

# Title: Garage management system

**College name:** Emeraldheightscollegeforwomen

**College code:** bru12

**Team ID:** 29BF83EF285F960ADCB4DE9C5FB5BF3A

**Team members: 5**

## **Team Leader Name: Anitha.M**

**Email:** arunanitha4499@gmail.com

**Team member:** Sneha.N

**Email:** chinnuammuzz09@gmail.com

**Team member:** shruthika

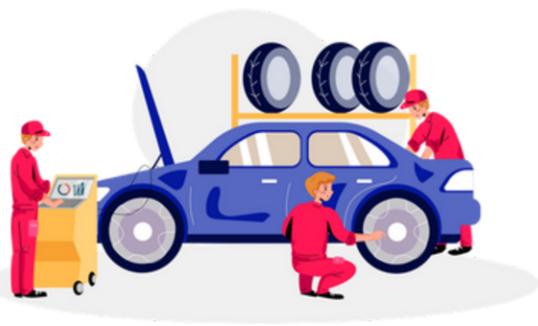
**Email:** sruthishanmugam6@gmail.com

**Team member:** Nivetha

**Email:** mnivetha150@gmail.com

**Team member:** Sowmiya

**Email:** sowmiyaooty123@gmail.com



## **Project overview:**

The GarageManagement System is a web-based application designed to streamline daily garage operations, including vehicle service tracking, customer management, billing, and

inventory management. The system automates key tasks, reduces manual errors, and enhances the customer experience with efficient service handling.

# Objectives:

## 1. Streamline Garage Operations

Automate day-to-day tasks like job scheduling, vehicle check-in/out, and work order management.

## 2. Efficient Customer Management

Maintain customer details, vehicle history, and service preferences for better customer relationships.

## 3. Track Vehicle Service History

Record and retrieve previous repairs, maintenance, and parts replaced to ensure consistent service quality.

## 4. Inventory & Spare Parts Management

Monitor stock levels of spare parts and tools, track usage, and send alerts when items run low.

## 5. Billing & Invoicing Automation

Generate accurate bills and invoices quickly, reducing errors and improving customer trust.

## 6. Improve Time & Resource Management

Assign tasks to mechanics based on availability and skill, reducing idle time and improving productivity.

## 7. Data Analysis & Reporting

Generate reports on revenue, customer trends, and mechanic performance for informed decision-making.

## 8. Enhance Customer Satisfaction

Provide faster service, accurate estimates, and reminders for upcoming maintenance, improving customer loyalty.

## 9. Secure Record Keeping

Keep all operational data safe, reducing paperwork and ensuring easy access to information when needed.

## 10. Scalability for Business Growth

Allow the garage to handle more customers and vehicles as the business expands, without losing efficiency.

### Module1: Salesforce

#### Activity 1: 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:



The image shows the sign-up form for the Salesforce Developer Edition. At the top is the Salesforce logo. Below it, the text "Sign up for your Salesforce Developer Edition" and "A full-featured copy of the Platform, for free". A descriptive message follows: "Complete the form to start your free trial. Our team will be in touch to help you make the most of your trial." The form itself has five input fields: "First Name\*" with placeholder "Your first name", "Last Name\*" with placeholder "Your last name", "Email\*" with placeholder "Your email address", "Role\*" with placeholder "Your job role" and a dropdown arrow, and "Company\*" with placeholder "Optional company name - We'll remember it".

**1:First Name & Last Name**

**2:Email**

**3:Role: Developer**

**4:Company: College Name**

**5:Country: India**

**5:Postal Code: PIN Code**

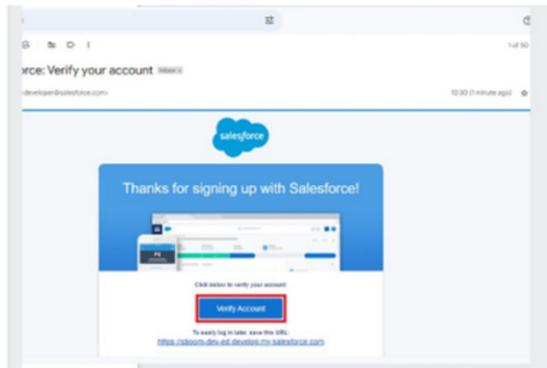
**Username: Should be a combination of your name and company.**

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

Click Sign Me Up after filling these.

## Activity 2: Account Activation

Go to the inbox of the email that you used while signing up. Click on the Verify Account to activate your account. The email may take 5–10 minutes.

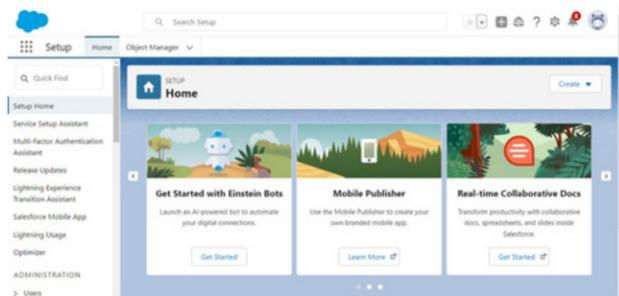


Click on Verify Account

Give a password and answer a security question and click on change password.

A screenshot of the "Change Your Password" page in the Salesforce Setup interface. The page title is "Change Your Password". It asks for a new password that includes at least 8 characters, 1 letter, and 1 number. Two input fields are shown: "New Password" and "Confirm New Password", both with red boxes around them. Below these are "Security Question" and "Answer" fields, also with red boxes around them. A large red box surrounds the entire form area. At the bottom is a blue "Change Password" button.

Then you will redirect to your salesforce setup page.



## Module2: object

### Activity 1: Create Customer Details Object

To create an object:

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

## Activity 2:Create Appointment Object

To create an object:

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

## Activity 3:Create Service records Object

To create an object:

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

## Activity 4: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.
2. Enter the label name >> Billing details and feedback
3. Plural label name >> Billing details and feedback
4. Enter Record Name Label and Format  
Record Name >> Billing details and feedback Name  
Data Type >> Auto Number  
Display Format >> bill-{000}  
Starting number >> 1
5. Click on Allow reports and Track Field History,
6. Allow search >> Save.

## Module3: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.

#### 1. 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.

#### 1. 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.

#### 1. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

#### 1. Lightning Page Tabs

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

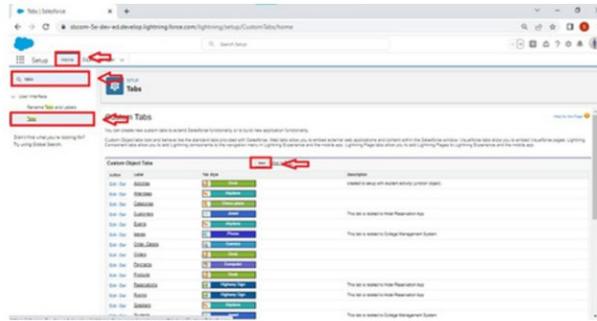
Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show

up in the Available Tabs list when you customise the tabs for your apps.

# Activity 1: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)



Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

Make sure that the Append tab to users' existing personal customizations is checked.

Click save





## Activity 2: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 .

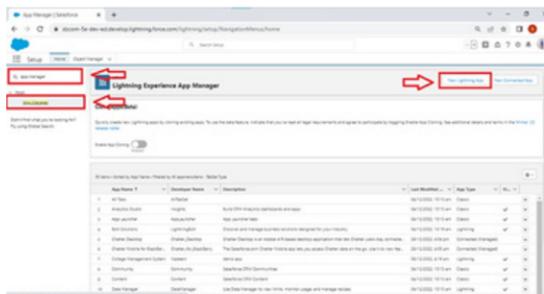
## Module4: The Lightning App

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.

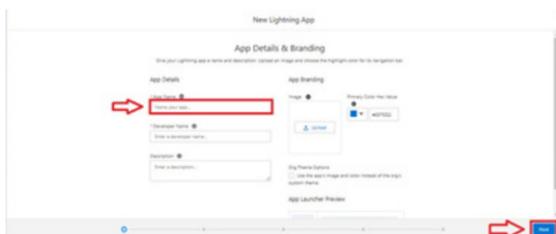
## Activity 1:Create a Lightning App

To create a lightning app page:

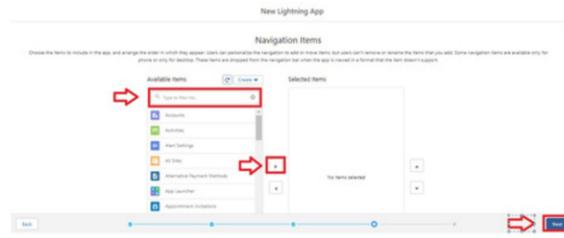
Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.



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

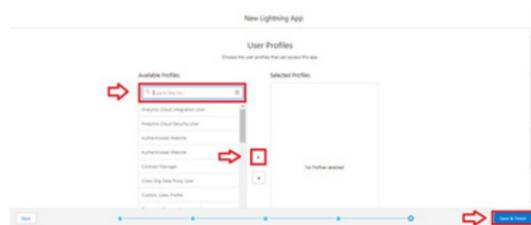


To Add Navigation Items:



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.

To Add User Profiles:



Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

## Module5: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

#### Standard Fields

#### Custom Fields

#### Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

Created By

Owner

Last Modified

## Field Made During object Creation

### Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organiser or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

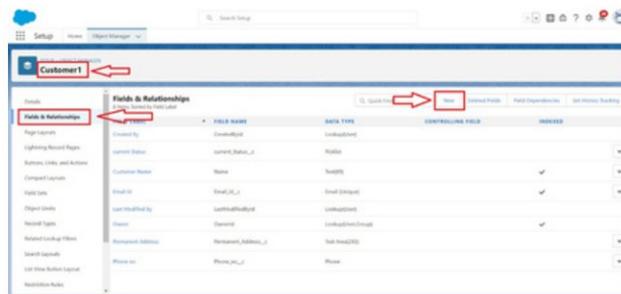
## Activity 1:Creation of fields for the Customer Details object

### 1. To create fields in an object:

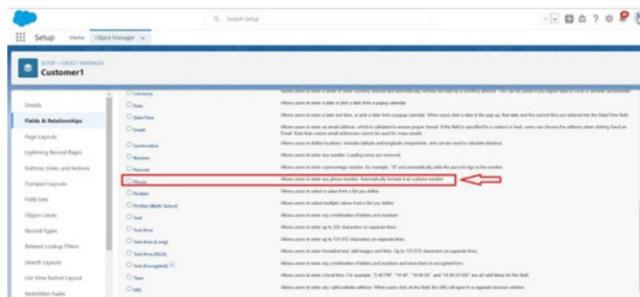
Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object



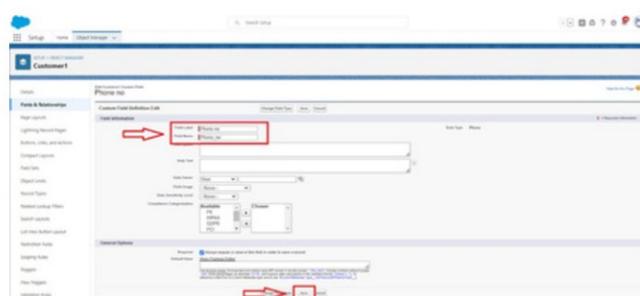
Now click on “Fields & Relationships” >> New



Select Data Type as a “Phone”



Click on next.



Creation of fields for the Customer Details object 1. To create fields in an object: Go to setup >>

click on Object Manager >> type object name(Customer Details) in search bar >>  
click on the object.

Now click on “Fields & Relationships” >> New

Select Data Type as a “Phone”

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:

Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >>  
click on the object.

Now click on “Fields & Relationships” >> New

Select Data type as a “Email” and Click on Next

Fill the Above as following:

Field Label : Gmail

Field Name : gets auto generated

Click on Next >> Next >> Save and new.

## Activity 2: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.



The screenshot shows the Salesforce Object Manager interface. At the top, there are tabs for Setup, Home, and Object Manager. The search bar contains the text 'Appointment'. On the right side, there are buttons for 'SchemaBuilder' and 'Create'. The main area displays a table with the following data:

Label	API Name	Type	Description	Last Modified	Deployed
Appointment	Appointment_A	Custom Object		25/08/2013	
Appointment Category	AppointmentCategory	Standard Object			
Appointment Location	AppointmentLocation	Standard Object			
Appointment Status	AppointmentStatus	Standard Object			

Now click on “Fields & Relationships” >> New

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment__Date__c	Date		
Appointment Name	Name	Auto Number		✓

Select “Look-up relationship” as data type and click Next.

Specify the type of information that the custom field will contain.

**Data Type**

- None Selected Select one of the data types below.
- Auto Number A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.
- Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
- Roll-Up Summary A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.
- Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.
- Master-Detail Relationship Creates a special type of parent-child relationship between this object (the child, or “detail”) and another object (the parent, or “master”) where:
  - The relationship field is required on all detail records.
  - The ownership and sharing of a detail record are determined by the master record.

Select the related object “Customer Details” and click next.

Next >> Next >> Save.

### Creation of Lookup Fields

#### Creation of Lookup Field on Appointment Object :

Go to setup >> click on Object Manager >> type object name( Appointment ) in the search bar >> click on the object.

Now click on “Fields & Relationships” >> New

Select “Look-up relationship” as data type and click Next.

Select the related object “Customer Details” and click next.

Next >> Next >> Save.

Note: Make sure you complete Activity 4 Before continuing.

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

Go to setup >> click on Object Manager >> type object name( Service records ) in search bar >> click on the object.

Now click on “Fields & Relationships” >> New

Select “Look-up relationship” as data type and click Next.

Select the related object “Appointment” and click next.

Make it a required field so click on Required.

**Lookup Options**

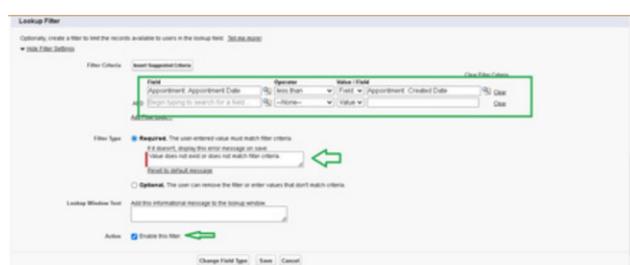
Related To	Appointment	Child Relationship Name	Service_records
Related List Label	Service records		
Required	<input checked="" type="checkbox"/> Always require a value in this field in order to save a record		
What to do if the lookup record is deleted?	<input type="radio"/> Clear the value of this field. You can't choose this option if you make this field required. <input checked="" type="radio"/> Don't allow deletion of the lookup record that's part of a lookup relationship.		

Scroll down for Lookup Filter and click on Show filter settings.

Now add the filter criteria.

Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date

Filter type should be Required.



Error Message : Value does not match the criteria.

Enable the filter by click on Active.

Next >> Next >> Save

#### 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.

## Activity 3:Creation of Checkbox Fields

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

Go to setup >> click on Object Manager >> type object name( Appointment ) in search bar >> click on the object.

Now click on “Fields & Relationships” >> New.

Select “Check box” as data type and click

Next.

Give the Field Label : Maintenance service

Field Name : is auto populated

Default value : unchecked

Click on next >> next >> save

## 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.

1. Give the Field Label : Maintenance service
2. Field Name : is auto populated
3. Default value : unchecked

Step 2. Enter the details Step 2 of 4

Field Label  

Default Value  Checked  Unchecked 

Field Name  

Description

Help Text

Auto add to custom report type  Add this field to existing custom report types that contain this entry 

Previous Next Cancel 

1. Click on next >> next >> save.

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

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

## **Activity 4: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 :

Go to setup >> click on Object Manager >> type object name( Appointment ) in the search bar >> click on the object.

Now click on “Fields & Relationships” >> New.

Select “Currency” as data type and click Next.

Give the Field Label : Service Amount

Field Nme : is auto populated

Click on next

Give read only for all the profiles in field level security for profile

Click on next > > save

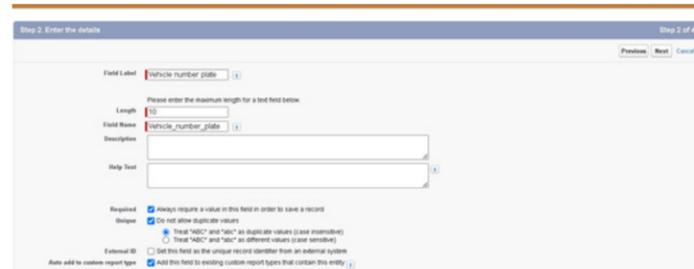
### Creation of Currency Field on Billing details and feedback Object :

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

## Activity 6: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



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.
8. Click on next >> next >> save

## **Activity 7: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



Click Next

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.

## Activity 8: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. Click on fields & relationship >> click on New. Select Data type as "Formula" and click Next.
2. Give Field Label and Field Name as "service date" and select formula return type as "Date" and click next
- 3.
- 4.



Insert field formula should be : CreatedDate

click "Check Syntax"

Click next >> next >> Save

## Module6: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

## Activity 1:To create a validation rule to an Appointment Object

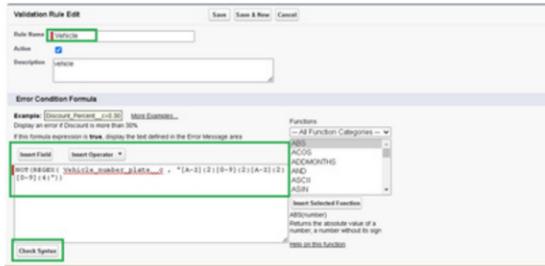
1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.



Enter the Rule name as “ Vehicle ”.

Insert the Error Condition Formula as : -

NOT(REGEX( Vehicle\_number\_plate\_\_c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{0}"))



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.

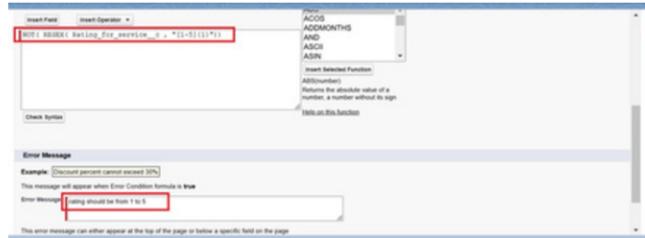


## Activity 2:To create a validation rule to an Billing details and feedback Object

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}"))



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.



## Module7: Duplicate rule

### Activity 1: 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

This screenshot shows the Matching Rules page in the Salesforce setup. It displays a list of existing matching rules. A green arrow points to the 'Matching Rules' link in the sidebar, and another green arrow points to the 'New Rule' button at the top of the list table.

Select the object as Customer details and click Next.

This screenshot shows the first step of creating a new matching rule, titled 'Step 1: Select object'. It asks to select the object for the rule. A green arrow points to the 'Object' dropdown menu, which has 'Customer Details' selected. The page also includes 'Next' and 'Cancel' buttons.

### 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.

This screenshot shows the Matching Rules page in the Salesforce setup, identical to the previous one but with a different URL. It displays a list of existing matching rules. A green arrow points to the 'Matching Rules' link in the sidebar, and another green arrow points to the 'New Rule' button at the top of the list table.

- Select the object as Customer details and click Next.

Matching Rule  
New Matching Rule

Step 1 of 2

Select the object to which this matching rule applies.

Object: Customer Details

Next Cancel

Help for this Page

- Give the Rule name : Matching customer details
- Unique name : is auto populated
- Define the matching criteria as
- Field Matching Method

1. Gmail Exact

2. Phone Number Exact

- Click save.

- After Saving Click on Activate

Rule Details

Object: Customer Details

Rule Name: matching\_Customer\_details

Unique Name: matching\_Customer\_det

Description:

Matching Criteria

Field: Email, Matching Method: Exact, Match Both Fields: AND

Field: Phone Number, Matching Method: Exact, Match Both Fields: AND

Field: Name, Matching Method: Exact, Match Both Fields: AND

Field: City, Matching Method: Exact, Match Both Fields: AND

Field: State, Matching Method: Exact, Match Both Fields: AND

Field: Zip, Matching Method: Exact, Match Both Fields: AND

Save Cancel

Matching Rule Detail

Object: Customer Details

Rule Name: matching\_Customer\_details

Unique Name: matching\_Customer\_det

Description: Excludes records where Email field is null OR Email field is null AND Phone Number field is null

Matching Status: Active

Created By: 000012 20090203 10:15 AM

Modified By: 000012 20090203 10:15 AM

Help for this Page

## Activity 2: To create a Duplicate rule to an Customer details Object

- Go to quick find box in setup and search for Duplicate rules.
- Click on Duplicate rule >> click on New Rule >> select customer details object.

All Duplicate Rules

What Are Duplicate Rules?

View: 24 Duplicate Rules

Rule Name	Description	Matching Rule	Action	Last Modified By	Last Modified Date
Customer Details Rule	Identify records that duplicate other accounts	Account	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify contacts that duplicate other contacts and leads	Lead	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify leads that duplicate other leads and contacts	Lead	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify accounts that duplicate other accounts	Account	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify contacts that duplicate other contacts and leads	Lead	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify leads that duplicate other leads and contacts	Lead	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify accounts that duplicate other accounts	Account	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify contacts that duplicate other contacts and leads	Lead	<input checked="" type="checkbox"/>	000012	20090203
Customer Details Rule	Identify leads that duplicate other leads and contacts	Lead	<input checked="" type="checkbox"/>	000012	20090203

1. Give the Rule name as : Customer Detail duplicate
2. Scroll a little in Matching rule section
3. Select the matching rule : Matching customer details
4. And Click on save.
5. After saving the Duplicate Rule, Click on Activate.

## 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.

## Module8:Types of profiles in salesforce

### 1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

### 1. Custom Profiles:

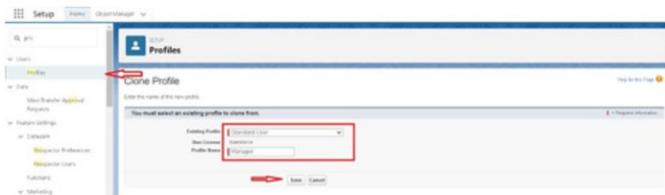
Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

## Activity 1:Manager Profile

To create a new profile:

Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.



While still on the profile page, then click Edit.



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



1. 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.

Custom Object Permissions									
	Basic Access				Data Administration				
	Read	Create	Edit	Delete	View All	Modify All			
Appointments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Billing details and feedback	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Customer Details	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Environments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Laptops	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Service records	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
SessionData	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

1. Changing the session times out after should be “ 8 hours of inactivity”.
2. Change the password policies as mentioned :
3. User passwords expire in should be “ never expires ”.
4. Minimum password length should be “ 8 ”, and click save.

## Activity 2: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.

	Basic Access						Data Administration					
	Read	Create	Edit	Delete	View All	Modify All	Read	Create	Edit	Delete	View All	Modify All
Appointments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Billing details and feedback	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer Details	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Environments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Laptops	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service records	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
SessionData	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

And click save

## Module9: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.

### Activity 1:Creating Manager Role

Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.

The screenshot shows the Salesforce Setup interface. In the left sidebar, under 'Object Manager', 'Roles' is selected and highlighted with a red box. In the main content area, the title 'Understanding Roles' is displayed. Below it, there is a sample role hierarchy diagram. At the top of the hierarchy is 'Executive Staff' (CEO, President, CFO, VP, Sales). Arrows point down to three regional directors: 'Western Sales Director', 'Eastern Sales Director', and 'International Sales Director'. Arrows from these directors point to their respective sales reps: 'Western Sales Rep', 'CA Sales Rep', 'OR Sales Rep' under Western; 'NY Sales Rep', 'MA Sales Rep' under Eastern; and 'Asian Sales Rep', 'European Sales Rep' under International. A note next to the hierarchy says: 'View & edit data, roll-up forecasts, & generate reports for all users below or at same level. Can't access data of users above or at same level.' At the bottom right of the main area, the 'Set Up Roles' button is highlighted with a red box. There is also a checkbox for 'Don't show this page again'.

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

The screenshot shows the 'Your Organization's Role Hierarchy' page. At the top, there is a 'Collapse All' and 'Expand All' button, with 'Expand All' highlighted with a red box. Below this, the 'Nick Enterprises' organization is listed. Under 'Nick Enterprises', several roles are listed with 'Edit | Del | Assign' buttons: 'CFO', 'HR', 'Manager', 'On Site Emp', 'Remote Emp', and 'Sales Rep'. Under each of these, there is another 'Add Role' button, which is also highlighted with a red box. The 'Manager' role has its 'Add Role' button specifically highlighted.

1. Give Label as “Manager” and Role name gets auto populated. Then click on Save.

Role Edit

Label: Manger

Role Name: Manger

This role reports to: CEO

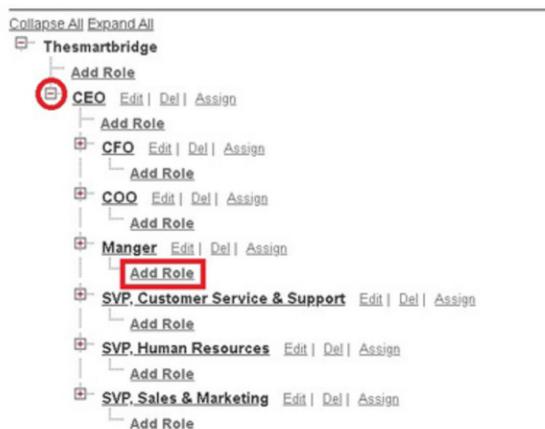
Role Name as displayed on reports:

**Save** **Save & New** **Cancel**

## Activity 2:roles

Creating another two roles under manager

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.



1. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

## Module10: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.

## Activity 1:Create User

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
3. First Name : Niklaus
4. Last Name : Mikaelson
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Manager

10. User licence : Salesforce

11. Profiles : Manager

Save

## Activity 2: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.

## Module11: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.

## Activity 1:Creating New Public Group

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

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

## Module12: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.

### Sharing Rules:

Sharing rules are used to extend access to records for users who meet specific criteria.

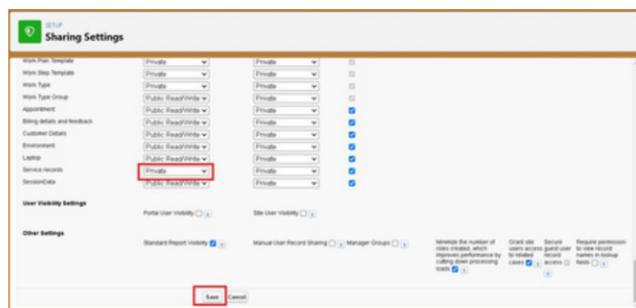
They can be used to grant read-only or read-write access to records owned by other users.

### Manual Sharing:

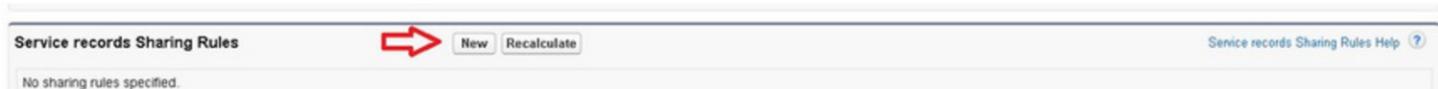
Administrators and record owners can manually share specific records with other users or groups.

## Activity 1: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.

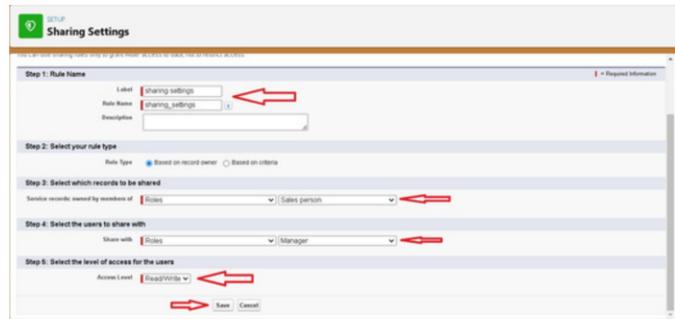


1. Click on save and refresh.
2. Scroll down a bit, Click new on Service records sharing Rules.



1. Give the Label name as " Sharing setting"
2. Rule name is auto populated.
3. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"

4. In step 4: share with, select “ Roles ” >> “ Manager ”
5. In step 5 : Change the access level to “ Read / write ”.
6. Click on save.

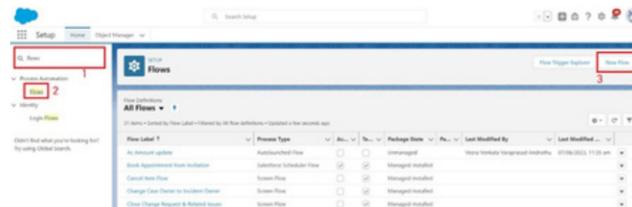


## Module13:Flows

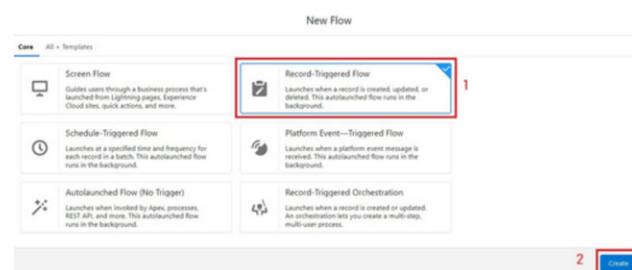
In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

## Activity 1:Create a Flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

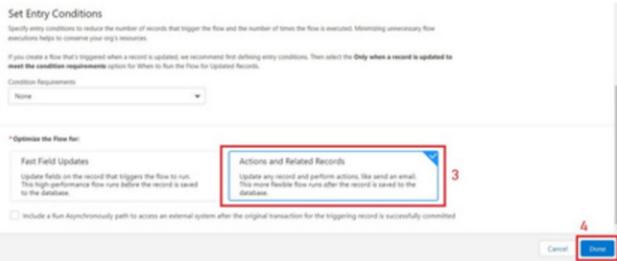


1. Select the Record-triggered flow and Click on Create.

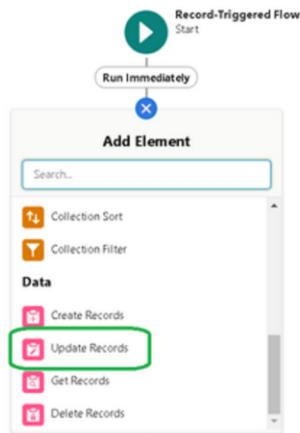


1. Select the Object as “Billing details and feedback”in the Drop down list.
2. Select the Trigger Flow when: “A record is Created or Updated”.
3. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.





- Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.

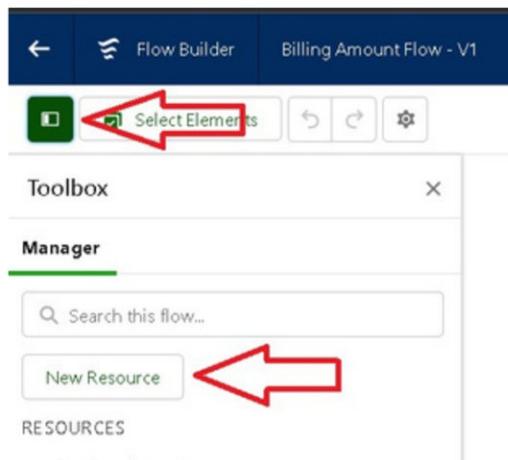


- Give the Label Name : Amount Update
- Api name : is auto populated

- Set a filter condition : All Conditions are met(AND) Field :
- Payment\_Status\_\_c Operator : Equals Value : Completed And
- Set Field Values for the Billing details and feedback Record
- 
-

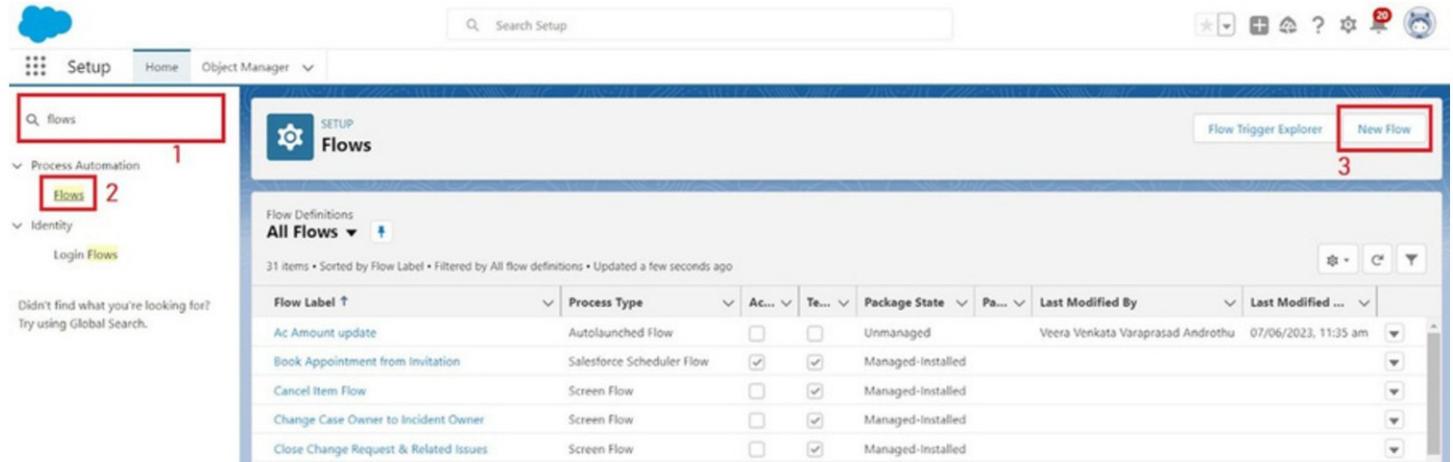
6. Field : Payment\_Paid\_\_c
7. Value : {!\$Record.Service\_records\_\_r.Appointment\_\_r.Service\_Amount\_\_c}
8. Click On Done

Before creating another Element. Create a New Resource from Toolbox form top left.



## Create a Flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.



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

## New Flow

Core All + Templates

1

Screen Flow Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.	Record-Triggered Flow Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
Schedule-Triggered Flow Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.	Platform Event—Triggered Flow Launches when a platform event message is received. This autolaunched flow runs in the background.
Autolaunched Flow (No Trigger) Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.	Record-Triggered Orchestration Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

2

Create

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

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

\*Object  
Billing details and feedback

Configure Trigger

\*Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

Done

## Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

None

## \*Optimize the Flow for:

### Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved to the database.

### Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.

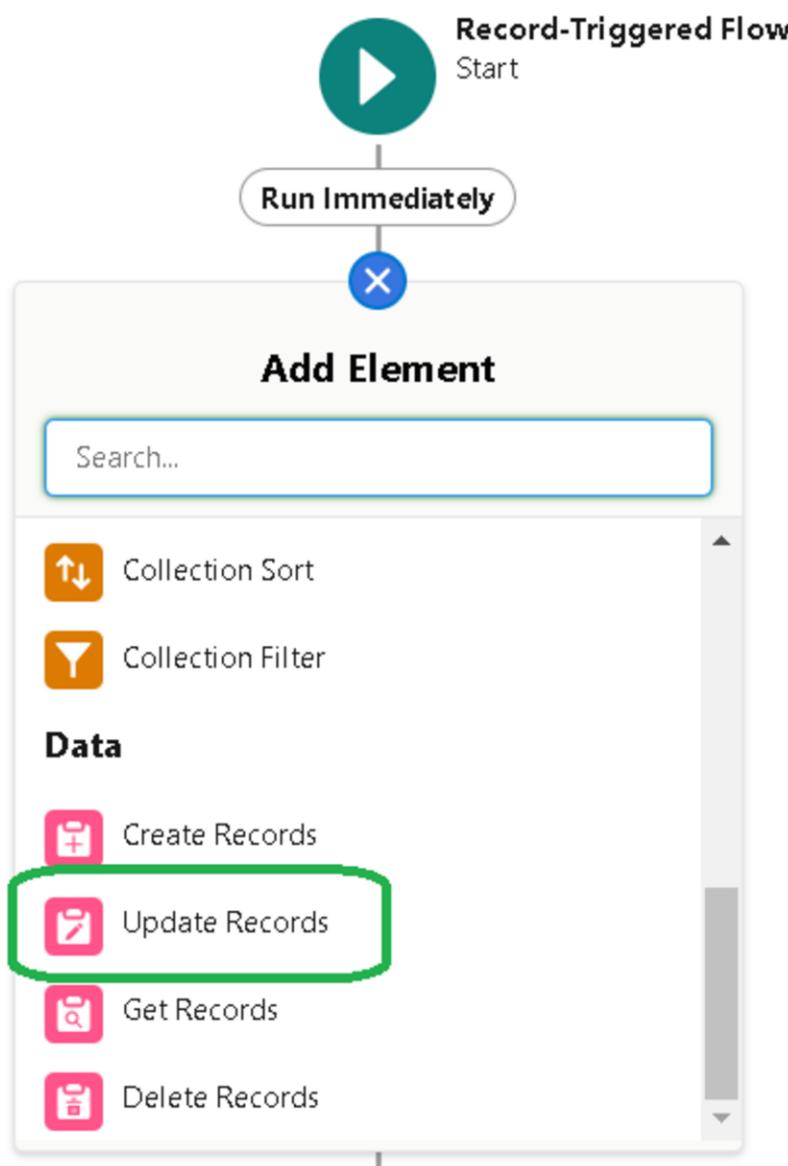
3

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

Cancel Done

4

- 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
- Api name : is auto populated

## Edit Update Records

Update Salesforce records using values from the flow.

\*Label

Amount Update

\*API Name

Amount\_Update

Description

### \* How to Find Records to Update and Set Their Values

- Use the billing details and feedback record that triggered the flow
- Update records related to the billing details and feedback record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

## Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Cancel

Done

## Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Field

Payment\_Status\_\_c

Operator

Equals

Value

Completed

+ Add Condition

## Set Field Values for the Billing details and feedback Record

Field

Payment\_Paid\_\_c

Value

\$Record > Service records > Appointment > Service A...

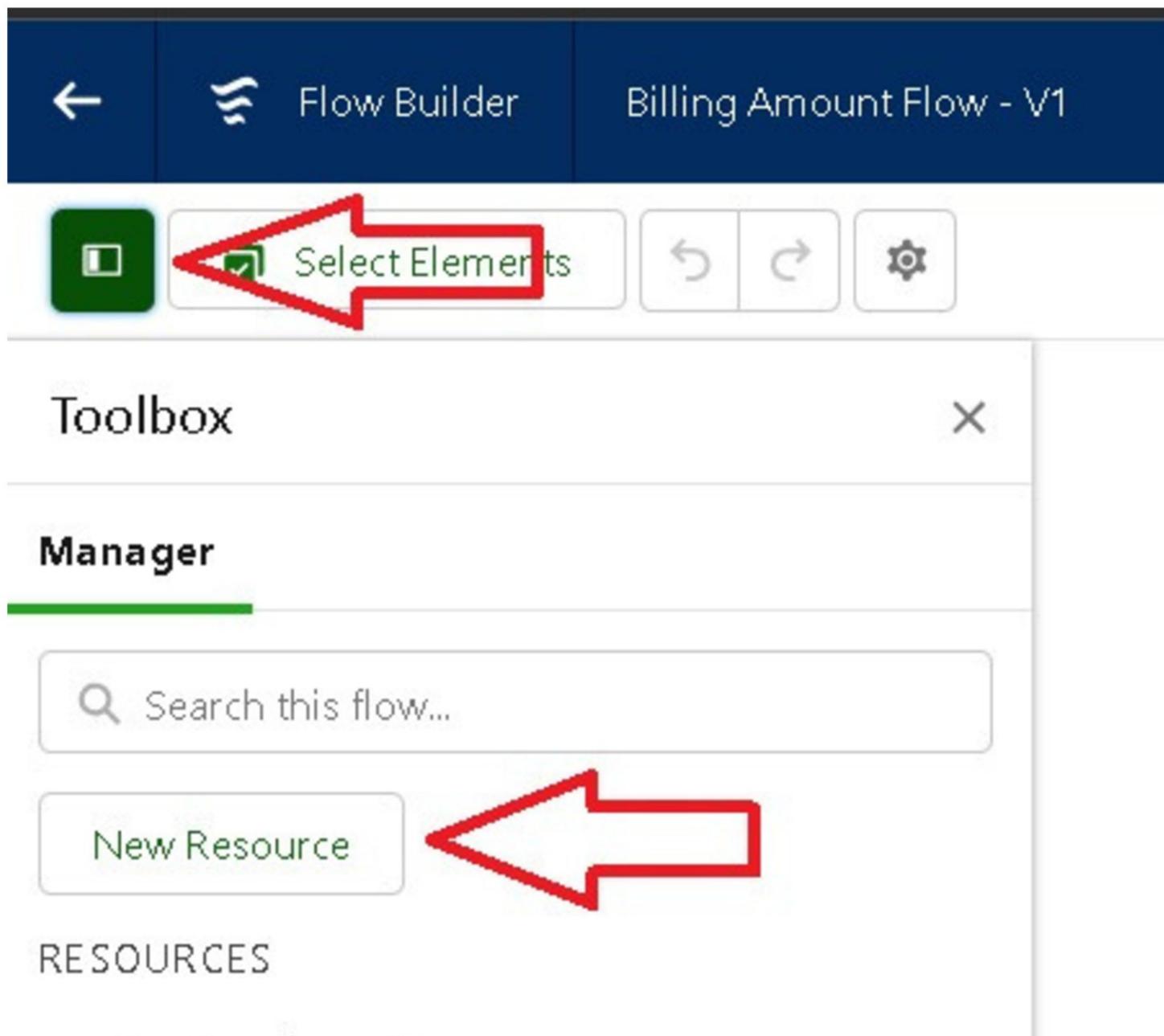
+ Add Field

Cancel

Done

1. Set a filter condition : All Conditions are met(AND)
2. Field : Payment\_Status\_\_c

3. Operator : Equals Value : Completed And Set Field Values for the Billing details and
4. feedback Record
5. Field : Payment\_Paid\_c
6. Value : {!\$Record.Service\_records\_\_r.Appointment\_\_r.Service\_Amount\_\_c} Click On
7. Done. Before creating another Element. Create a New Resource form Toolbox form
8. top left.



1. Click on the New Resource, And select Variable.
2. Select the resource type as text template.
3. Enter the API name as " alert".
4. Change the view as Rich Text ? View to Plain Text.
5. In body field paste the syntax that given below.

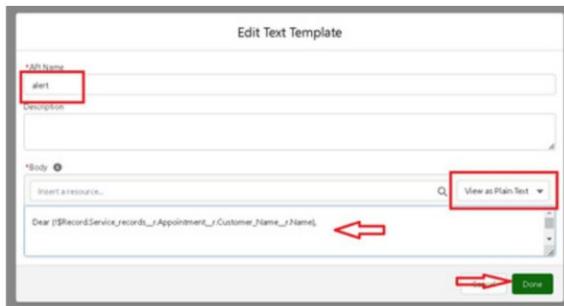
Dear {!\$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated,

and it helps us continue to provide top-notch services to you and all our valued customers.  
Amount paid : {!\$Record.Payment\_Paid\_\_c}

Thank you for Coming .

1. Click done.



## Create a Flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with a search bar containing 'flows' (Step 1). Under 'Process Automation', there's a 'Flows' link (Step 2). The main area is titled 'Flows' and shows a list of 'Flow Definitions'. At the top right of this list, there's a 'New Flow' button (Step 3). The list includes several entries like 'Ac Amount update', 'Book Appointment from Invitation', etc., with columns for 'Flow Label', 'Process Type', 'Last Modified By', and 'Last Modified Date'.

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

## New Flow

Core All + Templates

1

Screen Flow Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.	Record-Triggered Flow Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
Schedule-Triggered Flow Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.	Platform Event—Triggered Flow Launches when a platform event message is received. This autolaunched flow runs in the background.
Autolaunched Flow (No Trigger) Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.	Record-Triggered Orchestration Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

2

Create

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

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

\*Object  
Billing details and feedback

Configure Trigger

\*Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

Done

## Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

None

## \*Optimize the Flow for:

### Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved to the database.

### Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.

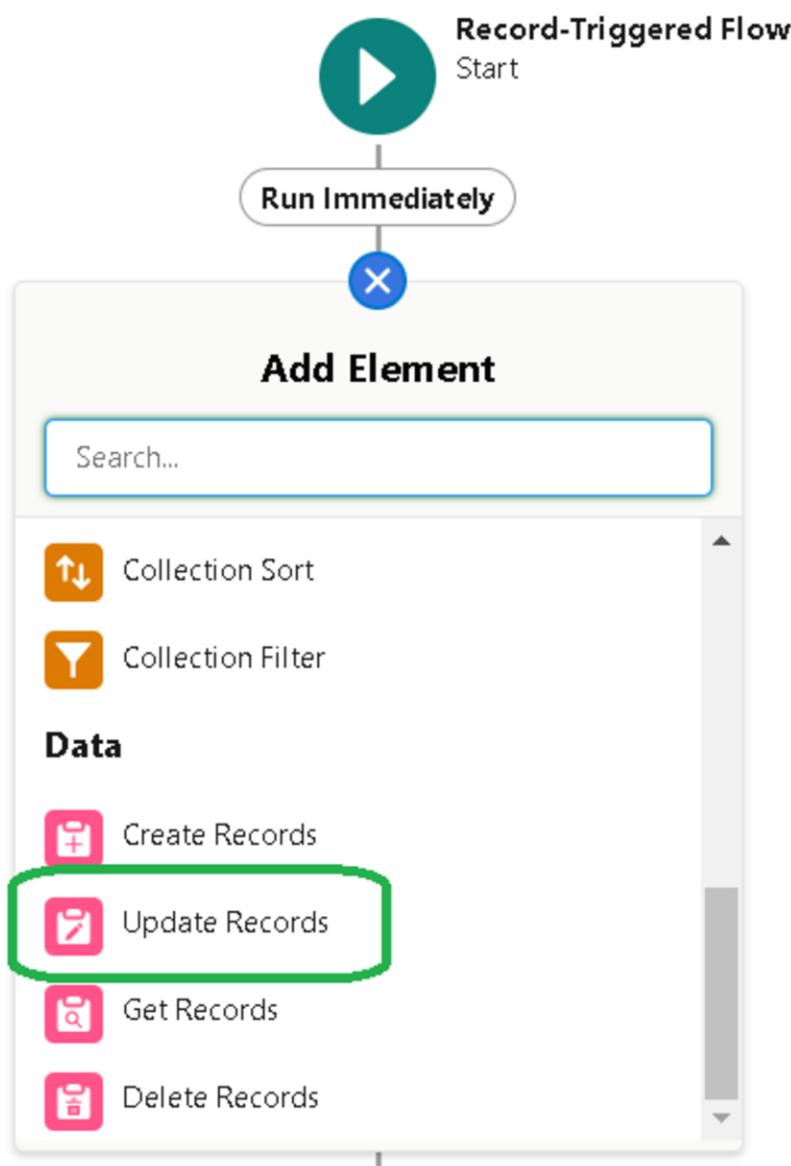
3

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

Cancel Done

4

- 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
- Api name : is auto populated

## Edit Update Records

Update Salesforce records using values from the flow.

\*Label

Amount Update

\*API Name

Amount\_Update

Description

### \* How to Find Records to Update and Set Their Values

- Use the billing details and feedback record that triggered the flow
- Update records related to the billing details and feedback record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

## Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Cancel

Done

## Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Field

Payment\_Status\_\_c

Operator

Equals

Value

Completed

+ Add Condition

## Set Field Values for the Billing details and feedback Record

Field

Payment\_Paid\_\_c

Value

\$Record > Service records > Appointment > Service A...



+ Add Field

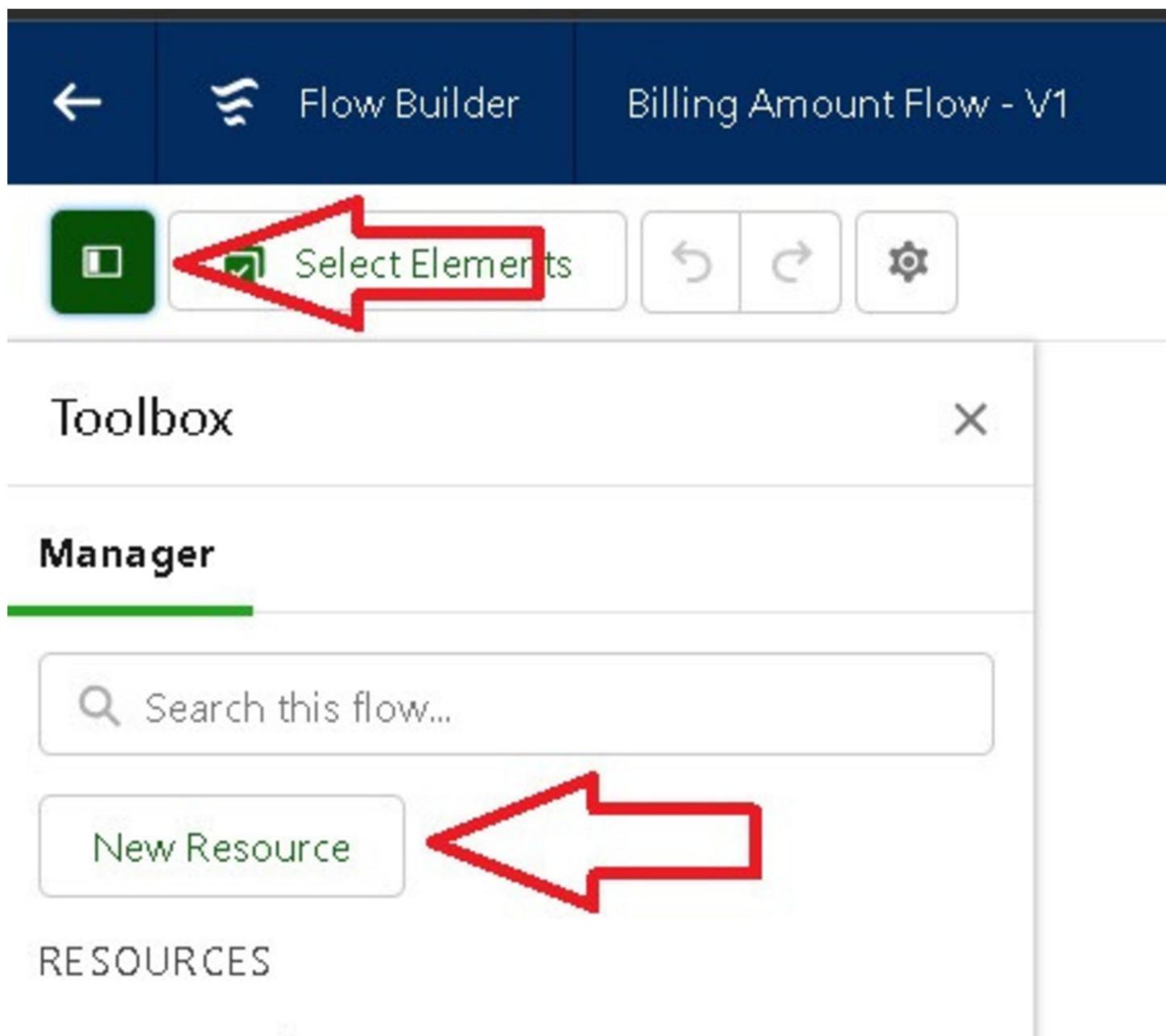
Cancel

Done

1. Set a filter condition : All Conditions are met(AND)
2. Field : Payment\_Status\_\_c

3. Operator : Equals Value : Completed And Set Field Values for the Billing
4. details and feedback Record
5. Field : Payment\_Paid\_\_c
6. Value : {!\$Record.Service\_records\_\_r.Appointment\_\_r.Service\_Amount\_\_c}
7. Click On Done.
- 8.
- 9.

Before creating another Element. Create a New Resource from Toolbox form top left.



1. Click on the New Resource, And select Variable.
2. Select the resource type as text template.
3. Enter the API name as " alert".
4. Change the view as Rich Text ? View to Plain Text.
5. In body field paste the syntax that given below.

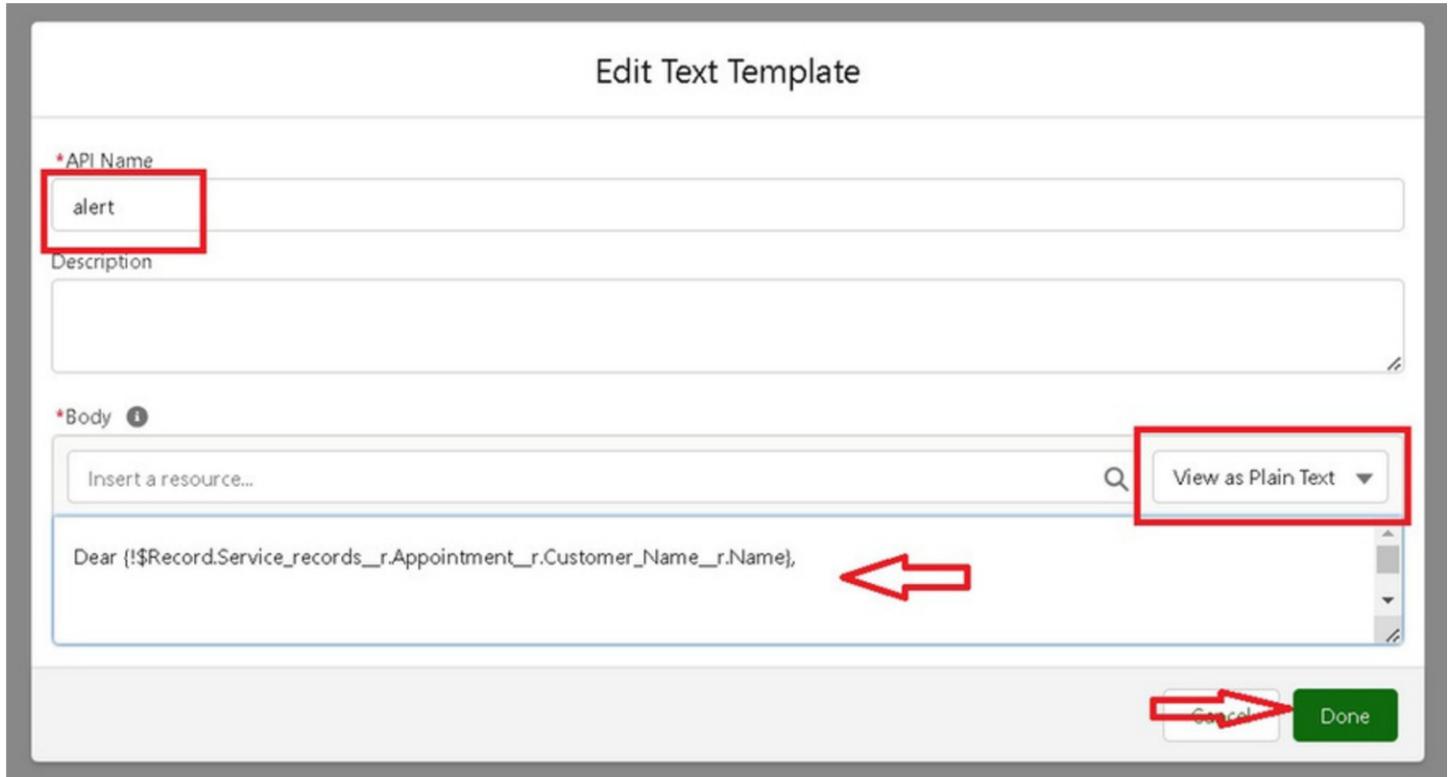
Dear {!\$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!\$Record.Payment\_Paid\_\_c}

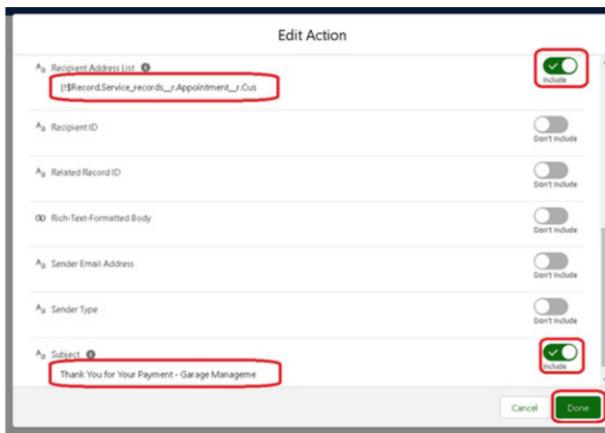
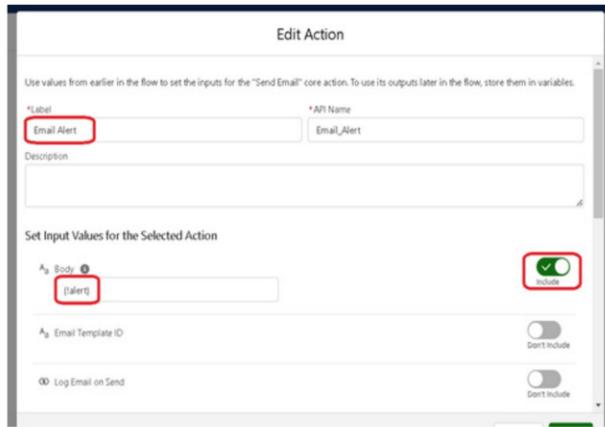
Thank you for Coming .

1. Click done.

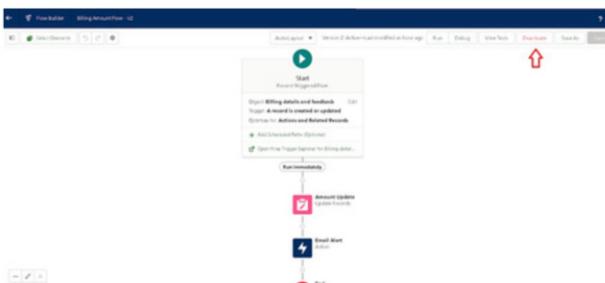
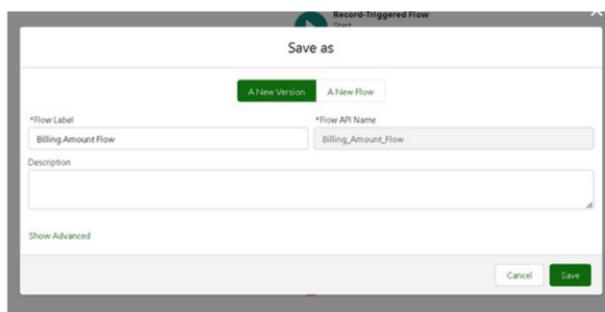


Now Click on Add Element , select Action.

1. Their action bar will be opened in that search for “ send email ” and click on it.
2. Give the label name as “ Email Alert”
3. API name will be auto populated.
4. Enable the body in set input values for the selected action.
5. Select the text template that created , Body : {!alert}
6. Include recipient address list select the email form the record.
7. RecipientAddressList: {!\$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Gmail\_\_c}
8. Include subject as “ Thank You for Your Payment - Garage Management”.
9. Click done.

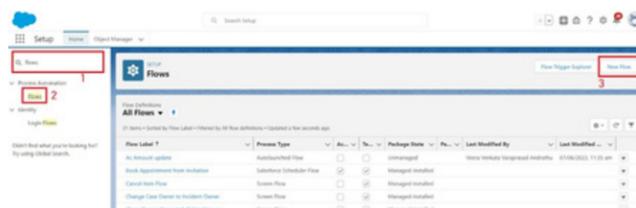


1. Click on save. Give the Flow label , Flow Api name will be autopopulated.
2. And click save, and click on activate.

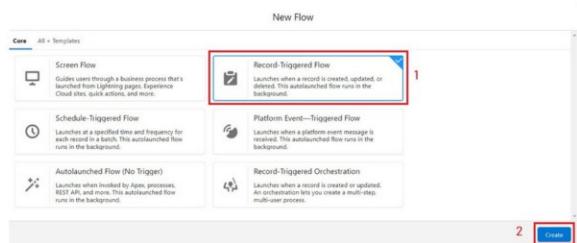


## Activity 2:Create another Flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.

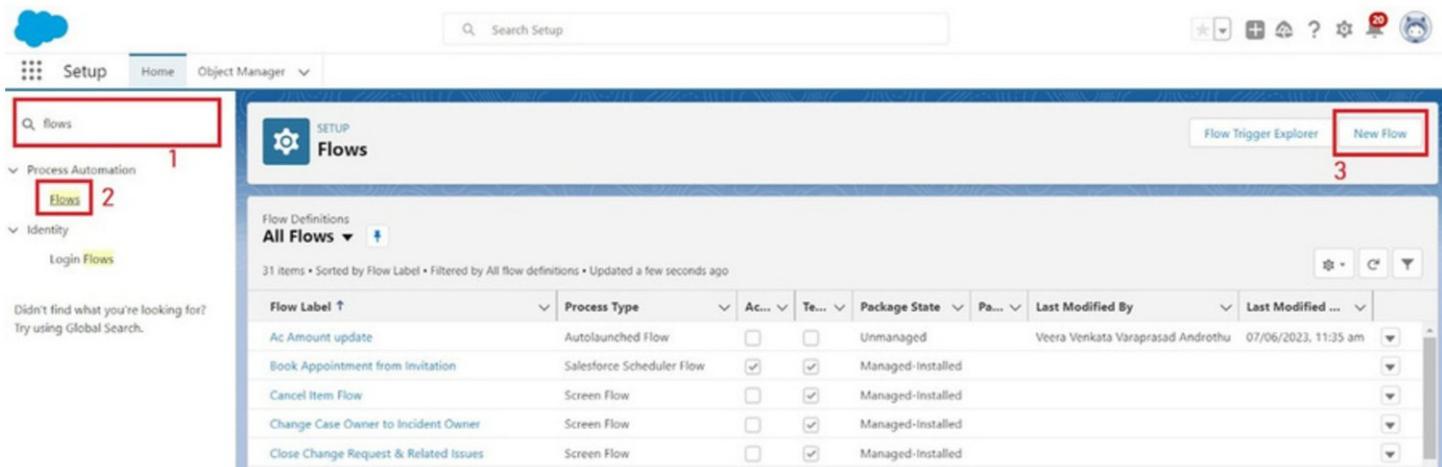


1. Select the Record-triggered flow and Click on Create.



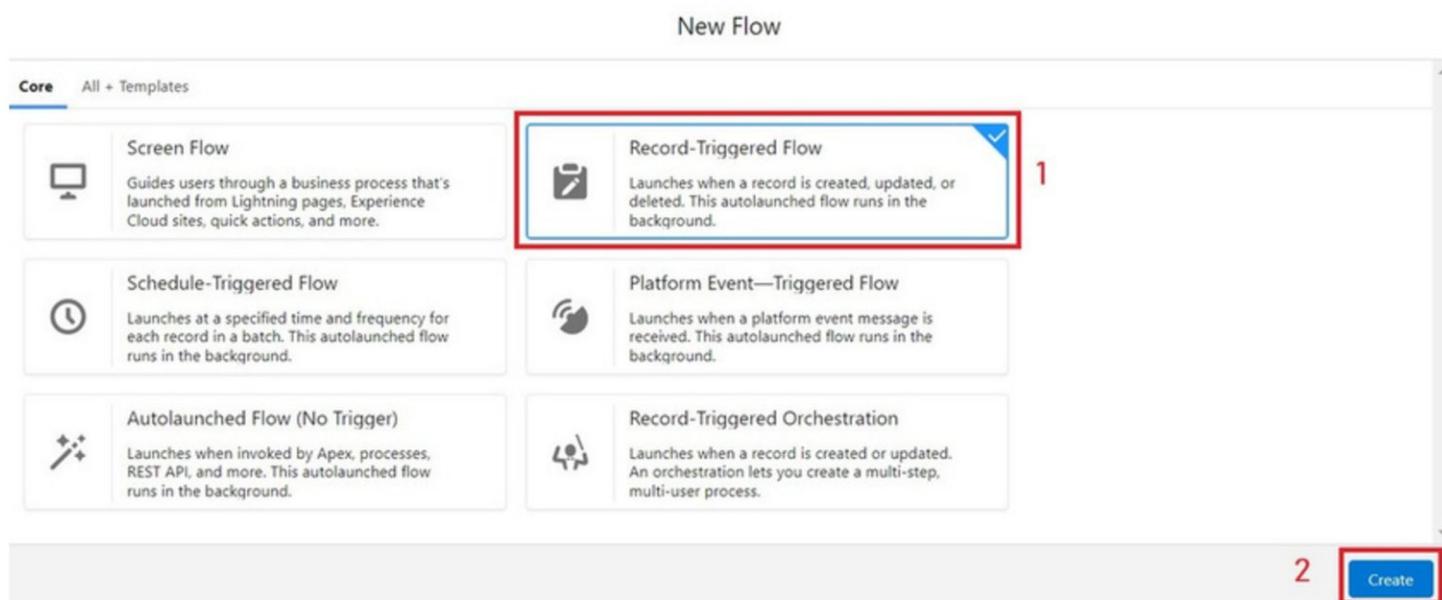
## Create another Flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.



1.

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



Select the Object as " Service records" in the Drop down list.

1. Select the Trigger Flow when: "A record is Created or Updated".
2. Select the Optimise the flow for: "Actions and Related Records" and Click on Done.
3. Under the Record-triggered Flow Click on "+" Symbol and In the Drop down List select the "Update records Element".
4. Set a filter condition : All Conditions are met(AND)
5. Field : **Quality\_Check\_Status\_\_c**

6. Operator : **Equals**
7. Value : **True**
8. And Set Field Values for the Billing details and feedback Record
9. Field : **Service\_Status\_\_c**
10. Value : **Completed**

The screenshot shows the 'Set Filter Conditions' section with a condition 'Quality\_Check\_Status\_\_c Equals True'. Below it, the 'Set Field Values for the Service record Record' section shows 'Service\_Status\_\_c' set to 'Completed'.

1. Click On **Done**.
2. Click on **save**
3. Given the Flow label as **Update Service Status**, Flow Api name will be auto populated.
4. And click save, and click on **activate**.

## Module14: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.

## Activity 1: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 ”.

```

AmountDistributionHandler // AmountDistributionHandler.apex
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 //
5 //        for(Appointment__c app : listApp){
6 //            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
7 //                app.Service_Amount__c = 10000;
8 //            }
9 //            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
10 //                app.Service_Amount__c = 5000;
11 //            }
12 //            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
13 //                app.Service_Amount__c = 8000;
14 //            }
15 //            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
16 //                app.Service_Amount__c = 7000;
17 //            }
18 //            else if(app.Maintenance_service__c == true){
19 //                app.Service_Amount__c = 2000;
20 //            }
21 //        }
22 //    }
23 //}
24 //
25 //}
26 //
27 //}
28 //
29 //}
30 //
31 //}
32 //
33 //}

```

```

AmountDistributionHandler.apex // AmountDistributionHandler.apex
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        }
21    }
22 }

```

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.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;

}

```

```

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;

}

}

}

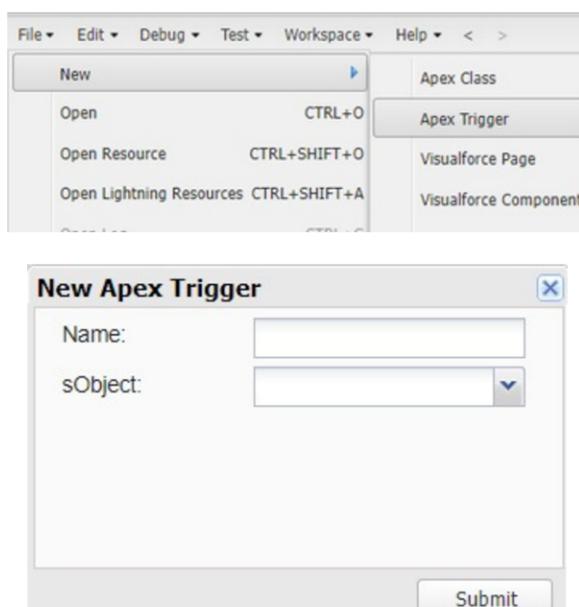
}

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 :

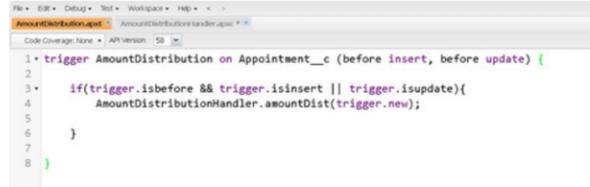
The syntax for creating trigger is :

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.

### 1. Handler for the Appointment Object



```
AmountDistributionHandler.apl  
Code Coverage: None • API Version: 50  
trigger AmountDistribution on Appointment__c (before insert, before update) {  
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
        AmountDistributionHandler.amountDist(trigger.new);  
    }  
}
```

Code:

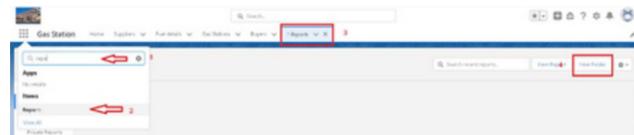
```
trigger AmountDistribution on Appointment__c (before insert, before update) {  
  
if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
  
AmountDistributionHandler.amountDist(trigger.new);  
  
}  
}
```

## Module15:Reports

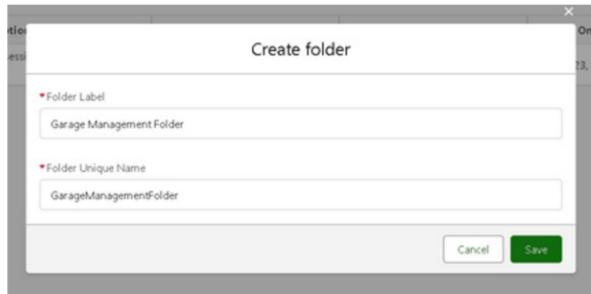
1. Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.
2. Types of Reports in Salesforce
  1. Tabular
  2. Summary
  3. Matrix
  4. Joined Reports

## Activity 1:create a report folder

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.

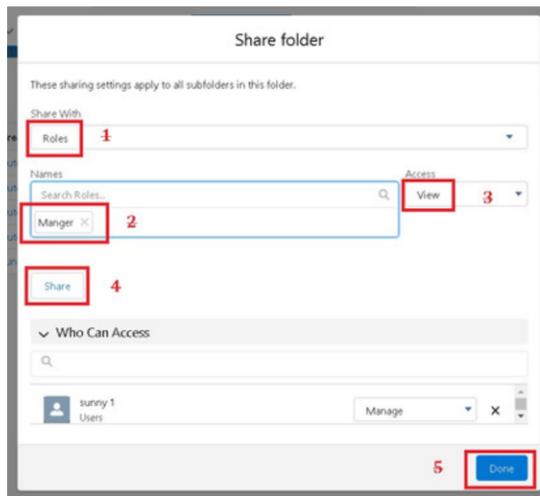


1. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
2. Click save.



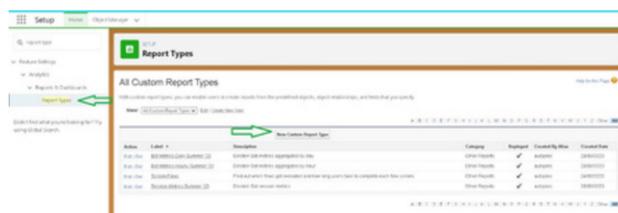
## Activity 2:Sharing a report folder

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.

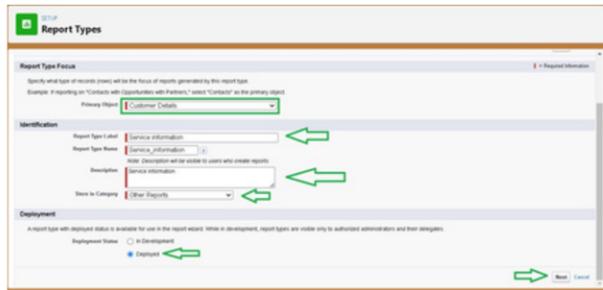


## Activity 3:Create Report Type

1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.



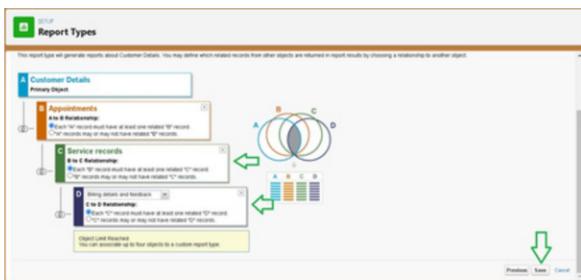
1. Select the Primary object as “ Customer details” .
2. Give the Report type Label as “ Service information ”
3. Report type Name is autopopulated.
4. Keep the Description as same.
5. Select Store in Category as “ other Reports ”
6. Select the deployment status as “ Deployed ”, click on Next.



1. now , Click on Related object box.
2. Click on Select Object, choose Appointment Object as shown in fig.



1. Again Click to relate another object.
2. And select the related object as “ service records”.
3. Repeat the process and select the related object as “ Billing details and feedback”.
4. And click on save.



## Activity 4: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.

1. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.

## Create Report

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

Try to fill every field in each record for better experience.

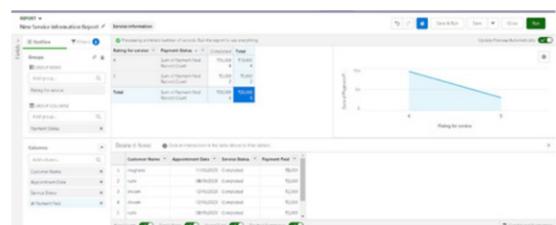
1.

Go to the app >> click on the reports tab

2. Click New Report.

1. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.

1. 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
2. Remove the unnecessary fields.
3. Select the fields that mentioned below in GROUP ROWS section.
  1. Rating for Service
4. Select the fields that mentioned below in GROUP ROWS section.
  1. Payment Status
5. Click on Add Chart , Select the Line Chart.
6. Click on save, Give the report Name : New Service information Report
7. Report unique Name is auto populated.
8. Select the folder the created and Click on save.

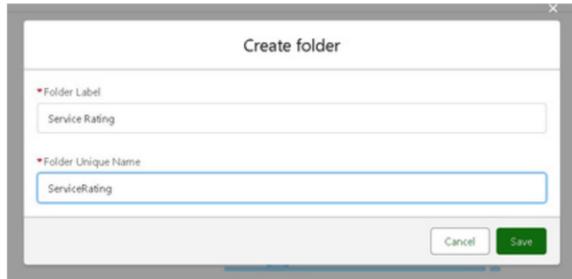


## Module16:Dashboards

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.

## Activity 1:Create Dashboard Folder

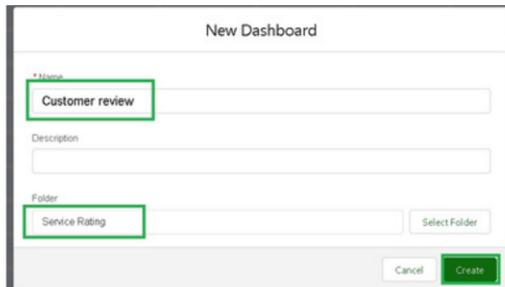
1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.



1. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

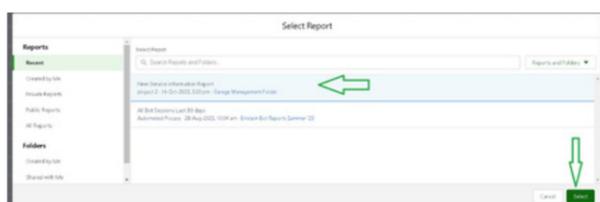
## Activity 2:Create Dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.



1. Select add component.

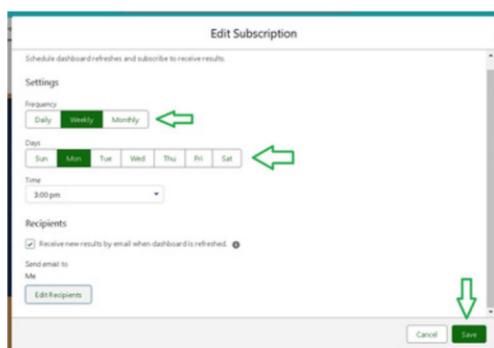
1. Select a Report and click on select.



1. Select the Line Chart. Change the theme.
2. Click Add then click on Save and then click on Done.
3. Preview is shown below.



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.



## Module17:User Adoption

### Activity 1:creating records

To create a record in the follow objects follow these steps

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

### creating records

To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for “ **Garage Management** ” and click on it.
3. Click on the “ **Consumer details** tab”.

- Click on new and fill the details as shown below figs, and click save.

**New Customer Detail**

\* = Required Information

**Information**

\* Customer Name  
Mac

Phone number  
5678765567

Gmail  
mac@gmail.com

Owner  
Annapurna SmartBridge

**Cancel** **Save & New** **Save**

Now, Create the Appointment Record

- Click on the “**Appointment** tab”.
- Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
- Match the validation while entering the vehicle number plate.
- Select the services you need.
- Click on save to see the Service Amount.

Appointments

Owner  
Annapurna SmartBridge

Appointment Name  
app-016

Customer Details  
Mac

Appointment Date  
13/11/2024

Maintenance service

Vehicle number plate  
T330EJ044

**Save**

Now, Create a service Record

- Click on the “**Service record** tab”.
- Enter the Appointment, and started is selected as default.
- Click on save.

New Service record

Information

\* = Required Information

Service Record Name: Appointment app-016

Owner: Annapurna SmartBridge

Quality Check Status:

Service Status: Started

Cancel Save & New Save

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

Service Record Name: ser-010

Owner: Annapurna SmartBridge

\* Appointment: app-016

Quality Check Status:

Service Status: Started

service date: 18/11/2024  
This field is calculated upon save

Created By: Annapurna SmartBridge, 18/11/2024, 4:32 pm

Last Modified By: Annapurna SmartBridge, 18/11/2024, 4:34 pm

Cancel Save Save & New

1. Now automatically Service status will be moved to completed.

Related Details

Service Record Name: ser-010

Owner: Annapurna SmartBridge

Appointment: app-016

Quality Check Status:

Service Status: Completed

service date: 18/11/2024

Created By: Annapurna SmartBridge, 18/11/2024, 4:32 pm

Last Modified By: Annapurna SmartBridge, 18/11/2024, 4:34 pm

## Conclusion

The Garage Management System successfully streamlines and automates the daily operations of a garage or automobile service center. By digitizing key processes such as vehicle service tracking, customer information management, billing, and inventory handling, it reduces manual effort, minimizes errors, and improves overall efficiency.

This project demonstrates how technology can replace traditional paper-based methods, ensuring quick data retrieval, accurate record-keeping, and better customer satisfaction. It also provides secure data storage, user-friendly navigation, and scope for future enhancements such as online booking, SMS/email notifications, and integration with payment gateways.



