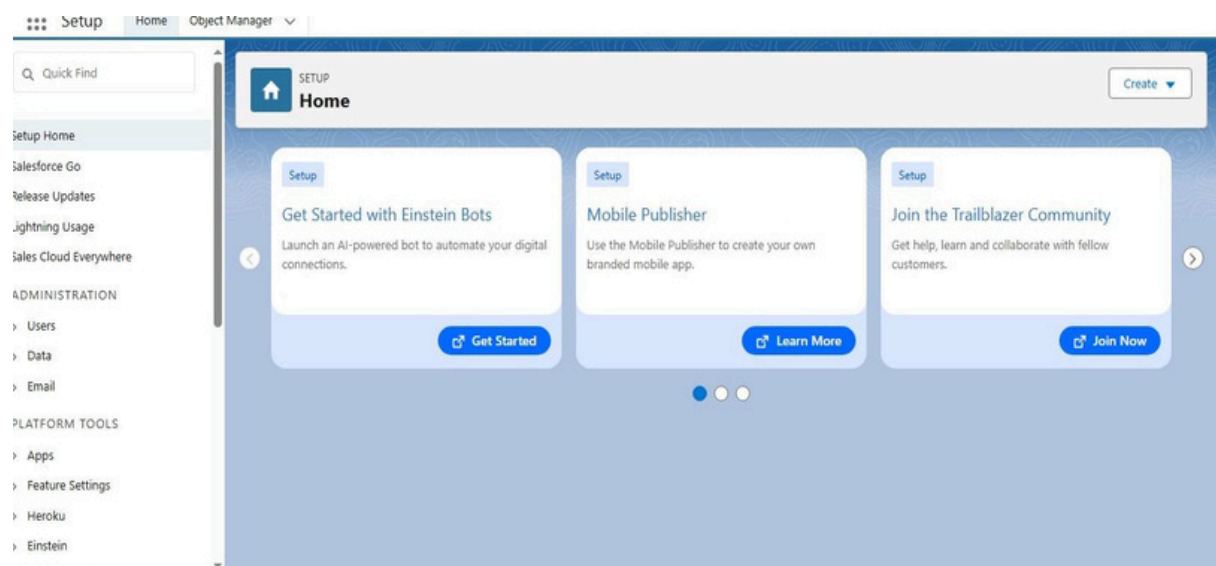

reCAPTCHA
Privacy - Terms

Sign Me Up

1. Give the details for requirement fields
2. Then click the “Sign me up”



3. Click on reset password
4. Give a password and answer the security question and click on change password
5. Then you will redirect your salesforce setup page



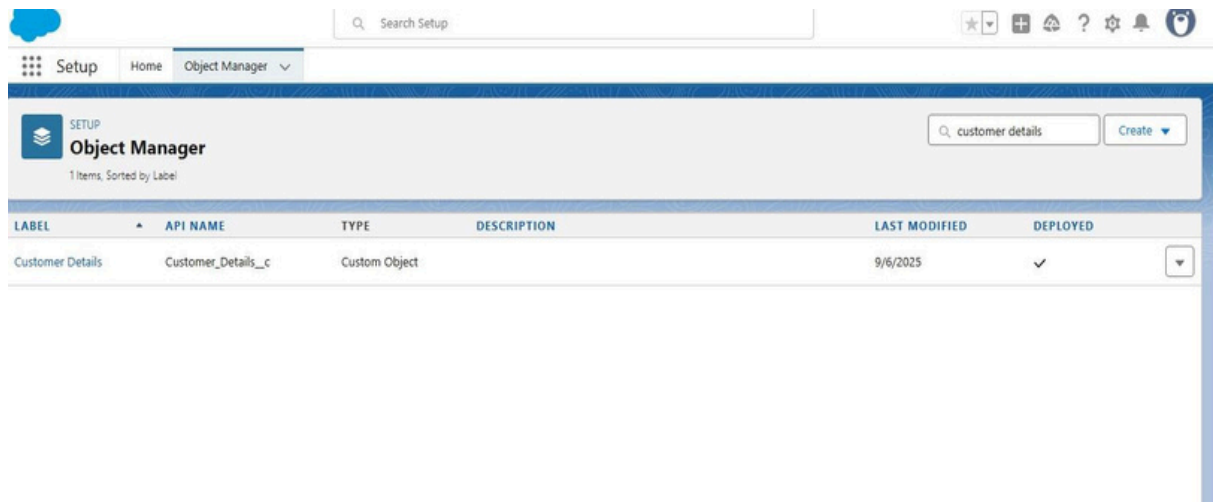
Create the objects:

- Create the customer detail object

The screenshot shows the 'New Custom Object' page in Salesforce Setup. The page has a navigation bar with 'Setup', 'Home', and 'Object Manager'. The main heading is 'New Custom Object'. Below this is a warning message: 'Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)'. The 'Custom Object Definition Edit' section contains the following fields:

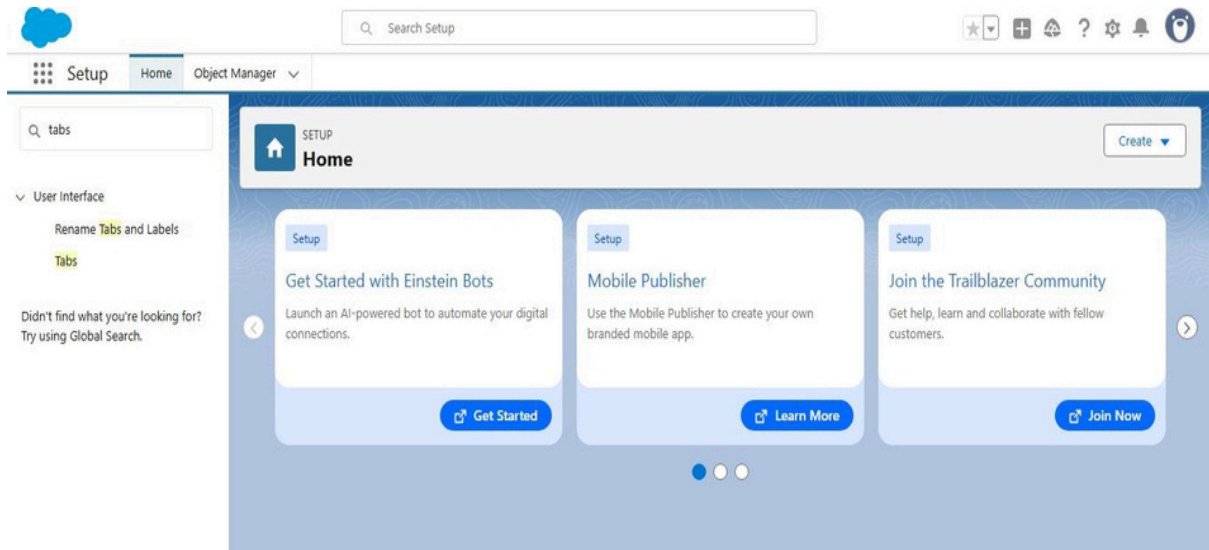
- Label:** Customer Details (Example: Account)
- Plural Label:** Customer Details (Example: Accounts)
- Starts with vowel sound:** ☐
- Object Name:** Customer_Details (Example: Account)
- Description:** (Empty text area)

- From the setup page >> click on object manager>>click on create>>click on custom 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
Click on Allow reports and Track Field History,
Allow search >> Save.
- You will see your created object in object manager



- Follow the same procedure to create the objects named Appointment, Service records and Billing details and feedback.

Create the Tabs :



- 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

SETUP
Tabs

Custom Tabs Help for this Page

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

| Action | Label | Tab Style | Description |
|--------|----------------------------------------------|-------------|-------------|
| | Appointments | Airplane | |
| | Billing details and feedback | Can | |
| | Customer Details | Alarm clock | |
| | Service records | Books | |

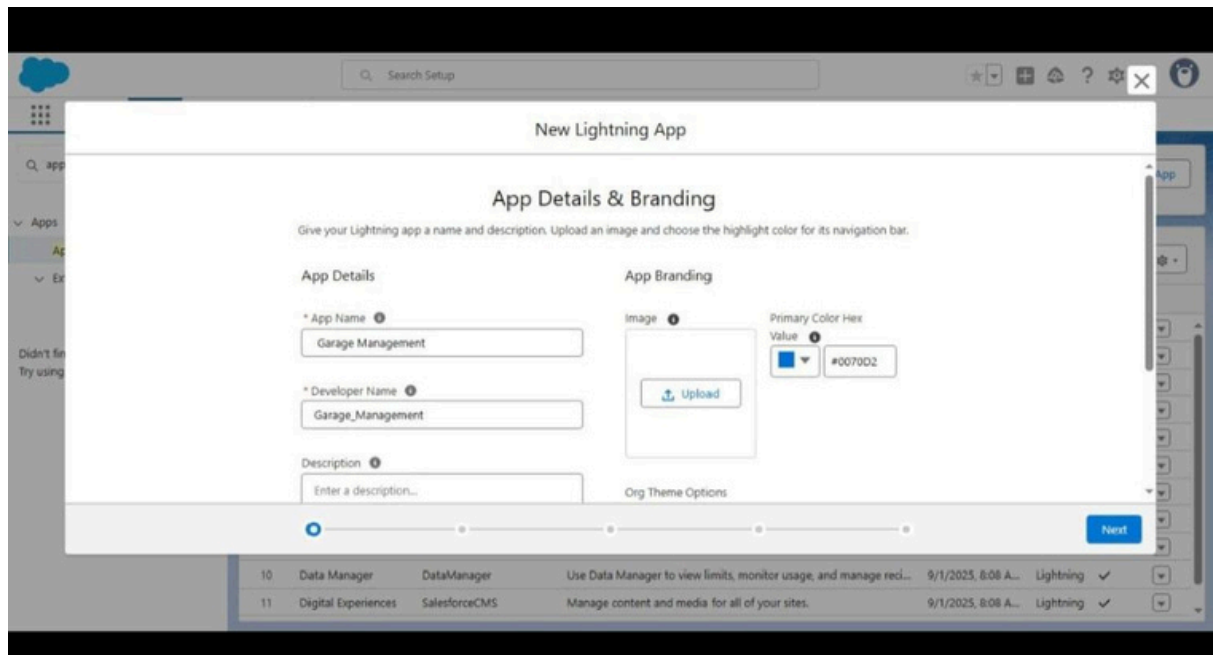
Web Tabs What Is This?

No Web Tabs have been defined

- Now create the Tabs for the remaining Objects, they are “ Appointments, Service records,Billing details and feedback”.
- Follow the same steps as mentioned in Activity -1

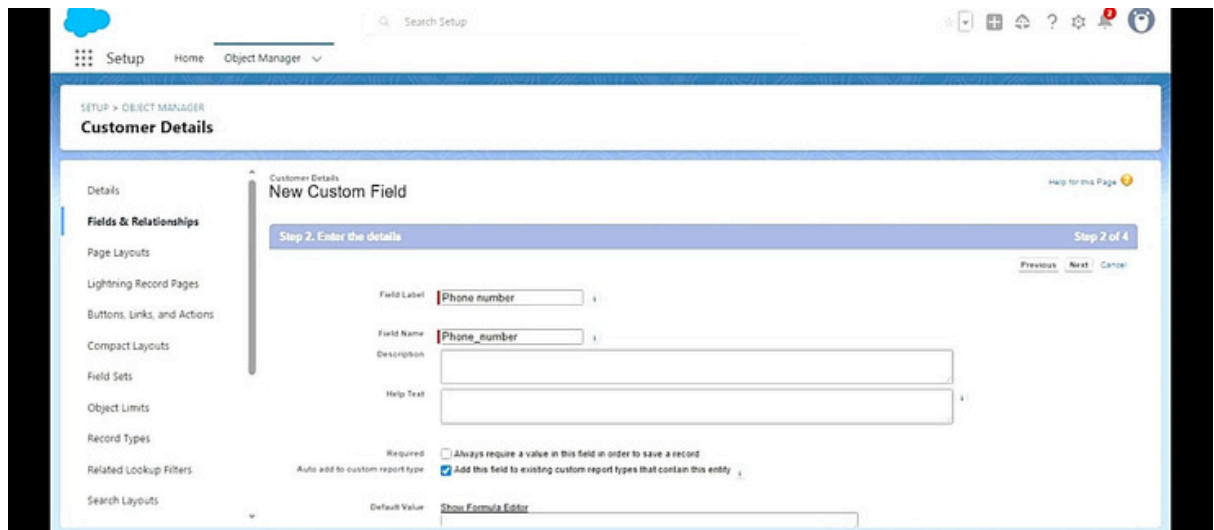
Create the Lightning app

- 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.



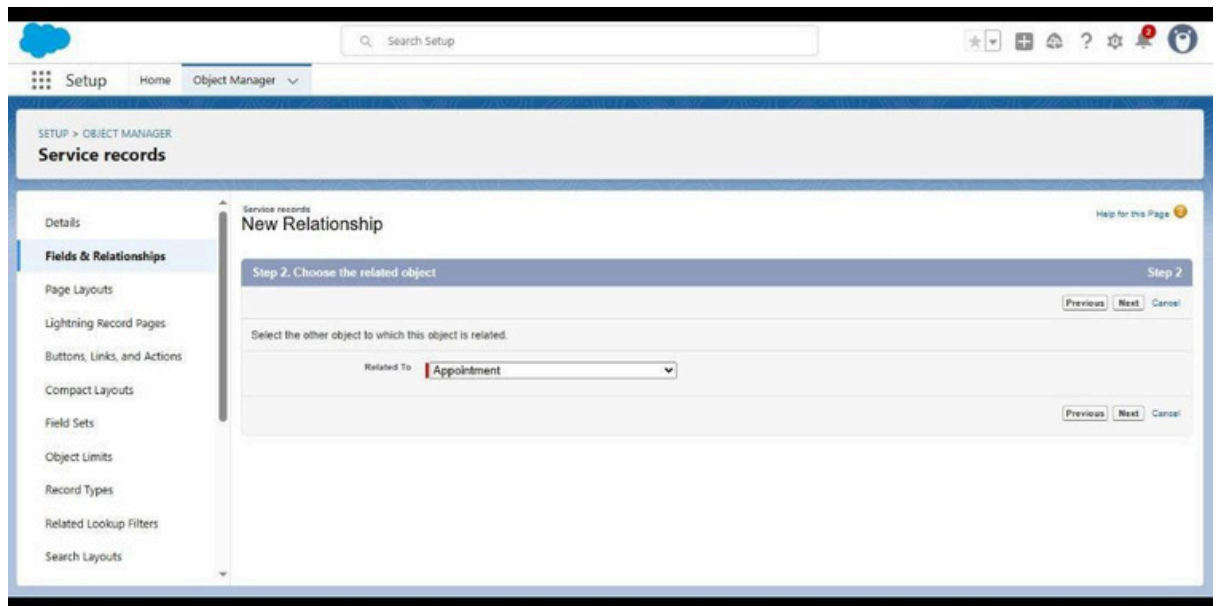
- Then add the navigations items

Create the Fields :

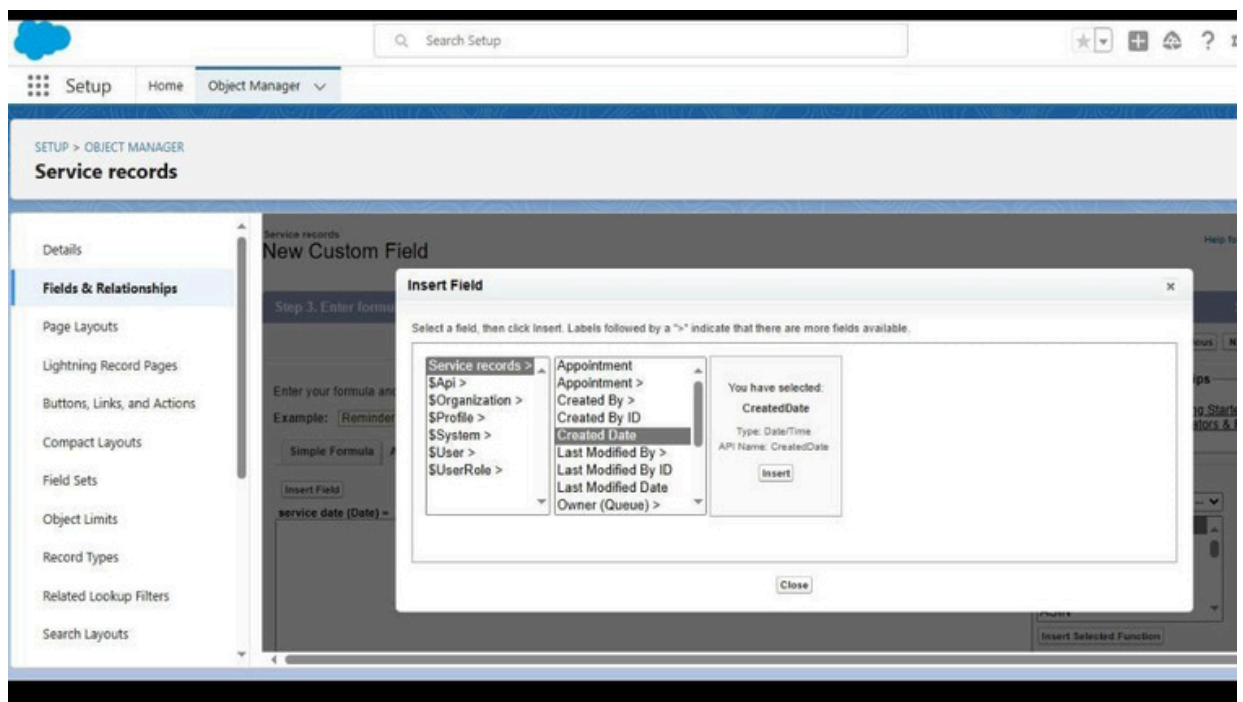


- click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
- Click on Field & Relationships>>New
- Select the data type as “Phone”
- Field Label: Phone number

- Field Name : gets auto generated then Click on Next >> Next >>save & new
- Same like as create the object named “Gmail”
- Then Create the Lookup Field on Service records Object
- Select “Look-up relationship” as data type and click Next
- Select the related object “ Appointment ” and click next

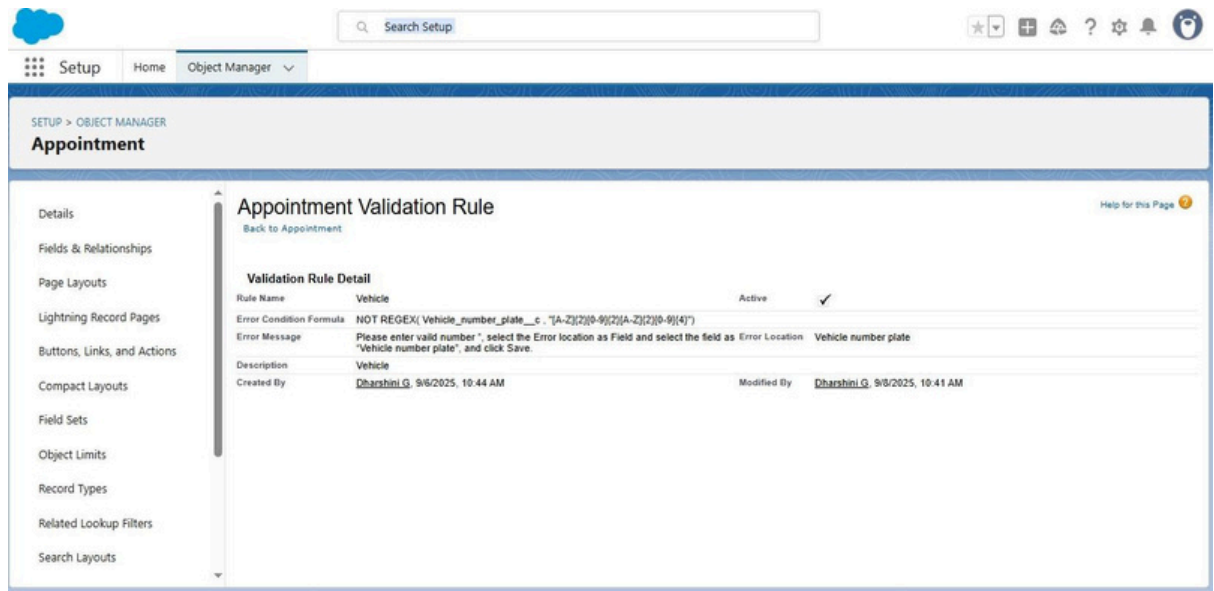


- Make it a required field so click on Required.
- Then create the Checkbox Field, date fields, currency fields, text fields, picklist fields and formula field for service records object

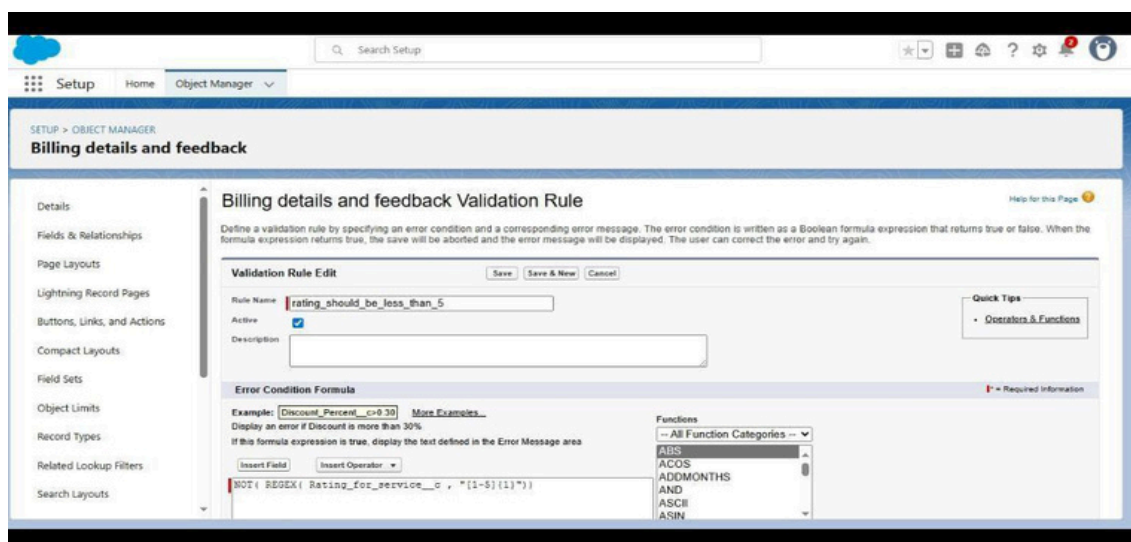


Create a Validation Rule :

- In Appointment object click the validation rules
- 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}"))



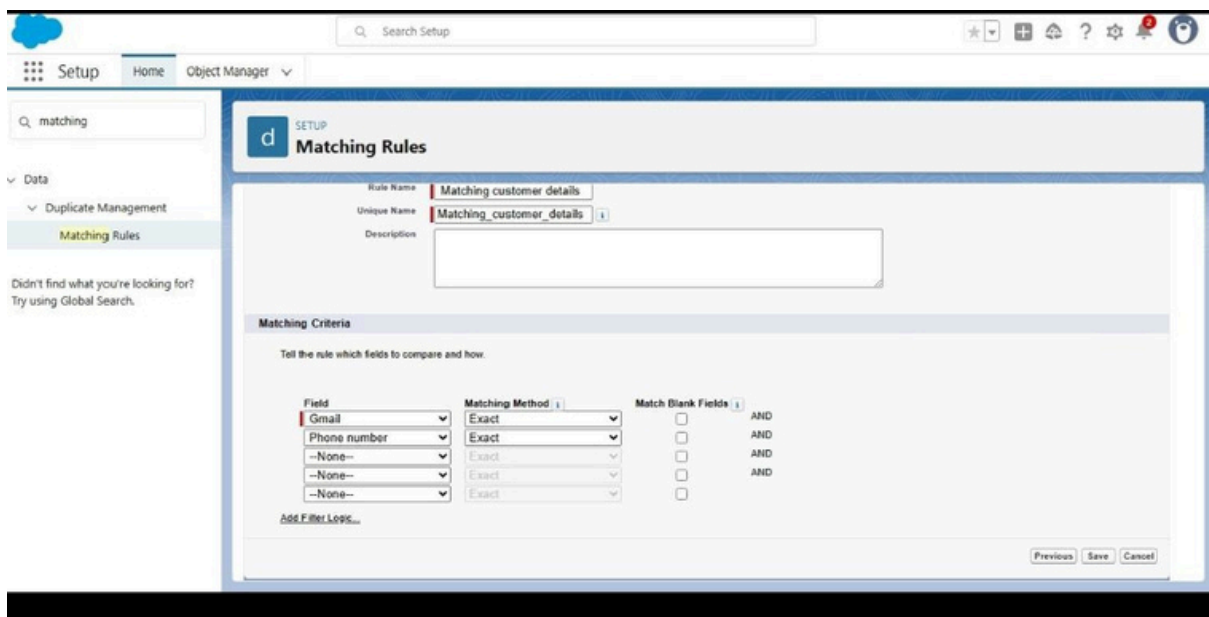
- Same as To create a validation rule to an Billing details and feedback Object
- Enter the Rule name as “ rating_should_be_less_than_5”.
- Insert the Error Condition Formula as : NOT(REGEX(Rating_for_service__c , "[1-5]{1}"))
- Click save.



Create a Duplicate rules:

- In Quick find box search for Matching rule.
- Give the Rule name : Matching customer details
- Define the match criteria as

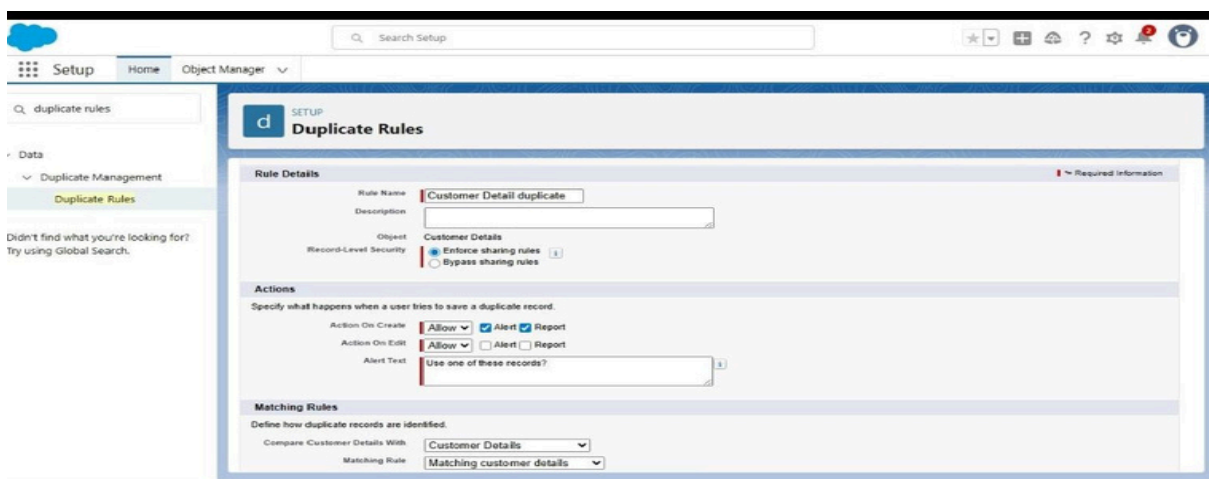
Field 1.Gmail
 2.Phone number
- Click save
- Then click on the Activation



The screenshot shows the Salesforce Setup interface for creating a Matching Rule. The left sidebar has a search bar with 'matching' entered and a navigation menu with 'Data' > 'Duplicate Management' > 'Matching Rules' highlighted. The main content area is titled 'Matching Rules' and contains the following fields and sections:

- Rule Name:** Matching customer details
- Unique Name:** Matching_customer_details
- Description:** (Empty text box)
- Matching Criteria:**
 - Tell the rule which fields to compare and how.
 - A table with columns: Field, Matching Method, Match Blank Fields, and a logical connector (AND/OR).
 - Row 1: Field = Gmail, Matching Method = Exact, Match Blank Fields = ☐, Connector = AND
 - Row 2: Field = Phone number, Matching Method = Exact, Match Blank Fields = ☐, Connector = AND
 - Row 3: Field = --None--, Matching Method = Exact, Match Blank Fields = ☐, Connector = AND
 - Row 4: Field = --None--, Matching Method = Exact, Match Blank Fields = ☐, Connector = AND
- Buttons:** Previous, Save, Cancel

- Now create a Duplicate rule for Customer details object



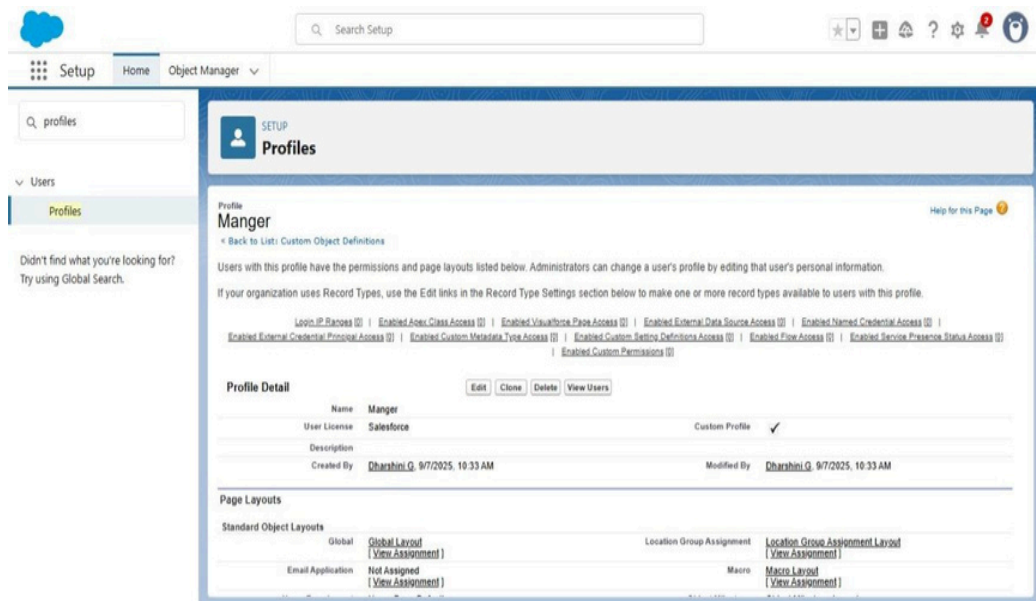
The screenshot shows the Salesforce Setup interface for creating a Duplicate Rule. The left sidebar has a search bar with 'duplicate rules' entered and a navigation menu with 'Data' > 'Duplicate Management' > 'Duplicate Rules' highlighted. The main content area is titled 'Duplicate Rules' and contains the following sections:

- Rule Details:**
 - Rule Name:** Customer Detail duplicate
 - Description:** (Empty text box)
 - Object:** Customer Details
 - Record-Level Security:** ☒ Enforce sharing rules, ☐ Bypass sharing rules
- Actions:**
 - Specify what happens when a user tries to save a duplicate record.
 - Action On Create:** Allow (selected), Alert (checked), Report (checked)
 - Action On Edit:** Allow (selected), Alert (unchecked), Report (unchecked)
 - Alert Text:** Use one of these records? (Text box)
- Matching Rules:**
 - Define how duplicate records are identified.
 - Compare Customer Details With:** Customer Details
 - Matching Rule:** Matching customer details

Manager profile

To create a new profile

- 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.



- Scroll down to Custom Object Permissions and Give access permissions for Appointments, Billing details and feedback , service records and customer details object.
- Then create a sales person profile like as Manager profile.

Create Manager Role

- In 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.

Setup Roles

Role Edit

New Role

Label

Role Name

This role reports to

Role Name as displayed on reports

- Create the another rule under the manager then give the lable name as “sales person” and click save

CreatetheUser :

- Go to setup >> type users in quick find box >> select users >> click New user.
- Fill the fields

Setup Users

New User

User Edit

General Information

First Name

Last Name

Alias

Email

Username

Nickname

Title

Company

Department

Division

Role

User License

Profile

Active ☒

Marketing User ☐

Offline User ☐

Knowledge User ☐

Flow User ☐

Service Cloud User ☐

Site.com Contributor User ☐

Site.com Publisher User ☐

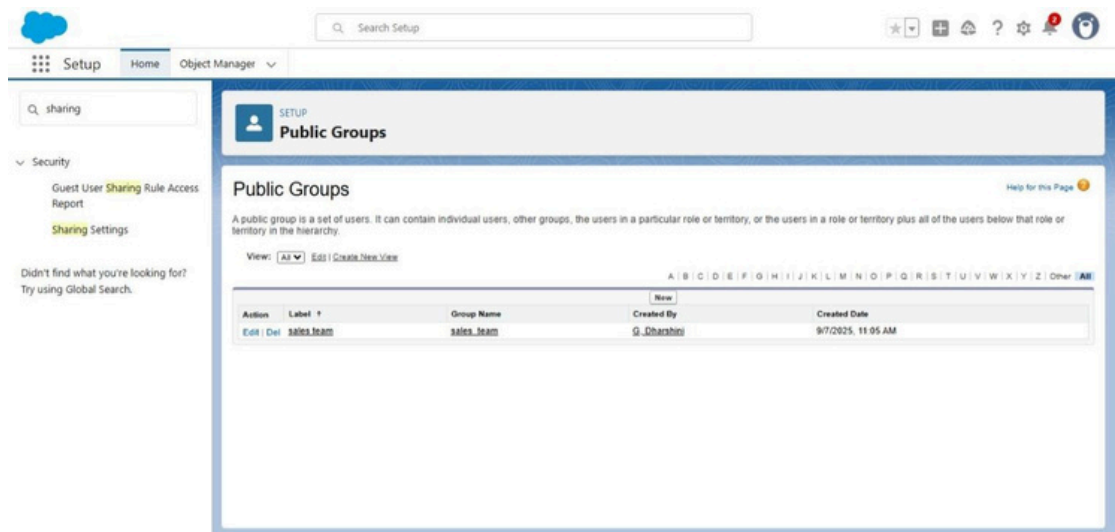
WDC User ☐

- Create another users using the above steps.

CreatingNewPublicGroup :

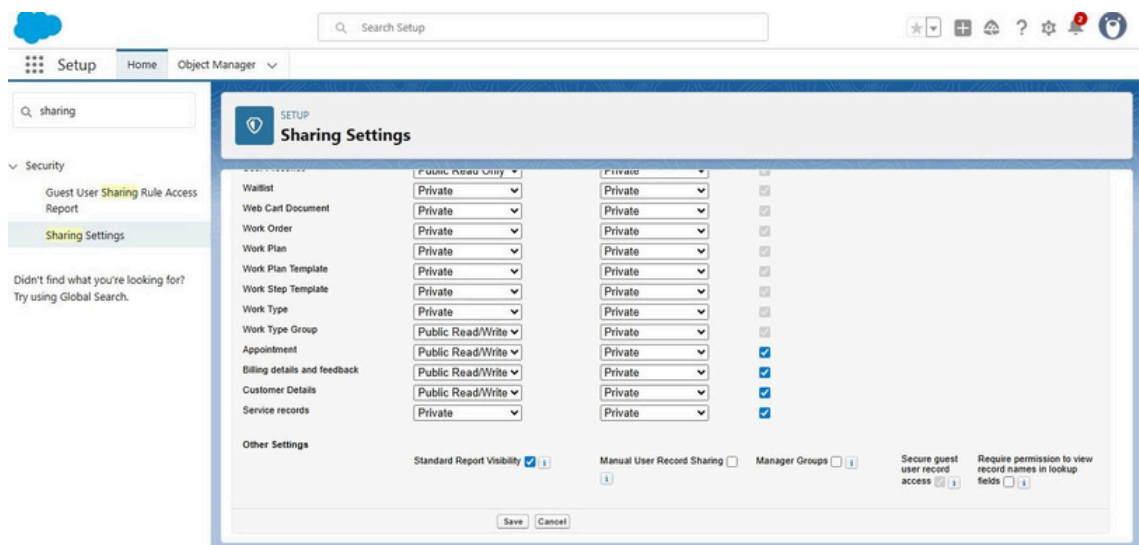
- Go to setup >> type users in quick find box >> select public groups >> click New.

- Label name as “Sales team”



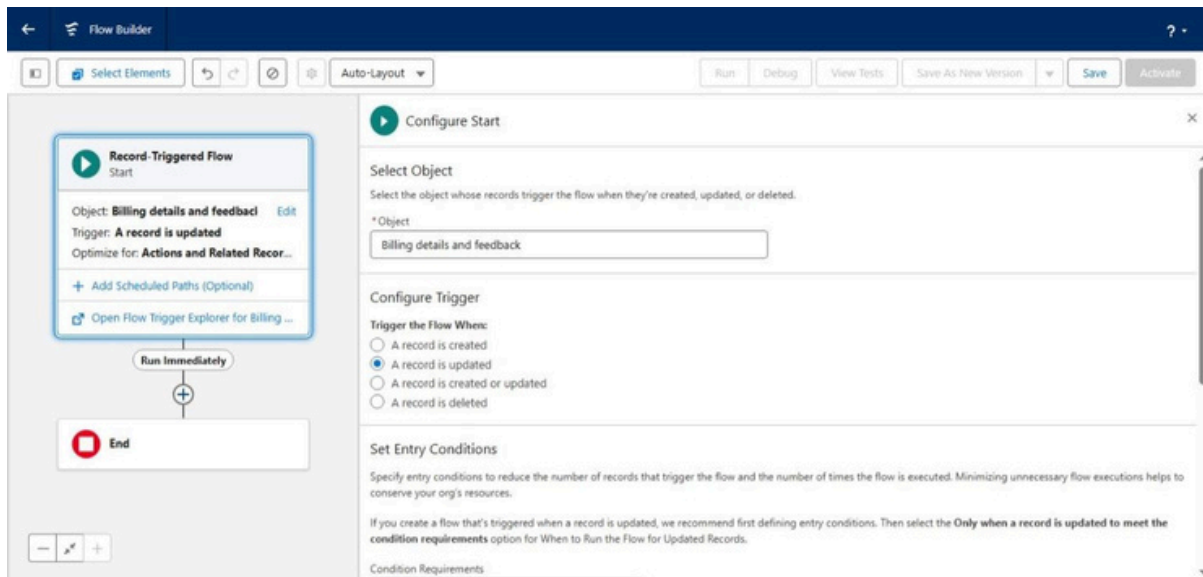
Creating the Sharing settings :

- In quick find box select the share setting records
- Change the OWD setting of the service record object to private

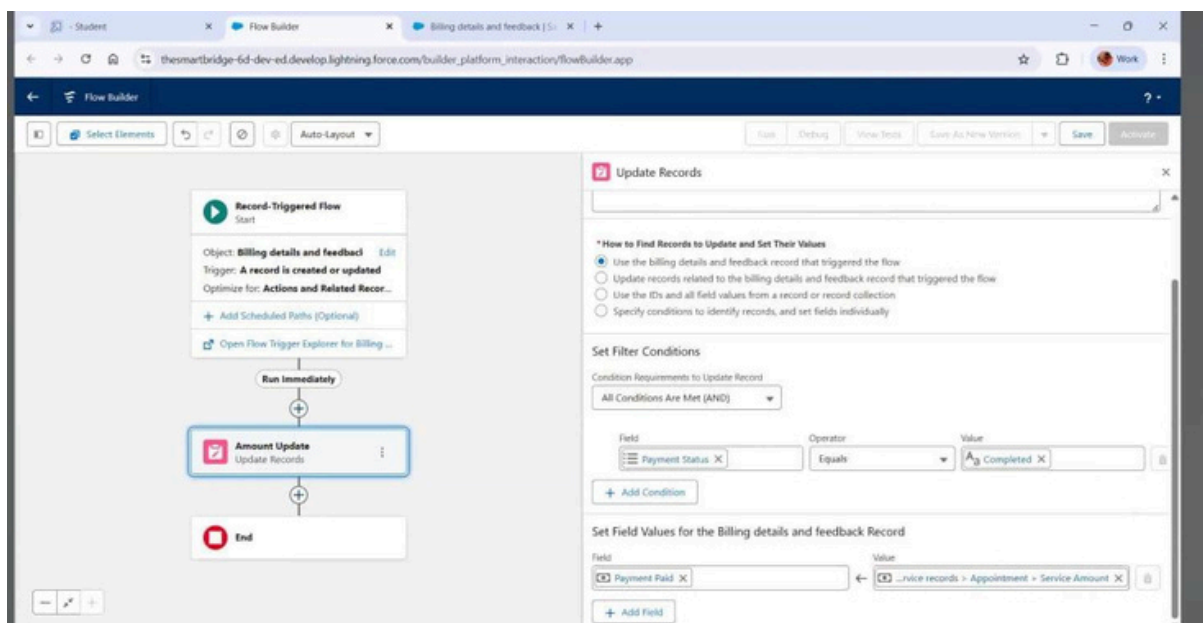


Create a Flow :

- Type flow in Quick find box then click the new flow
- Select the object as “Billing details and feedback”



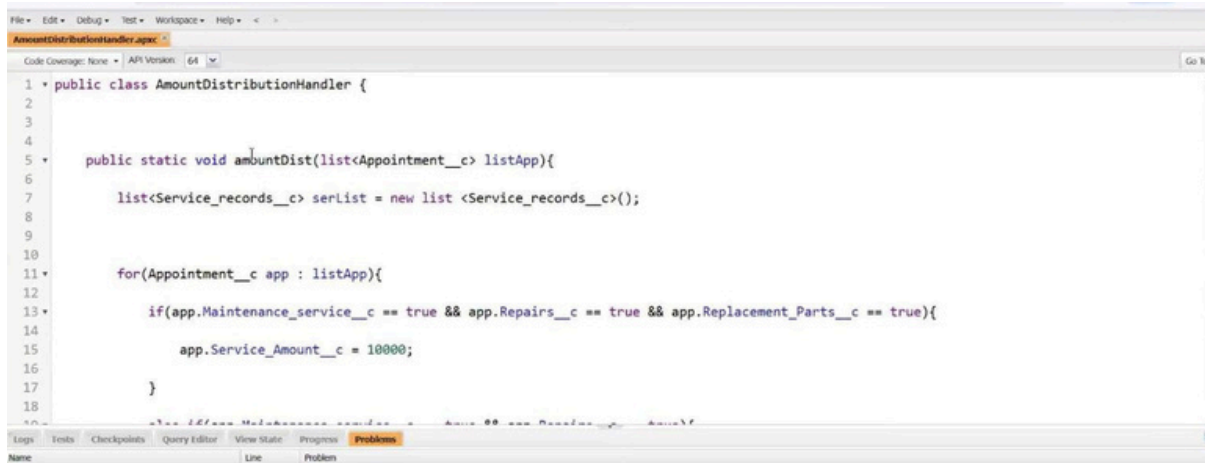
j



- Create the table name as “Amount update”
- Give the details same as above the picture.
- Then create the another Flows

Apex Trigger :

- Click the Developer console Then create new Apex class
- Name the class as “AmountDistributorHandler”
- 4. Copy the code in the instruction page.

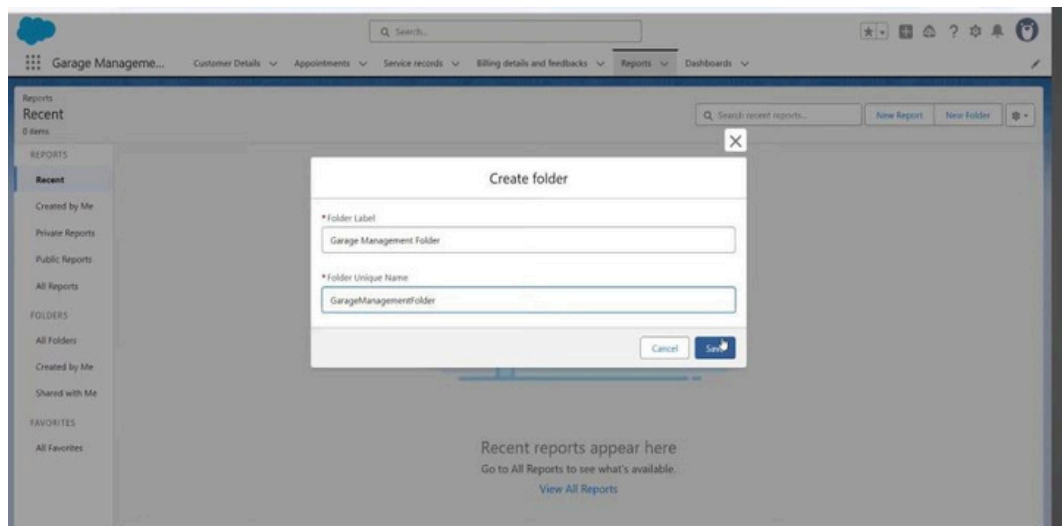


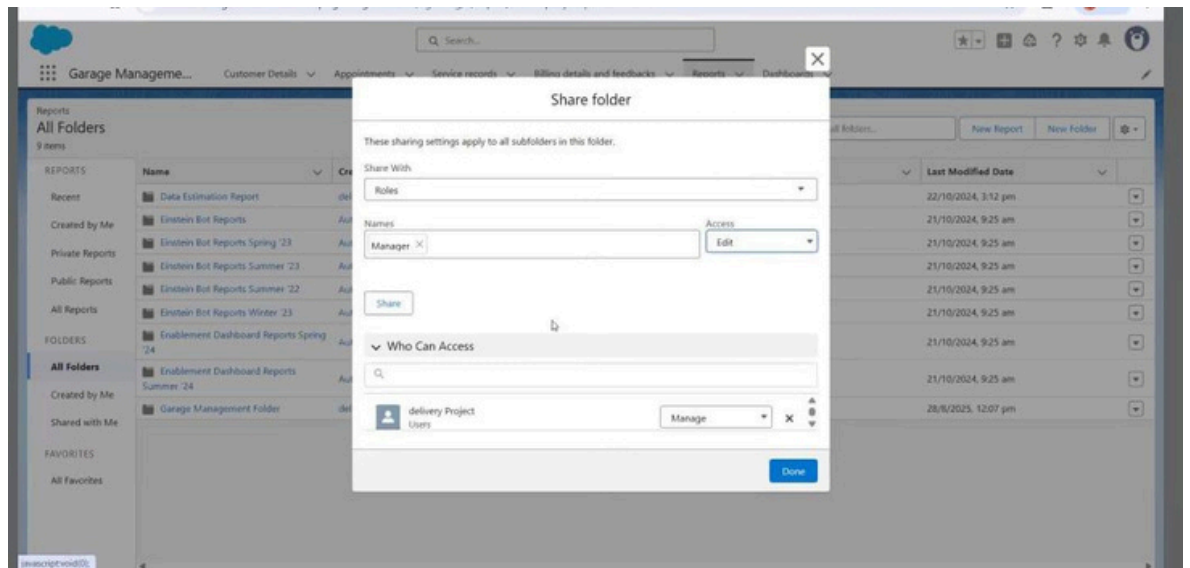
```
1 public class AmountDistributionHandler {
2
3
4
5 public static void amountDist(list<Appointment__c> listApp){
6
7     list<Service_records__c> serlist = new list <Service_records__c>();
8
9
10
11     for(Appointment__c app : listApp){
12
13         if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
14
15             app.Service_Amount__c = 10000;
16
17         }
18
19     }
20 }
```

- Then the create the Apex Trigger and copy the code in instruction page.

CreateaReportFolder :

- 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 then Click save

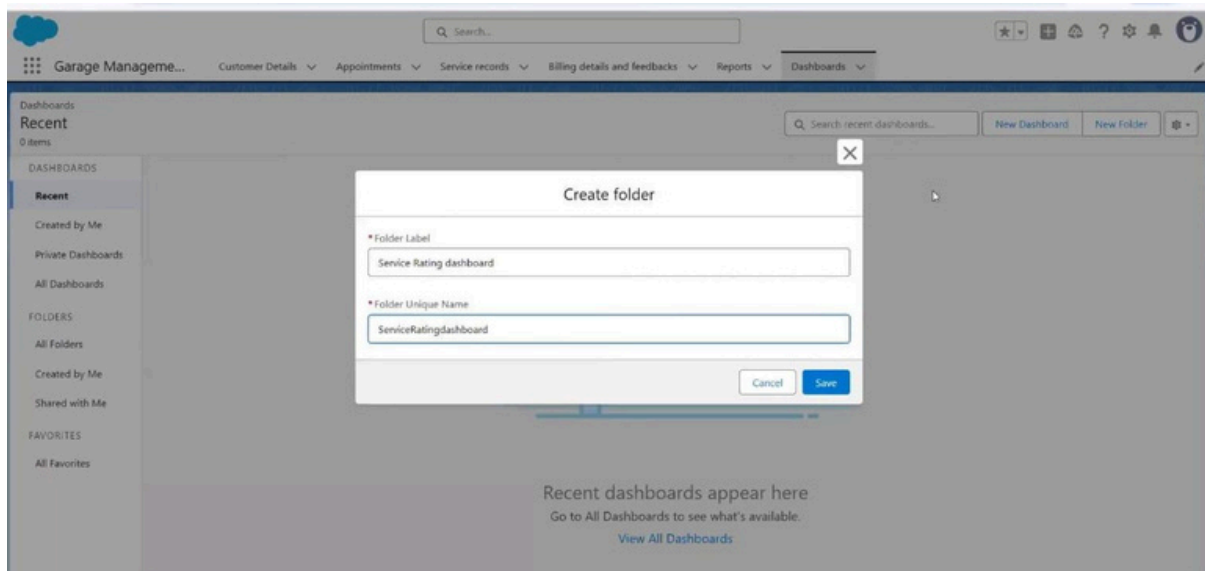




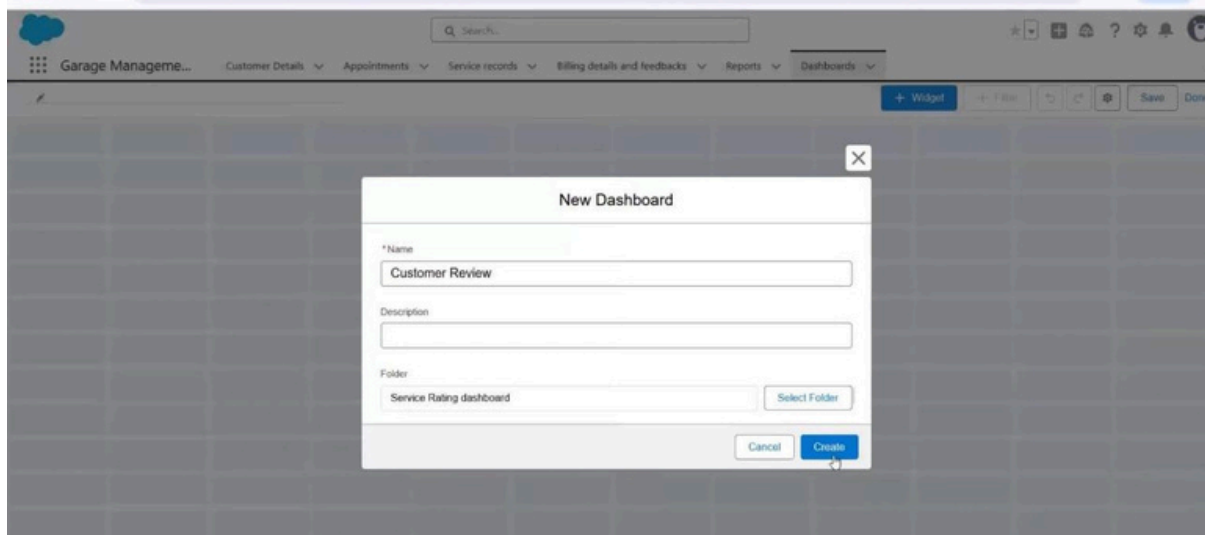
- Then create the Report type and Report

Create Dashboard Folder :

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

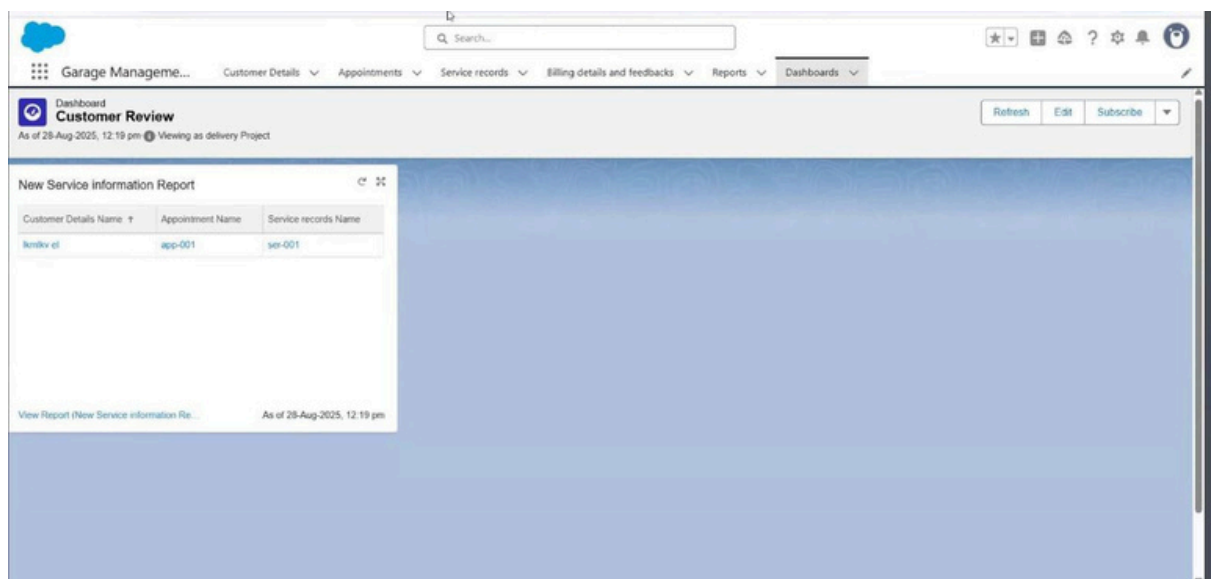


- Now create the dashboard



Creating Records :

- To create a record in the follow objects follow these steps.
- Click on the app launcher located at the left side of the screen.
- Search for “ Garage Management” and click on it.
- Click on the “ Consumer details tab” and fill the details then click save.



- Then create the Appointment record and service record.
- Now automatically Service status will be moved to completed.

Conclusion :

The Garage Management System makes garage work easier by storing customer and vehicle details in one place. It saves time, reduces mistakes, and helps in giving better service. This project is useful for managing daily garage activities in a simple and effective way