

Garage Management system

College Name: BISHOP APPASAMY COLLEGE OF

ARTS AND SCIENCE

College Code: BRU3G

TEAM ID:

TEAM MEMBERS: Team Leader Name:

George Benedict.A

Email: g.benedict2205@gmail.com

Team Member 1: yuvaraj.a

Email: yuvastyle02@gmail.com

Team Member 2: KUMARAN .R

Email: kumaranpoongodi7@gmail.com

Team Member 3: BASIL SABU

Email: basilsabu286@gmail.com

1 .INTRODUCTION

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.

DEVELOPMENT PHASE

Creating Developer Account: By using this URL

- [Developer Edition with Agentforce and Data Cloud - Salesforce.com](#)

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.

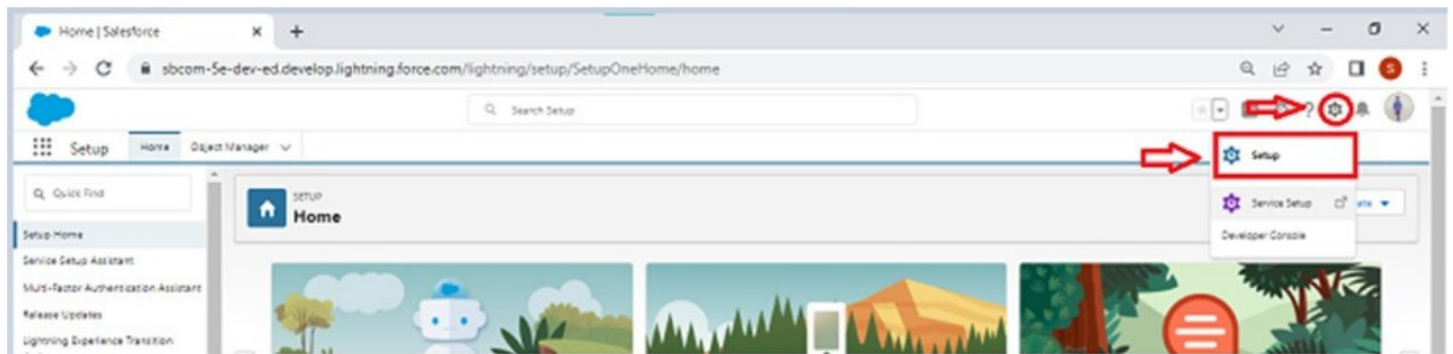
Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

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:

Object

1. 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:
 - Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
 - 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.
-
- To Navigate to Setup page:
- Click on gear icon ? click setup.



-
-

To create an object:

- From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.



- On Custom object defining page:
 - Enter the label name, plural label name, click on Allow reports, Allow search.

New Custom Object | Salesforce

sbcom-5e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new

Setup Home Object Manager

SETUP New Custom Object

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information Required Information

The singular and plural labels are used in lists, page layouts, and records.

Label Example: Account

Plural Label Example: Accounts

Starts with novel sound ☐

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 window using a Visualforce page

Context 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 Example: AccountName

Date Type

Optional Features

☒ Allow Reports ☐ Allow Activities ☐ Track Field History ☐ Allow in Chatter Groups ☐ Enable Licensing

- 1.
- 2.

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 [What is this?](#)

☐ 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

- 1.

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.

Lightning Page Tabs

1.

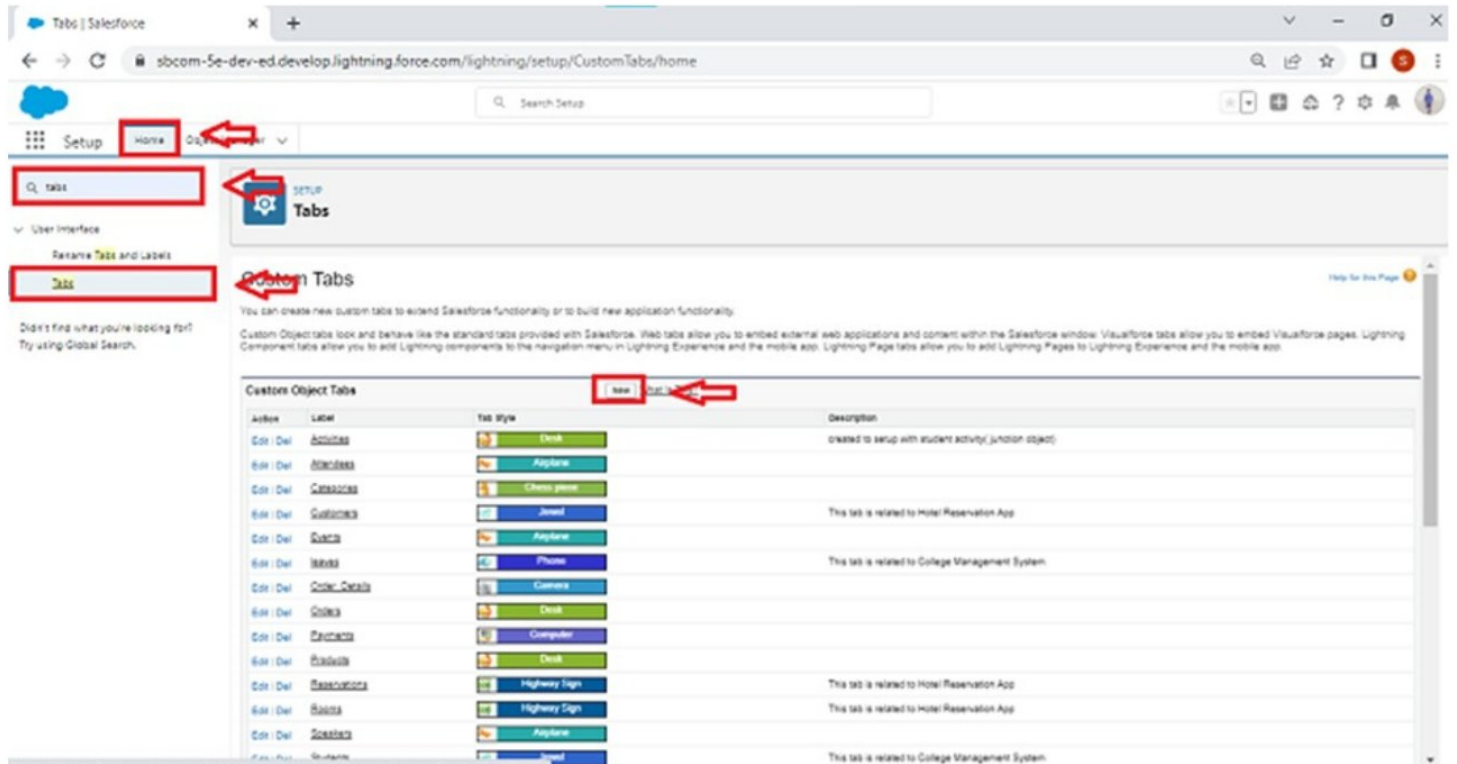
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

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)



1. 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 .
2. Make sure that the Append tab to users' existing personal customizations is checked.
3. Click save.

New Custom Object Tab

Help for this Page 

Step 1. Enter the Details

Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object

Tab Style

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link

Enter a short description.

Description

[Next](#) [Cancel](#)

Tab Style Selector

[Create your own style](#)

Hide styles which are used on other tabs

 Airplane	 Alarm clock	 Apple	 Balls
 Bank[1]	 Bell	 Big top	 Boat[1]
 Books	 Bottle	 Box	 Bridge
 Building	 Building Block	 Caduceus	 Camera
 Can	 Car	 Castle	 CD/DVD
 Cell phone	 Chalkboard	 Chess piece	 Chip
 Circle	 Compass	 Computer	 Credit card
 CRT TV	 Cup	 Desk[1]	 Diamond
 Dice	 Factory	 Fan	 Flag
 Form	 Gears	 Globe	 Guitar
 Hammer	 Hands	 Handsaw	 Headset
 Heart[1]	 Helicopter	 Hexagon	 Highway Sign
 Hot Air Balloon	 Insect	 IP Phone	 Jewel
 Keys	 Laptop	 Leaf	 Lightning

[Save](#)

[Cancel](#)

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.

Custom App	 <input type="checkbox"/> Include Tab
Platform (standard__Platform)	<input type="checkbox"/>
Sales (standard__Sales)	<input type="checkbox"/>
Service (standard__Service)	<input type="checkbox"/>
Marketing (standard__Marketing)	<input type="checkbox"/>
Sample Console (standard__ServiceConsole)	<input type="checkbox"/>
High Volume Customer Portal User	<input type="checkbox"/>
Authenticated Website User	<input type="checkbox"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>

Analytics Studio (standard__Insights)	<input type="checkbox"/>
Sales Console (standard__LightningSalesConsole)	<input type="checkbox"/>
Service Console (standard__LightningService)	<input type="checkbox"/>
Sales (standard__LightningSales)	<input type="checkbox"/>
Lightning Usage App (standard__LightningInstrumentation)	<input type="checkbox"/>
Digital Experiences (standard__SalesforceCMS)	<input type="checkbox"/>
Queue Management (standard__QueueManagement)	<input type="checkbox"/>
Bolt Solutions (standard__LightningBolt)	<input type="checkbox"/>
Data Manager (standard__DataManager)	<input type="checkbox"/>
Salesforce Scheduler Setup (standard__LightningScheduler)	<input type="checkbox"/>

☒ Append tab to users' existing personal customizations

 [Previous](#) [Save](#) [Cancel](#)

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

Creation of fields for the Customer Details object

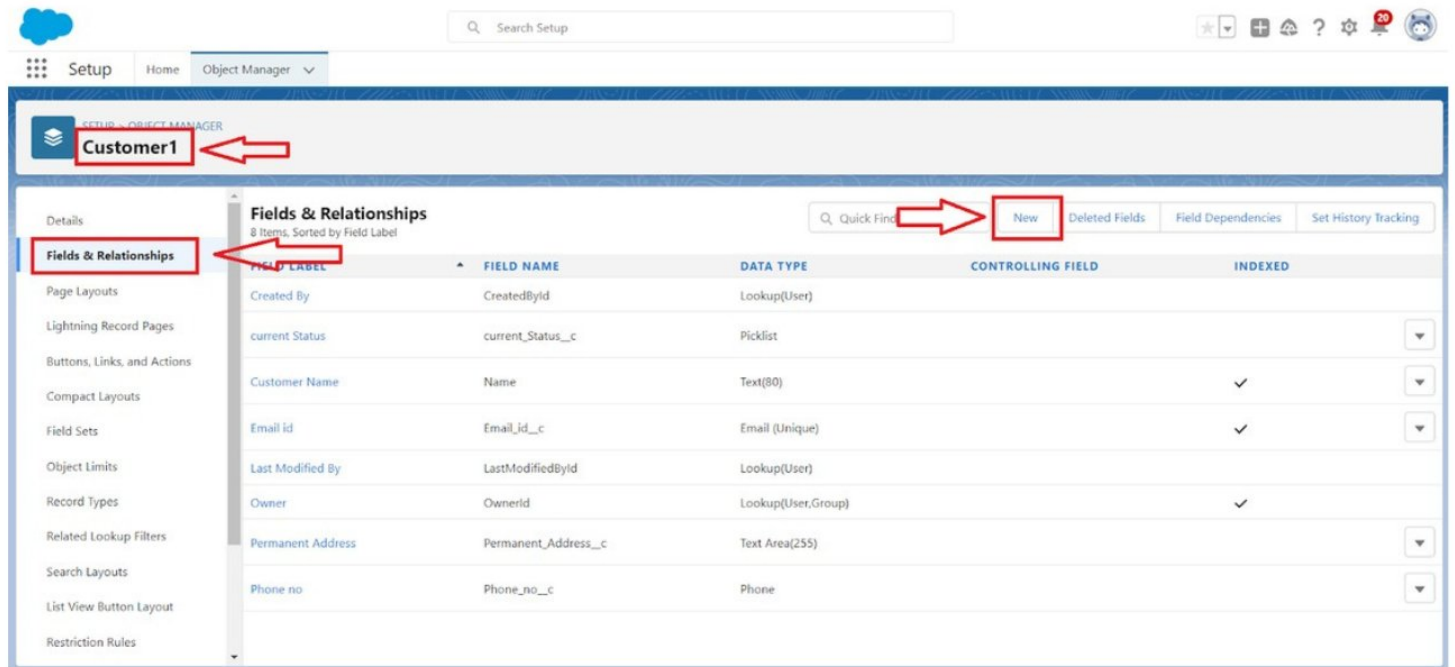
1. To create fields in an object:

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



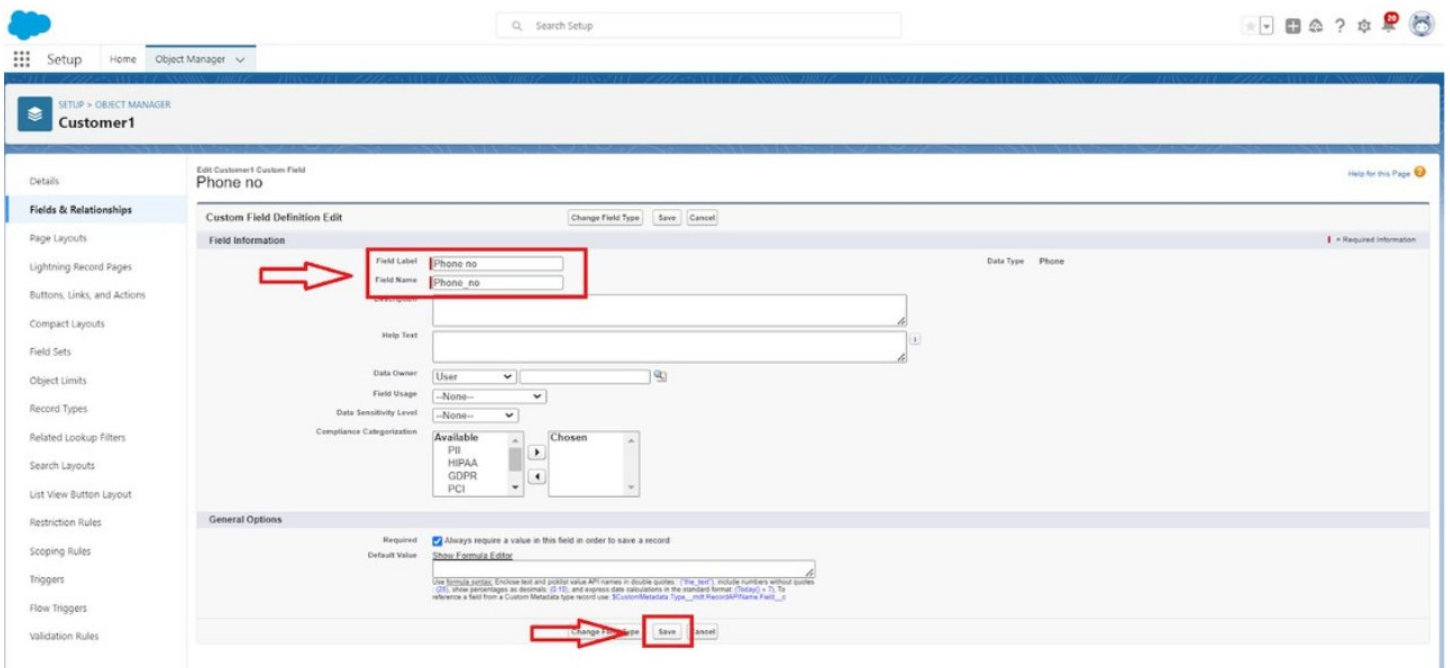
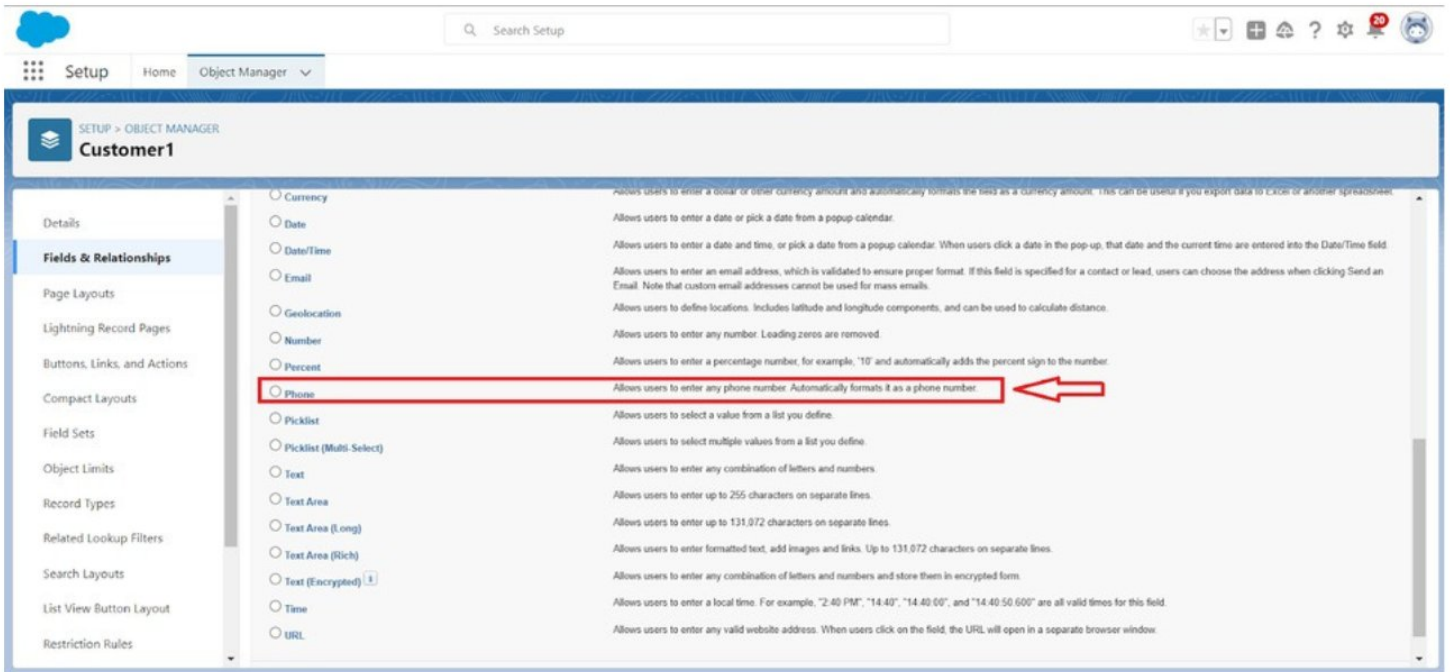
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Customer	Customer	Standard Object			
Customer Details	Customer_Details__c	Custom Object		05/10/2023	✓

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



FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
current Status	current_Status__c	Picklist		
Customer Name	Name	Text(80)		✓
Email id	Email_Id__c	Email (Unique)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Permanent Address	Permanent_Address__c	Text Area(255)		
Phone no	Phone_no__c	Phone		

1. Select Data Type as a “Phone”



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:
 - Field Label : Gmail
 - Field Name : gets auto generated
 - Click on Next >> Next >> Save and new.

Validation rule

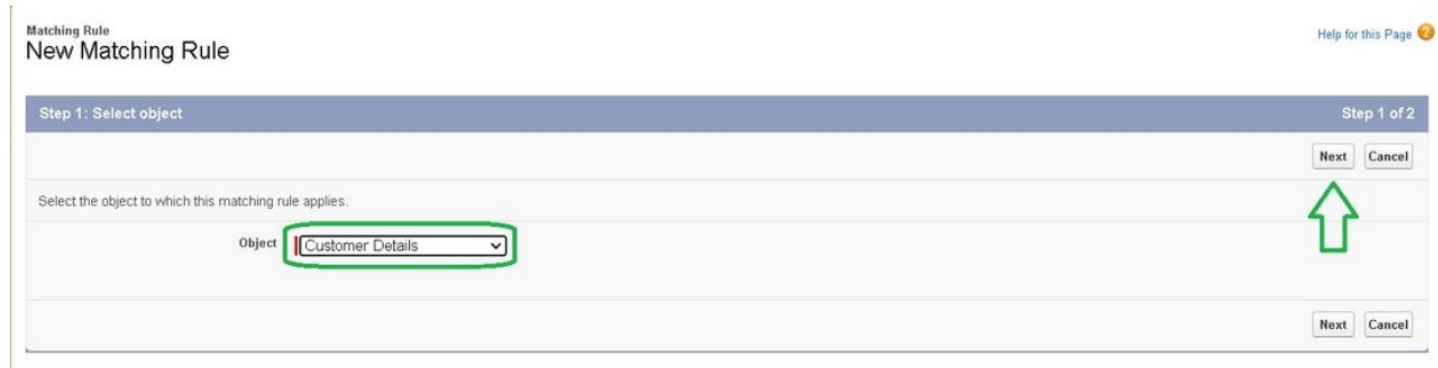
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

To create a 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.



1. Select the object as Customer details and click Next.



1. Give the Rule name : Matching customer details
2. Unique name : is auto populated
3. Define the matching criteria as
4. Field Matching Method

1. Gmail Exact

2. Phone Number Exact

1. Click save.
2. After Saving Click on Activate.

SaveCancel

Rule Details

Object

Customer Details

Rule Name

matching Customer deta

Unique Name

matching_Customer_det

Description

Matching Criteria

Tell the rule which fields to compare and how.

Field

Gmail

Phone Number

--None--

--None--

--None--

Matching Method

Exact

Exact

Exact

Exact

Exact

Match Blank Fields

☐

☐

☐

☐

☐

AND

AND

AND

AND

AND

Add Filter Logic...

SaveCancel

Matching Rule

matching Customer details

Help for this Page

Matching Rule Detail

EditDeleteCloneActivate

Object

Customer Details

Rule Name

matching Customer details

Unique Name

matching_Customer_details

Description

Matching Criteria

{Customer Details: Gmail EXACT MatchBlank = FALSE} AND {Customer Details: Phone_Number EXACT MatchBlank = FALSE}

Status

Inactive

Created By

project.2, 25/09/2023, 10:15 am

Modified By

project.2, 10/10/2023, 3:32 pm

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.

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.

ne

d

e

Information

* Customer Name ↶

Mac

Phone number ↶

5678765567

Gmail ↶

mac@gmail.com

Owner

Annapurna SmartBridge

Cancel Save & New Save

Now, Create the Appointment Record

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

Garage Manageme...

Customer Details

Appointments

Service records

Billing details and feedback

Reports

Dashboards

Appointment

app-016

Appointment Name

app-016

Owner

Annapurna SmartBridge

Customer Details

Mac

* Appointment Date

13/11/2024

Maintenance service

☒

Repairs

☒

Replacement Parts

☐

Service Amount

* Vehicle number plate

TS30EU0443

Created By

Annapurna SmartBridge, 18/11/2024, 3:28 pm

Cancel

Save

Last Modified By

Annapurna SmartBridge, 18/11/2024, 3:28 pm

Now, Create a service Record

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

New Service record

* = Required Information

Information

Service Record Name

Owner

Annapurna SmartBridge

* Appointment

app-016

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

Cancel


Save


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


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

Related **Details**


Service Record Name
ser-010



Appointment
[app-016](#) 


Quality Check Status
☒ 

Service Status
Completed 

service date
18/11/2024

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

Owner
 [Annapurna SmartBridge](#) 

Last Modified By
 [Annapurna SmartBridge](#), 18/11/2024, 4:34 pm

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”.

A screenshot of the Salesforce Developer Console. The top bar shows the file name 'AmountDistributionHandler.apex'. Below the toolbar, the code editor displays the following Apex code:

```
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;
```

Code:

```
public class AmountDistributionHandler {

    public static void amountDist(list<Appointment__c> listApp){

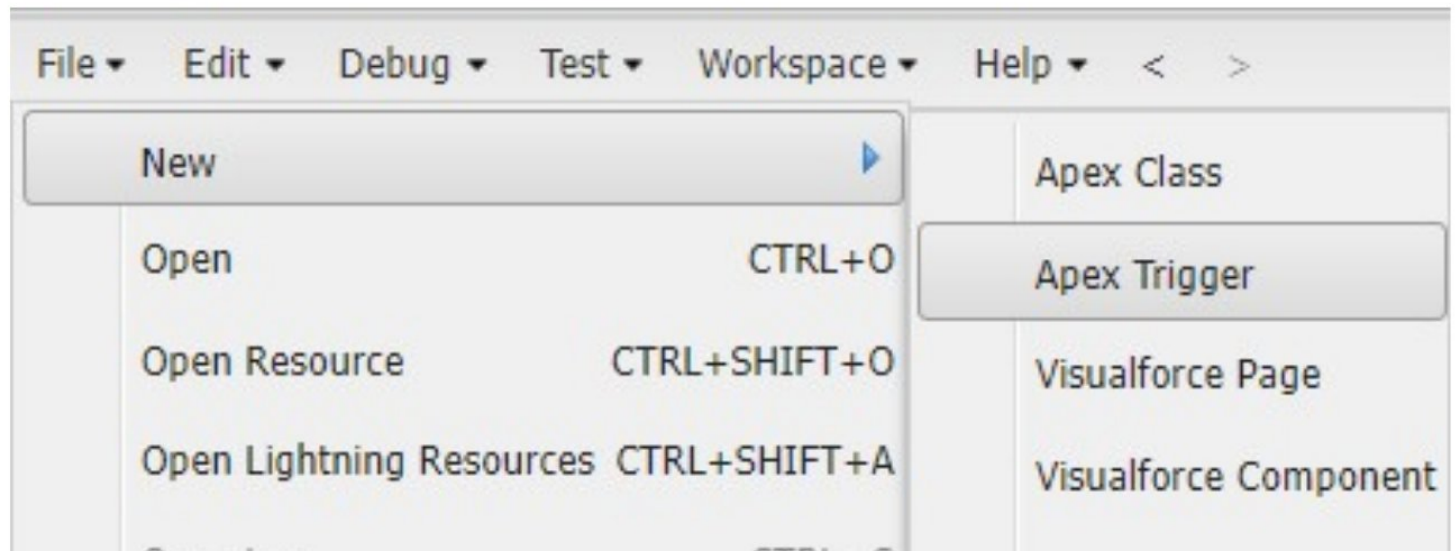
        list<Service_records__c> serList = new list <Service_records__c>();
```

```
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

A screenshot of the 'New Apex Trigger' dialog box. It has a title bar with a close button. Inside, there are two labels: 'Name:' and 'sObject:'. Next to 'Name:' is a text input field. Next to 'sObject:' is a dropdown menu with a downward arrow. At the bottom right of the dialog is a 'Submit' button.

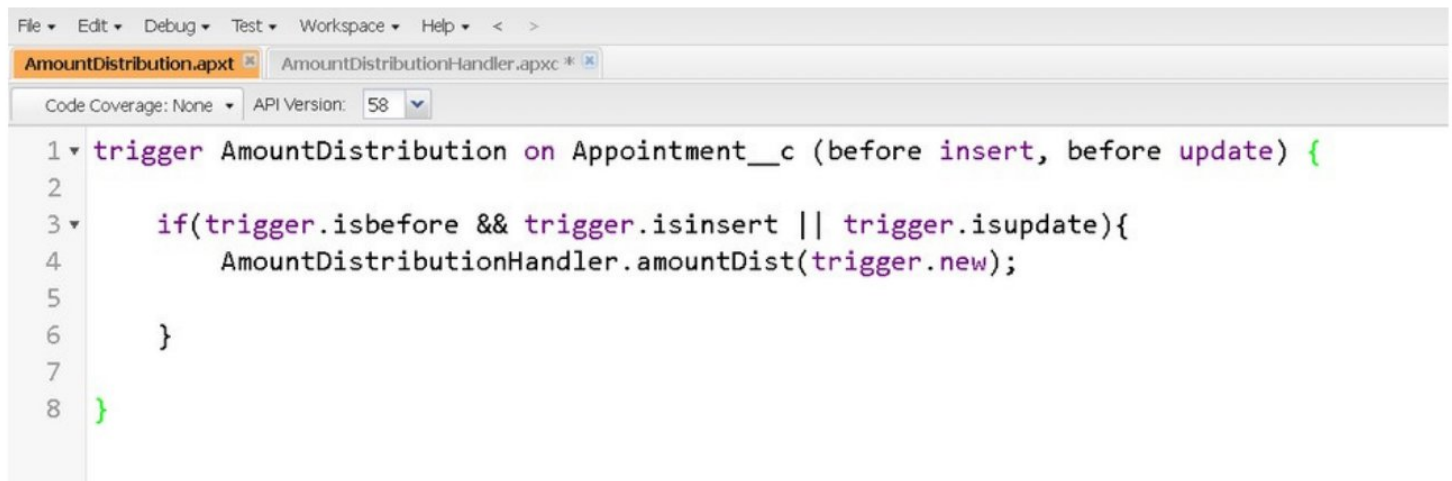
Syntax For creating trigger :

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

A screenshot of an IDE window showing a trigger definition. The window has a menu bar with File, Edit, Debug, Test, Workspace, and Help. Below the menu bar are two tabs: 'AmountDistribution.apxt' (active) and 'AmountDistributionHandler.apxc'. Below the tabs is a status bar showing 'Code Coverage: None' and 'API Version: 58'. The main editor area contains the following code:

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {  
2  
3     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
4         AmountDistributionHandler.amountDist(trigger.new);  
5     }  
6 }  
7  
8 }
```

Code:

```
trigger AmountDistribution on Appointment__c (before insert, before update) {
```

```
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
```

```
        AmountDistributionHandler.amountDist(trigger.new);
```

}

}