

# **Garage Management system**

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff. The **Garage Management System (GMS)** is a comprehensive software solution designed to streamline and optimize the operations of automotive repair facilities, service centers, and garages. It provides an array of features tailored to meet the needs of mechanics, service advisors, and business owners, ensuring smoother workflows and higher customer satisfaction.

## **1. Appointment Scheduling:**

- a. Simplifies the booking process for customers.
- b. Enables staff to manage daily schedules efficiently, reducing downtime and improving resource allocation.

## **2. Vehicle Management:**

- a. Maintains detailed records of vehicles, including service history, repairs, and maintenance schedules.
- b. Tracks vehicle status during servicing for better communication with customers.

## **3. Customer Relationship Management(CRM):**

- a. Stores customer details and preferences.
- b. Sends service reminders, follow-ups, and promotional offers to build loyalty.

## **4. Inventory and Spare Parts Management:**

- a. Tracks spare parts stock levels, automates reorder processes, and prevents stockouts.
- b. Ensures that mechanics always have the necessary tools and parts on hand.

## **5. Billing and Invoicing:**

- a. Generates professional invoices quickly and accurately.
- b. Supports multiple payment methods, discounts, and tax calculations.

## **6. Work Order Management:**

- a. Creates detailed work orders with a list of tasks, estimated costs, and timelines.
- b. Helps staff prioritize jobs and ensure timely completion.

## **7. Reporting and Analytics:**

- a. Provides insights into key performance indicators like revenue, job completion rates, and customer feedback.
- b. Helps identify trends and areas for improvement.

# Salesforce

## Introduction:

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

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?"

## What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

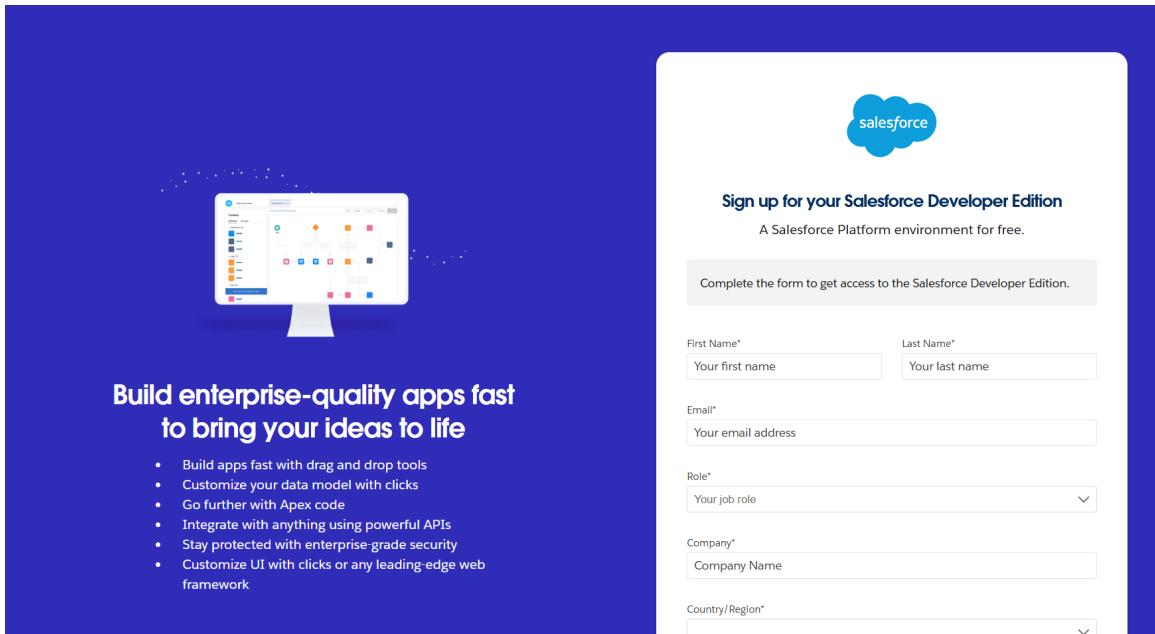
So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something.

## Creating Developer Account:

### Creating a developer org in salesforce.

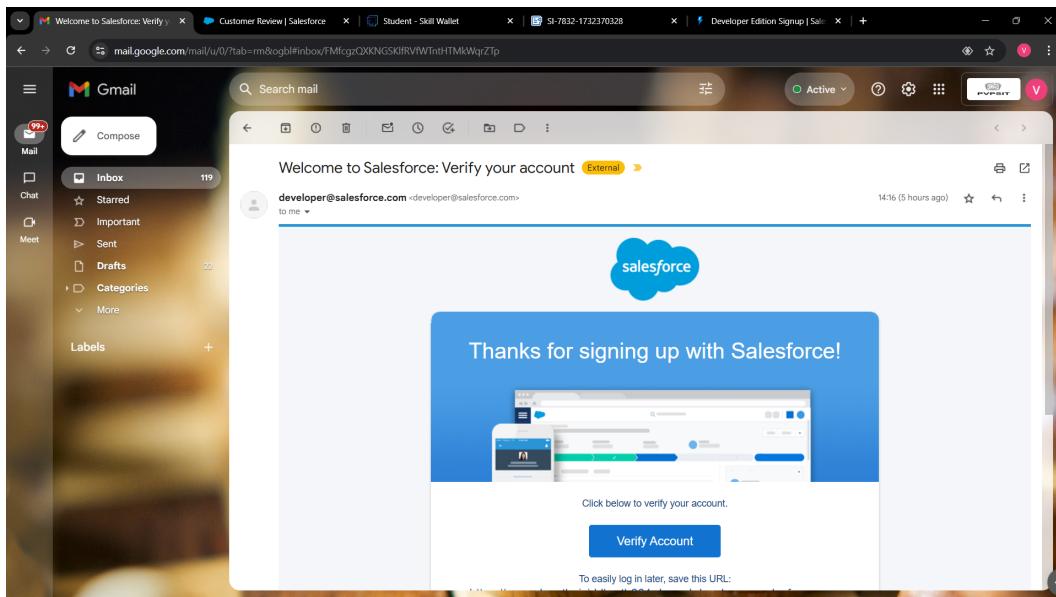
1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :
  1. First name & Last name
  2. Email
  3. Role : Developer
  4. Company : College Name
  5. County : India
  6. Postal Code : pin code
  7. Username : should be a combination of your name and company. This need not be an actual email id, you can give anything in the format : [username@organization.com](mailto:username@organization.com)

Click on sign me up after filling these.



## Account Activation

1. Go to the inbox of the email that you used while signingup. Click on the verify account to activate your account.



OBJECT

### **What Is an Object?**

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:

1. **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **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.

## **Create Customer Details Object**

### **To create an object:**

1. From the setup page >> Click on ObjectManager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> CustomerDetails
2. Plural label name >> Customer Details
3. Enter Record Name Label and Format
  1. Record Name >> CustomerName
  2. Data Type >> Text
1. Click on Allow reports and Track Field History,
2. Allow search >> Save.

## **Create Appointment Object**

### **To create an object:**

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

**Fields & Relationships**  
11 items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date		
Appointment Name	Name	Auto Number		✓
Created By	CreatedBy	Lookup(User)		
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Maintenance service	Maintenance_service__c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		✓
Repairs	Repairs__c	Checkbox		
Replacement Parts	Replacement_Parts__c	Checkbox		
Service Amount	Service_Amount__c	Currency(18, 0)		

## Create Servicerecords Object

**To create an object:**

- From the setup page >> Click on Object Manager >> Click on Create >> Click on CustomObject.
- Enter the label name >> Servicerecords
- 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

The screenshot shows the Salesforce Setup interface. In the top left is a blue cloud icon. To its right are tabs for 'Setup', 'Home', and 'Object Manager'. A search bar with the placeholder 'Search Setup' is located above a toolbar with various icons. The main content area has a header 'SETUP > OBJECT MANAGER' and 'Service records'. On the left is a sidebar with a 'Details' tab selected, listing various configuration options like 'Fields & Relationships', 'Page Layouts', and 'Buttons, Links, and Actions'. The main panel shows the 'Details' section for the 'Service records' object. It includes fields for 'API Name' (set to 'Service\_records\_\_c'), 'Custom' (checked), 'Singular Label' (set to 'Service records'), and 'Plural Label' (set to 'Service records'). On the right, there are sections for 'Enable Reports' (checked), 'Track Activities', 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), 'Help Settings', and 'Standard salesforce.com Help Window'. At the bottom right of the main panel are 'Edit' and 'Delete' buttons.

## Create Billingdetails and feedbackObject

### To create an object:

1. From the setup page >> Click on ObjectManager >> Click on Create >> Click on CustomObject.
1. Enter the label name >> Billingdetails and feedback
2. Plural label name >> Billing details and feedback
3. Enter RecordName Label and Format
  1. Record Name >> Billingdetails and feedbackName
  2. DataType >> Auto Number
  3. Display Format >> bill-{000}
  4. Starting number >> 1
1. Click on Allow reports and Track Field History,
2. Allow search >> Save.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(18, 0)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1) (Unique Case Insensitive)		✓
Service records	Service_records__c	Lookup(Service records)		✓

## Tabs

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

### Types of Tabs:

#### 1. Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

#### 2. WebTabs

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

#### 3. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

#### 4. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu

inLightning Experience and the mobile app.

### 5. **Lightning Page Tabs**

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other customtabs. Once created,they don't show up on the All Tabs page when you click the Plus icon that appearsto the right of your current tabs.Lightning Page tabs also don't show up in the Available Tabs list when you customise the tabs for your apps.

## **Creating a Custom Tab**

### **To createa Tab:(Customer Details)**

1. Go to setup page >> type Tabs in Quick Find bar >>click on tabs >> New (under customobject tab)
2. Select Object(Customer Details) >> Select the tab style >> Next (Add to profilespage) keep it as default >>Next (Add to Custom App) uncheck the include tab .
3. Make sure that the Append tab to users' existing personalcustomizations is checked.
4. Click save.

## **Creating RemainingTabs**

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

Action	Label	Tab Style	Description
Edit   Del	Appointments		
Edit   Del	Billing details and feedback		
Edit   Del	Customer Details		
Edit   Del	Service records		

## The Lightning App

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

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

## Create a Lightning App

To create a lightning app page:

1. Go to setup page >> search "app manager" in quick find >> select "app manager" >> click on New lightning App.
2. Fill the app name in app details as Garage Management Application >> Next >> (Appoption page) keep it as default >> Next >> (UtilityItems) keep it as default >> Next.

3. To Add Navigation Items:
4. 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.
5. To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

The screenshot shows the 'Navigation Items' configuration screen in the Lightning App Builder. On the left, a sidebar lists 'App Settings' sections: App Details & Branding, App Options, Utility Items (Desktop Only), and Navigation Items (which is selected). Below these are 'User Profiles'. The main area is titled 'Navigation Items' and contains a description: 'Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.' A large list of 'Available Items' is shown on the left, and a list of 'Selected Items' is on the right. Arrows between the lists allow items to be moved between them.

## Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

### Types of Fields

1. Standard Fields
2. Custom Fields

## Creation of fields for the CustomerDetails object

1. To create fields in an object:
  - a. Go to setup >> click on Object Manager >> type object name (Customer Details) in

- searchbar >> clickon the object.
- b. Nowclick on “Fields& Relationships” >>New
  - c. Select Data Type as a “Phone”
  - d. Click on next.
  - e. Fill the Above as following:
    1. Field Label:Phone number
    2. Field Name : gets auto generated
    3. Click on Next >>Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

**2. To create another fields in an object:**

- a. Go to setup >>click on ObjectManager >> type object name(Customer Details) in searchbar >> clickon the object.
- b. Nowclick on “Fields& Relationships” >>New
- c. Select Data type as a “Email”and Click on Next
- d. Fill the Above as following:
  - e. Field Label : Gmail
  - f. Field Name : gets auto generated
  - g. Click on Next >>Next >> Save and new.

The screenshot shows the Salesforce Object Manager interface for the 'Customer Details' object. The left sidebar has 'Fields & Relationships' selected. The main area displays a table titled 'Fields & Relationships' with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains six rows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
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		▼

## Creation of Lookup Fields

### Creation of Lookup Field on Appointment Object :

1. Go to setup>> click on Object Manager>> type objectname( Appointment ) in the search bar >> click on the object.
2. Now click on “Fields& Relationships” >>New
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ CustomerDetails ” and click next.
5. Next >> Next >>Save.

Note: Make sure you complete Activity4 Before continuing.

**Creation of Lookup Field on Service recordsObject :**

1. Go to setup >> clickon Object Manager>> type objectname( Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >>New
3. Select “Look-up relationship”as data type and click Next.
4. Select the relatedobject “ Appointment ” and clicknext.
5. Make it a required field so click on Required.
6. Scroll down for LookupFilter and click on Show filter settings.
7. Now add the filter criteria.
8. Field : Appointment: Appointment Date >>Operator : less than >>select field >>Appointment: Created Date
9. Filter type should be Required.
10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.

12. Next >> Next >>Save.

#### **Creation of Lookup Field on Billing details and feedback Object :**

1. Go to setup >> click on ObjectManager >> type object name( Billing detailsand feedback ) in searchbar >> click on the object.
2. Now click on “Fields& Relationships” >>New.
3. Select “Look-up relationship” as data type and click Next.
4. Select the relatedobject “ Servicerecords” and click next.
5. Next >> Next >>Save & new.

## **Creation of Checkbox Fields**

#### **Creation of Checkbox Field on Appointment Object :**

1. Go to setup >> click on ObjectManager >> type object name( Appointment ) in searchbar >> click on the object.
2. Now click on “Fields & Relationships” >>New.
3. Select “Check box” as data type and click Next.
4. Give the Field Label: Maintenance service
5. Field Name : is auto populated
6. Default value : unchecked
7. Click on next >>next >> save.

#### **Creation of Another Checkbox Field on Appointment Object :**

1. Repeat the steps form 1 to 3.
2. Give the Field Label : Repairs
3. Field Nme : is auto populated
4. Default value : unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkboxwith given names
7. Give the Field Label : Replacement Parts
8. Field Nme : is auto populated

9. Default value : unchecked

10. Click on next >>next >> save.

The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Fields & Relationships' and displays 11 items, sorted by Field Label. The fields listed are:

Field Name	Label	Type
Appointment Date	Appointment_Date_c	Date
Appointment Name	Name	Auto Number
Created By	CreatedById	Lookup(User)
Customer Details	Customer_Details__c	Lookup(Customer Details)
Last Modified By	LastModifiedById	Lookup(User)
Maintenance service	Maintenance_service__c	Checkbox
Owner	OwnerId	Lookup(User,Group)
Repairs	Repairs__c	Checkbox
Replacement Parts	Replacement_Parts__c	Checkbox
Service Amount	Service_Amount__c	Currency(18, 0)
Vehicle number plate	Vehicle_number_plate__c	Text(10)

### Creation of Checkbox Field on Service records Object :

1. Go to setup >> click on Object Manager >> type object name( Service records ) in search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New.
3. Select "Check box" as data type and click Next.
4. Give the Field Label : Quality Check Status
5. Field Nme : is auto populated
6. Default value : unchecked
7. Click on next >> next >> save

## Creation of date Fields

### Creation of Date Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name( Appointment ) in the search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New.

3. Select “Date”as data type and click Next.
4. Give the Field Label : Appointment Date
5. Field Nme : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >>next >> save.

Field	Field Name	Type
Appointment Date	Appointment_Date_c	Date
Appointment Name	Name	Auto Number
Created By	CreatedById	Lookup(User)
Customer Details	Customer_Details__c	Lookup(Customer Details)
Last Modified By	LastModifiedById	Lookup(User)
Maintenance service	Maintenance_service__c	Checkbox
Owner	OwnerId	Lookup(User,Group)
Repairs	Repairs__c	Checkbox
Replacement Parts	Replacement_Parts__c	Checkbox
Service Amount	Service_Amount__c	Currency(18, 0)
Vehicle number plate	Vehicle_number_plate__c	Text(10)

## Creation of Currency Fields

### Creation of Currency Field on Appointment Object :

1. Go to setup >> click on ObjectManager >> type object name( Appointment ) in the search bar >> clickon the object.
2. Now click on “Fields& Relationships” >>New.
3. Select “Currency” as data type and click Next.
4. Give the Field Label : Service Amount
5. Field Nme : is auto populated
6. Click on next
7. Give read only for all the profiles in field levelsecurity for profile.
8. Click on next >> save.

**Creation of Text Fields**

1. Go to setup >>click on Object Manager >>type object name( Appointment ) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >>New.
3. Select “Text” as data type and click Next.
4. Give the Field Label : Vehicle numberplate
5. Field Name : is auto populated
6. Length : 10
7. Make field as Required and Unique.
8. Click on next >>next >> save.

**Creation of Text Fields in Billing details and feedback object :**

1. Go to setup >> click on Object Manager >>type object name( Billing detailsand feedback ) in searchbar >> click on the object.
2. Now click on “Fields& Relationships” >>New.
3. Select “text” as data type and click Next.

4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required and Unique.
8. Click on next >> next >> save

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Billing details and feedback'. On the left, there's a sidebar with links like 'Page Layouts', 'Lightning Record Pages', etc. The main content area is titled 'Fields & Relationships' and shows a table with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(18, 0)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1) (Unique Case Insensitive)		✓
Service records	Service_records__c	Lookup(Service records)		✓

## Creation of Picklist Fields

### Creation of Picklist Fields in Service records object :

1. Go to setup >>click on ObjectManager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Picklist" and click Next.
4. Enter Field Label as "Service Status",under values select "Enter values,with each valuesseparated by a new line" and enter valuesas shown below.
5. Thevalues are: Started,Completed.
6. Click Next.
7. Next >>Next >> Save.

## **Creation of Picklist Fields in Billingdetails and feedbackobject :**

1. Go to setup >>click on ObjectManager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Picklist" and click Next.
4. Enter Field Label as "PaymentStatus", under values select "Enter values,with each values separated by a new line" and enter values as shown below.
5. The values are: Pending, Completed.
6. Click Next.
7. next >> next >> save .

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Billing details and feedback'. On the left, there's a sidebar with various options like 'Page Layouts', 'Lightning Record Pages', etc. The main content area is titled 'Fields & Relationships' and shows a table with 8 items. The table columns are 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The data in the table is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(18, 0)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1) (Unique Case Insensitive)		✓
Service records	Service_records__c	Lookup(Service records)		✓

## **Creating Formula Field in Service recordsObject**

1. Go to setup >>click on ObjectManager >> type object name(Service records) in searchbar >> click on the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Formula" and click Next.
4. Give Field Label and Field Name as "servicedate" and select formula return type as "Date" and click next.
5. Insert field formula should be : CreatedDate
6. click "CheckSyntax".
7. Click next >>next >> Save

The screenshot shows the Salesforce Setup interface under the Object Manager for the Service records object. The left sidebar lists various setup categories. The main area displays the 'Fields & Relationships' section, which contains a table of fields. The columns in the table are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

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		

## Validation rule

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

## To create a validation rule to an Appointment Object

1. Go to the setup page >> click on objectmanager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> clickNew.
3. Enter the Rule name as " Vehicle ".
4. Insert the ErrorCondition Formula as :-
  1. NOT(REGEX( Vehicle\_number\_plate\_\_c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))
  6. Enter the Error Message as "Please enter valid number ", select the Error location as Field and select the field as "Vehicle numberplate", and click Save.

SETUP > OBJECT MANAGER  
**Appointment**

Appointment Validation Rule

Validation Rule Detail

Rule Name	Vehicle	Active
Error Condition Formula	NOT(REGEX( Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}")))	Error Location
Error Message	Please enter valid number	Vehicle number plate
Description		
Created By	Shankar_Vemana	Modified By
	23/11/2024, 3:19 pm	Shankar_Vemana
		23/11/2024, 3:19 pm

## To create a validation rule to an Service records Object

1. Go to the setup page >> click on objectmanager >> From drop down click edit for Servicerecords object.
2. Click on the validation rule >> clickNew.
3. Enter the Rule name as " service\_status\_note ".
4. Insert the ErrorCondition Formula as :-

NOT( ISPICKVAL( Service\_Status\_c , "Completed") )

Enter the Error Message as "still it is pending", select the Error location as Field and select the field as "Service status", and click Save.

SETUP > OBJECT MANAGER  
**Service records**

Service records Validation Rule

Validation Rule Detail

Rule Name	service_status_note	Active
Error Condition Formula	NOT( ISPICKVAL( Service_Status_c , "Completed") )	Error Location
Error Message	still it is pending	Service Status
Description		
Created By	Shankar_Vemana	Modified By
	23/11/2024, 3:21 pm	Shankar_Vemana
		23/11/2024, 3:21 pm

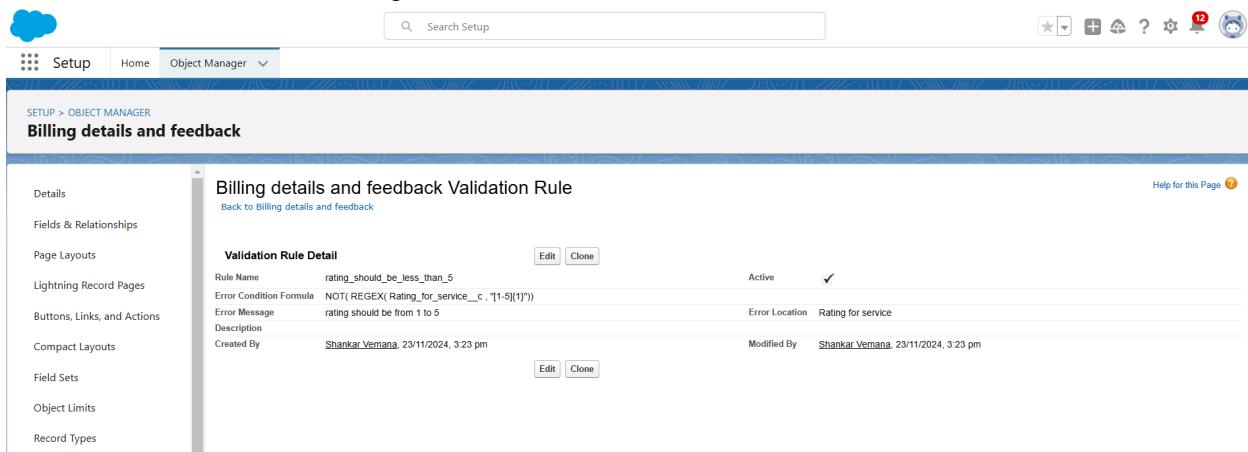
## To create a validation rule to an Billing

## detailsandfeedback Object

1. Go to the setup page >> clickon object manager>> From drop down clickedit for Billingdetails and feedbackobject.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating\_should\_be\_less\_than\_5”.
4. Insert the ErrorCondition Formula as :-

```
NOT( REGEX( Rating_for_service_c, "[1-5]{1}" ))
```

Enter the Error Messageas “rating shouldbe from 1 to 5”, select the Error locationas Field and select the field as “Rating for Service”, and click Save.



## Duplicate rule

### To create a matching rule to an Customerdetails Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matchingrule >> click on New Rule.
3. Select the object as Customerdetails and click Next.
4. Give the Rule name : Matchingcustomer details

5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method
  - a. Gmail Exact
  - b. Phone Number Exact
8. Click save.
9. After Saving Click on Activate.

## 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.

The screenshot shows the Salesforce Setup interface with the following details:

- Left Sidebar:** Shows a search bar with "Search Setup" and a navigation menu with "Setup" selected. Other items include "Home" and "Object Manager".
- Middle Content Area:**
  - Section Header:** "d SETUP Duplicate Rules"
  - Rule Details:**
    - Name:** Customer Detail duplicate
    - Description:** Customer Details
    - Object:** Customer Details
    - Record-Level Security:** Enforce sharing rules
    - Action On Create:** Allow
    - Action On Edit:** Allow
    - Alert Text:** Use one of these records?
    - Active:** ✓
    - Matching Rule:** Matching customer details (Mapped)
    - Conditions:** (Customer Details: Email EXACT MatchBlank = FALSE) AND (Customer Details: Phone\_number EXACT MatchBlank = FALSE)
  - Buttons:** Edit, Delete, Clone, Deactivate
  - Order:** 1 of 1 [Reorder]
  - Operations:** Operations On Create (✓ Alert, ✓ Report), Operations On Edit (Alert, Report)
  - Modified By:** Shankar Vemana, 23/11/2024, 3:25 pm

## 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.

## Manager Profile

### To create a new profile:

- Goto 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 CustomApp settings as default for the Garagemanagement.
- 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”.

- f. Change the password policies as mentioned :
- g. User passwords expire in should be “ never expires ”.
- h. Minimum password length should be “ 8 ”, and click save.

**Profile Manager**

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Profile Detail		Custom Profile	
Name	Manager	User License	Salesforce
Description		Created By	Shankar Vemana, 23/11/2024, 2:27 pm
Modified By		Modified By	Shankar Vemana, 23/11/2024, 3:30 pm

**Page Layouts**

Standard Object Layouts	Global	Global Layout [View Assignment]	Invoice	Invoice Layout [View Assignment]
Email Application	Not Assigned	[View Assignment]	Invoice Line	Invoice Line Layout [View Assignment]
Home Page Layout	DE Default	[View Assignment]	Lead	Lead Layout [View Assignment]
Account	Account Layout [View Assignment]	Profile: Manager - Salesforce - Developer Edition	Legal Entity	Legal Entity Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]		Location	Location Layout [View Assignment]
Appointment Invitation	Appointment Invitation Layout [View Assignment]		Location Group	Location Group Layout [View Assignment]
Asset	Asset Layout		Location Group Assignment	Location Group Assignment Layout

## Sales person Profile

1. Goto setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Salesforce Platform User) >> enter profile name (sales person) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the GArage management.
4. 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.
5. 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

- i. Go to quick find >> Search for Roles >> click on set up roles.
- ii. Click on Expand All and click on add role under whom this role works.
- iii. Give Label as "Manager" and Role name gets auto populated. Then click on Save.

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The sidebar on the left contains links for 'Users', 'Feature Settings', 'Sales' (with sub-links for Contracts, Opportunities, and Cases), 'Service', 'Case Teams', and 'Contact Roles'. The main content area is titled 'Creating the Role Hierarchy' and displays a hierarchical list of roles. The hierarchy starts with 'Prasad V Poturi Siddhartha Institute of Technology' and branches into various roles like CEO, CFO, COO, Manager, sales\_person, etc. Each role entry includes 'Edit | Del | Assign' buttons. A 'Help for this Page' link is located in the top right corner of the main content area.

## Creating another roles

- a. Go to quick find >>Search for Roles >> click on set up roles.
- b. Click plus on CEO role, and click add role under manager.
- c. Give Label as “salesperson” 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
  - a. First Name : Niklaus

- b. Last Name : Mikaelson
- c. Alias : Give a AliasName
- d. Email id : Give your Personal Emailid
- e. Username : Username should be in this form:text@text.text
- f. Nick Name : Give a Nickname
- g. Role : Manager
- h. User licence : Salesforce
- i. Profiles : Manager
- j. save.

The screenshot shows the Salesforce Setup interface. The left sidebar is titled "Setup" and includes sections for Home, Object Manager, and various administrative tools. Under "User Management Settings", the "Users" section is selected. The main content area is titled "Users" and shows a list of users. A specific user, "Niklaus Mikaelson", is selected and displayed in detail. The "User Detail" section shows the following information:

Name	Niklaus Mikaelson	Role	Manager
Alias	mika	User License	Salesforce
Email	vemana.vbs2139@gmail.com [Verify]	Profile	Manager
Username	vemana123@gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	kun	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>
Department		Flow User	<input type="checkbox"/>
Division		Service Cloud User	<input type="checkbox"/>
Address		Site.com Contributor User	<input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	<input type="checkbox"/>
Locale	English (India)	WDC User	<input type="checkbox"/>
Language	English	Mobile Push Registrations	<input checked="" type="checkbox"/>
Delegated Approver	Manager	Data.com User Type	<input checked="" type="checkbox"/>
Receive Approval Request Emails	Only if I am an approver	Accessibility Mode (Classic Only)	<input type="checkbox"/> <input checked="" type="checkbox"/>
Expiration ID		Debug Mode	<input type="checkbox"/> <input checked="" type="checkbox"/>

At the bottom right of the detail screen, there is a link "Hide Contact Details on Charts".

## creating another users

1. Repeat the steps and create another user using
  - a. Role : sales person
  - b. User licence : Salesforce Platform

### c. Profile : sales person

Note : create atleast 3 userswith these permissions

Action	Full Name	Alias	Username	Role	Active	Profile
<a href="#">Edit</a>	Baji	baji	baji12@gmail.com	sales_person	✓	sales_person
<a href="#">Edit</a>	Chatter Expert	Chatter	chatv.00day00000g8cezmai.pcpxfqm8kdeo@chatter.salesforce.com		✓	Chatter Free User
<a href="#">Edit</a>	Fasi	fasi	fasiun@gmail.com	sales_person	✓	sales person
<a href="#">Edit</a>	Mikaelson_Niklaus	mika	yemana123@gmail.com	Manager	✓	Manager
<a href="#">Edit</a>	Phani	phan	phan12@gmail.com	sales_person	✓	sales person
<a href="#">Edit</a>	User_Integration	integ	integration@00day00000g8cezmai.com		✓	Analytics_Cloud_Integration_User
<a href="#">Edit</a>	User_Security	sec	insightssecurity@00day00000g8cezmai.com		✓	Analytics_Cloud_Security_User
<a href="#">Edit</a>	Yemana_Shankar	SVema	process123@company.com		✓	System Administrator

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

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with categories such as Users, Feature Settings, and Company Settings. Under 'Users', 'Public Groups' is selected. The main content area is titled 'Public Groups' and shows a group named 'sales team'. It has fields for Label ('sales team'), Group Name ('sales\_team'), and 'Grant Access Using Hierarchies' (which is checked). Below this, there's a table for users, with one row for 'sales\_person'. At the bottom, it says 'Group: sales team - Salesforce - Developer Edition'.

## 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.

## 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 Servicerecords Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Click new on Servicerecords sharing Rules.
5. Give the Labelname as "Sharing setting"
6. Rule name is auto populated.
7. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person "
8. In step 4: share with, select " Roles " >> " Manager "
9. In step 5 : Change the access level to " Read / write ".

10. Click on save.

The screenshot shows the Salesforce Sharing Settings page. The left sidebar has a search bar and navigation links for Setup, Home, and Object Manager. Under Security, there are links for Guest User Sharing Rule Access Report and Sharing Settings. A message says "Didn't find what you're looking for? Try using Global Search." The main content area is titled "Sharing Settings". It lists several sharing rule categories: Work Step Template Sharing Rules, Work Type Sharing Rules, Work Type Group Sharing Rules, Appointment Sharing Rules, Billing details and feedback Sharing Rules, Customer Details Sharing Rules, and Service records Sharing Rules. The "Service records Sharing Rules" section is expanded, showing a table with one row. The table columns are Action, Criteria, Shared With, and Access Level. The row shows "Edit | Del" under Action, "Owner in Role: sales person" under Criteria, "Role: Manager" under Shared With, and "ReadWrite" under Access Level.

## Flows

## Create a Flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the Record-triggered flow and Click on Create.
3. Select the Object as "Billing details and feedback" in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimizer the flow for: "Actions and Related Records" and Click on Done.
6. 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
7. API name : is auto populated
8. Set a filter condition : All Conditions are met(AND)
9. Field : Payment\_Status\_c
10. Operator : Equals
11. Value : Completed
12. And Set Field Values for the Billing details and feedback Record
13. Field : Payment\_Paid\_c
14. Value : {!\$Record.Service\_records\_r.Appointment\_r.Service\_Amount\_c}
15. Click On Done. Before creating another Element. Create a New Resource from Toolbox form top left.
16. Click on the New Resource, And select Variable.
17. Select the resource type as text template.
18. Enter the API name as " alert".
19. Change the view as Rich Text ? View to Plain Text.
20. In body field paste the syntax that given below.

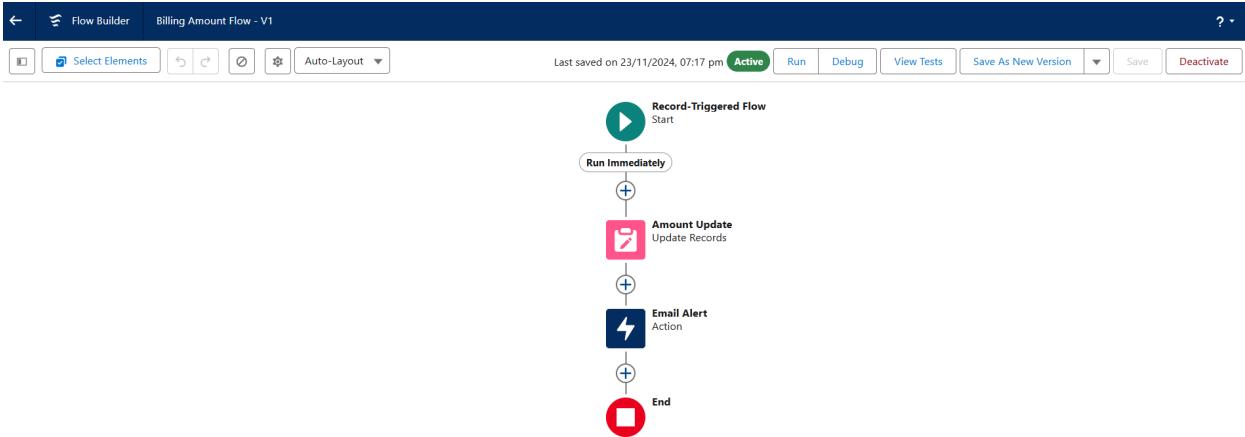
Dear {!\$Record.Service\_records\_r.Appointment\_r.Customer\_Name\_r.Name},

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 .

1. Click done.
2. Now Click on Add Element, select Action.
3. Their action bar will be openedin that search for “ send email ” and click on it.
4. Give the label name as “ Email Alert”
5. API name will be auto populated.
6. Enable the body in set inputvalues for the selected action.
7. Select the text template that created , Body : {!alert}
8. Include recipient addresslist select the email form the record.
9. RecipientAddressList:  
    {!\$Record.Service\_records\_r.Appointment\_\_r.Customer\_Name\_\_r.Gmail\_\_c}
10. Includesubject as “ Thank You for Your Payment - Garage Management”.
- 11.Click done.
- 12.Click on save. Give the Flow label , Flow Api name will be autopopulated.
- 13.And click save, and click on activate.



## 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:

1. insert
2. update
3. delete
4. merge
5. upsert
6. 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:

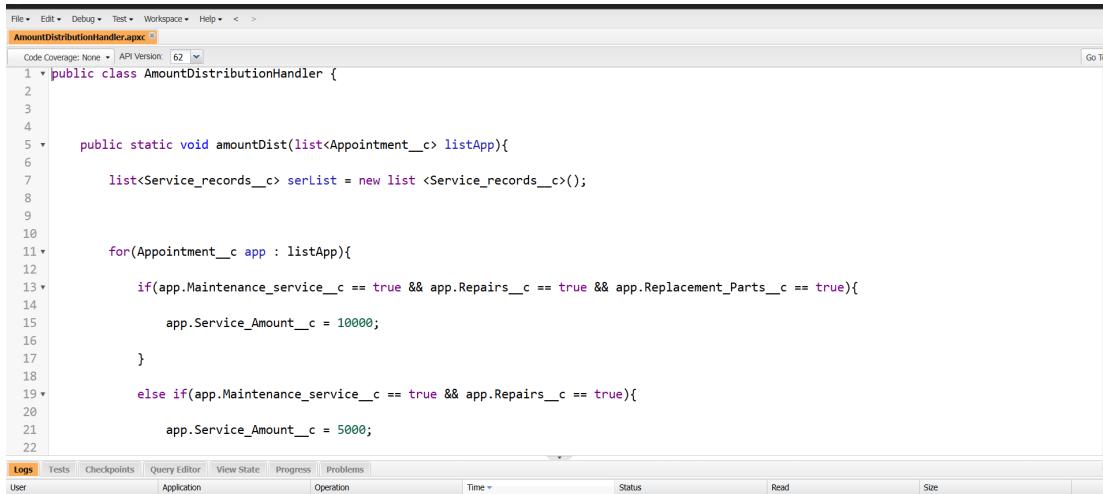
**Before Trigger:** This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets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

Use Case : This use case works for Amount Distribution for each Service the customer selected for their 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".



The screenshot shows the Salesforce Developer Console interface. The title bar says "AmountDistributionHandler.apxc". The code editor contains the following Apex code:

```
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    for(Appointment__c app : listApp){  
11        if(app.Maintenance_Service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){  
12            app.Service_Amount__c = 10000;  
13        }  
14        else if(app.Maintenance_Service__c == true && app.Repairs__c == true){  
15            app.Service_Amount__c = 5000;  
16        }  
17    }  
18  
19    else if(app.Maintenance_Service__c == true && app.Repairs__c == true){  
20        app.Service_Amount__c = 5000;  
21    }  
22}
```

The bottom of the screen shows a logs table with columns: Logs, Tests, Checkpoints, Query Editor, ViewState, Progress, Problems. The logs section is currently active.

```
AmountDistribution.apex | AmountDistributionHandler.apex | 
Code Coverage: None | API Version: 58 |
12     }
13     else if(app.Maintenance_service_c == true && app.Replacement_Parts_c == true){
14         app.Service_Amount_c = 8000;
15     }
16     else if(app.Repairs_c == true && app.Replacement_Parts_c == true){
17         app.Service_Amount_c = 7000;
18     }
19     else if(app.Maintenance_service_c == true){
20         app.Service_Amount_c = 2000;
21     }
22     else if(app.Repairs_c == true){
23         app.Service_Amount_c = 3000;
24     }
25     else if(app.Replacement_Parts_c == true){
26         app.Service_Amount_c = 5000;
27     }
28 }
29 }
30 }
31 }
```

## 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.Repairsc == 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 consoleand you will be navigatedto a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the triggername and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment\_c

### **Syntax For creating trigger :**

The syntax for creatingtrigger is :

Trigger [triggername] on [objectname]( Before/After event)

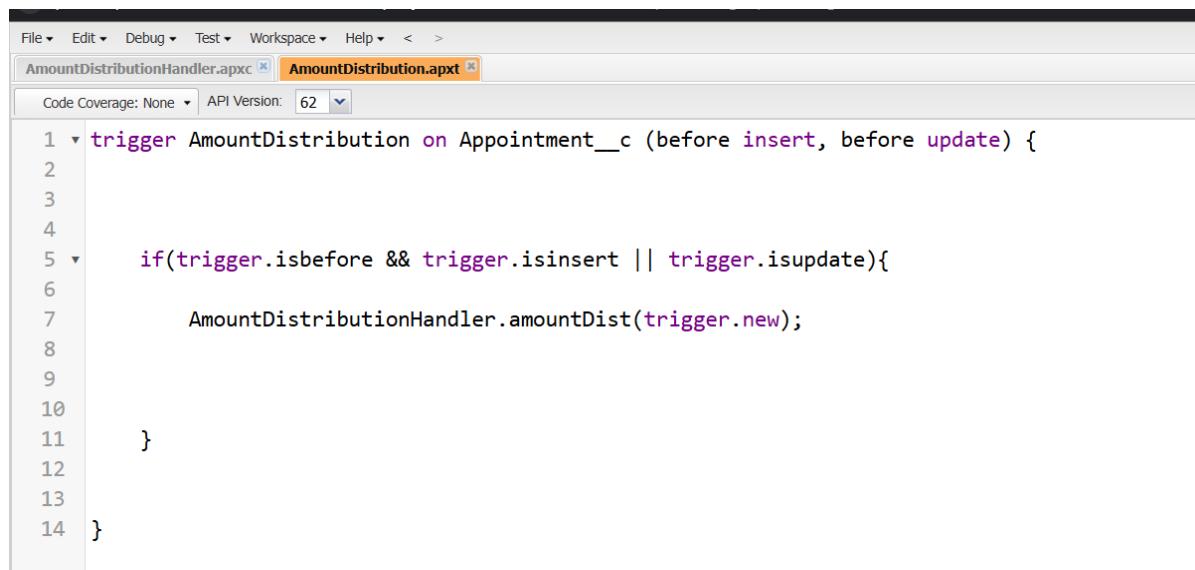
{

```
}
```

In this project , trigger is called whenever the particular record sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

## Code:

```
trigger AmountDistribution on Appointment_c (before insert,before update) {  
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
        AmountDistributionHandler.amountDist(trigger.new);  
    }  
}
```



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >  
AmountDistributionHandler.apxc □ AmountDistribution.apxt □  
Code Coverage: None ▾ API Version: 62 ▾  
1 trigger AmountDistribution on Appointment__c (before insert, before update) {  
2  
3  
4  
5     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
6         AmountDistributionHandler.amountDist(trigger.new);  
7     }  
8  
9  
10    }  
11  
12  
13  
14 }
```

## 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.

3. Give the Folderlabel as “GarageManagement Folder”, Folderunique name will be auto populated.

4. 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.

## **Create Report Type**

1. Go to setup>> type users in quickfind box >>select Report Type >> click on Continue.

2. Click on new custom report type.

3. Select the Primaryobject as “Customer details”.

4. Give the Report type Label as “Serviceinformation”

5. Report type Name is autopopulated.

6. Keep the Description as same.

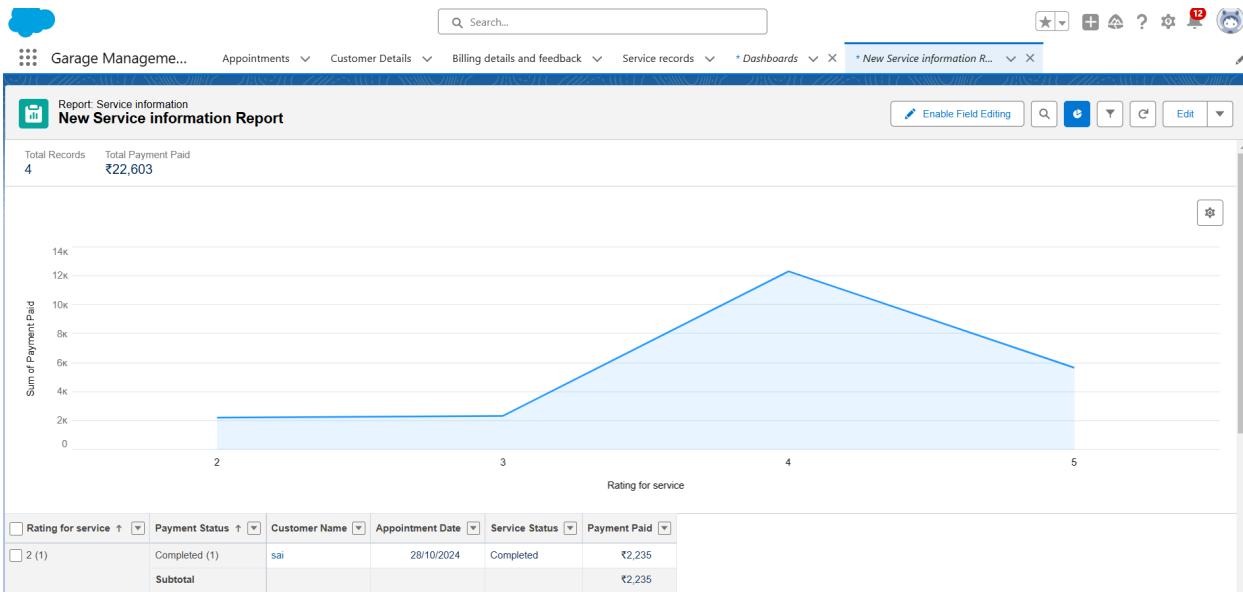
7. Select Store in Category as "other Reports"
8. Select the deployment status as "Deployed", click on Next.
9. now, Click on Related object box.
10. Click on Select Object, choose Appointment Object as shown in fig
11. Again Click to relate another object.
12. And select the related object as "service records".
13. Repeat the process and select the related object as "Billing details and feedback".
14. 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.

3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.
4. Their outline pane is opened already, select the fields that mentioned below in column section.
  - a. Customer name
  - b. Appointment Date
  - c. Service Status
  - d. Payment paid
  - e. Remove the unnecessary fields.
- f. Select the fields that mentioned below in GROUP ROWS section.
  - i. Rating for Service
- g. Select the fields that mentioned below in GROUP ROWS section.
  - i. Payment Status
- h. Click on Add Chart, Select the Line Chart.
- i. Click on save, Give the reportName : New Service information Report
- j. Report unique Name is auto populated.
- k. Select the folder the created and Click on save.



## Dashboards

Dashboards help you visually understand changing business conditions so you can make

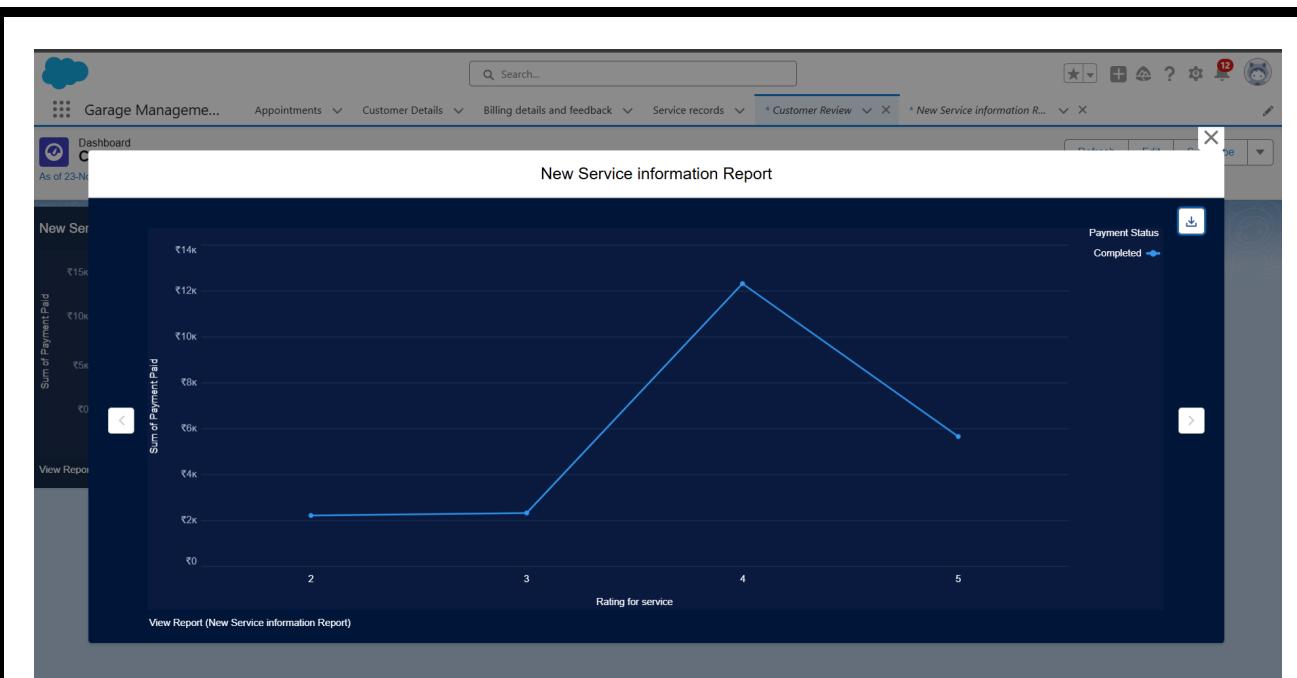
decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

## Create 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.
6. Follow the same steps, from milestone 15, and activity 2, and provide the sharing settings for the folder that just created.

## Create 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.
3. Select add component.
4. Select a Report and click on select.
5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.



## Subscription:

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

Edit Subscription

Schedule dashboard refreshes and subscribe to receive results.

**Settings**

Frequency:  Weekly  Monthly  Daily

Days:  Mon  Tue  Wed  Thu  Fri  Sat

Time: 6:00 pm

**Recipients**

⚠ Recipients see the same report data as the person running the report.

Receive new results by email when dashboard is refreshed. ⓘ

Send email to: Me

[Edit Recipients](#)

[Unsubscribe](#) [Cancel](#) [Save](#)