

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.

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.

- **Appointment Scheduling:**
 - Simplifies the booking process for customers.
 - Enables staff to manage daily schedules efficiently, reducing downtime and improving resource allocation.
- **Vehicle Management:**
 - Maintains detailed records of vehicles, including service history, repairs, and maintenance schedules.
 - Tracks vehicle status during servicing for better communication with customers.
- **Customer Relationship Management (CRM):**
 - Stores customer details and preferences.
 - Sends service reminders, follow-ups, and promotional offers to build loyalty.
- **Inventory and Spare Parts Management:**
 - Tracks spare parts stock levels, automates reorder processes, and prevents stockouts.
 - Ensures that mechanics always have the necessary tools and parts on hand.

- **Billing and Invoicing:**
 - Generates professional invoices quickly and accurately.
 - Supports multiple payment methods, discounts, and tax calculations.
- **Work Order Management:**
 - Creates detailed work orders with a list of tasks, estimated costs, and timelines.
 - Helps staff prioritize jobs and ensures timely completion.
- **Reporting and Analytics:**
 - Provides insights into key performance indicators like revenue, job completion rates, and customer feedback.
 - 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 like this:

<https://youtu.be/r9EX3IGde5k>

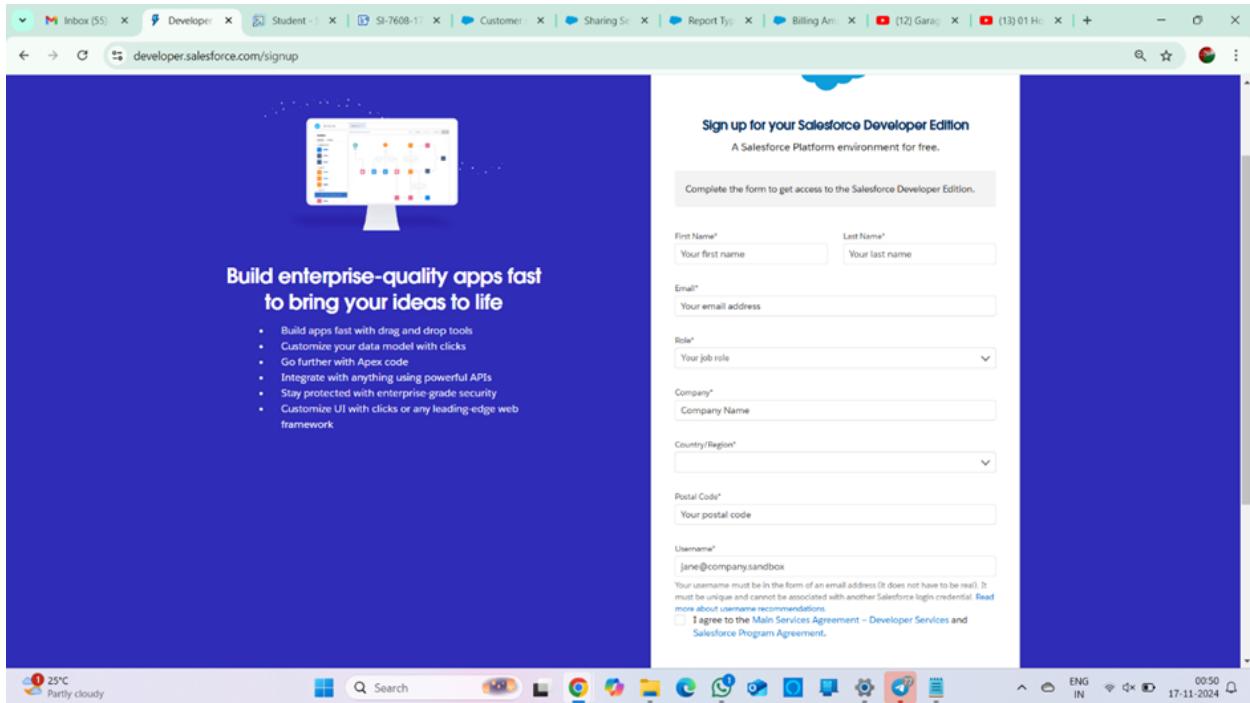
Creating Developer Account:

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

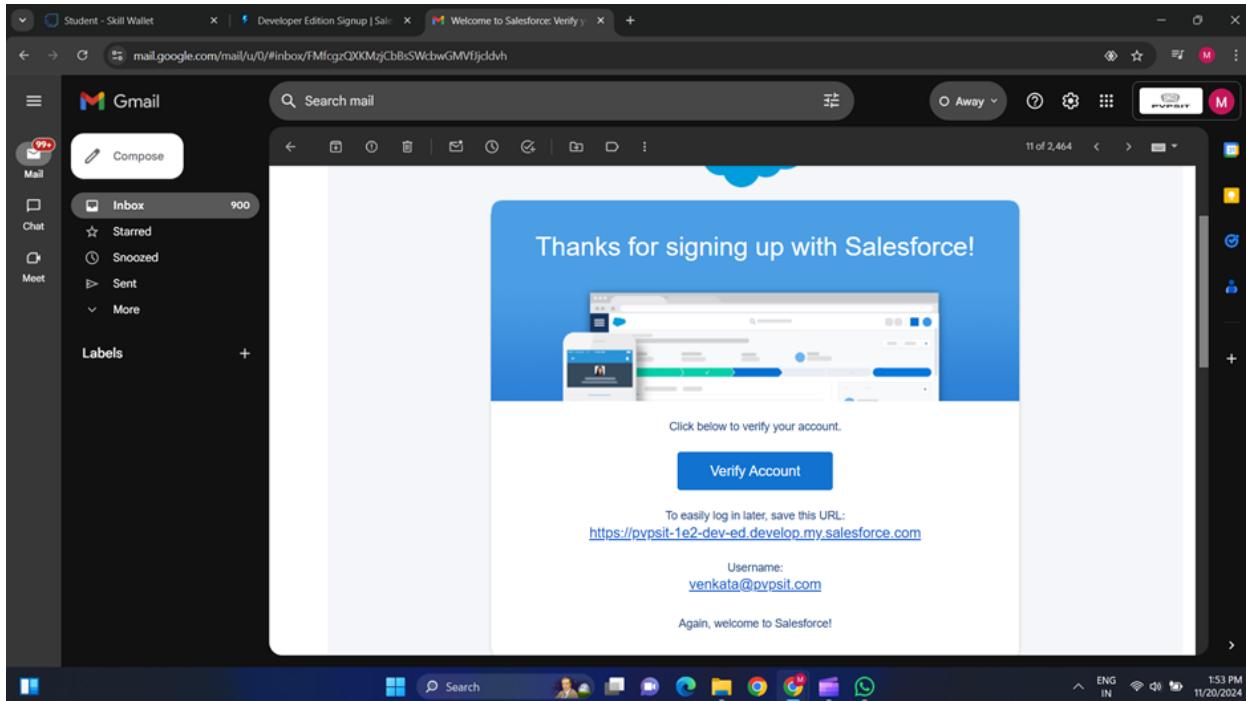
This need not be an actual email id, you can give anything in the format :
username@organization.com

Click on sign me up after filling these.



Account Activation

1. Go to the inbox of the email that you used while signing up. 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 DetailsObject

To create an object:

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

1. Enter the label name >> Customer Details
2. Plural label name >> Customer Details
3. Enter Record Name Label and Format
 - Record Name >> Customer Name
 - Data Type >> Text
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

Create Appointment Object

To create an object:

1. From the setup page >> Click on Object Manager >> 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
 - Record Name >> Appointment Name
 - Data Type >> Auto Number
 - Display Format >> app-{000}
 - Starting number >> 1
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

New Custom Object

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Example: Account
Plural Label: Example: Accounts
Share with owner account:

The Object Name is used when referencing the object via the API.

Object Name: Example: Account
Description:

Context-Sensitive Help Setting: Open the standard Salesforce.com Help & Training window Open a mobile using a Visualforce page

Contact Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key tabs, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Example: Account Name
Data Type: Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.
Display Format: Example: A-(000) Ser-1 Then
Starting Number:

Optional Features

Allow Reports
 Allow Activities
 Track Field History
 Allow Chatter Groups
 Enable Licensing

Object Classification

If these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.

Allow Bulk API Access
 Allow Streaming API Access

Deployment Status

In Development
 Deployed

Search Status

What this setting is enabled, your users can find records of this object type when they search. Learn more.

Allow Search

Object Creation Options (Available only when custom object is first created)

Add Notes and Attachments section last in default page layout
 Launch New Custom Tab Wizard after saving this custom object

Save | Save & New | Cancel

Create Service records Object

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Service records
2. Plural label name >> Service records
3. Enter Record Name Label and Format
 - Record Name >>Service records Name
 - Data Type >> Auto Number
 - Display Format >> ser-{000}
 - Starting number >> 1
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Learn more](#) | [Don't show this message again](#)

Custom Object Definition Edit
Save Save & New Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: <input type="text" value="Service records"/> Example: Account Plural Label: <input type="text" value="Service records"/> Example: Accounts Starts with vowel sound: <input type="checkbox"/>

The Object Name is used when referencing the object via the API.

Object Name: <input type="text" value="Service_records"/> Example: Account
--

Description:

Context-Sensitive Help Setting: Open the standard Salesforce.com Help & Training window Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: <input type="text" value="Service recordsName"/> Example: Account Name

Data Type: Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Display Format: Example: A-{0000} [What Is This?](#)

Starting Number:

Optional Features

- Allow Reports
- Allow Activities
- Track Field History
- Allow in Chatter Groups
- Enable Learning [Learn more](#)

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#).

- Allow Sharing
- Allow Bulk API Access
- Allow Streaming API Access

Deployment Status

- In Development
- Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. [Learn more](#).

- Allow Search

Object Creation Options (Available only when custom object is first created)

- Add Notes and Attachments related list to default page layout
- Launch New Custom Tab Wizard after saving this custom object

Save Save & New Cancel

Create Billing details and feedback Object

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> Billing details and feedback
 2. Plural label name >> Billing details and feedback
 3. Enter Record Name Label and Format
 - Record Name >> Billing details and feedback Name
 - Data Type >> Auto Number
 - Display Format >> bill-{000}
 - Starting number >> 1
 2. Click on Allow reports and Track Field History,
 3. Allow search >> Save.

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Learn more](#) [View object message history](#)

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label	<input type="text" value="Billing details and feedback"/> Example: Account
Home Label	<input type="text" value="Billing details and feedback"/> Example: Accounts
Starts with vowel sound	<input type="checkbox"/>

The Object Name is used when referencing the object via the API.

Object Name	<input type="text" value="Billing_details_and_feedback"/> Example: Account
-------------	--

Description

Context-Sensitive Help Setting: Open the standard Salesforce.com Help & Training window Open a window using a Visualforce page

Content Name:

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	<input type="text" value="Billing details and feedback"/> Example: Account Name
Date Type	<input checked="" type="button" value="Auto Number"/> <small>Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.</small>
Display Format	<input type="text" value="bill-(000)"/> Example: A-(0000) What Is This?
Starting Number	<input type="text" value="1"/>

Optional Features

- Allow Reports
- Allow Activities
- Track Field History
- Allow in Chatter Groups
- Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#).

- Allow Sharing
- Allow Bulk API Access
- Allow Streaming API Access

Deployment Status

- In Development
- Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. [Learn more](#).

- Allow Search

Object Creation Options (Available only when custom object is first created)

- Add Notes and Attachments related list to default page layout
- Launch New Custom Tab Wizard after saving this custom object

Buttons:

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. Web Tabs

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 in Lightning 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 custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to 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.

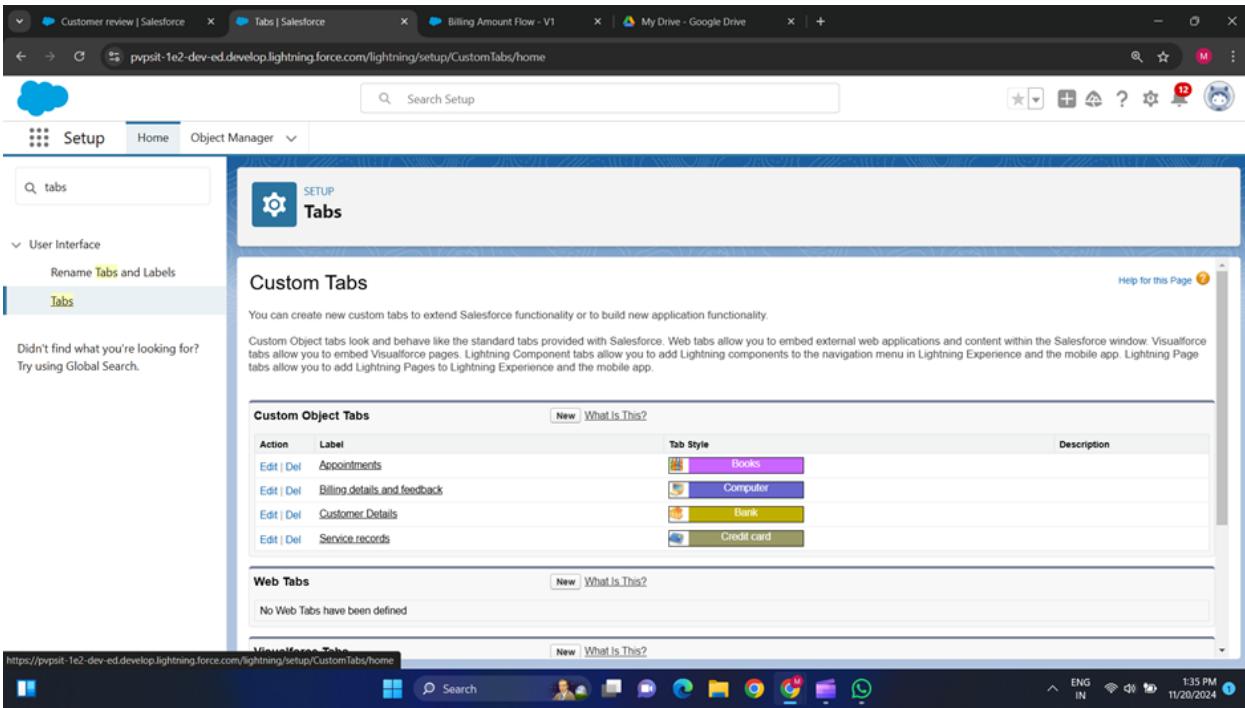
Creating a Custom Tab

To create a Tab:(Customer Details)

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

Creating Remaining Tabs

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



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 >> (App option page) keep it as default >> Next >> (Utility Items) 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 Salesforce App Manager interface. On the left, there's a sidebar with categories like Apps, External Client Apps, and User Interface. A search bar at the top has 'app m' typed into it. The main area is titled 'Lightning Experience App Manager' and displays a table of 24 items. The columns are: App Name, Developer Name, Description, Last Modified, App Type, and Visibility. One row, 'Garage Management Application' by 'Garage_Management_Applicati...', is highlighted with a red box.

App Name	Developer Name	Description	Last Modified	Ap...	Vi...
App Launcher	AppLauncher	App Launcher tabs	19/11/2024, 2:27 pm	Classic	✓
Automation	FlowsApp	Automate business processes and repetitive tasks.	19/11/2024, 2:34 pm	Lightning	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions designed...	19/11/2024, 2:31 pm	Lightning	✓
Business Rules Engine	ExpressionSetConsole	Create and maintain business rules that perform c...	19/11/2024, 2:27 pm	Lightning	✓
Community	Community	Salesforce CRM Communities	19/11/2024, 2:27 pm	Classic	✓
Content	Content	Salesforce CRM Content	19/11/2024, 2:27 pm	Classic	✓
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, a...	19/11/2024, 2:27 pm	Lightning	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	19/11/2024, 2:27 pm	Lightning	✓
Garage Management Application	Garage_Management_Applicati...	View Adoption and Usage Metrics for Lightning Ex...	19/11/2024, 3:46 pm	Lightning	✓
Lightning Usage App	LightningInstrumentation	Track sales and marketing efforts with CRM objects.	19/11/2024, 2:27 pm	Lightning	✓
Marketing CRM Classic	Marketing	Marketing	19/11/2024, 2:27 pm	Classic	✓

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 Customer Details object

1. To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on "Fields & Relationships" >> New
3. Select Data Type as a "Phone"
4. Click on next.
5. Fill the Above as following:

- Field Label: Phone number
- Field Name : gets auto generated
- 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:

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:
5. Field Label : Gmail
6. Field Name : gets auto generated
7. Click on Next >> Next >> Save and new.

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 object name(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 “ Customer Details” and click next.
 5. Next >> Next >> Save.

Note: Make sure you complete Activity 4 Before continuing.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
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		

Creation of Lookup 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 “Look-up relationship” as data type and click Next.
4. Select the related object “ Appointment ” and click next.
5. Make it a required field so click on Required.
6. Scroll down for Lookup Filter 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.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. Under 'Service records', the 'Fields & Relationships' tab is active. A table lists eight fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment_c	Lookup(Appointment)		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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		

Creation of Lookup Field on Billing details and feedback Object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) 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 related object " Service records" and click next.
5. Next >> Next >> Save & new.

The screenshot shows the Salesforce Object Manager Fields & Relationships page. The left sidebar lists various setup options like Details, Page Layouts, Lightning Record Pages, etc. The main area displays a table of fields for the 'Billing details and feedback' object. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

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 Checkbox Fields

Creation of Checkbox Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) 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 : 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 from 1 to 3.
2. Give the Field Label : Repairs
3. Field Name : is auto populated
4. Default value : unchecked
5. Click on next >> next >> save.

6. Follow the same and create another checkbox with 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.

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.

Creation of Currency Fields

Creation of Currency 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 “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 level security for profile.
 8. Click on next >> save.

Creation of Currency Field on Billing details and feedback Object :

1. Follow the same steps as mentioned above in Billing details and feedback Object.
2. Change the label name as mentioned.
3. Give the Field Label : Payment Paid
4. Field Nme : is auto populated

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 number plate
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 details and feedback) in 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 : 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

Creation of Picklist Fields

Creation of Picklist Fields in Service records object :

1. Go to setup >> click on Object Manager >> 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 value separated by a new line" and enter values as shown below.
5. The values are: Started, Completed.
6. Click Next.
7. Next >> Next >> Save.

Creation of Picklist Fields in Billing details and feedback object :

1. Go to setup >> click on Object Manager >> 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 "Payment Status", under values select "Enter values, with each value separated by a new line" and enter values as shown below.
5. The values are: Pending, Completed.
6. Click Next.
7. Next >> Next >> Save.

Creating Formula Field in Service records Object

1. Go to setup >> click on Object Manager >> 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 "Formula" and click Next.
4. Give Field Label and Field Name as "service date" and select formula return type as "Date" and click next.

5. Insert field formula should be : CreatedDate
6. click "Check Syntax" .
7. Click next >> next >> Save.

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 object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " Vehicle ".
4. Insert the Error Condition 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 number plate", and click Save.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. Under the 'Appointment' object, the 'Validation Rules' section is displayed. A single validation rule is listed:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	22505A0509 MAREEDU VENKATA VARUN, 19/11/2024, 4:20 pm

To create a validation rule to an Service records Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Service records object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " service_status_note ".
4. Insert the Error Condition Formula as :-
5.NOT(ISPICKVAL(Service_Status__c , "Completed"))
6. 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.

The screenshot shows the Salesforce Setup page with the 'Object Manager' tab selected. Under 'Service records', the 'Validation Rules' section is displayed. A table lists one rule named 'service_status_note' located on 'Service Status' with the message 'still it is pending'. The table includes columns for Rule Name, Error Location, Error Message, Active status, and Modified By.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
service_status_note	Service Status	still it is pending	✓	22505A0509 MAREEDU VENKATA VARUN, 19/11/2024, 4:22 pm

To create a validation rule to an Billing details and feedback Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating_should_be_less_than_5”.
4. Insert the Error Condition Formula as :-

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

Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”, and click Save.

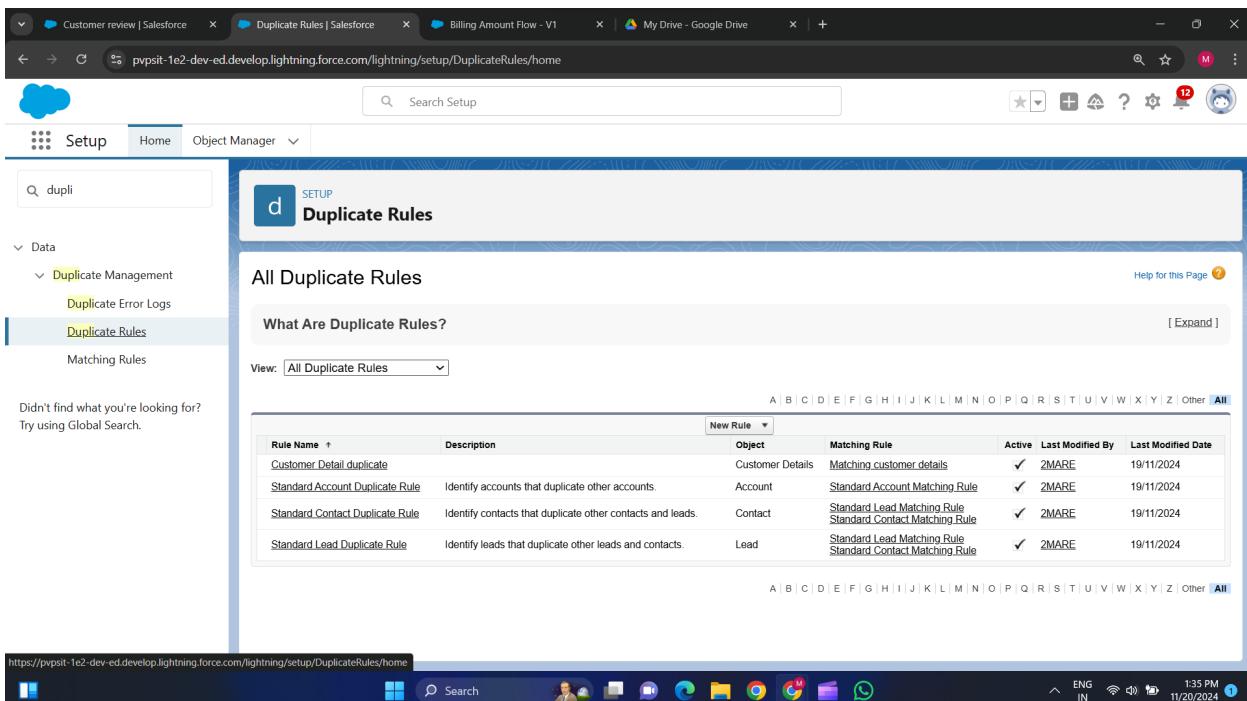
Duplicate rule

To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
 2. Click on matching rule >> click on New Rule.
 3. Select the object as Customer details and click Next.
 4. Give the Rule name : Matching customer details
 5. Unique name : is auto populated
 6. Define the matching criteria as

Field	Matching Method
Gmail	Exact
Phone Number	Exact

 7. Click save.
 8. 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 'Duplicate Rules' page open. The page title is 'Customer Details Duplicate Rule' and the rule name is 'Customer Detail duplicate'. The 'Duplicate Rule Detail' section includes fields for Rule Name (Customer Detail duplicate), 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 (checked), Matching Rule (Matching customer details, Mapped), and Operations On Create (Alert checked, Report checked). The 'Matching Criteria' section contains the formula: (Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE). The 'Conditions' section shows Created By (22505A0509 MARREEDU VENKATA VARUN, 19/11/2024, 4:34 pm) and Modified By (22505A0509 MARREEDU VENKATA VARUN, 19/11/2024, 4:34 pm). The bottom of the page has standard edit, delete, clone, and deactivate buttons.

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

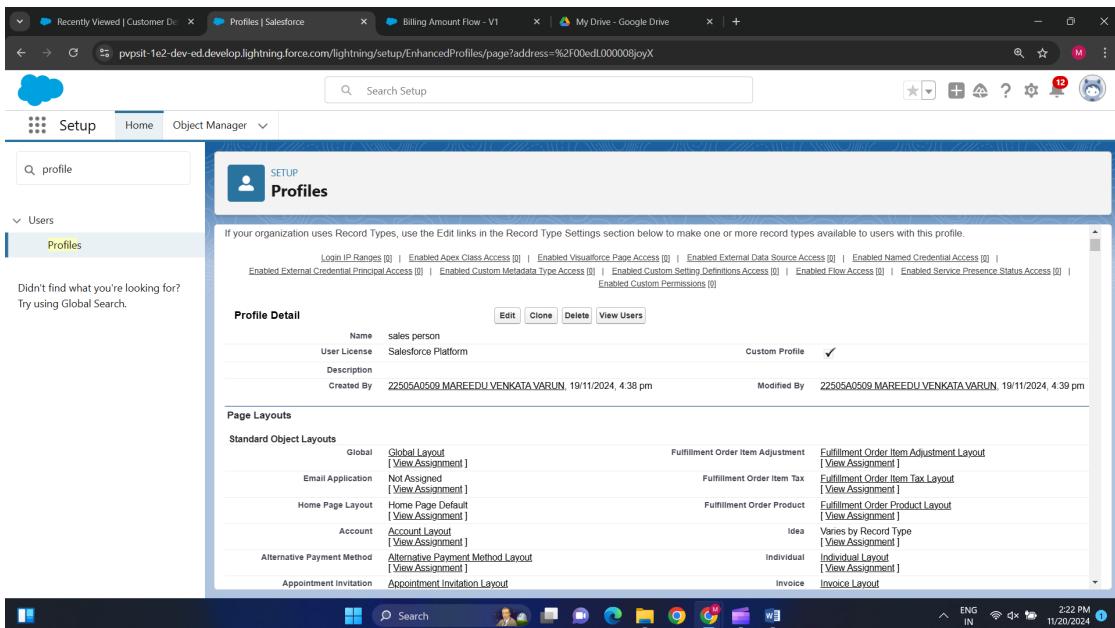
To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> 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. Changing the session times out after should be “ 8 hours of inactivity”.
 6. Change the password policies as mentioned :
 7. User passwords expire in should be “ never expires ”.
 8. Minimum password length should be “ 8 ”, and click save.

Sales person Profile

1. Go to 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

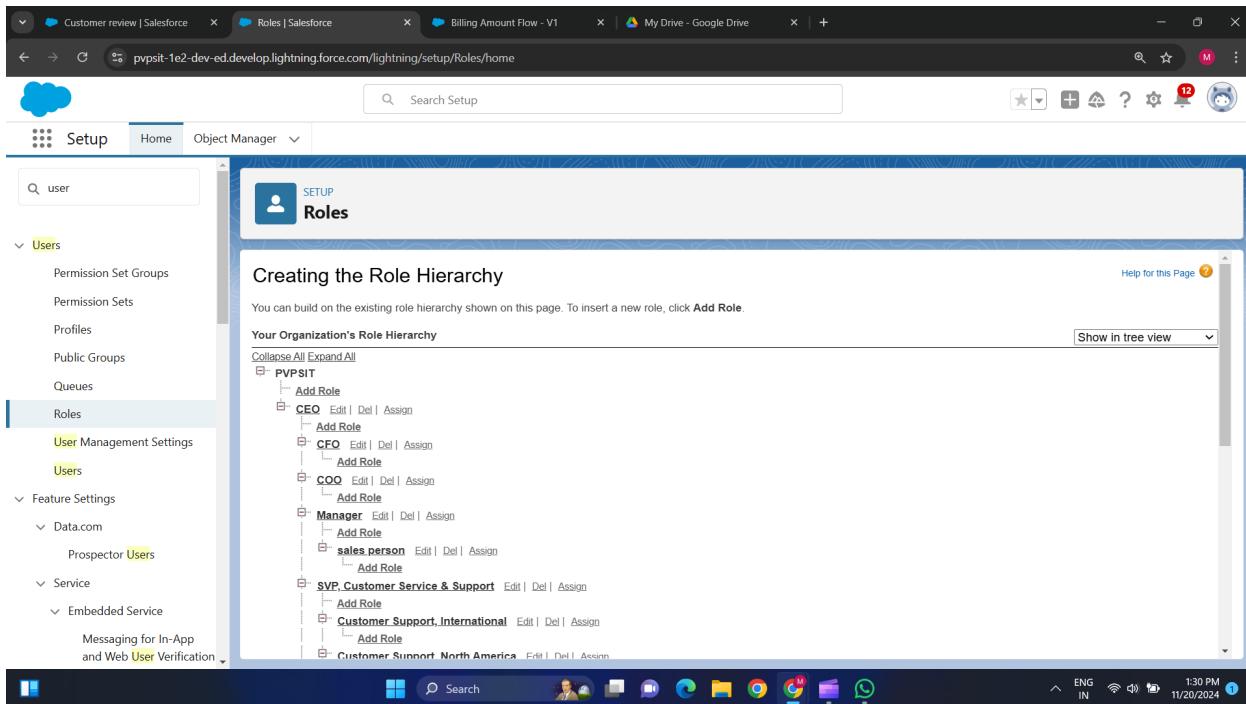
A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating Manager Role

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

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 "sales person" and Role name gets auto populated. Then click on Save.



Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Create User

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
 - a. First Name : Niklaus
 - b. Last Name : Mikaelson
 - c. Alias : Give a Alias Name
 - d. Email id : Give your Personal Email id
 - 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
3. Save.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00dd00000frifuan.dhotwxclwfe5@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/> Edit	kollipara.adithya	akoll	adithya@kollipara.com	sales person	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/> Edit	MAREEDU VENKATA VARUN_22505A0509	2MARE	venkata@npvpsit.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/> Edit	Mikaelson_Niklaus	nmika	niklaus@mikaelso.com	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/> Edit	User_Integration	integ	integration@00dd00000frifuan.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightssecurity@00dd00000frifuan.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User
<input type="checkbox"/> Edit	varun Venkata	yvaru	venkata@varun.com	sales person	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/> Edit	venkata_pavan	pvenk	pavan@venkata.com	sales person	<input checked="" type="checkbox"/>	sales person

creating another users

1. Repeat the steps and create another user using

1. Role : sales person
2. User licence : Salesforce Platform
3. Profile : sales person

Note : create atleast 3 users with these permissions.

The screenshot shows the Salesforce Setup interface under the 'Users' section. On the left, there's a sidebar with various options like Permission Set Groups, Profiles, Public Groups, Queues, Roles, and User Management Settings. The 'Users' option is selected. The main content area displays the details of a user with ID 22505A0509, name MAREEDU VENKATA VARUN, and email 22505a0509@pvpst.ac.in. The 'User Detail' section includes fields for Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, and Delegated Approver. To the right of the detail table, there's a grid of checkboxes indicating user roles and licenses such as Salesforce, System Administrator, Active, Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, WDC User, Mobile Push Registrations, and Data.com User Type. Most checkboxes are checked. At the bottom of the page, there are tabs for Edit, Sharing, Change Password, and View Summary, along with a 'User Profile Help' link. The browser address bar shows the URL for the ManageUsers page.

Public groups

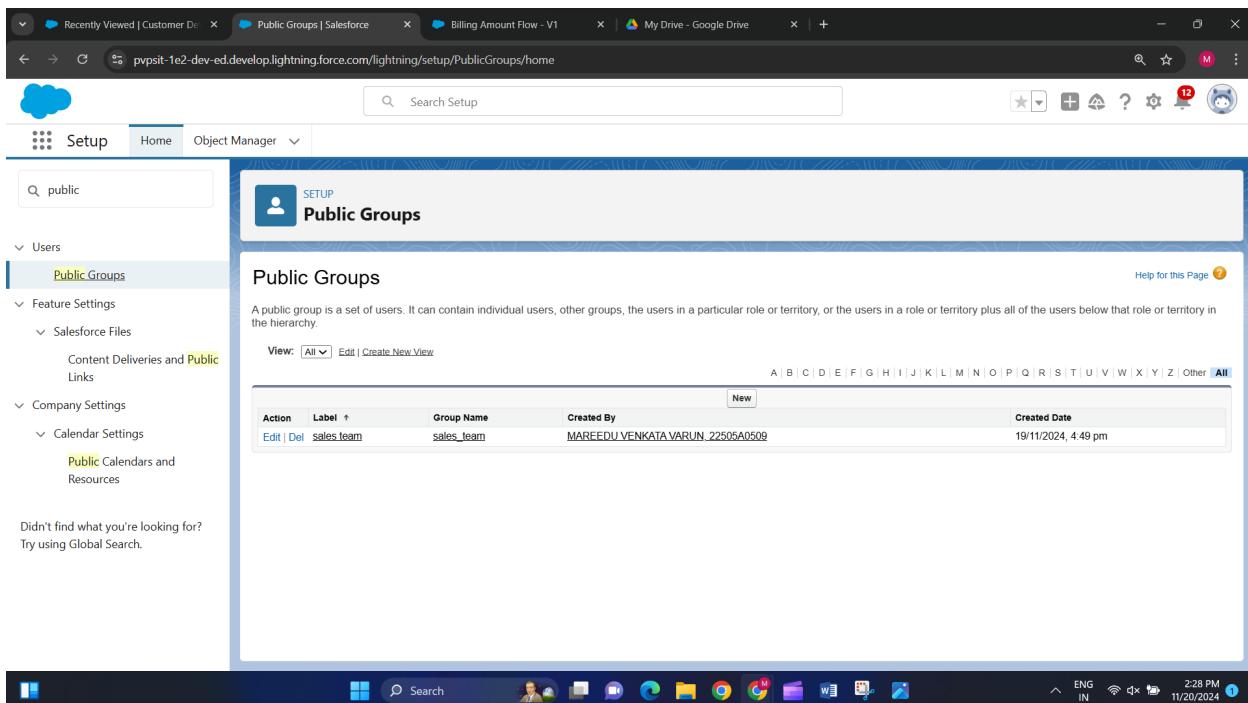
Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings. By creating and using public groups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have appropriate 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 Public Groups page. The left sidebar has a search bar and navigation links for Users, Feature Settings, Company Settings, and Public Calendars and Resources. The main area title is '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 table with one row:

Action	Label	Group Name	Created By	Created Date
Edit Del	sales team	sales_team	MAREEDU VENKATA VARUN_22505A0509	19/11/2024, 4:49 pm

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 Service records Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
5. Give the Label name 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.

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 Optimize 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 form 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.

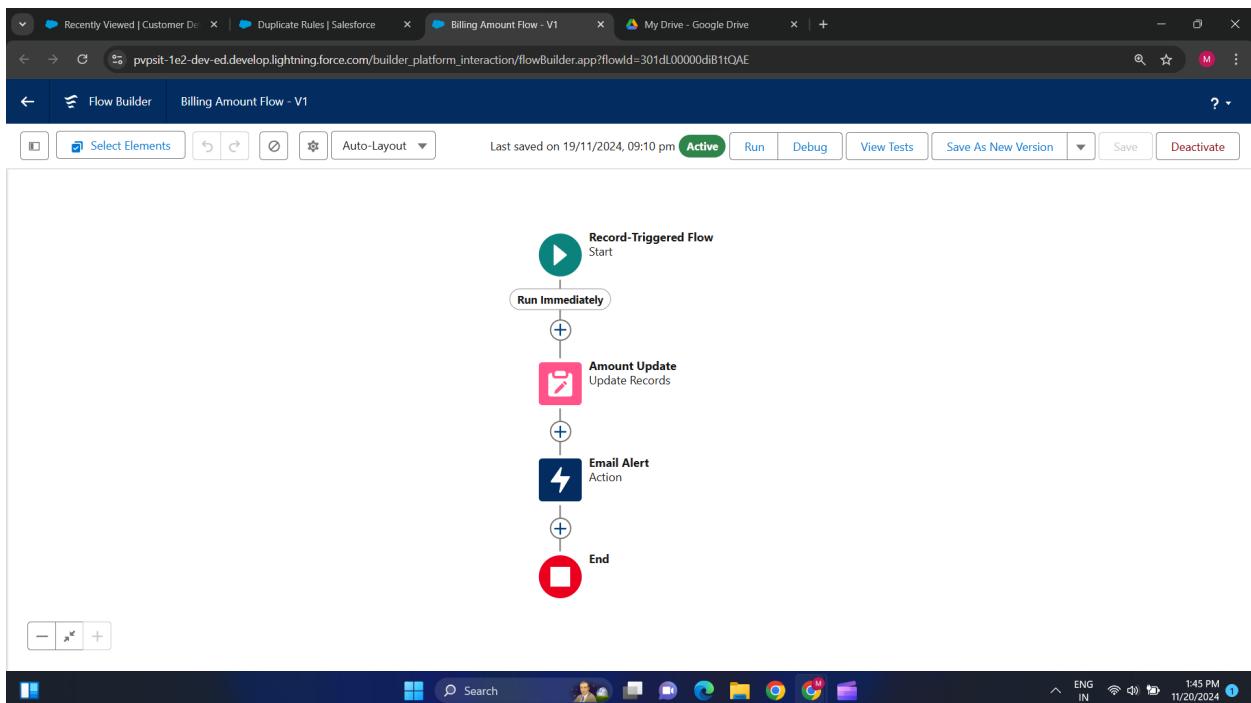
Amount paid : {!\$Record.Payment_Paid__c}

Thank you for Coming .

23. Click done.
24. Now Click on Add Element , select Action.
25. Their action bar will be opened in that search for " send email " and click on it.
26. Give the label name as " Email Alert"
27. API name will be auto populated.
28. Enable the body in set input values for the selected action.
29. Select the text template that created , Body : {!alert}
30. Include recipient address list select the email form the record.
31. RecipientAddressList:
 {!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
32. Include subject as " Thank You for Your Payment - Garage Management".
33. Click done.

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

35. 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:

- insert
- update
- delete
- merge
- upsert
- undelete

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

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

access, go to Triggers.

There are primarily two types of Apex Triggers:

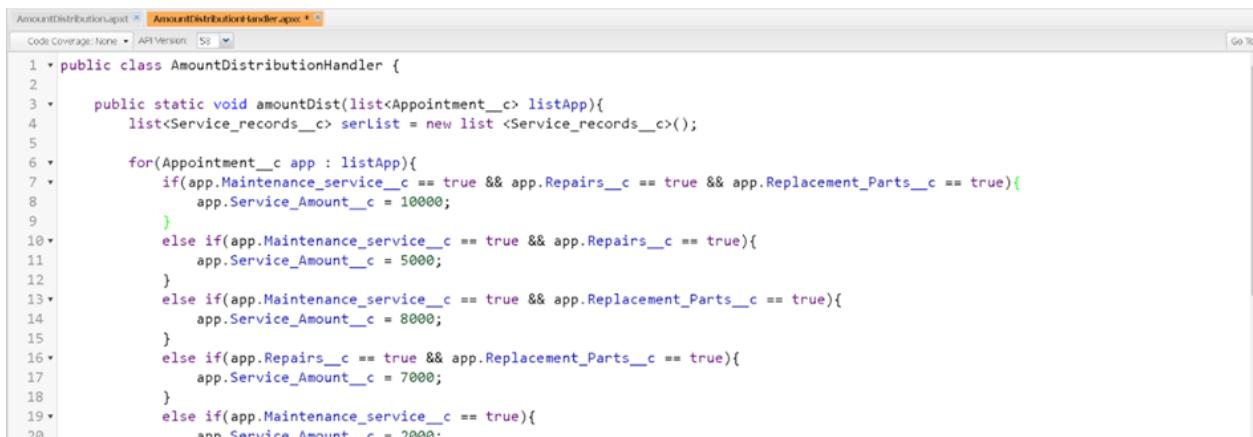
Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready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

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

Apex handler

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

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



```
AmountDistribution.apxc * AmountDistributionHandler.apc * 
Code Coverage: None API Version: 58 Go To
1 public class AmountDistributionHandler {
2
3     public static void amountDist(List<Appointment__c> listApp){
4         List<Service_Records__c> serList = new List <Service_Records__c>();
5
6         for(Appointment__c app : listApp){
7             if(app.Maintenance_Service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
8                 app.Service_Amount__c = 10000;
9             }
10            else if(app.Maintenance_Service__c == true && app.Repairs__c == true){
11                app.Service_Amount__c = 5000;
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        }
23    }
}
```

```
12     }
13     else if(app.Maintenance_service_c == true && app.Repairs_c == true){
14         app.Service_Amount_c = 10000;
15     }
16     else if(app.Maintenance_service_c == true && app.Replacement_Parts_c == true){
17         app.Service_Amount_c = 5000;
18     }
19     else if(app.Repairs_c == true && app.Replacement_Parts_c == true){
20         app.Service_Amount_c = 8000;
21     }
22     else if(app.Maintenance_service_c == true){
23         app.Service_Amount_c = 2000;
24     }
25     else if(app.Repairs_c == true){
26         app.Service_Amount_c = 3000;
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.Repairs__c == true){  
    app.Service_Amount__c = 3000;  
}  
  
else if(app.Replacement_Parts__c == true){  
    app.Service_Amount__c = 5000;  
}  
  
}  
}  
}
```

Trigger Handler :

How to create a new trigger :

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

Syntax For creating trigger :

The syntax for creating trigger is :

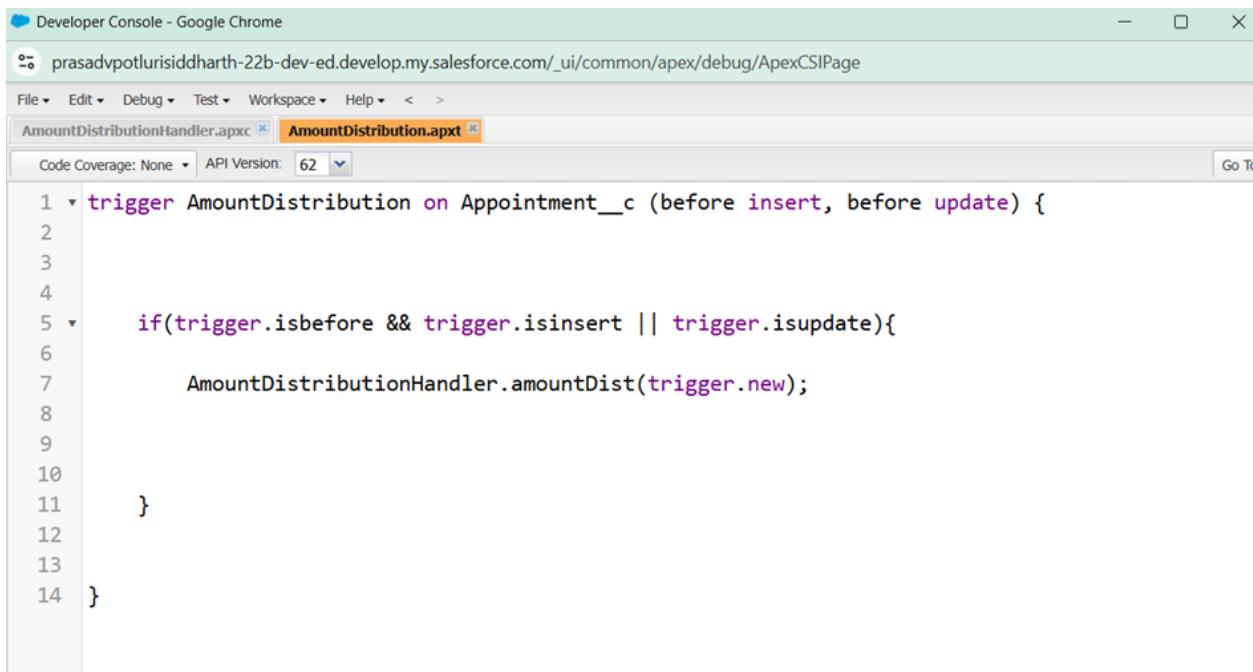
Trigger [trigger name] on [object name](Before/After event)

```
{  
}
```

In this project , trigger is called whenever the particular records 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);  
    }  
}
```



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `prasadvpotlurisiddharth-22b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tabs at the top are "AmountDistributionHandler.apxc" and "AmountDistribution.apxt". The code editor displays the trigger code with line numbers 1 through 14. The code is identical to the one shown above.

```
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.

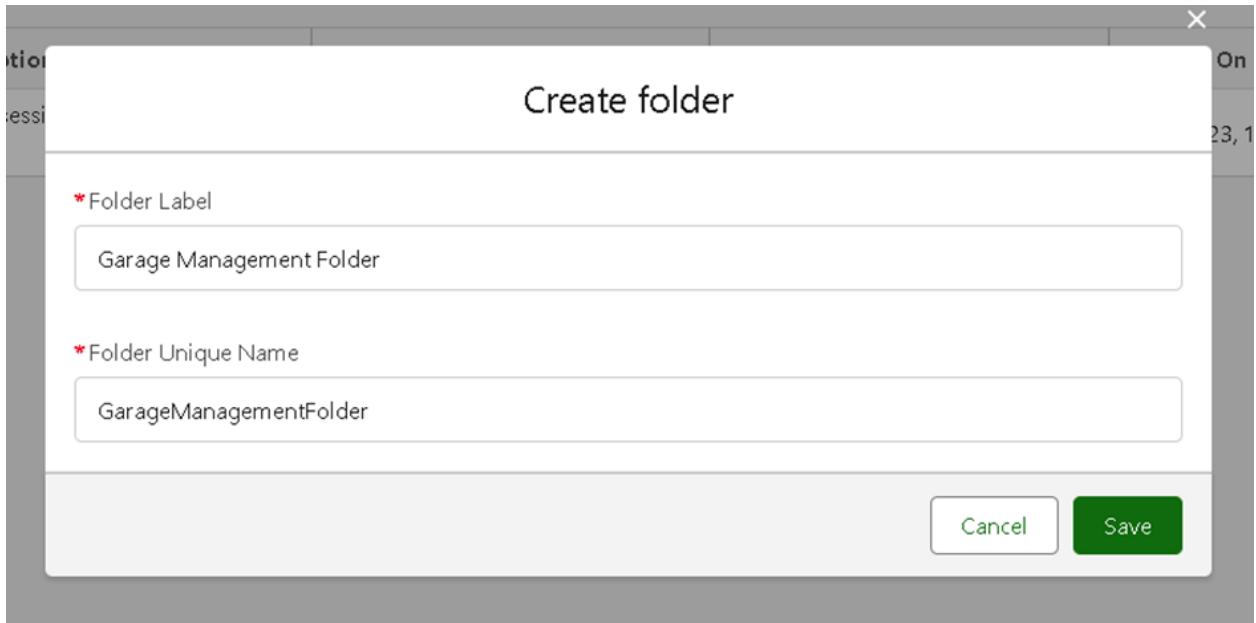
Types of Reports in Salesforce

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 Folder label as "Garage Management Folder", Folder unique 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 quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.
3. Select the Primary object as "Customer details" .
4. Give the Report type Label as "Service information "
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

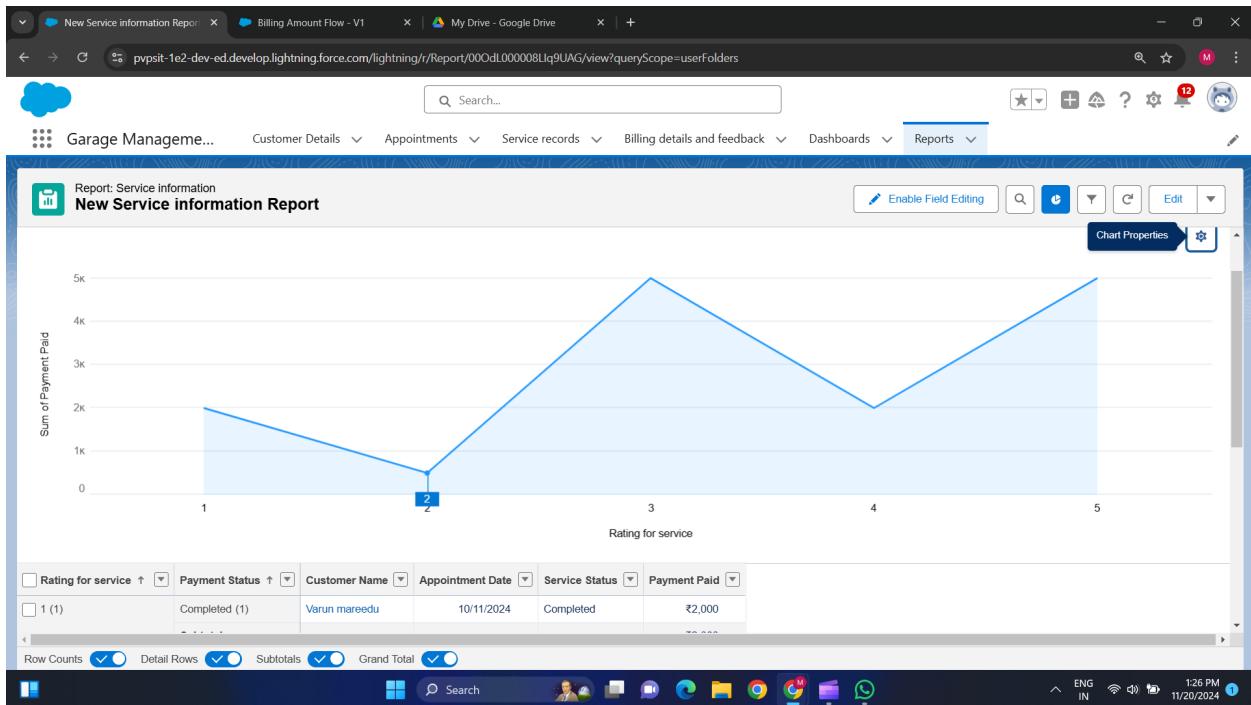
Note : Before creating report, create latest "10" records in every object.

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.
 1. Customer name
 2. Appointment Date
 3. Service Status
 4. Payment paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
 1. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
 1. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report

10. Report unique Name is auto populated.

11. Select the folder the created and Click on save.

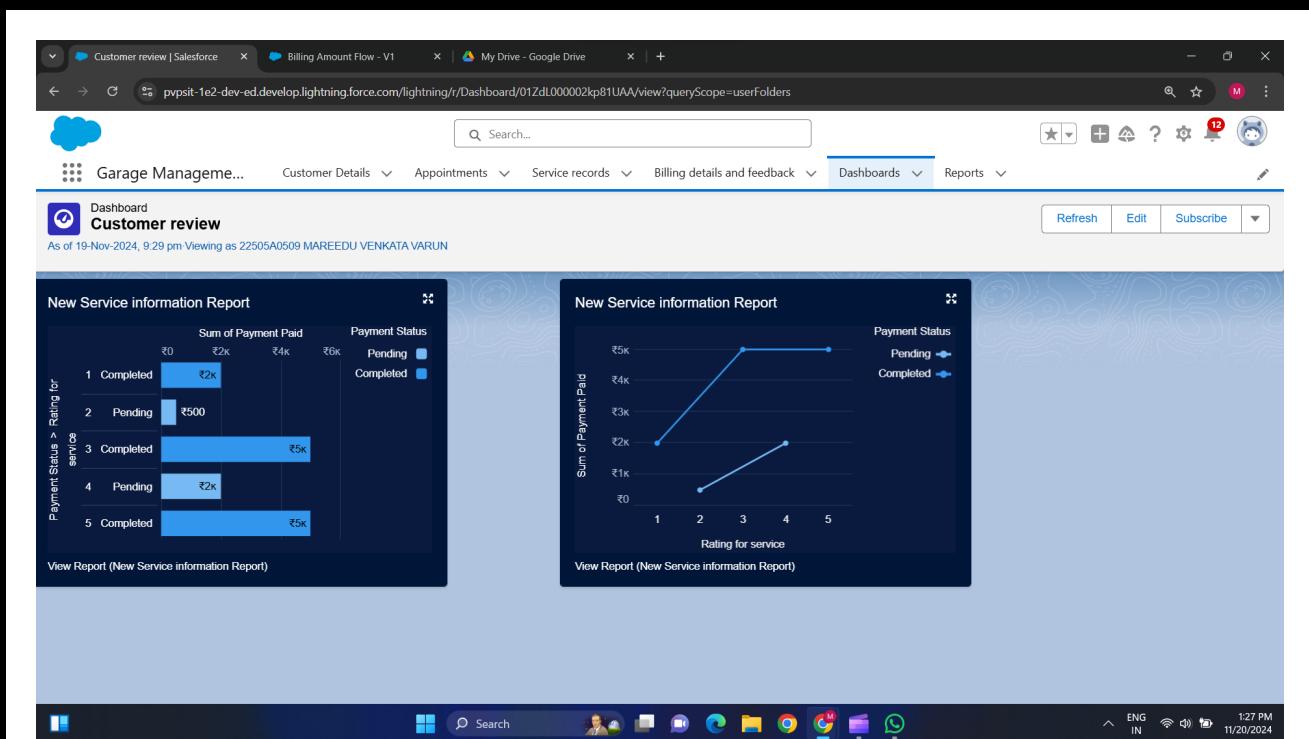


Dashboards

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, form 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.

