

# Garage Management System

Name: Vyshnavi Challa

College Name: Annamacharya Institute of Technology& Sciences,Rajampet

Mail ID: vyshnavichalla984@gmail.com

## *Project Overview:*

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 centres, 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 stock outs.
- b. Benstock biomechanics 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 prioritise jobs and ensures timely completion.

### **Creating a Salesforce Developer Account :**

Creating a Salesforce Developer account is the first step to building and testing Salesforce applications. This guide walks you through the process.

#### **Step-by-Step Instructions**

1.  Go to the Salesforce Developer Signup Page
  - URL: <https://developer.salesforce.com/signup>
2.  Fill in the Signup Form

Field	What to Enter
First Name	Your actual first name
Last Name	Your actual last name
Email	Your valid email address
Role	Select <b>Developer</b>
Company	Enter your <b>College Name</b>
Country	Select <b>India</b>
Postal Code	Enter your area <b>PIN Code</b>
Username	A combination of your name and company, in email format doesn't have to be a real email

3.  Submit the Form
  - After entering all the details, click on the **Sign Me Up** button.

**Sign up for your Developer Edition**

A free Salesforce Platform environment with Agentforce and Data Cloud

First name	Last name
Vyshnavi	Challa

Job title	Work email
Developer	vyshnavichalla984@salesforce.com

Company	Country/Region
AU	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.

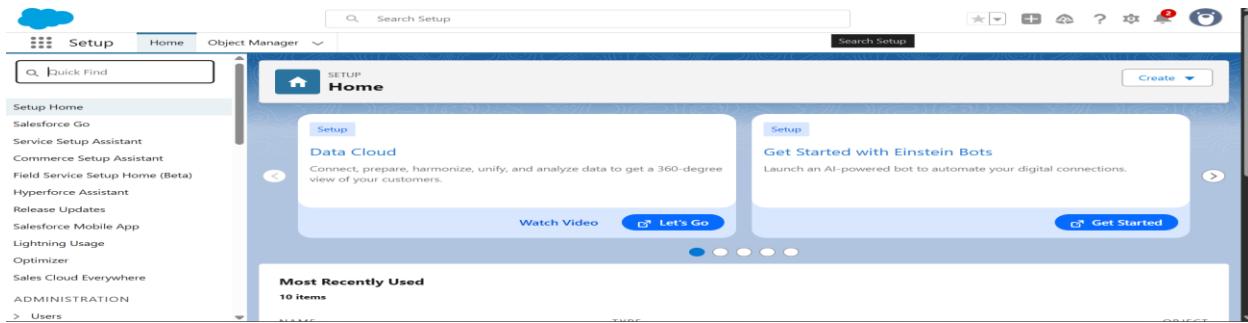
#### 4. Verify Your Email

- You'll receive a verification email from Salesforce.
- Open the email and click on the **verification link** to activate your Developer org.

#### 5. Next Steps

After account creation:

- Log in to your Salesforce Developer org at <https://login.salesforce.com>
- Begin creating objects, flows, Apex code, or Lightning components for your project



#### Creating the Customer Details Custom Object in Salesforce :

This section explains how to create a custom object called **Customer Details** in your Salesforce Developer Org.

◆ Step-by-Step Instructions

1.  Navigate to Object Manager

- Click on the **Setup** icon () in the top-right corner.
- In the Quick Find box, search for **Object Manager**.
- Click on **Object Manager**.

2.  Create a Custom Object

- In Object Manager, click **Create → Custom Object**.

3.  Fill in Object Details

Field	Value
Label	Customer Details
Plural Label	Customer Details
Object Name	Auto-filled (Customer_Details)
Record Name Label	Customer Name
Data Type	Text

4.  Enable the Following Options

-  **Allow Reports**
-  **Track Field History**
-  **Allow Search**

5.  Save the Object

- After entering the above details and selecting the options, click on **Save**.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. The main page title is 'Customer Detail'. On the left, there's a sidebar with various tabs like 'Details', 'Fields & Relationships', 'Page Layouts', etc. The 'Details' tab is active. The main content area shows the 'Customer Detail' object configuration. It includes fields for 'API Name' (Customer\_Detail\_\_c), 'Custom' (selected), 'Singular Label' (Customer Detail), and 'Plural Label' (Customer Details). On the right, there are sections for 'Enable Reports' (checked), 'Track Activities', 'Track Field History' (checked), 'Deployment Status' (Deployed), and 'Help Settings' (Standard salesforce.com Help Window). There are 'Edit' and 'Delete' buttons at the top right.

## Creating the Appointment Custom Object in Salesforce :

This guide details how to create a custom object named **Appointment** in Salesforce to manage customer service bookings and appointments.

### ◆ Step-by-Step Instructions

#### 1. Navigate to Object Manager

- Click the **Setup** gear icon at the top-right corner of the screen.
- In the **Quick Find** box, type and select **Object Manager**.
- Click **Create → Custom Object**.

#### 2. Fill in Object Information

Field	Value
Label	Appointment
Plural Label	Appointments
Object Name	Auto-filled (Appointment)

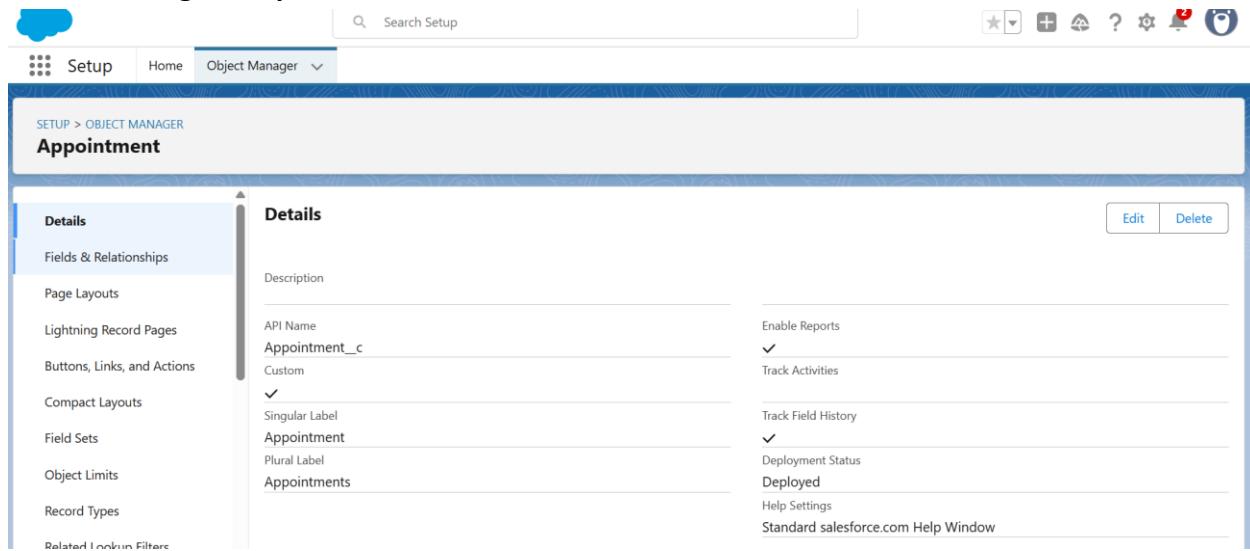
Record Name Label	Appointment Name
Data Type	Auto Number
Display Format	app-{000}
Starting Number	1

### 3. Enable Object Options

- **Allow Reports**
- **Track Field History**
- **Allow Search**

### 4. Save the Object

After entering all required details, click on the Save button.



## Creating the Service Records Custom Object in Salesforce :

The **Service Records** object is used to store details of services provided for each appointment. Follow these steps to create it in your Salesforce Developer Org.

### ◆ Step-by-Step Instructions

#### 1. Navigate to Object Manager

- From the **Setup** page (click the gear icon in the top-right),

- Go to **Object Manager**.
- Click on **Create → Custom Object**.

## 2. Enter Object Details

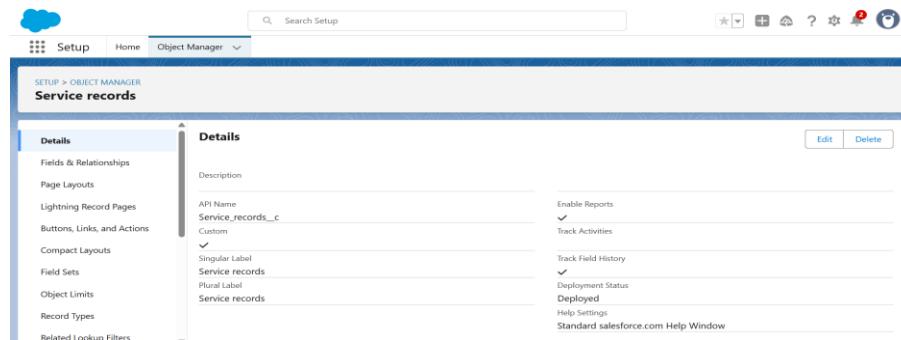
Field	Value
Label	Service records
Plural Label	Service records
Object Name	Auto-filled (Service_records)
Record Name Label	Service records Name
Data Type	Auto Number
Display Format	ser-{000}
Starting Number	1

## 3. Enable the Following Options

- **Allow Reports**
- **Track Field History**
- **Allow Search**

## 4. Save the Object

- Click on the **Save** button to finish creating the object.



**Creating the Billing Details and Feedback Custom Object in Salesforce :**

The **Billing Details and Feedback** object is designed to capture customer billing amounts, payment details, and service feedback. This guide outlines the steps to create this object in your Salesforce Developer Org.

◆ Step-by-Step Instructions

1.  Navigate to Object Manager

- Go to the **Setup** page (click the  gear icon).
- In the **Quick Find** box, search for **Object Manager**.
- Click on **Object Manager**.
- Select **Create → Custom Object**.

2.  Enter Object Details

Field	Value
Label	Billing details and feedback
Plural Label	Billing details and feedback
Object Name	Auto-filled (Billing_details_and_feedback)
Record Name Label	Billing details and feedback Name
Data Type	Auto Number
Display Format	bill-{000}
Starting Number	1

3.  Enable Object Options

-  **Allow Reports**
-  **Track Field History**
-  **Allow Search**

4.  Save the Object

- Click the **Save** button to finish creating the custom object.

The screenshot shows the Salesforce Setup interface. At the top, there's a blue header bar with the Salesforce logo, a search bar labeled 'Search Setup', and various navigation icons. Below the header, the 'Object Manager' tab is selected. The main content area has a title 'SETUP > OBJECT MANAGER' and 'Billing details and feedback'. On the left, a sidebar titled 'Details' lists various configuration options like 'Fields & Relationships', 'Page Layouts', and 'Buttons, Links, and Actions'. The main right panel also has a 'Details' section with fields for 'Description', 'API Name' (set to 'Billing\_details\_and\_feedback\_\_c'), and other settings like 'Enable Reports' (checked) and 'Track Activities'. There are 'Edit' and 'Delete' buttons at the bottom right of the main panel.

## Creating a Custom Tab for Customer Details Object in Salesforce :

To allow users to easily access the Customer Details records, a **Custom Tab** is created in Salesforce. This guide outlines the step-by-step procedure to create and configure it.

---

### ◆ Step-by-Step Instructions

#### 1. Navigate to Tabs

- Go to the **Setup** page (click the gear icon in the top right).
- In the **Quick Find** box, type and select **Tabs**.

#### 2. Create a New Custom Object Tab

- Under the **Custom Object Tabs** section, click on **New**.

#### 3. Configure the Custom Tab

Field	Action
Object	Select Customer Details
Tab Style	Choose any style icon (e.g., contact, user)

Click **Next** to continue.

#### 4. Add to Profiles

- On the **Add to Profiles** page, **keep the default visibility settings**.
- Click **Next**.

#### 5. Add to Custom Apps

- On the **Add to Custom Apps** page:
  - **Uncheck** the option **Include Tab** for all apps.
  - Ensure that **Append tab to users' existing personal customizations** is **checked**.

Click **Save** to complete the tab creation.

#### Result

- The custom tab for Customer Details is now created.
- Users will be able to view and access records from the tab when it's included in a custom app.

### **Creating Tabs for Remaining Custom Objects in Salesforce :**

In this section, we will create tabs for the following custom objects:

- **Appointments**
- **Service records**
- **Billing details and feedback**

These tabs allow users to access object records easily through the Salesforce UI.

#### Step-by-Step Instructions

Repeat the steps below **for each object**: Appointments, Service records, and Billing details and feedback.

##### 1. Navigate to Tabs

- Go to the **Setup** page ( icon at the top-right).
- In the **Quick Find** box, type and click on **Tabs**.
- Under **Custom Object Tabs**, click **New**.

##### 2. Configure Tab Details

Field	Value
Object	Select one of the remaining objects: <ul style="list-style-type: none"><li>• Appointments</li><li>• Service records</li><li>• Billing details and feedback</li></ul>
Tab Style	Choose any icon style to represent the object

Click **Next**.

### 3. Add to Profiles

- Keep default visibility settings.
- Click **Next**.

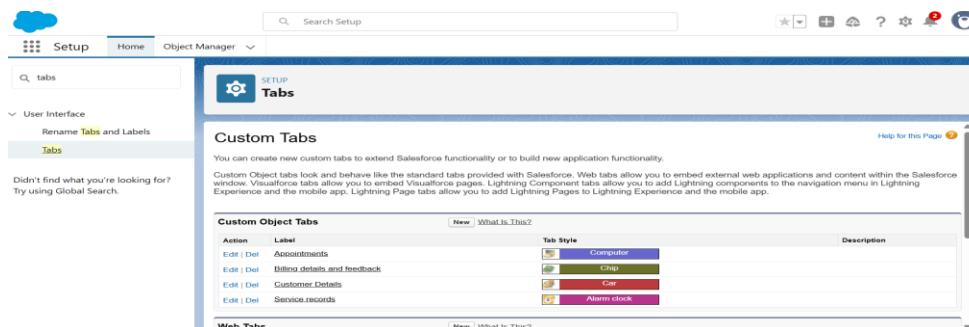
### 4. Add to Custom Apps

- **Uncheck** the **Include Tab** option for all listed apps.
- Ensure **Append tab to users' existing personal customizations** is **checked**.

Click **Save**.

Repeat

- Repeat these steps individually for all 3 objects:
  - Appointments
  - Service records
  - Billing details and feedback



## Creating a Lightning App for Garage Management System :

The Lightning App provides a centralized user interface to navigate and manage all Garage Management-related objects like customers, appointments, services, and billing.

### Step-by-Step Instructions

#### 1. Open the App Manager

- Go to **Setup** ( icon in the top-right).
- In the **Quick Find** box, search for **App Manager**.
- Click on **App Manager**.
- Click **New Lightning App** at the top-right.

#### 2. App Details

Field	Value
App Name	Garage Management Application

- Click **Next**.

#### 3. App Options

- Keep all **default settings** as-is.
- Click **Next**.

#### 4. Utility Items (Optional)

- Keep the **Utility Items page** as default.
- Click **Next**.

#### 5. Add Navigation Items

- In the **Available Items** panel, search and add the following objects:
  - Customer Details
  - Appointments
  - Service records
  - Billing details and feedback
  - Reports

- Dashboards
- Select each item and click the arrow to move them to the **Selected Items** panel.
- Click **Next**.

## 6. Assign User Profiles

- In the **Assign to User Profiles** screen:
  - Search for System Administrator.
  - Select it and click the arrow to move it to the selected profiles list.
- Click **Save & Finish**.

### Result

You've successfully created the **Garage Management Lightning App**, which provides a user-friendly navigation interface for:

- Customer Details
- Appointments
- Service Records
- Billing & Feedback
- Reports and Dashboards

You can now access the app from the **App Launcher** in the top-left corner of your Salesforce UI.

The screenshot shows the Lightning App Builder interface with the following details:

- Header:** Includes back, forward, and search icons, followed by "Lightning App Builder", "App Settings", "Pages", "Garage Management Application", and a help icon.
- Left Sidebar (App Settings):**
  - App Details & Branding** (selected)
  - App Options
  - Utility Items (Desktop Only)
  - Navigation Items
  - User Profiles
- Main Content Area:**
  - App Details & Branding:**
    - App Name:** Garage Management Application
    - Developer Name:** Garage\_Management\_Application
    - Description:** Enter a description...
  - App Details:**
    - Image:** Placeholder image with an "Upload" button.
    - Primary Color Hex Value:** #0070D2
  - App Branding:**
    - Image:** Placeholder image with an "Upload" button.
    - Primary Color Hex Value:** #0070D2
  - Org Theme Options:** A checkbox labeled "Use the app's image and color instead of the org's custom theme".
  - App Launcher Preview:** Shows a preview card with the "GM" logo and the text "Garage Management Appl...".

## **Creating Fields for the Customer Details Object in Salesforce :**

This guide explains how to add custom fields such as **Phone Number** and **Email (Gmail)** to the Customer Details object using the Salesforce Object Manager.

### ◆ **1. Navigate to the Object**

- Go to **Setup** (  icon in top right)
- Click on **Object Manager**
- In the **Quick Find/Search** bar, type Customer Details
- Click on the Customer Details object

### ◆ **2. Add a Phone Number Field**

#### **Steps:**

1. Inside the object, click on **Fields & Relationships**
2. Click **New**
3. Select **Data Type** as Phone → Click **Next**

#### **Fill the Field Details:**

Field	Value
<b>Field Label</b>	Phone number
<b>Field Name</b>	Auto-generated

4. Click **Next**
5. Click **Next** again to accept visibility and layout settings
6. Click **Save & New** (to add another field)

### ◆ **3. Add a Gmail Field**

#### **Steps:**

1. After saving the previous field, you're already on the New Field screen.
2. Select **Data Type** as Email → Click **Next**

#### **Fill the Field Details:**

Field	Value
Field Label	Gmail
Field Name	Auto-generated

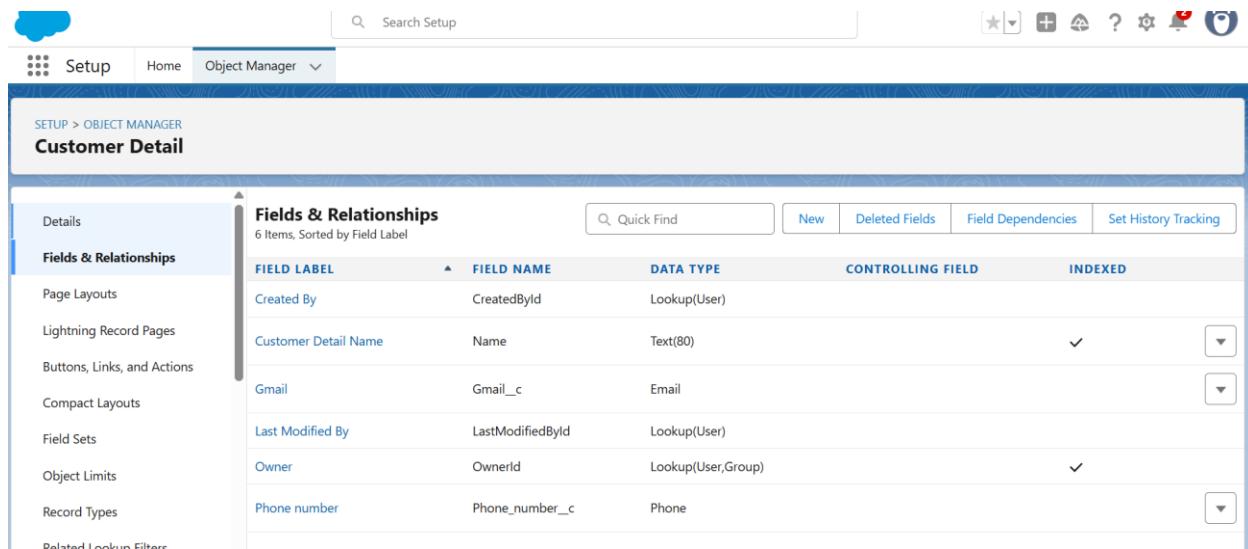
3. Click **Next**
4. Click **Next** again
5. Click **Save**

 **Outcome:**

You have successfully added the following fields to the **Customer Details** object:

- Phone number (Phone)
- Gmail (Email)

These fields are now available for use in record pages, flows, validation rules, and reports.



The screenshot shows the Salesforce Object Manager interface for the 'Customer Detail' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Fields & Relationships' and displays a table of fields. The table columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are: Created By (CreatedBy), Customer Detail Name (Name), Gmail (Gmail\_\_c), Last Modified By (LastModifiedById), Owner (OwnerId), and Phone number (Phone\_number\_\_c). The 'INDEXED' column contains checkmarks for most fields except 'Phone number'.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Customer Detail 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		

### **Creating Lookup Fields in Salesforce :**

Lookup relationships are used to establish parent-child connections between different objects. This section walks through creating lookup fields to connect:

- **Appointment → Customer Details**

- **Service Records → Appointment**
- **Billing Details and Feedback → Service Records**
  - ◆ **1. Lookup Field on Appointment Object**

**Purpose:**

To associate an appointment with a specific customer.

**Steps:**

1. Go to **Setup → Object Manager**
2. Search for and click on the **Appointment** object
3. Click on **Fields & Relationships**
4. Click **New**
5. Select **Lookup Relationship** → Click **Next**
6. For Related Object, select **Customer Details** → Click **Next**
7. Click **Next** on all remaining steps
8. Click **Save**

 **Result:** A lookup field is created that links each Appointment record to a Customer Details record.

- ◆ **2. Lookup Field on Service Records Object**

**Purpose:**

To associate each service record with a specific appointment.

**Steps:**

1. Go to **Setup → Object Manager**
2. Search for and click on the **Service Records** object
3. Click on **Fields & Relationships**
4. Click **New**
5. Select **Lookup Relationship** → Click **Next**
6. For Related Object, select **Appointment** → Click **Next**

### **Additional Settings:**

- Check the box for **Required Field**
- Scroll down to **Lookup Filter** section
- Click **Show Filter Settings**

### **Filter Criteria:**

**Field**            **Value**

**Field**            Appointment: Appointment Date

**Operator**        Less than

**Value**            Appointment: Created Date

**Filter Type**      Required

**Error Message** Value does not match the criteria.

**Active**             Checked

7. Click **Next → Next → Save**

 **Result:** A filtered, required lookup is created linking Service Records to valid Appointment records.

◆ **3. Lookup Field on Billing Details and Feedback Object**

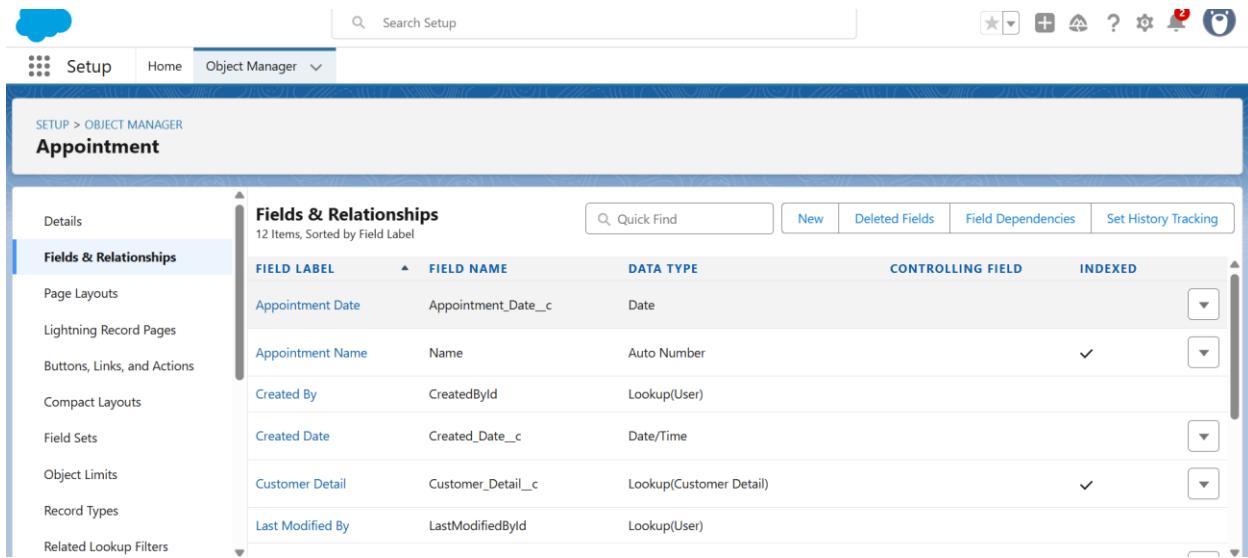
### **Purpose:**

To link each billing record and customer feedback entry to a specific service record.

### **Steps:**

1. Go to **Setup → Object Manager**
2. Search for and click on **Billing Details and Feedback**
3. Click on **Fields & Relationships**
4. Click **New**
5. Select **Lookup Relationship** → Click **Next**
6. Select **Service Records** as the related object → Click **Next**
7. Click **Next → Next → Save & New**

 **Result:** A lookup field is created that connects billing and feedback records to their respective service.



The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. Under the 'Appointment' object, the 'Fields & Relationships' section is active. The table lists various fields with their labels, names, data types, controlling fields, and indexing status. Key fields shown include 'Appointment Date' (Date), 'Appointment Name' (Name), 'Created By' (Lookup(User)), 'Created Date' (Date/Time), 'Customer Detail' (Lookup(Customer Detail)), and 'Last Modified By' (Lookup(User)).

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)		
Created Date	Created_Date_c	Date/Time		
Customer Detail	Customer_Detail_c	Lookup(Customer Detail)		✓
Last Modified By	LastModifiedBy	Lookup(User)		

## Creating Checkbox Fields in Salesforce :

Checkbox fields are Boolean (true/false) values that are useful for tracking statuses or binary options in your records. This guide covers how to add custom checkbox fields to the **Appointment** and **Service Records** objects.

- ◆ **1. Checkbox Fields on Appointment Object**

 **1.1 Maintenance Service**

**Purpose:** To track whether a maintenance service is required.

**Steps:**

1. Navigate to **Setup** → **Object Manager**
2. Search and select **Appointment**
3. Go to **Fields & Relationships**
4. Click **New**
5. Select **Checkbox** as the Data Type → Click **Next**
6. Field Label: Maintenance Service  
Field Name: (*auto-populated*)  
Default Value: Unchecked
7. Click **Next** → **Next** → **Save**

## **1.2 Repairs**

**Purpose:** To indicate if repair service is needed.

**Repeat steps 1–4 from above**, then:

6. Field Label: Repairs  
Field Name: *(auto-populated)*  
Default Value: Unchecked
7. Click **Next → Next → Save**

## **1.3 Replacement Parts**

**Purpose:** To track if replacement parts are needed for a service.

**Repeat steps 1–4 again**, then:

6. Field Label: Replacement Parts  
Field Name: *(auto-populated)*  
Default Value: Unchecked
7. Click **Next → Next → Save**

### ◆ **2. Checkbox Field on Service Records Object**

Quality Check Status :

**Purpose:** To indicate if a quality check has been completed for the service.

**Steps:**

1. Navigate to **Setup → Object Manager**
2. Search and select **Service Records**
3. Go to **Fields & Relationships**
4. Click **New**
5. Select **Checkbox** as the Data Type → Click **Next**
6. Field Label: Quality Check Status  
Field Name: *(auto-populated)*  
Default Value: Unchecked
7. Click **Next → Next → Save**

SETUP > OBJECT MANAGER

## Appointment

**Maintenance service**

Custom Field Definition Detail

Field Label	Maintenance service	Object Name	Appointment
Field Name	Maintenance_service	Data Type	Checkbox
API Name	Maintenance_service__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Vyshnavi Challa, 7/12/2025, 3:11 AM	Modified By	Vyshnavi Challa, 7/12/2025, 3:11 AM

### Creating Date Field in Salesforce :

Date fields are used to capture calendar dates, useful for scheduling appointments, tracking events, or logging service deadlines.

- ◆ **1. Date Field on Appointment Object**

#### **Field: Appointment Date**

**Purpose:** To store the date on which an appointment is scheduled.

#### **Steps to Create:**

1. Navigate to **Setup**.
2. Click on **Object Manager** from the top menu.
3. In the **Quick Find** bar, type and select **Appointment**.
4. In the left panel, click on **Fields & Relationships**.
5. Click **New** to create a new field.
6. Select **Data Type** as **Date** → Click **Next**.
7. Fill in the field details:
  - **Field Label:** Appointment Date

- **Field Name:** *(auto-populated)*
- Check the **Required** checkbox to ensure this field must be filled in every time a record is created.

8. Click **Next** → **Next** → **Save**.

 **Security & Usage Notes:**

- Making it a **required** field ensures that every appointment record must have a valid date.
- This field can be used in **report filters**, **flow conditions**, or **lookup filters** (e.g., validating past appointments).

### **Creation of Currency Fields in Salesforce :**

Currency fields are used to store monetary values such as billing amounts, service charges, or payments made. These are essential in financial tracking within the application.

◆ **1. Currency Field on Appointment Object**

 **Field: Service Amount**

**Purpose:** To capture the amount charged for a service during the appointment.

 **Steps to Create:**

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Search and select **Appointment**.
4. Go to **Fields & Relationships** → Click **New**.
5. Select **Currency** as the **Data Type** → Click **Next**.
6. Enter field details:
  - **Field Label:** Service Amount
  - **Field Name:** *(auto-generated)*
7. Click **Next**.
8. In **Field-Level Security**, set the field to **Read-Only** for all profiles (ensure users can see but not edit this field).

9. Click **Next** → **Save**.
  - ◆ **2.** Currency Field on Billing details and feedback Object
-  **Field: Payment Paid**

**Purpose:** To store the payment made by the customer for services rendered.

 **Steps to Create:**

  1. Follow steps 1–5 from the previous section, but search and select the **Billing details and feedback** object.
  2. Field details:
    - **Field Label:** Payment Paid
    - **Field Name:** (*auto-generated*)
  3. Click **Next**.
  4. In **Field-Level Security**, set it to **Read-Only** for all profiles.
  5. Click **Next** → **Save**.

**Creation of Text Fields in Salesforce :**

Text fields are used to capture short string-based data such as license plate numbers, ratings, or reference notes.

  - ◆ **1.** Text Field on Appointment Object-  **Field: Vehicle Number Plate**

**Purpose:** To capture the customer's vehicle registration number for service tracking.

 **Steps to Create:**

  1. Go to **Setup**.
  2. Click on **Object Manager**.
  3. Type **Appointment** in the search bar and click on the object.
  4. Go to **Fields & Relationships** → Click **New**.
  5. Select **Text** as the **Data Type** → Click **Next**.
  6. Fill in the details:

- **Field Label:** Vehicle Number Plate
- **Field Name:** (*auto-generated*)
- **Length:** 10
- **Field Attributes:**
  - Check **Required**
  - Check **Unique**

7. Click **Next** → **Next** → **Save**.

- ◆ **2.** Text Field on Billing details and feedback Object

### **Field: Rating for Service**

**Purpose:** To collect customer feedback in the form of a single-character rating (e.g., 1–5 or A–E).

#### **Steps to Create:**

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Type **Billing details and feedback** in the search bar and click on the object.
4. Go to **Fields & Relationships** → Click **New**.
5. Select **Text** as the **Data Type** → Click **Next**.
6. Fill in the details:
  - **Field Label:** Rating for Service
  - **Field Name:** (*auto-generated*)
  - **Length:** 1
  - **Field Attributes:**
    - Check **Required**

7. Click **Next** → **Next** → **Save**.

### **Creation of Picklist Fields in Salesforce :**

Picklist fields are used to provide users with a list of predefined values to choose from, ensuring consistency and data accuracy across records.

- ◆ **1. Picklist Field on Service Records Object**

### **Field: Service Status**

**Purpose:** To track the current status of the service being performed.

#### **Steps to Create:**

1. Navigate to **Setup**.
  2. Click on **Object Manager**.
  3. Type **Service Records** in the search bar and click on the object.
  4. Click on **Fields & Relationships** → Click **New**.
  5. Choose **Picklist** as the **Data Type** → Click **Next**.
  6. Fill in the field details:
    - **Field Label:** Service Status
    - **Values:**
      - nginx
      - CopyEdit
      - Started
      - Completed
    - Choose “**Enter values, with each value separated by a new line.**”
  7. Click **Next**.
  8. Field-Level Security: Keep as default → Click **Next**.
  9. Add to page layouts: Keep as default → Click **Save**.
- ◆ **2. Picklist Field on Billing Details and Feedback Object**

### **Field: Payment Status**

**Purpose:** To track the payment status after service completion.

#### **Steps to Create:**

1. Go to **Setup**.
2. Click on **Object Manager**.

3. Type **Billing Details and Feedback** in the search bar and click on the object.
4. Click on **Fields & Relationships** → Click **New**.
5. Choose **Picklist** as the **Data Type** → Click **Next**.
6. Fill in the field details:
  - **Field Label:** Payment Status
  - **Values:**
  - nginx
  - CopyEdit
  - Pending
  - Completed
  - Choose “**Enter values, with each value separated by a new line.**”
7. Click **Next**.
8. Field-Level Security: Keep as default → Click **Next**.
9. Add to page layouts: Keep as default → Click **Save**.

#### **Creation of Formula Field in Salesforce :**

Formula fields are read-only fields that automatically calculate values based on other fields or functions. In this case, the formula field will extract and store the **date** when the **Service Record** was created.

- ◆ Object: Service Records
- █ Field: service date (Formula Field)

**Purpose:** To capture the **creation date** of the service record automatically.

#### **Steps to Create the Formula Field:**

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Type **Service Records** in the search bar → Click on the object.
4. Select **Fields & Relationships** → Click **New**.
5. Select **Formula** as the **Data Type** → Click **Next**.

 **Formula Field Configuration:**

- **Field Label:** service date
- **Field Name:** service\_date (auto-generated)
- **Formula Return Type:** Date → Click **Next**

 **Enter the Formula:**

- **Formula Expression:**
- plaintext
- CopyEdit
- CreatedDate
- Click on **Check Syntax** to validate the formula.
- No errors should be present.

 **Final Steps:**

1. Click **Next**.
2. Assign **Field-Level Security** as needed (default is fine).
3. Click **Next** to add the field to page layouts.
4. Click **Save**.

**Creation of Validation Rule in Salesforce :**

Validation rules verify that the data entered by users meets the standards you specify before the record can be saved. This rule ensures that the **Vehicle Number Plate** follows a valid Indian registration format.

 **Object:** Appointment **Validation Rule:** Vehicle Number Plate Format

**Purpose:** To ensure the **Vehicle Number Plate** entered follows a valid format (e.g., AP01AB1234).

 **Steps to Create the Validation Rule:**

1. Go to **Setup**.

2. Click on **Object Manager**.
3. Type **Appointment** in the search bar and click on it.
4. In the left sidebar, click on **Validation Rules**.
5. Click the **New** button.

 Validation Rule Configuration:

- **Rule Name:** Vehicle
- **Error Condition Formula:**
  - plaintext
  - CopyEdit
  - NOT(REGEX(Vehicle\_number\_plate\_\_c, "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))
- **Error Message:**
  - typescript
  - CopyEdit
  - Please enter valid number
- **Error Location:**
  - Choose: **Field**
  - Field: **Vehicle number plate**

 Explanation of REGEX Pattern:

- [A-Z]{2} → 2 uppercase letters (e.g., AP)
- [0-9]{2} → 2 digits (e.g., 01)
- [A-Z]{2} → 2 uppercase letters again (e.g., AB)
- [0-9]{4} → 4 digits (e.g., 1234)

 **Example Valid Format:** TS08GH4567

**Validation Rules**  
1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	Vyshnavi Challa, 7/13/2025, 9:28 AM

### Creation of Validation Rule in Salesforce :

This validation rule ensures the **Rating for Service** entered by the user falls within a valid range from **1 to 5** only.

- ◆ Object: Billing details and feedback

#### ■ **Validation Rule: Rating Should Be Less Than 5**

**Purpose:** To ensure that the **Rating for Service** field accepts only numbers between **1** and **5**.

#### ✓ Steps to Create the Validation Rule:

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Type **Billing details and feedback** in the search bar and click on it.
4. In the left sidebar, click on **Validation Rules**.
5. Click the **New** button.

#### ⚙ Validation Rule Configuration:

- **Rule Name:**  
rating\_should\_be\_less\_than\_5
- **Error Condition Formula:**
- plaintext

- CopyEdit
- NOT(REGEX(Rating\_for\_service\_\_c, "[1-5]{1}"))
- **Error Message:**
- CSS
- CopyEdit
- Rating should be from 1 to 5
- **Error Location:**
  - Choose: **Field**
  - Field: **Rating for Service**

### **Creation of Matching Rule in Salesforce :**

A **Matching Rule** identifies duplicate records based on defined criteria. In this case, it ensures that **Customer Details** records are uniquely identified using the customer's **Gmail** and **Phone Number**.

◆ Object: Customer Details

 Matching Rule: Matching Customer Details

**Purpose:** To prevent duplicate customer records in the system by matching based on **Email (Gmail)** and **Phone Number**.

 Steps to Create the Matching Rule:

1. Go to **Setup**.
2. In the **Quick Find Box**, type and select **Matching Rules**.
3. Click the **New Rule** button.

 Matching Rule Configuration:

- **Object:**  
Customer Details
- Click **Next**.

- **Rule Label:**

Matching Customer Details

- **Unique Name:**

(Auto-populated)

📌 Define Matching Criteria:

Field	Matching Method
-------	-----------------

Gmail	Exact Match
-------	-------------

Phone Number	Exact Match
--------------	-------------

📌 Use the **Add Row** option to include both fields and their corresponding methods.

The screenshot shows the Salesforce Setup interface with the 'Matching Rules' page selected. The page title is 'Matching Rules' under the 'SETUP' tab. The main content area is titled 'All Matching Rules' and contains a section 'What Are Matching Rules?'. Below this is a table listing matching rules, with one row highlighted. The table columns are: Action, Rule Name, Object, Status, Description, Last Modified Date, and Last Modified By. The highlighted row shows 'Matching customer details' for the 'Customer Detail' object, which is active and described as a matching rule for account records. The 'Last Modified Date' is 7/12/2025 and 'Last Modified By' is 'OEPIC'.

Action	Rule Name	Object	Status	Description	Last Modified Date	Last Modified By
Del   Deactivate	Matching customer details	Customer Detail	Active	Matching rule for account records. <a href="#">More info</a>	7/12/2025	OEPIC
Deactivate	Standard Account Matching Rule	Account	Active	Matching rule for account records. <a href="#">More info</a>	7/3/2025	OEPIC
Deactivate	Standard Contact Matching Rule	Contact	Active	Matching rule for contact records. <a href="#">More info</a>	7/3/2025	OEPIC
Deactivate	Standard Lead Matching Rule	Lead	Active	Matching rule for lead records. <a href="#">More info</a>	7/3/2025	OEPIC

## Creation of Duplicate Rule in Salesforce :

Duplicate Rules work alongside Matching Rules to determine what happens when a user tries to create or edit a record that matches an existing one. This documentation outlines how to set up a **Duplicate Rule** using the previously created **Matching Rule**.

Object: Customer Details

📋 Duplicate Rule: Customer Detail Duplicate

**Purpose:** Prevent users from creating duplicate customer records based on **Gmail** and **Phone Number**, ensuring clean and accurate data.

 Steps to Create the Duplicate Rule:

1. Go to **Setup**.
2. In the **Quick Find Box**, type and select **Duplicate Rules**.
3. Click on the **New Rule** button.
4. From the object list, select **Customer Details**.

 Duplicate Rule Configuration:

- **Rule Label:**  
Customer Detail Duplicate
- **Unique Name:**  
*(Auto-populated)*
- **Record-Level Security:**  
Choose whether to enforce or bypass sharing rules (default settings are usually sufficient for internal enforcement).

 Matching Rule Association:

1. Scroll down to the **Matching Rule** section.
2. Select the matching rule you previously created:  
Matching customer details

 Save and Activate:

1. Click **Save** to save the rule.
2. After saving, click on **Activate** to enable the rule in your Salesforce instance.

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

- Header:** Search Setup, Home, Object Manager.
- Left Sidebar:** Data, Duplicate Management, Duplicate Error Logs, **Duplicate Rules** (selected), Matching Rules.
- Middle Content:**
  - Section:** What Are Duplicate Rules?
  - View:** All Duplicate Rules
  - Table:** A list of duplicate rules with columns: Rule Name, Description, Object, Matching Rule, Active, Last Modified By, Last Modified Date.
- Bottom:** Navigation links for letters A-Z and Other.

## Creating the Manager Profile in Salesforce:

This document outlines the step-by-step process to create a **Manager Profile** in Salesforce by cloning a Standard User profile and customizing permissions, app settings, session timeout, and password policies.

### Steps to Create Manager Profile

#### 1. Clone a Profile

- Go to **Setup**.
- In the **Quick Find Box**, type **Profiles**.
- Click on **Profiles**.
- Locate and click **Clone** next to the **Standard User** profile.
- In the **Profile Name** field, enter: Manager.
- Click **Save**.

#### 2. Edit Manager Profile Settings

After saving, you will be redirected to the Manager profile page.

Click **Edit** to customize the following settings:

##### Custom App Settings

- Scroll to **Custom App Settings**.

- Set the default app to **Garage Management** by selecting it as the default and ensuring visibility is checked.

#### Custom Object Permissions

Provide the following access levels for the custom objects:

Object Name	Read	Create	Edit	Delete	View All	Modify All
Appointments	✓	✓	✓	✓	✓	✓
Billing Details & Feedback	✓	✓	✓	✓	✓	✓
Service Records	✓	✓	✓	✓	✓	✓
Customer Details	✓	✓	✓	✓	✓	✓

#### Session Settings

- Scroll down to **Session Settings** section.
- Set **Session Timeout** to: 8 hours of inactivity.

#### Password Policies

Adjust the password settings:

Setting	Value
User passwords expire in	Never Expires
Minimum password length	8 characters

Click **Save** to apply all changes.

Action	Profile Name	User License	Custom
<input type="checkbox"/> Edit   Del  ...	sales_person	Salesforce Platform	✓
<input type="checkbox"/> Edit   Del  ...	Salesforce API Only System Integrations	Salesforce Integration	✓
<input type="checkbox"/> Edit   Clone	Silver Partner User	Silver Partner	□
<input type="checkbox"/> Edit   Clone	Solution Manager	Salesforce	□
<input type="checkbox"/> Edit   Clone	Standard Platform User	Salesforce Platform	□
<input type="checkbox"/> Edit   Clone	Standard User	Salesforce	□
<input type="checkbox"/> Edit   Clone	System Administrator	Salesforce	□

## Creating the Sales Person Profile in Salesforce :

This document outlines the step-by-step process to create a **Sales Person** profile by cloning the **Salesforce Platform User** profile and configuring it with appropriate app settings and object permissions specific to the Garage Management System.

### Steps to Create Sales Person Profile

#### 1. Clone the Profile

- Navigate to **Setup**.
- In the **Quick Find Box**, type **Profiles**.
- Click on **Profiles**.
- Locate and click **Clone** next to the **Salesforce Platform User** profile.
- In the **Profile Name** field, enter: Sales Person.
- Click **Save**.

#### 2. Edit Sales Person Profile Settings

After saving, you'll be taken to the profile overview page.

Click **Edit** to begin customization.

#### Custom App Settings

- Scroll to the **Custom App Settings** section.
- Set the default app to **Garage Management**.

- Ensure the **Garage Management App** is visible and selected as the default app.

#### Custom Object Permissions

Provide access permissions for the following custom objects:

Object Name	Read	Create	Edit	Delete	View All	Modify All
Appointments	✓	✓	✓	✗	✗	✗
Billing Details & Feedback	✓	✓	✓	✗	✗	✗
Service Records	✓	✓	✓	✗	✗	✗
Customer Details	✓	✓	✓	✗	✗	✗

- ◆ You may update the access matrix as per your organization's policies.

#### Final Step

Click **Save** to apply the changes.

#### Creating the Manager Role in Salesforce:

This documentation outlines the process of creating a **Manager Role** within the Salesforce role hierarchy for the **Garage Management System**.

#### Steps to Create the Manager Role

##### 1. Navigate to Roles Setup

- Go to **Setup**.
- In the **Quick Find Box**, type: Roles
- Click on **Set Up Roles**.

##### 2. Access the Role Hierarchy

- On the **Roles** page, click **Expand All** to view the full role hierarchy.

##### 3. Add a New Role

- Decide under which existing role the new **Manager** role should report.
- Click **Add Role** beneath the desired supervisor/parent role.

##### 4. Define the Role

- In the **Label** field, enter: Manager

- The **Role Name** will be auto-populated based on the label.
- Optionally, add a **Role Description** for clarity (e.g., "Responsible for overseeing service operations and staff").

## 5. Save the Role

- Click **Save** to create the role and insert it into the hierarchy.

## Creating Additional Roles under Manager in Salesforce :

To create two new roles — **Sales Person** and another (e.g., Technician or Assistant) — under the **Manager** role within the role hierarchy for the **Garage Management System** in Salesforce.

### Steps to Create a New Role under Manager

#### 1. Navigate to Role Setup

- Go to **Setup**.
- In the **Quick Find** box, type: Roles
- Click on **Set Up Roles**.

#### 2. Open the Role Hierarchy

- On the Roles page, click **Expand All** to display the full hierarchy.
- Locate the **CEO** role at the top.
- Click the **plus icon (+)** next to **CEO** to expand the subtree.
- Find the **Manager** role.

#### 3. Add Role under Manager

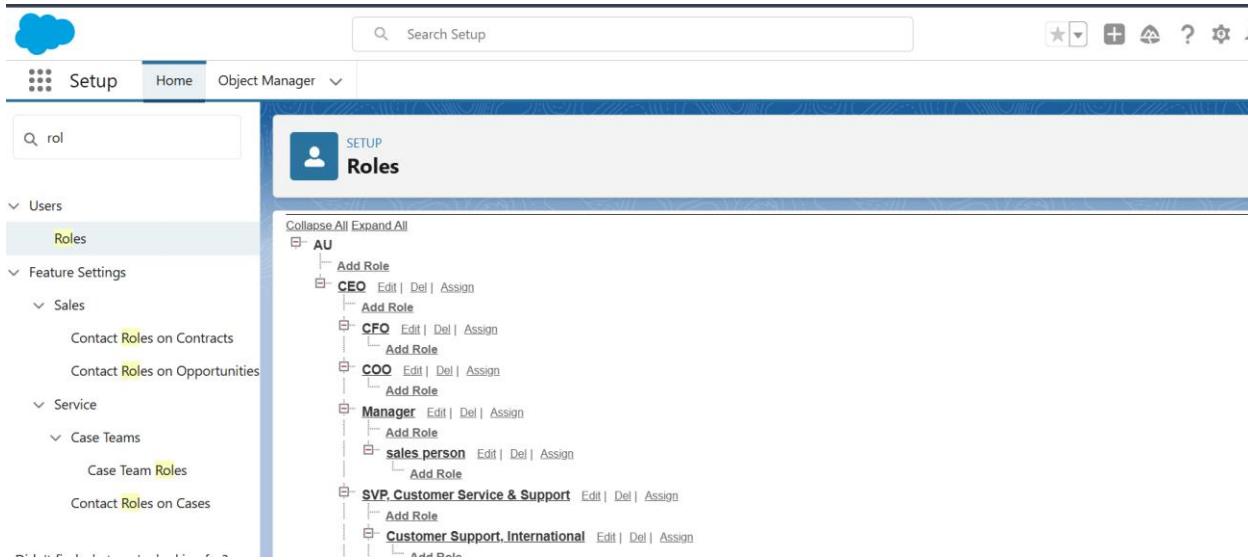
- Click **Add Role** under the **Manager** role.

#### 4. Define the Role

- In the **Label** field, enter: Sales Person
- The **Role Name** will auto-populate.
- Optionally, enter a **Description** like "Handles sales and interacts with customers".

#### 5. Save the Role

- Click **Save**.



## Creating a User in Salesforce :

To create a new user account with the **Manager** role and **Manager** profile in the Salesforce environment for the Garage Management System.

### Steps to Create a New User

#### 1. Navigate to Users Setup

- Go to **Setup**.
- In the **Quick Find box**, type: Users
- Click on **Users** under **Users** section.

#### 2. Add a New User

- Click on the **New User** button at the top.

#### 3. Fill in User Details

Field	Value / Description
-------	---------------------

First Name Niklaus

Last Name Mikaelson

Alias A short alias name (e.g., n.mika)

Email Enter a valid personal email address

Username Must follow format: text@text.text (e.g., niklaus.mikaelson@garage.com)

Nickname Any nickname (e.g., nik)

Role Select: Manager

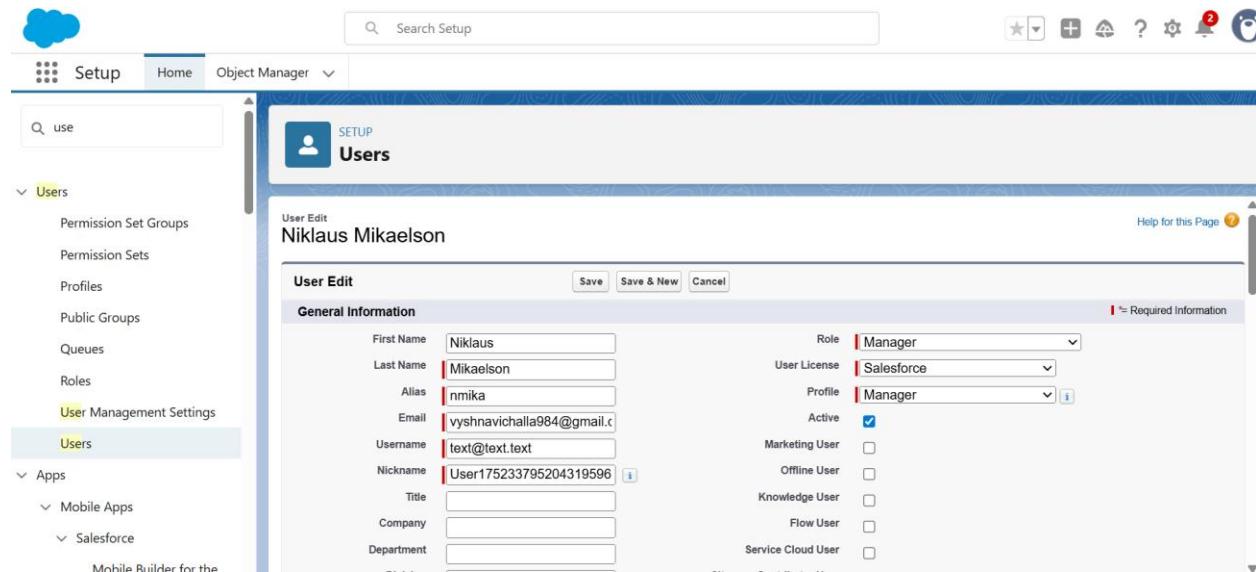
User License Select: Salesforce

Profile Select: Manager

#### 4. Save the User

- After filling in the fields, click **Save**.

 Salesforce will send a welcome email to the user with login setup instructions.



#### Creating Multiple Users in Salesforce (Sales Person Role) :

To create **three users** in the Salesforce Garage Management System with the **Sales Person** role, **Salesforce Platform** user license, and **Sales Person** profile.

Steps to Create Users

##### 1. Navigate to Users Setup

- Go to **Setup**.

- In the **Quick Find box**, type: Users
- Click on **Users** under **Users** section.

## 2. Create First User

- Click **New User**.
- Enter details.

## 3. Create Second User

Click **New User** again, and enter details.

## 4. Create Third User

Click **New User** again, and enter:

Three users have been successfully created with the following configurations:

- **Role:** Sales Person
- **User License:** Salesforce Platform
- **Profile:** Sales Person

Each user will receive an email to log in and set their password.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chella_Vyshnavi	vys	vyshnavichalla984509@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Chatter Expert	Chatter	chatty.00dg000006bybxuae.phis8c74fz0@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	EPIC_OrgFarm	OEPIG	epic.47814a80f39@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	garadi_deepika	dgarai	deep@gar.com	sales_person	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	ila_sakshi	sila	sakshi@fec.com	sales_person	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Mikaelson_Niklaus	nnika	text@text.text	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	User_Integration	integ	integration@00dg000006bybxuae.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	insightssecurity@00dg000006bybxuae.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

## Creating a New Public Group in Salesforce :

To create a **Public Group** named **Sales Team** and add the **Sales Person** role as its member in the Garage Management System project.

## Steps to Create the Public Group

### 1. Navigate to Public Groups

- Go to **Setup**.
- In the **Quick Find box**, type: Public Groups.
- Click on **Public Groups** under the **Users** section.
- Click on the **New** button at the top.

### 2. Fill Group Details

Field              Value

Label              Sales Team

Group Name      Auto-populated

### 3. Add Members

- Scroll to the **Search** dropdown → select **Roles**.
- From **Available Members**, find and select **Sales Person**.
- Click the **Add** button to move it to **Selected Members**.

### 4. Save Group

- Click **Save** to finish creating the public group.

The screenshot shows the Salesforce Setup interface. The left sidebar navigation bar is visible, with 'Users' expanded and 'Public Groups' selected. The main content area has a title 'Public Groups' with a subtitle 'A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.' Below this is a 'View' dropdown set to 'All' with options 'Edit' and 'Create New View'. At the bottom of the page, there is a table with one row:

Action	Label	Group Name	Created By	Created Date
<a href="#">Edit</a>   <a href="#">Delete</a>	sales_team	sales_team	Challa Vyshnavi	7/12/2025, 10:24 AM

## **Creating Sharing Settings in Salesforce :**

To configure **Object-Level Security** for the Service Records object using **Sharing Settings**.

The goal is to:

- Set **OWD (Organization-Wide Default)** access for Service Records to **Private**
- Create a **Sharing Rule** to allow **Managers** to access records owned by **Sales Persons**

Step-by-Step Process

### **1. Access Sharing Settings**

- Go to **Setup**.
- In the **Quick Find** box, type: Sharing Settings.
- Click on **Sharing Settings** under the **Security** section.
- Click **Edit** at the top of the page.

### **2. Modify OWD (Organization-Wide Default)**

- Scroll to **Service Records** in the OWD list.
- Set the **Default Internal Access** to: Private.
- Click **Save**.
- Refresh the page to apply changes.

### **3. Create a New Sharing Rule**

Navigate to Sharing Rules

- Scroll down to the **Service Records** section.
- Click on **New** to create a new rule.

Enter Rule Details

Field            Value

Label Name    Sharing setting

Rule Name      Auto-populated

### **4. Define Rule Logic**

### Step 3: Select Records to Share

- **From:**Roles
- **Member:**Sales Person

### Step 4: Share With

- **To:**Roles
- **Member:**Manager

### Step 5: Access Level

- **Access Level:**Read/Write

### Final Step

- Click **Save** to apply the sharing rule.

## **Creating a Record-Triggered Flow for Payment Acknowledgment in Salesforce :**

To create an automated **Record-Triggered Flow** that:

- Updates the **Payment Paid** field when the **Payment Status** is "Completed"
- Sends a **Thank You email** to the customer with the amount paid

### Step-by-Step Instructions

#### 1. Navigate to Flows

- Go to **Setup**.
- In the **Quick Find** box, type Flow.
- Click on **Flows**.
- Click **New Flow**.

#### 2. Choose Flow Type

- Select **Record-Triggered Flow**.
- Click **Create**.

#### 3. Configure Trigger

Setting	Value
---------	-------

<b>Object</b>	Billing details and feedback
<b>Trigger Flow When</b>	A record is Created or Updated

**Optimize the Flow For** Actions and Related Records

- Click **Done**.

#### 4. Add Update Records Element

- Click the “+” icon under the Record-Triggered Start element.
- Select **Update Records**.

**Fill in Details:**

Field	Value
-------	-------

<b>Label</b>	Amount
	Update

**API Name** Auto-populated

**Filter Condition (All Conditions Are Met - AND):**

Field	Operator Value
-------	----------------

Payment_Status__c	Equals	Completed
-------------------	--------	-----------

**Set Field Values to Update:**

Field	Value
-------	-------

Payment_Paid__c	!\$Record.Service_records__r.Appointment__r.Service_Amount__c
-----------------	---

- Click **Done**.

#### 5. Create a Text Template Resource

- Open **Toolbox** from top left.
- Click **New Resource**.
- Select **Resource Type**: Text Template
- API Name: alert
- Change view to **Plain Text**.

- **Body:**

text

CopyEdit

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 .

- Click **Done**.

## 6. Add Send Email Action

- Click the “+” icon.
- Select **Action**.
- In the Action panel, search for **Send Email** and click on it.

### Configure Email Action:

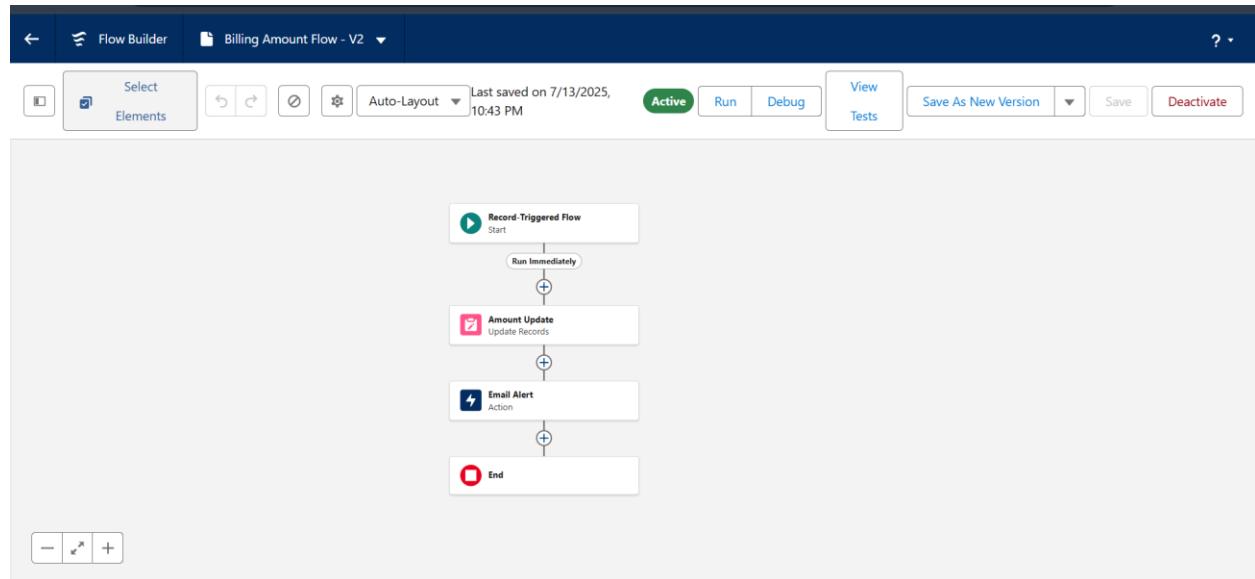
Field	Value
Label	Email Alert
API Name	Auto-populated
Body	!alert
RecipientAddressList	{!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
Subject	Thank You for Your Payment - Garage Management

- Click **Done**.

## 7. Save and Activate the Flow

- Click **Save**.

- Flow Label: Payment Email Flow (you can choose your own name)
- API Name: Auto-populated
- Click **Save** again.
- Click **Activate**.



### Flow Documentation: Update Service Status Automatically :

To automate the process of updating the **Service\_Status\_\_c** field to Completed when **Quality\_Check\_Status\_\_c** is marked as True in the **Service Records** object.

#### Step-by-Step Instructions

##### 1. Navigate to Flows

- Go to **Setup**.
- In the **Quick Find** box, type: Flow.
- Click on **Flows** under **Process Automation**.
- Click **New Flow**.

##### 2. Select Flow Type

- Choose **Record-Triggered Flow**.
- Click **Create**.

### 3. Configure Start Element

Setting	Value
Object	Service Records
Trigger Flow When	A record is Created or Updated

**Optimize the Flow For** Actions and Related Records

- Click **Done**.

### 4. Add Update Records Element

- Click on the "+" icon under the Record-Triggered Start.
- Select **Update Records** from the menu.

**Fill in the Details:**

Field	Value
Label	Update Service Status

**API Name** Auto-populated

**Configure Filter Conditions:**

Field	Operator	Value
Quality_Check_Status__c	Equals	True

**Set Field Values to Update:**

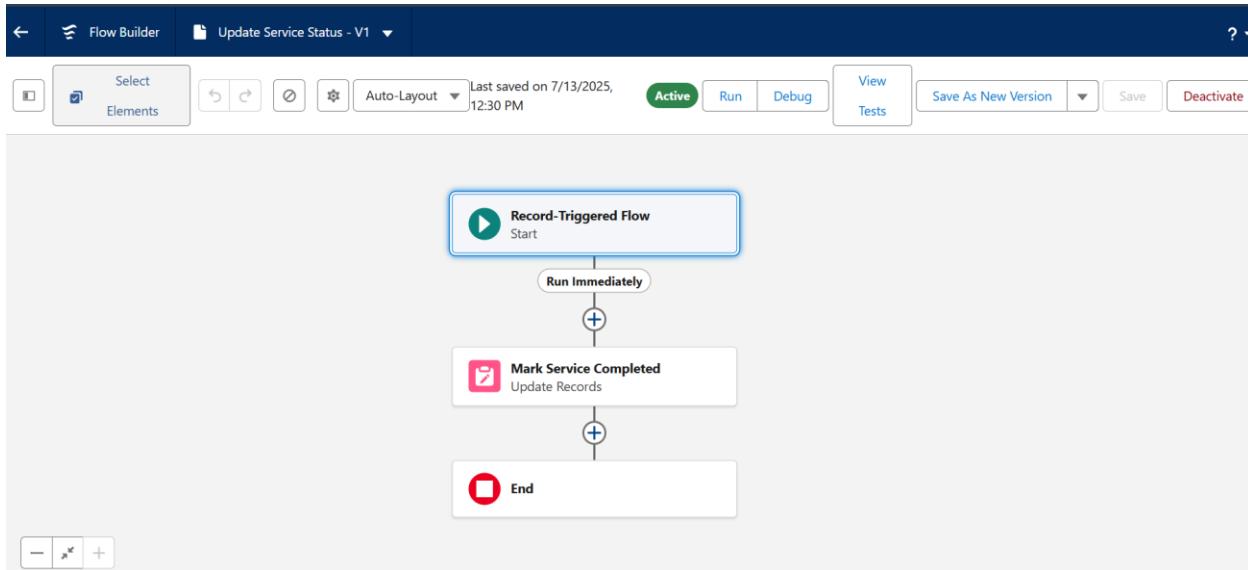
Field	Value
Service_Status__c	Completed

- Click **Done**.

### 5. Save and Activate the Flow

- Click **Save**.
- **Flow Label:** Update Service Status
- **API Name:** Auto-populated

- Click **Save** again.
- Click **Activate**.



### **Apex Handler and Trigger Documentation :**

#### **Use Case: Amount Distribution Based on Selected Services**

To automate the distribution of service cost based on the services selected (Maintenance, Repairs, Replacement Parts) by the customer during their vehicle appointment. This is done using an **Apex Class** and a **Trigger** on the **Appointment\_\_c** object.

#### **Step 1: Open Developer Console**

1. Login to your Salesforce Trailhead account.
2. Click the **Gear Icon** (top-right corner).
3. Select **Developer Console** – a new console window opens.

#### **Step 2: Create Apex Class – AmountDistributionHandler**

1. In Developer Console:
  - Click **File > New > Apex Class**.
  - Enter class name: **AmountDistributionHandler**.
  - Click **OK**.

#### **Paste the following code:**

apex

## CopyEdit

```
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;  
            }  
        }  
    }  
}
```

### Step 3: Create Trigger – AmountDistribution

#### 1. In the **Developer Console**:

- Click **File > New > Apex Trigger**.
- Enter:
  - **Trigger Name:** AmountDistribution
  - **sObject:** Appointment\_\_c
- Click **Submit**.

#### Use the following trigger code:

apex

## CopyEdit

```
trigger AmountDistribution on Appointment__c (before insert, before update) {
    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
        AmountDistributionHandler.amountDist(Trigger.new);
    }
}
```

Developer Console - Google Chrome  
orgfarm-fda56f0e83-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

AmountDistributionHandler.apxc \* AmountDistribution.apxt \*

Code Coverage: None API Version: 64

```
1 public class AmountDistributionHandler {
2     public static void amountDist(List<Appointment__c> listApp){
3         List<Service_records__c> serList = new List<Service_records__c>();
4         for(Appointment__c app : listApp){
5             if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
6                 app.Service_Amount__c = 10000;
7             }
8             else if(app.Maintenance_service__c == true && app.Repairs__c == true){
9                 app.Service_Amount__c = 5000;
10            }
11            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
12                app.Service_Amount__c = 8000;
13            }
14            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
15                app.Service_Amount__c = 7000;
16            }
17            else if(app.Maintenance_service__c == true){
18                app.Service_Amount__c = 2000;
19            }
20            else if(app.Repairs__c == true){
21                app.Service_Amount__c = 3000;
22            }
23            else if(app.Replacement_Parts__c == true){
24                app.Service_Amount__c = 5000;
25            }
        }
    }
}
```

Developer Console - Google Chrome  
orgfarm-fda56f0e83-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

AmountDistributionHandler.apxc \* AmountDistribution.apxt \*

Code Coverage: None API Version: 64

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {
2     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
3         AmountDistributionHandler.amountDist(trigger.new);
4     }
5 }
```

**Create a Report Folder :**

 **Salesforce Garage Management System – Reporting Module**

To organize and store all the related reports of the Garage Management System inside a dedicated report folder named “**Garage Management Folder**”.

### **Step-by-Step Instructions**

#### **Step 1: Open Reports Tab**

1. Click on the **App Launcher** ( nine-dot icon) at the top-left corner of Salesforce.
2. In the search bar, type “**Reports**”.
3. Click on the **Reports** tab from the results.

#### **Step 2: Create a New Report Folder**

1. Once in the Reports section, click on the “**New Folder**” button on the left sidebar.

#### **Step 3: Enter Folder Details**

- In the **Folder Label**, enter:  
**Garage Management Folder**
- The **Folder Unique Name** will be auto-generated.

#### **Step 4: Save the Folder**

- Click the **Save** button.

### **Sharing a Report Folder**

#### **Folder: Garage Management Folder**

#### **Salesforce Garage Management System**

To share the **Garage Management Folder** with the **Manager** role and grant them **View** access.

### **Step-by-Step Instructions**

#### **Step 1: Open the Reports Section**

1. Click on the **App Launcher** ( grid icon) from the top-left corner.
2. Type “**Reports**” in the search bar.
3. Click on the **Reports** tab to open the reports section.

#### **Step 2: Locate the Report Folder**

1. In the left pane, click on “**All Folders**”.

2. Find the folder named “**Garage Management Folder**”.
3. Click the **drop-down arrow (▼)** next to it.
4. Select “**Share**” from the menu.

### **Step 3: Share Folder with Role**

1. In the **Share With** field, select: **Roles**
2. In the **Name** field, search and select: **Manager**
3. In the **Access** field, choose: **View**

### **Step 4: Save Sharing Settings**

1. Click the **Share** button to confirm.
2. Then click **Done** to finish.

## **Documentation: Create a Custom Report Type**

### **Module: Garage Management System (Salesforce SmartInternz)**

To create a **Custom Report Type** using the primary object **Customer Details** and relate it with **Appointment**, **Service Records**, and **Billing Details and Feedback** objects.

#### **Step-by-Step Instructions:**

##### **Step 1: Navigate to Report Types**

1. Click on the **Gear Icon** (Setup) in the top-right corner.
2. In the **Quick Find** box, type “**Report Types**”.
3. Click on **Report Types**.
4. Click **Continue** if prompted.

##### **Step 2: Create New Custom Report Type**

1. Click on the **New Custom Report Type** button.

#### **Fill the following details:**

- **Primary Object:** Customer Details
- **Report Type Label:** Service Information
- **Report Type Name:** (auto-populated)

- **Description:** Keep as same or copy Report Type Label
  - **Store in Category:** Other Reports
  - **Deployment Status:** Deployed
2. Click **Next.**

### **Step 3: Add Related Objects**

#### **Add 1st Related Object:**

1. In the **Object Relationships** section, click the “+” (**Related Object box**).
2. Choose: Appointment

#### **Add 2nd Related Object:**

1. Click on the next “+” symbol next to Appointment.
2. Choose: Service Records

#### **Add 3rd Related Object:**

1. Again, click the “+” symbol next to Service Records.
2. Choose: Billing Details and Feedback

### **Step 4: Save the Report Type**

1. After adding all related objects, click the **Save** button.

#### **Documentation: Create Service Information Report :**

#### **Module: Garage Management System (Salesforce SmartInternz)**

To create a detailed **line chart report** using the custom report type **Service Information**, and visualize service and billing status by customer.

#### **Prerequisite:**

Before proceeding, ensure:

- You have created **10 sample records** in **each object**:
  - Customer Details

The screenshot shows the Garage Management software interface. The top navigation bar includes a cloud icon, the title "Garage Management", and links for "Customer Details", "Appointments", "Service records", "Billing details and feedback", "Reports", and "Dashboards". A search bar and a toolbar with various icons are also present. The main content area displays a "Recently Viewed" list titled "Customer Details" with 11 items, updated a few seconds ago. The list shows customer names from 1 to 10: Mac, Nisha, Kiran, Pooja, Rahul, Priya, Arjun, Harini, Aditya Verma, and Sneha Reddy. Each item has a checkbox and a small edit icon.

	Customer Detail Name
1	Mac
2	Nisha
3	Kiran
4	Pooja
5	Rahul
6	Priya
7	Arjun
8	Harini
9	Aditya Verma
10	Sneha Reddy

## ▪ Appointment

The screenshot shows the Garage Management software interface, similar to the previous one but with a different list. The top navigation bar and search bar are identical. The main content area displays a "Recently Viewed" list titled "Appointments" with 11 items, updated a few seconds ago. The list shows appointment names from 1 to 10: app-011, app-010, app-009, app-008, app-007, app-006, app-005, app-004, app-003, and app-002. Each item has a checkbox and a small edit icon. The first item, "app-011", is highlighted with a blue border.

	Appointment Name
1	app-011
2	app-010
3	app-009
4	app-008
5	app-007
6	app-006
7	app-005
8	app-004
9	app-003
10	app-002

## ▪ Service Records

The screenshot shows a Salesforce application interface for 'Garage Management'. The top navigation bar includes tabs for Customer Details, Appointments, Service records (which is the active tab), Billing details and feedback, Reports, and Dashboards. A search bar and various icons are also present. The main content area is titled 'Service records' and 'Recently Viewed'. It displays a list of 11 items, each with a checkbox and a name: ser-011, ser-010, ser-009, ser-008, ser-007, ser-006, ser-005, ser-004, ser-003, ser-002, and ser-001. A sidebar on the right provides filtering and sorting options.

#### ▪ Billing Details and Feedback

The screenshot shows a Salesforce application interface for 'Garage Management'. The top navigation bar includes tabs for Customer Details, Appointments, Service records, Billing details and feedback (which is the active tab), Reports, and Dashboards. A search bar and various icons are also present. The main content area is titled 'Billing details and feedback' and 'Recently Viewed'. It displays a list of 10 items, each with a checkbox and a name: bill-025, bill-026, bill-024, bill-023, bill-022, bill-021, bill-020, bill-019, bill-018, and bill-017. A sidebar on the right provides filtering and sorting options.

- Fill all fields in each record for complete visibility and testing.

### **Step-by-Step Instructions:**

#### **Step 1: Go to Reports**

1. Navigate to your Salesforce App.
2. Click on the **Reports** tab.
3. Click the **New Report** button.

#### **Step 2: Select the Report Type**

1. In the **report type selector**, choose the **Category**: Other Reports.
2. Search for the custom report type: Service Information.
3. Select **Service Information**, then click **Start Report**.

### **Step 3: Customize Report Columns**

#### **In the Outline Pane:**

- Under **Columns**, remove unnecessary fields (click "x" next to unwanted fields).
- **Add the following columns:**
  - Customer Name
  - Appointment Date
  - Service Status
  - Payment Paid

### **Step 4: Grouping Rows**

#### **Under GROUP ROWS, add:**

- Rating for Service
- Payment Status

This allows the report to be grouped and visualized based on customer satisfaction and payment tracking.

### **Step 5: Add Line Chart**

1. Click on **Add Chart** (top-right in report builder).
2. From the chart types, select **Line Chart**.
3. Ensure the chart plots values based on Payment Paid over Rating for Service.

### **Step 6: Save the Report**

1. Click on the **Save & Run** or **Save** button.
2. In the save dialog:
  - **Report Name:** New Service Information Report
  - **Report Unique Name:** auto-populated

- **Folder:** Select Garage Management Folder (created earlier)
3. Click **Save**.

The screenshot shows a Salesforce report interface. At the top, there's a navigation bar with a cloud icon, a search bar, and various menu options like 'Customer Details', 'Appointments', 'Service records', 'Billing details and feedback', 'Reports', and 'Dashboards'. The main area displays a report titled 'Report: Service information' and 'New Service information Report'. The report has a header row with columns for 'Rating for service' (with values 0, 3, 4, 5), 'Payment Status', 'Customer Detail Name', 'Appointment Date', 'Service Status', and 'Amount'. Below this is a data grid. The first section shows 3 services: Harini (Completed, 7/3/2025, 1,000.00), Pooja (Completed, 7/8/2025, 8,000.00), and a subtotal of 9,000.00. The second section shows 4 services: Sneha Reddy (Started, 7/2/2025, 1,200.00) and Priya (Started, 7/6/2025, 1,200.00), with a subtotal of 2,400.00. There are also 'Subtotal' rows for each section.

	Rating for service	Payment Status	Customer Detail Name	Appointment Date	Service Status	Amount
3 (2)	Pending (1)	Harini	7/3/2025	Completed	1,000.00	
	<b>Subtotal</b>				1,000.00	
	Completed (1)	Pooja	7/8/2025	Completed	8,000.00	
	<b>Subtotal</b>				8,000.00	
	<b>Subtotal</b>				9,000.00	
4 (3)	Pending (2)	Sneha Reddy	7/2/2025	Started	1,200.00	
		Priya	7/6/2025	Started	1,200.00	
	<b>Subtotal</b>				2,400.00	

## Documentation: Create & Share Dashboard Folder :

### Module: Garage Management System – Salesforce SmartInternz

To create a new **Dashboard Folder** named **Service Rating dashboard** and set up proper sharing permissions for role-based access.

#### Step-by-Step Instructions:

##### Step 1: Create Dashboard Folder

1. Click on the **App Launcher** (grid icon in the top-left corner).
2. Search for and select **Dashboards**.
3. In the **Dashboard** tab, click on **New Folder** (top-right corner).
4. In the popup window:
  - **Folder Label:** Service Rating dashboard
  - **Folder Unique Name:** Auto-populated
5. Click **Save**.

## Step 2: Share the Dashboard Folder

Follow the same procedure used for sharing the report folder in Reports Milestone and Activity 2:

1. In the **Dashboard tab**, click on **All Folders** (left sidebar).
2. Find the folder **Service Rating dashboard**.
3. Click on the **Dropdown Arrow (▼)** next to the folder name.
4. Select **Share**.
5. In the **Share With** dropdown, select: Roles.
6. In the **Name** field, search for and select: Manager.
7. Set the **Access Level**: View.
8. Click **Share**, then click **Done**.

The screenshot shows the Salesforce interface for the Garage Management System. The top navigation bar includes links for Customer Details, Appointments, Service records, Billing details and feedback, Reports, and Dashboards. The main content area displays the 'All Folders > Service Rating dashboard' view. A table lists one item: 'Customer review' under the 'Name' column, which is linked to 'Service Rating dashboard' under 'Folder'. Other columns include 'Created By' (Vyshnavi Challa), 'Created On' (7/14/2025, 6:41 AM), and 'Subscribed' (indicated by a checkmark). The left sidebar has sections for DASHBOARDS (Recent, Created by Me, Private Dashboards, All Dashboards) and FOLDERS.

### Documentation: Create and Subscribe to Dashboard :

#### Module: Garage Management System – Salesforce SmartInternz

To create a **Dashboard** using an existing report, configure its visual chart, and subscribe to receive weekly email updates.

#### Step-by-Step Instructions:

##### Step 1: Create Dashboard

1. Click on the **App Launcher** (grid icon in the top-left corner).

2. Search and select **Dashboards**.
3. In the **Dashboard tab**, click on **New Dashboard** (top-right corner).
4. In the popup:
  - **Dashboard Name:** Service Performance Dashboard
  - **Folder:** Select the previously created folder Service Rating dashboard
5. Click **Create**.

### **Step 2: Add Component (Report Chart)**

1. After the dashboard canvas opens, click **+ Add Component**.
2. A list of available reports appears. Select the report named **New Service Information Report**.
3. Click **Select**.
4. In the chart configuration:
  - **Chart Type:** Choose Line Chart
  - **Theme:** Select a theme (e.g., Light, Dark) based on preference.
5. Click **Add**.
6. Click **Save**.
7. Click **Done** to finalize the dashboard.

### **Step 3: Subscribe to the Dashboard**

1. On the Dashboard page, click the **Subscribe** button (top-right corner).
2. In the subscription settings:
  - **Frequency:** Select Weekly
  - **Day:** Choose Monday
3. Click **Save**.

