



# **PROJECT OF**

## **SALESFORCE DEVELOPER**

### **GARAGE MANAGEMENT SYSTEM**

**To optimize customer details, appointment, service records, and billing & feedback for automotive garages.**

By

Government Arts & Science College

Anthiyur-638 501

Department of Computer Science

Sales Force Developer with Agent Blazer Champion

Garage Management System

Team leader : Thiruganasambanthar M

Team Members

Uthayakumar T

Hariprakash K

Vivekananthan D

# PROJECT OVERVIEW

The Garage Management System (GMS) project on Salesforce is a CRM solution aimed at transforming the way automotive garages manage customer relationships, vehicle service records, and day-to-day operations. With centralized customer profiles, service histories, and appointment scheduling, the system empowers garages to deliver more personalized and efficient customer service. Automated reminders and service notifications help enhance customer engagement, ensuring regular maintenance and timely service updates to boost customer satisfaction.

In addition to customer management, the GMS CRM includes comprehensive vehicle tracking and work order management, allowing technicians and advisors to access detailed service histories, track service progress, and assign tasks seamlessly. An integrated parts inventory module ensures that essential components are in stock, with automated reordering for low-stock items to prevent service delays. Technicians can use mobile access to view work orders and inventory on the go, optimizing workflow and reducing downtime.

Finally, the GMS CRM supports billing and invoicing, providing a streamlined system for generating service estimates, digital invoices, and online payment options. With built-in analytics and reporting, managers gain insights into customer trends, revenue, and operational efficiency, enabling data-driven decisions for business growth. The system is designed to scale with the garage, offering robust data security and compliance, making it a future-proof solution for modern garage management needs.

# OBJECTIVES

- **Enhance Customer Retention and Loyalty:** By providing personalized service reminders, streamlined appointment scheduling, and post-service follow-ups, the GMS CRM aims to build long-term customer relationships and increase repeat visits.
- **Optimize Resource Allocation & Technician Efficiency:** Efficient scheduling, real-time work order tracking, and inventory management help maximize technician productivity, reduce wait times, and ensure that resources are allocated effectively for each service job.
- **Increase Revenue through Data-Driven Upselling:** With insights into customer preferences and service history, the CRM enables garages to offer tailored service recommendations and upsell additional maintenance packages, driving revenue growth.
- **Reduce Operational Costs through Automation:** By automating routine tasks like appointment reminders, parts reordering, and invoicing, the GMS CRM minimizes manual workload, reduces errors, and cuts down operational costs.
- **Improve Business Insights and Strategic Planning:** Advanced reporting and analytics provide managers with valuable insights into service trends, technician performance, and financial metrics, supporting informed decision-making and strategic business growth.

# KEY FEATURES

- **Customer details:**
  - Store and manage customer details including contact information, service history, and preferences.
  - Access comprehensive service records for customer insights and tailored service.
- **Appointment:**
  - Automated reminders and notifications to reduce no-shows.
  - Integrated calendar to optimize working time.
- **Service records:**
  - Track detailed service information including service type, parts used, and associated costs.
  - Link service records to specific customer appointments for easy tracking and reference.
- **Billing details:**
  - Generate and manage billing details, linking them to service records.
  - Track payment status and send timely payment reminders.
  - Automated invoice generation and email notifications for a streamlined payment process.
- **Feedback:**
  - Collect and manage customer feedback post-service.
  - Track ratings and comments to enhance service quality.
  - Analyze feedback trends for continuous improvement.

# DETAILED PROCESS

## Activity – 1

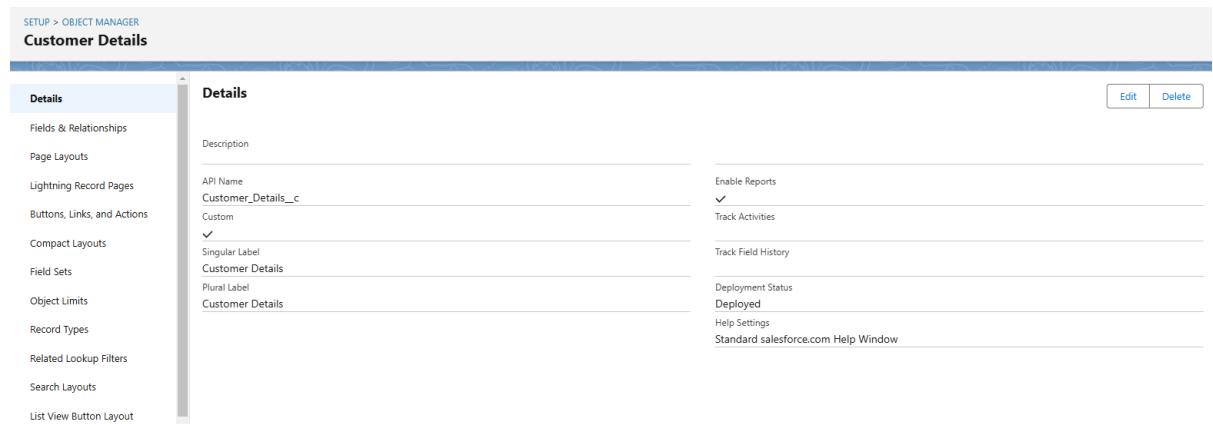
### Salesforce account Creation & Activation

- Create a Developer Account by providing all the required details and Activate the account.
- Go to <https://developer.salesforce.com/signup>.

## Activity – 2

### Object Creation

- Create all the required objects with the given label name and format.
- **Customer details Object**
  - Enter the label name >> Customer Details
  - Plural label name >> Customer Details
  - Enter Record Name Label and Format
  - Record Name >> Customer Name
  - Data Type >> Text
- 



- **Appointment Object**
  - Enter the label name >> Appointment
  - Plural label name >> Appointments

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

The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER Appointment'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits (which is selected), Record Types, Related Lookup Filters, and Search Layouts. The main 'Details' tab displays the following information:

Field	Value
Description	
API Name	Appointment_c
Custom	✓
Singular Label	Appointment
Plural Label	Appointments
Enable Reports	✓
Track Activities	
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Buttons 'Edit' and 'Delete' are located in the top right corner.

## - Service records Object

- Enter the label name >> Service records
- Plural label name >> Service records
- Enter Record Name Label and Format
- Record Name >>Service records Name
- Data Type >> Auto Number
- Display Format >> ser-{000}
- Starting number >> 1

The screenshot shows the Salesforce Object Manager interface for the 'Service records' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER Service records'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main 'Details' tab displays the following information:

Field	Value
Description	
API Name	Service_records_c
Custom	✓
Singular Label	Service records
Plural Label	Service records
Enable Reports	✓
Track Activities	
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Buttons 'Edit' and 'Delete' are located in the top right corner.

## - Billing details & Feedback Object

- Enter the label name >> Billing details and feedback

- Plural label name >> Billing details and feedback
- Enter Record Name Label and Format
- Record Name >> Billing details and feedback Name
- Data Type >> Auto Number
- Display Format >> bill-{000}
- Starting number >> 1

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

- Setup > Object Manager**
- Billing details and feedback** (Object Name)
- Details** (Tab selected)
- Fields & Relationships**
- Page Layouts**
- Lightning Record Pages**
- Buttons, Links, and Actions**
- Compact Layouts**
- Field Sets** (Selected)
- Object Limits**
- Record Types**
- Related Lookup Filters**
- Search Layouts**

**Details Panel:**

- Description: [Empty]
- API Name: Billing\_details\_and\_feedback\_\_c
- Custom: ✓
- Singular Label: Billing details and feedback
- Plural Label: Billing details and feedback
- Enable Reports: ✓
- Track Activities: [Empty]
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

## Activity – 3

### Custom Tabs Creation

- A tab is like a user interface that is used to build records for objects and to view the records in the objects.
- Create these Custom tabs for every Object by following below steps:
  - 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.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The 'Custom Tabs' section contains a table with the following data:

Action	Label	Tab Style	Description
Edit   Del	Appointments	Alarm clock	
Edit   Del	Billing details and feedback	Credit card	
Edit   Del	Customer Details	Gears	
Edit   Del	Service records	Books	

The 'Web Tabs' section displays the message: "No Web Tabs have been defined".

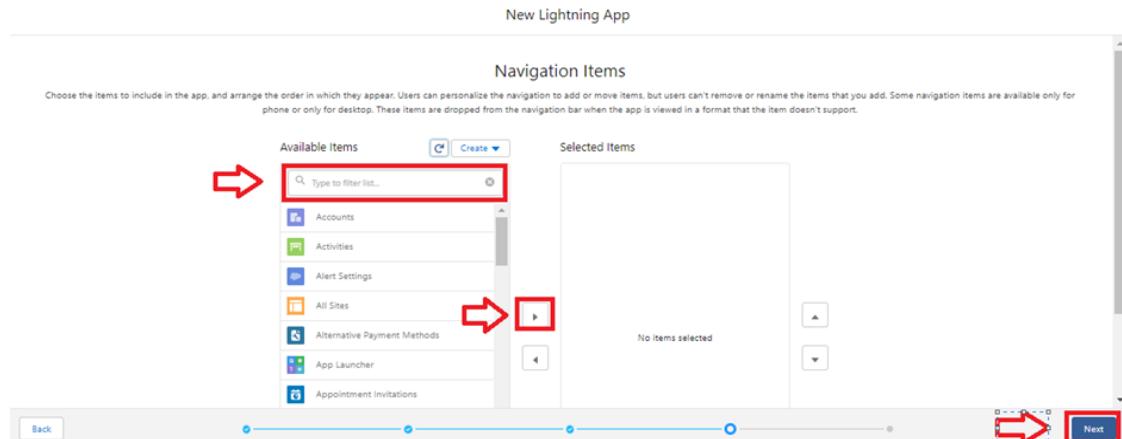
## Activity – 4

### Create a Lighting App

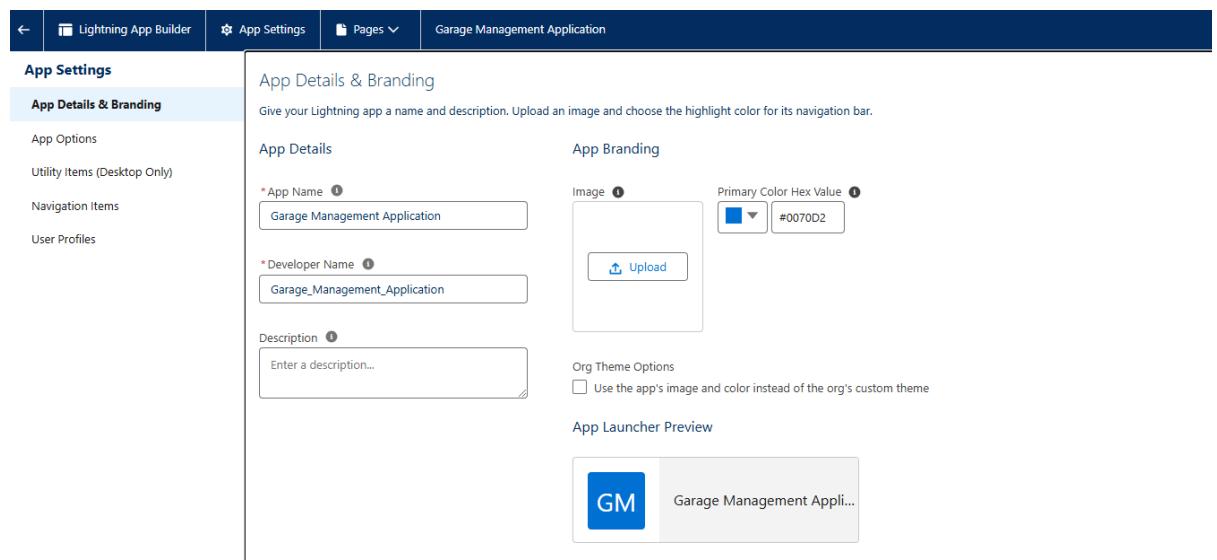
1.Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.

The screenshot shows the 'New Lightning App' setup page. The 'App Details & Branding' section includes fields for 'App Name' (with a red arrow pointing to the input field), 'Developer Name', and 'Description'. To the right, there's 'App Branding' with an 'Image' upload field and a color picker set to #0070D2. At the bottom right, a blue 'Next' button is highlighted with a red arrow.

2. To Add Navigation Items:



3. 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.



## Activity – 5

### Field Creation for every object:

Create fields for each and every object as per shown in below images.

## - Customer details object

The screenshot shows the Salesforce Object Manager interface for the 'Customer Details' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main area displays the 'Fields & Relationships' section with 8 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Gmail	Gmail_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Text(100)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone number	Phone_number_c	Phone		

## - Appointment object

The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The left sidebar lists various setup options. The main area displays the 'Fields & Relationships' section with 12 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date_c	Date		
Appointment Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Customer Details	Customer_Details_c	Lookup(Customer Details)		✓
Customer Name	Customer_Name_c	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service_c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		✓

## - Service records object

The screenshot shows the Salesforce Object Manager interface for the 'Service records' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area displays the 'Fields & Relationships' section with 8 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment_c	Lookup(Appointment)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Quality Check Status	Quality_Check_Status_c	Checkbox		
service date	service_date_c	Formula (Date)		
Service records Name	Name	Auto Number		✓
Service Status	Service_Status_c	Picklist		

## - Billing details & Feedback object

The screenshot shows the Salesforce Object Manager interface for the 'Billing details and feedback' object. The left sidebar lists various setup options. The main area displays the 'Fields & Relationships' section with 8 items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid_c	Currency(18, 0)		
Payment Status	Payment_Status_c	Picklist		
Rating for service	Rating_for_service_c	Text(1)		
Service records	Service_records_c	Lookup(Service records)		✓

## Activity – 6

### Validation Rules

#### - For Appointment

- 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]{4}"))

SETUP > OBJECT MANAGER  
**Appointment**

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

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	PANDETI NIKITHA, 7/12/2025, 10:17 PM

## - For Billing details & Feedback

- Enter the Rule name as “ rating\_should\_be\_less\_than\_5”.
- Insert the Error Condition Formula as :-
- NOT( REGEX( Rating\_for\_service\_\_c , "[1-5]{1}" ))
- Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”.

SETUP > OBJECT MANAGER  
**Billing details and feedback**

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

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rating_should_be_less_than_5	Rating for service	rating should be from 1 to 5	✓	PANDETI NIKITHA, 7/12/2025, 10:21 PM

## Activity – 7

### Duplication Rules

- **Matching Rule:**
- 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

- Create a matching rule for the Customer details Object with the given criteria.

All Matching Rules

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

Matching Rule Detail

Object	Customer Details
Rule Name	Matching customer details
Unique Name	Matching_customer_details
Description	(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)
Matching Criteria	
Status	Active
Created By	PANDETI NIKITHA, 7/12/2025, 10:25 PM
Modified By	PANDETI NIKITHA, 7/12/2025, 10:25 PM

## -Duplicate Rules:

Create a duplicate rule for the same customer details object which allows the duplication in input fields.

### Steps:

- Give the Rule name as : Customer Detail duplicate.
- Select the matching rule : Matching customer details.

The screenshot shows the 'Duplicate Rules' setup page. The rule name is 'Customer Detail duplicate'. It is set to match 'Customer Details' using the 'Matching customer details' rule. The matching criteria are '(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone\_number EXACT MatchBlank = FALSE)'. The rule was created by PANDETI NIKITHA on 7/12/2025, 10:29 PM.

The screenshot shows the 'All Duplicate Rules' setup page. It lists four standard duplicate rules: 'Customer Detail duplicate', 'Standard Account Duplicate Rule', 'Standard Contact Duplicate Rule', and 'Standard Lead Duplicate Rule'. Each rule is associated with its respective object (Customer Details, Account, Contact, Lead) and uses a standard matching rule.

## Activity – 8

### Creating Profiles

#### - Manager Profile

The screenshot shows the 'Clone Profile' dialog box. It asks for the name of the new profile ('Manager') and requires selecting an existing profile to clone from ('Standard User'). The 'Save' button is highlighted with a red arrow.

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



The screenshot shows the Salesforce Setup interface with the 'Profiles' page selected. The profile 'Sales person' is displayed, showing its details: Name (sales person), User License (Salesforce Platform), and Custom Profile (checked). The 'Profile Detail' section also lists Description, Created By (PANDETI NIKITHA, 7/12/2025, 10:46 PM), and Modified By (PANDETI NIKITHA, 7/16/2025, 6:43 AM). The 'Page Layouts' section shows assignments for Global, Lead, Email Application, Location, and Home Page Layout.

The screenshot shows the 'All Profiles' page in the Salesforce Setup interface. It displays a list of profiles, including 'Identify User', 'Manager', 'Marketing User', 'Minimum Access - API Only Integrations', 'Minimum Access - Salesforce', 'Partner App Subscription User', 'Partner Community Login User', 'Partner Community User', 'ReadOnly', 'sales person', and 'Salesforce API Only System Integrations'. The 'User License' column shows various options like Identify, Salesforce, Salesforce Integration, etc., and the 'Custom' column indicates which profiles are custom profiles.

## Activity – 9

### Roles & Role Hierarchy

#### Manager Role:

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

-Give Label as “Manager” and Role name gets auto populated.

#### Another Role:

-Click plus on CEO role, and click add role under manager.

-Give Label as “sales person” and Role name gets auto populated.

## Activity – 10

### Users

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.

### Create Users:

Fill in the fields :

1. First Name : Niklaus
2. Last Name : Mikaelson
3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: [text@text.text](#)
6. Nick Name : Give a Nickname
7. Role : Manager
8. User licence : Salesforce
9. Profiles : Manager

New User

User Edit Save Save & New Cancel

**General Information**

First Name	Niklaus
Last Name	Mikaelson
Alias	nmika
Email	
Username	Mikaelson@Niklaus
Nickname	nik
Title	
Company	
Department	
Division	

**Role** Manager User License Salesforce Profile Manager Active ✓

Marketing User   
 Offline User   
 Knowledge User   
 Flow User   
 Service Cloud User   
 Site.com Contributor User   
 Site.com Publisher User   
 WDC User   
 Data.com User Type —None—

## Create Another Users:

All Users

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>   Edit	Carrie_Nick	ncarr	nickcarrie@gmail.com	sales person	✓	sales person
<input type="checkbox"/>   Edit	Chatter_Expert	Chatter	chatty_00dgk000006jyvnuag_cb6kr0kpplb@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/>   Edit	EPIC_OrgFarm	QEPIC	epic_2536c0c9849e@orgfarm.salesforce.com		✓	System Administrator
<input type="checkbox"/>   Edit	Michael_John	imich	john@agent.com	sales person	✓	sales person
<input type="checkbox"/>   Edit	Mikaelson_Niklaus	nmika	mikaelson@agent.com	Manager	✓	Manager
<input type="checkbox"/>   Edit	NIKITHA_PANDETI	var	varmanikitha595123@agentforce.com		✓	System Administrator
<input type="checkbox"/>   Edit	User_Integration	integ	integration@00dgk000006jyvnuag.com		✓	Analytics Cloud Integration User
<input type="checkbox"/>   Edit	User_Security	sec	insightssecurity@00dgk000006jyvnuag.com		✓	Analytics Cloud Security User

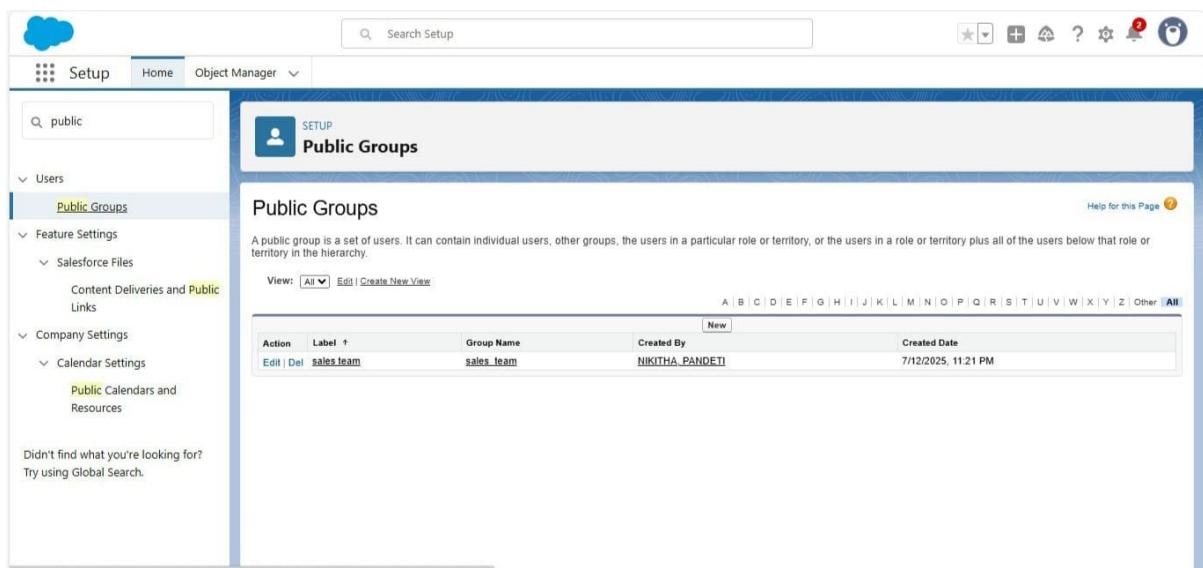
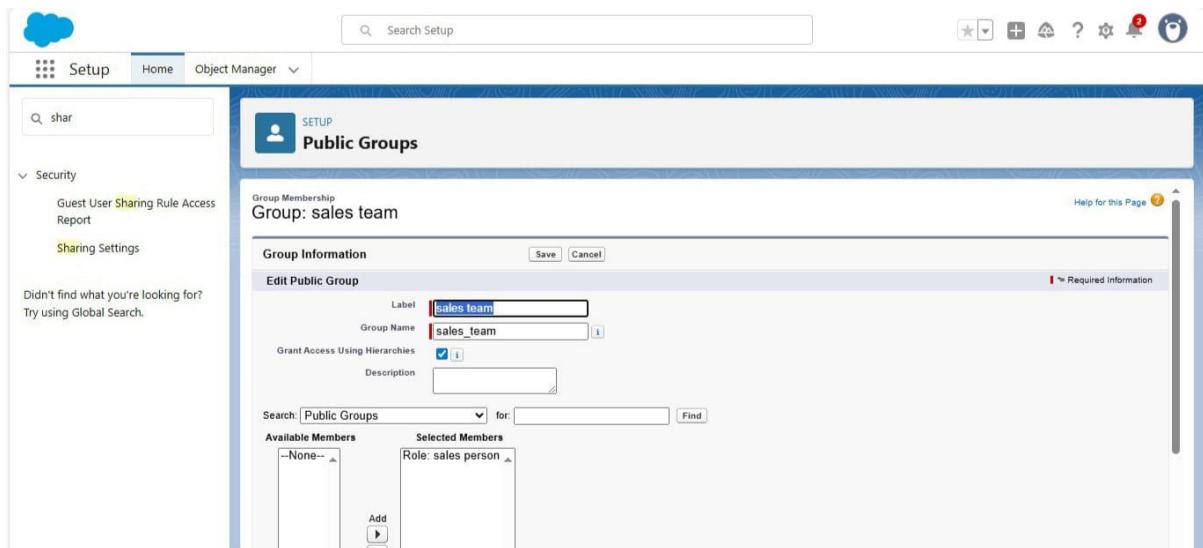
## Activity – 11

### Public Groups

- **Creating New Public Groups**
  - a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings.

#### Steps:

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.



## Activity – 12

**Sharing Settings:** control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy.

Steps:

-Change the OWD setting of the Service records Object to private as shown in fig.

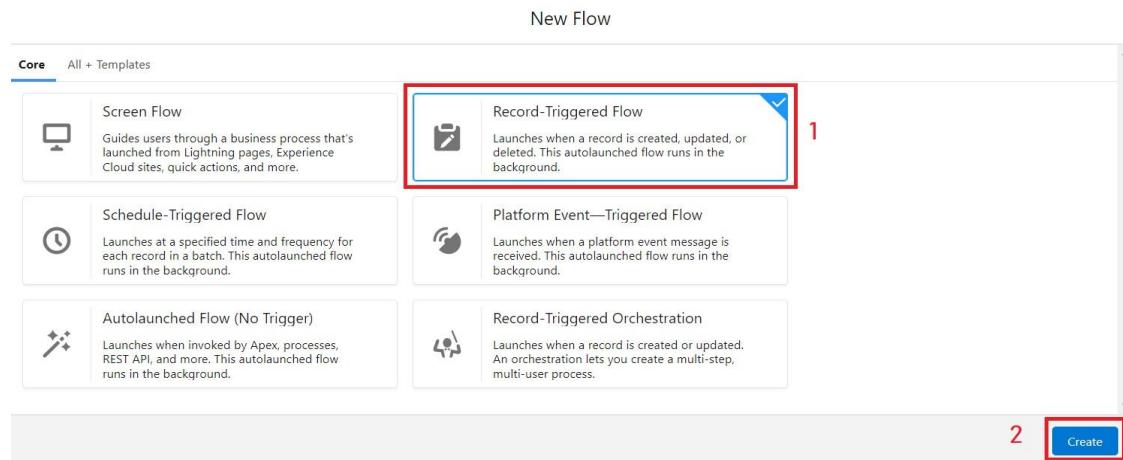
- Scroll down a bit, Click new on Service records sharing Rules

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.

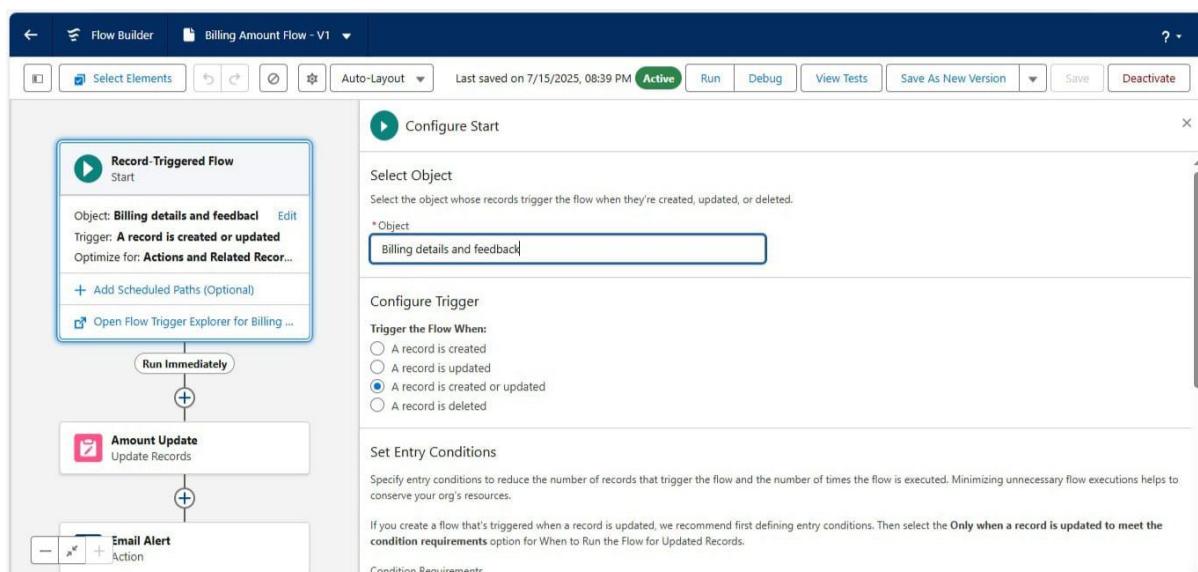
## Activity – 13

**Flows:** a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps.

- **Record-Trigger Flow** Select the Record-triggered flow and Click on Create.



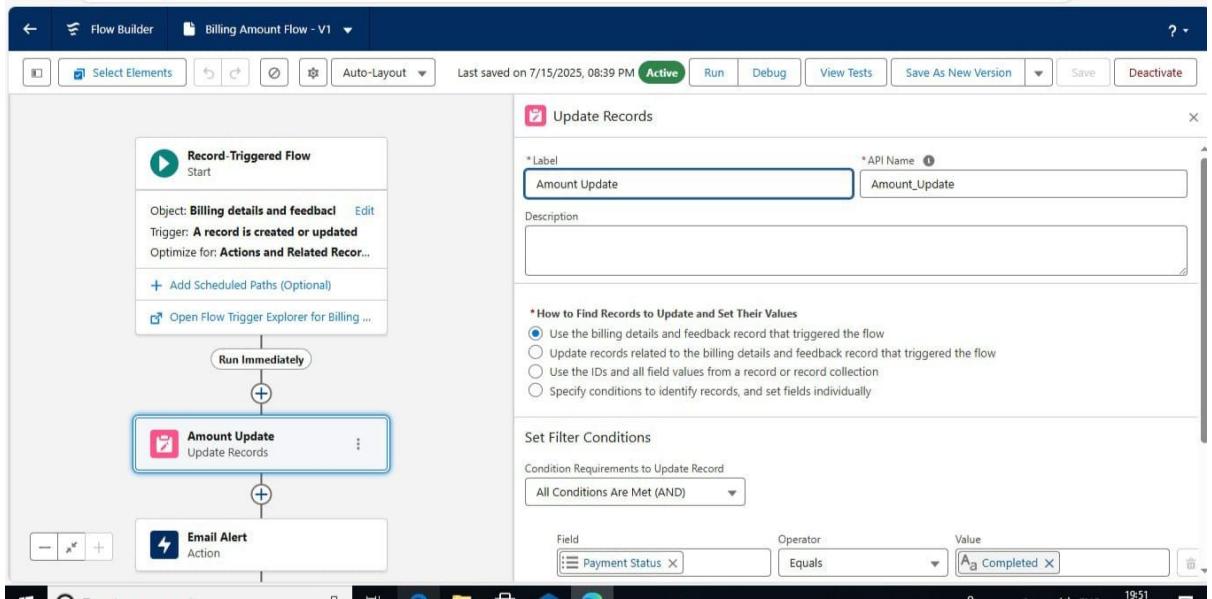
- Select the Object as "Billing details and feedback" in the Drop down list.
- Select the Trigger Flow when: "A record is Created or Updated".
- Select the Optimize the flow for: "Actions and Related Records" and Click on Done.



- **Amount Update Flow**

- Field : Payment\_Paid\_\_c
- Value :
 

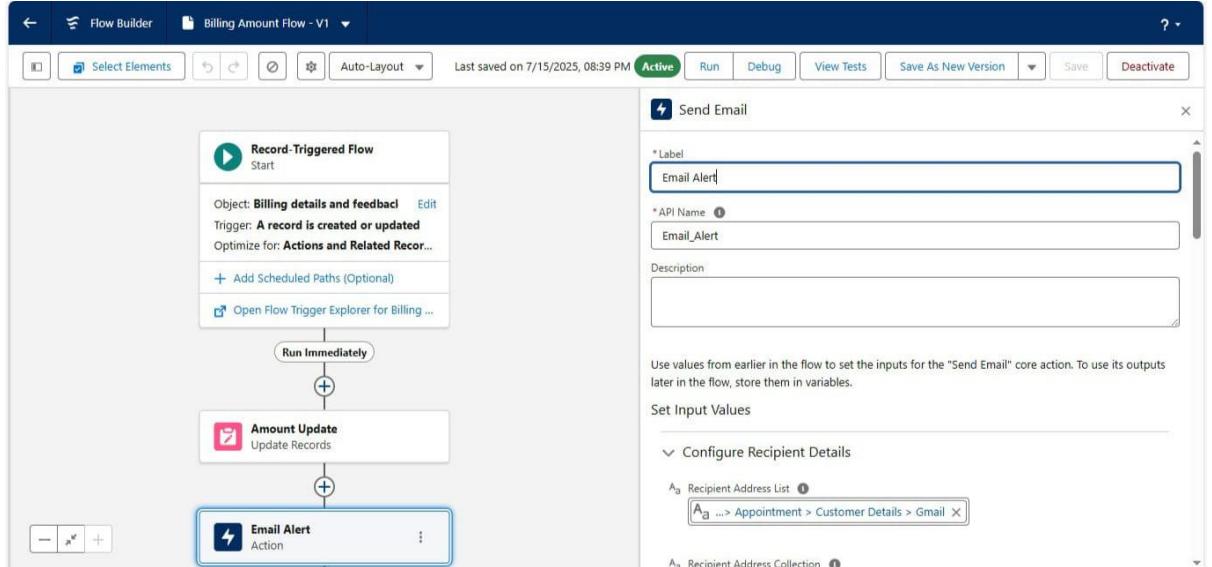
```
{!$Record.Service_records__r.Appointment__r.Service_Amount__c}
```



## - Email Alert Flow

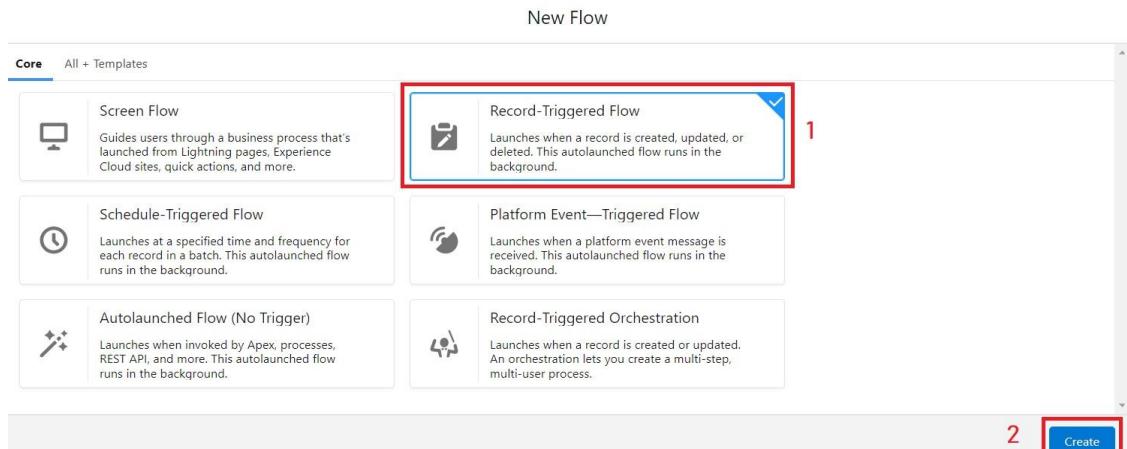
- RecipientAddressList:

```
{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail
__c}
```

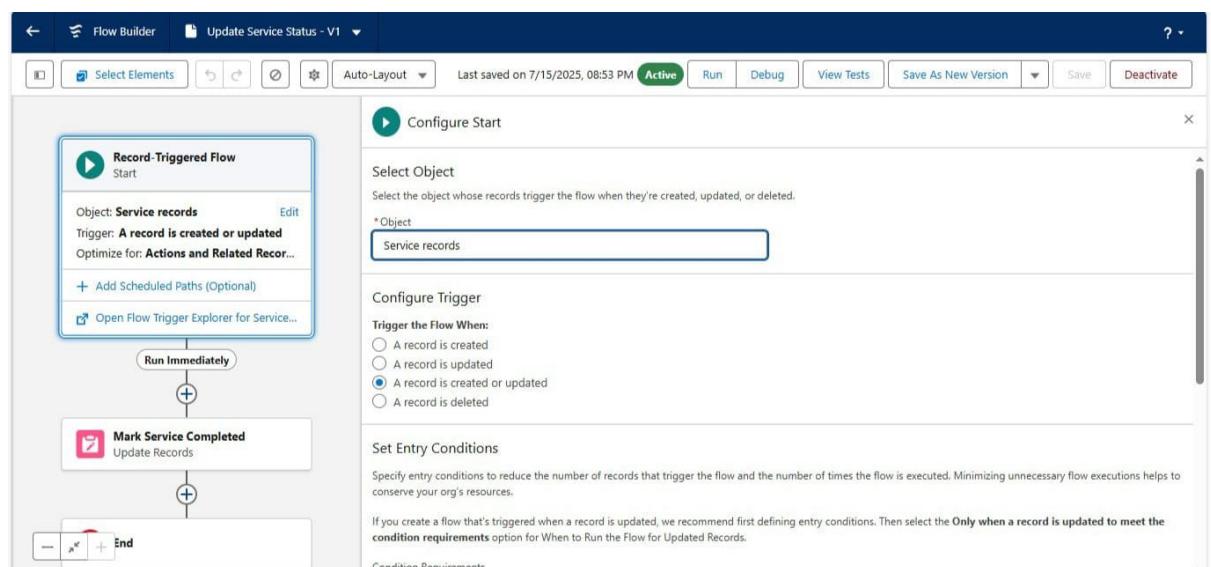


## Create Another Flow:

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

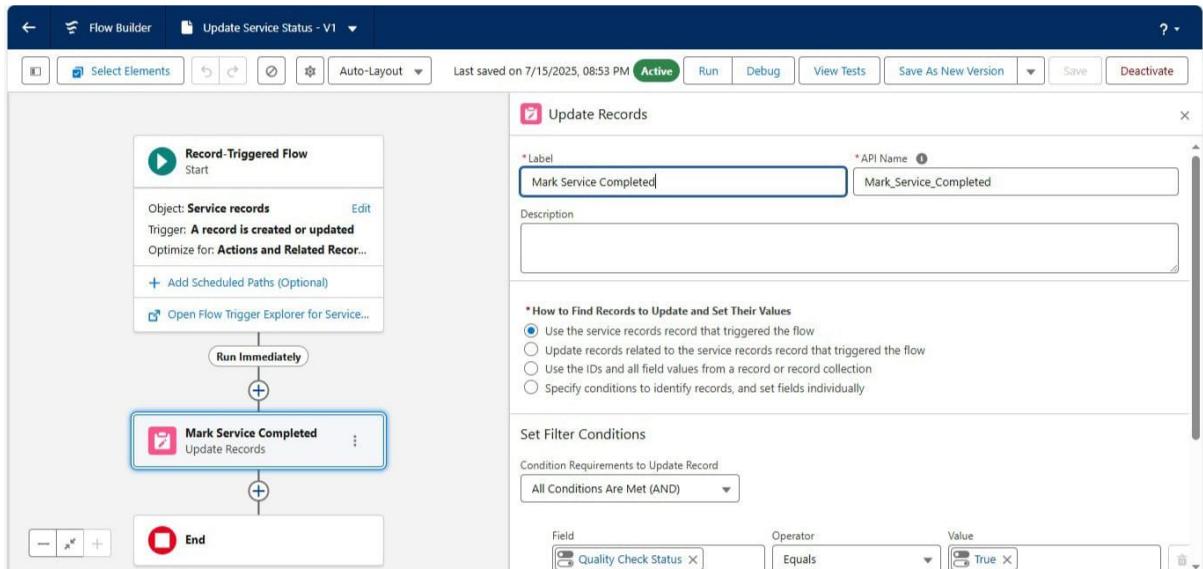


- 2.
3. Select the Object as "Service records" in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimise the flow for: "Actions and Related Records"



Set a filter condition : All Conditions are met(AND)

1. Field : Quality\_Check\_Status\_\_c
2. Operator : Equals
3. Value : True
4. And Set Field Values for the Billing details and feedback Record
5. Field : Service\_Status\_\_c
6. Value : Completed



## Activity – 14

### Triggers

1. Name the class as “AmountDistributionHandler”.

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 :

1. Name : AmountDistribution
2. sObject : Appointment\_\_c

Code:

```

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

if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
    AmountDistributionHandler.amountDist(trigger.new);

}

}

```

## - AmountDistributionHandler.apxc (Class)

The screenshot shows the Salesforce Developer Console interface with the URL [https://orgfarm-861d05e13d-dev-ed.develop.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://orgfarm-861d05e13d-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The tab bar shows "AmountDistributionHandler.apxc" is active. The code editor displays the following Apex class:

```
1 public class AmountDistributionHandler {
2     public static void amountDist(list<Appointment__c> listApp){
3         list<Service_Records__c> serList = new list<Service_Records__c>();
4         for(Appointment__c app : listApp){
5             if(app.Maintenance_Service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
6                 app.Service_Amount__c = 10000;
7             }
8             else if(app.Maintenance_Service__c == true && app.Repairs__c == true){
9                 app.Service_Amount__c = 5000;
10            }
11            else if(app.Maintenance_Service__c == true && app.Replacement_Parts__c == true){
12                app.Service_Amount__c = 8000;
13            }
14            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
15                app.Service_Amount__c = 7000;
16            }
17            else if(app.Maintenance_Service__c == true){
18                app.Service_Amount__c = 2000;
19            }
20            else if(app.Repairs__c == true){
21
22
23
24
25
26
27
28
29
2
30 }
```

The screenshot shows the continuation of the Salesforce Developer Console interface with the same URL and active tab "AmountDistributionHandler.apxc". The code editor displays the remaining part of the class:

```
10 }
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30 }
```

## - AmountDistribution.apxt (Trigger)

The screenshot shows the Salesforce Developer Console in Microsoft Edge. The URL is https://orgfarm-861d05e13d-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab is titled 'AmountDistributionHandler.apxc' and the file is 'AmountDistribution.apxt'. The code is:

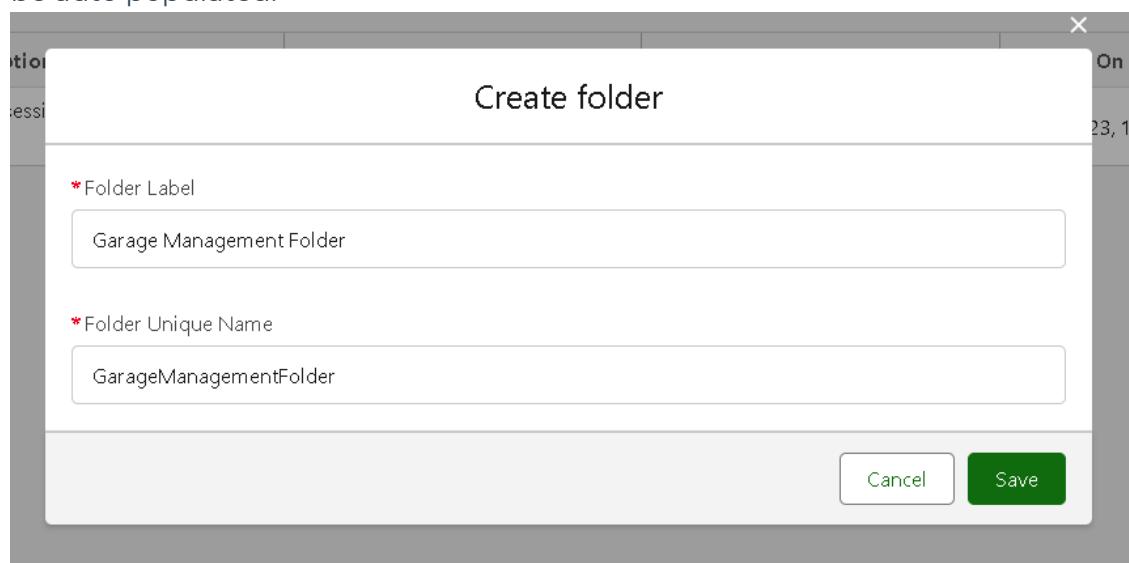
```
trigger AmountDistribution on Appointment__c (before insert, before update) {
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);
    }
}
```

The console interface includes a toolbar with File, Edit, Debug, Test, Workspace, Help, and various dropdown menus. Below the code editor is a logs panel with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The logs tab is selected.

## Activity – 15

### Reports

- **Report Folder Creation**
- Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.



	Name	Created By	Created On	Last Modified By	Last Modified Date
Recent	Einstein Bot Reports	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Created by Me	Einstein Bot Reports Spring '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Private Reports	Einstein Bot Reports Summer '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Public Reports	Einstein Bot Reports Summer '22	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
All Reports	Einstein Bot Reports Winter '23	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
FOLDERS	Enablement Dashboard Reports Spring '24	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
All Folders	Enablement Dashboard Reports Summer '24	Automated Process	7/3/2025, 12:12 PM	Automated Process	7/3/2025, 12:12 PM
Created by Me	Garage Management Folder	PANDETI NIKITHA	7/16/2025, 6:48 AM	PANDETI NIKITHA	7/16/2025, 6:48 AM

	Name	Description	Folder	Created By	Created On	Subscribed
Recent	New Service information Report		Garage Management Folder	PANDETI NIKITHA	7/17/2025, 7:24 AM	

## - Sharing a folder:

- Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.

These sharing settings apply to all subfolders in this folder.

Share With

Roles

Names: managed

Access: View

**Share**

Who Can Access

PANDETI NIKITHA  
Users

**Done**

## - Report Types:

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.

**Report Type Focus**

Specify what type of records (rows) will be the focus of reports generated by this report type.

Example: If reporting on "Contacts with Opportunities with Partners," select "Contacts" as the primary object.

**Identification**

Primary Object	Customer Details	<input type="text"/>	
Report Type Label	Service information	<input type="text"/>	
Report Type Name	Service_Information	<input type="text"/>	
Description	Service information	<input type="text"/>	
Store in Category	Other Reports	<input type="text"/>	

**Deployment**

A report type with deployed status is available for use in the report wizard. While in development, report types are visible only to authorized administrators and their delegates.

Deployment Status	<input type="radio"/> In Development	<input checked="" type="radio"/> Deployed	
-------------------	--------------------------------------	-------------------------------------------	--

**Next** **Cancel**

- 7.
8. now , Click on Related object box.
9. Click on Select Object, choose Appointment Object as shown in fig.

**New Custom Report Type**  
Service information

**Step 2 of 2**

Help for this Page

**Step 2. Define Report Records Set**

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

**A Customer Details**

**Primary Object**

**B**  **Select Object**

- Select Object--
- Activities
- Appointments**
- Duplicate Record Items

at one related "B" record.  
related "B" records.

**Diagram:** A Venn diagram with two overlapping circles labeled A and B. Below it is a diagram showing a blue circle (A) and an orange circle (B) with horizontal bars representing records.

**Previous** **Save** **Cancel**

- 10.

## Step 2. Define Report Records Set

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

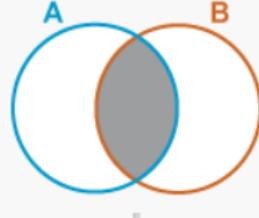
**A Customer Details**  
Primary Object

**B Appointments**

**A to B Relationship:**

- Each "A" record must have at least one related "B" record.
- "A" records may or may not have related "B" records.

(Click to relate another object)



A	B

11.

12. Again Click to relate another object.

13. And select the related object as " service records".

14. Repeat the process and select the related object as " Billing details and feedback".

**A Customer Details**  
Primary Object

**B Appointments**  
**A to B Relationship:**

- Each "A" record must have at least one related "B" record.
- "A" records may or may not have related "B" records.

**C Service records**  
**B to C Relationship:**

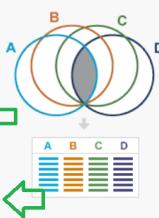
- Each "B" record must have at least one related "C" record.
- "B" records may or may not have related "C" records.

**D Billing details and feedback**

**C to D Relationship:**

- Each "C" record must have at least one related "D" record.
- "C" records may or may not have related "D" records.

Object Limit Reached  
You can associate up to four objects to a custom report type.



A	B	C	D

Previous Save Cancel

15.

- **New Service Information Report**
- In column section.
  - o Customer name
  - o Appointment Date
  - o Service Status
  - o Payment paid
- In GROUP ROWS section.
  - o Rating for Service
- In GROUP COLUMN section.
  - o Payment Status
- Click on Add Chart , Select the Line Chart.

Rating for service	Sum of Payment Paid	Record Count	Total
4	\$20,000	2	\$20,000
5	\$10,000	1	\$10,000
<b>Total</b>	<b>Sum of Payment Paid</b>	<b>Record Count</b>	<b>\$30,000</b>

Customer Name	Appointment Date	Service Status	Payment Paid
NickCarrie	7/16/2025	Completed	\$10,000
Mikaelson	7/15/2025	Completed	\$10,000
PANDETI NIKITHA	7/15/2025	Completed	\$10,000
			\$30,000

## Activity – 16

**Dashboard:** Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports.

## Create Dashboard Folder:

- The folder label as “Service Rating dashboard”.

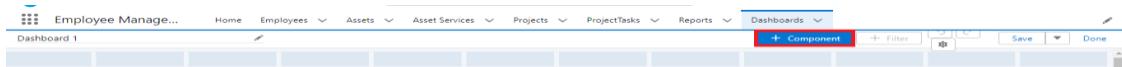
The screenshot shows the 'Garage Management' application interface. The top navigation bar includes links for Customer Details, Appointments, Service records, Billing details and feedback, Reports, and Dashboards. The 'Dashboards' tab is active. On the left, a sidebar menu lists categories like DASHBOARDS (Recent, Created by Me, Private Dashboards, All Dashboards), FOLDERS (All Folders, Created by Me, Shared with Me), and FAVORITES (All Favorites). The main content area displays a table with columns: Dashboard Name, Description, Folder, Created By, Created On, and Subscribed. A single row is shown: 'Customer review' (Description: 'Service Rating dashboard'), created by PANDETI NIKITHA on 7/17/2025, 7:51 AM.

## Create Dashboard:

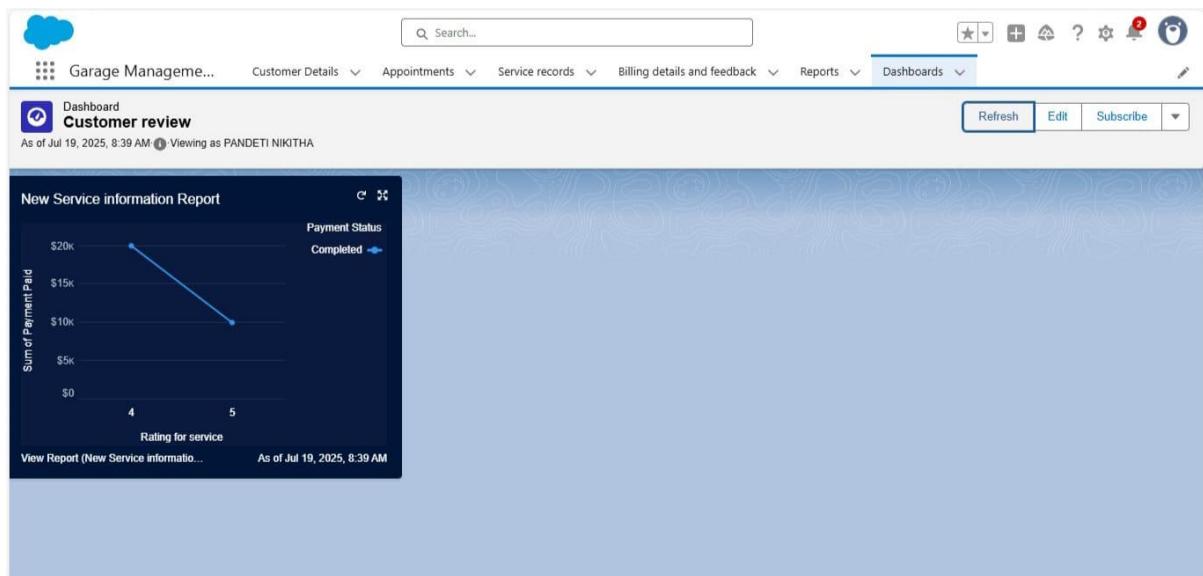
1. Give a Name and select the folder that created, and click on create.

The screenshot shows a 'New Dashboard' dialog box. It has fields for 'Name' (containing 'Customer review') and 'Folder' (containing 'Service Rating'). There are 'Description' and 'Select Folder' buttons below the folder field. At the bottom are 'Cancel' and 'Create' buttons, with 'Create' being the one highlighted.

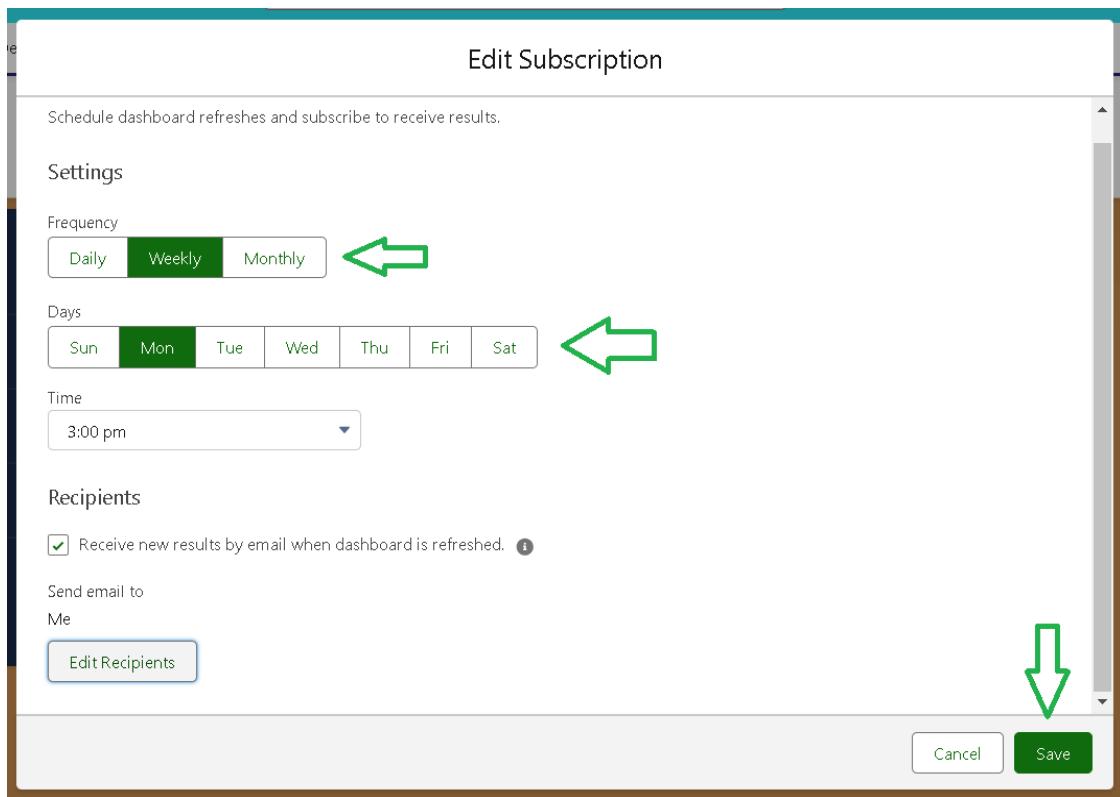
3. Select add component.



4. Select a Report and click on select.
5. Select the Line Chart. Change the theme.



6. Set the Frequency as " weekly ".
7. Set a day as monday.



## Activity – 17

### User Adoption:-

#### -Creating Records

##### - Contact Details:

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.

Garage Management

Recently Viewed

Customer Details

\* Customer Name: Mikaelson

Owner: PANDETI NIKITHA

Phone number: 09764234567

Gmail: mikael@gmail.com

Name: Mikaelson

Address: London

Created By: PANDETI NIKITHA / 7/17/2025, 8:04 AM

Cancel Save & New Save

### - Appointment Details:

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.

Appointments

Recently Viewed

Customer Details

Appointment Name: app-006

Owner: PANDETI NIKITHA

Appointment Date: 7/15/2025

Maintenance service:

Repairs:

Service Amount: \$10,000

Vehicle number plate: AP12AB1234

Cancel Save & New Save

### - Service Record Details:

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

Garage Management

Service records

Recently Viewed

3 items • Updated a few seconds ago

Service records Name  
1  ser-006  
2  ser-005  
3  ser-004

Edit ser-006

\* = Required Information

Service records Name: ser-006

Owner: PANDETI NIKITHA

\*Appointment: app-006

Quality Check Status:

Service Status: Completed

service date: 7/17/2025

This field is calculated upon save

Created By: PANDETI NIKITHA, 7/17/2025, 8:05 AM

Last Modified By: PANDETI NIKITHA, 7/17/2025, 8:05 AM

Cancel Save & New Save

### - Billing&Feedback Details:

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

Garage Management

Billing details and feedback

Recently Viewed

3 items • Updated a few seconds ago

Billing details and feedback  
1  bill-003  
2  bill-002  
3  bill-001

Edit bill-003

\* = Required Information

Billing details and feedback Name: bill-003

Owner: PANDETI NIKITHA

Service records: ser-006

Payment Paid: \$10,000

\*Rating for service: 4

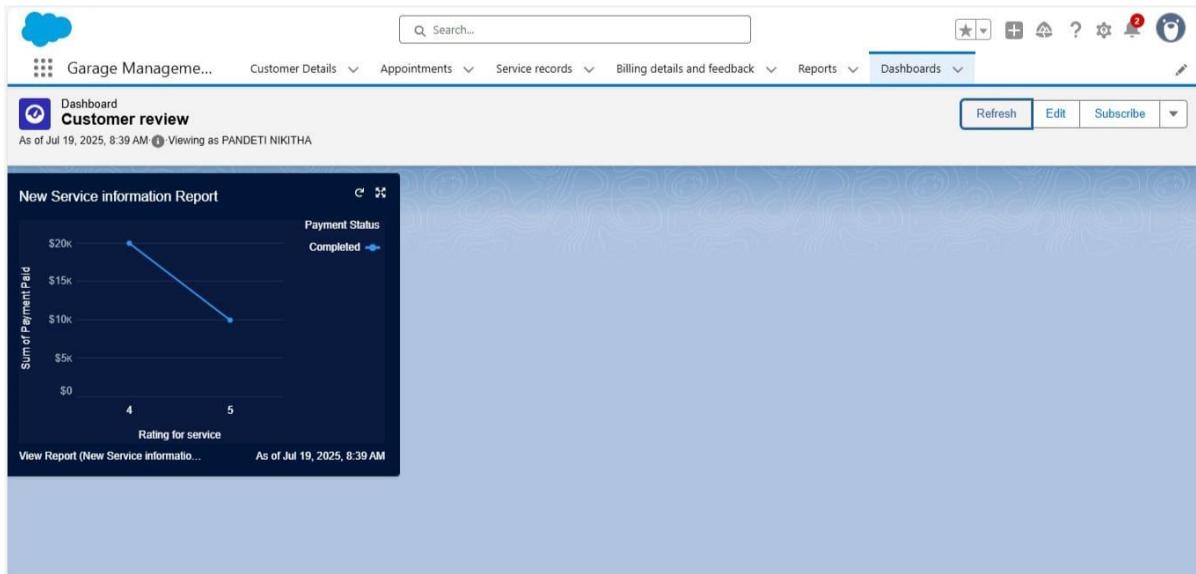
Payment Status: Completed

Created By: PANDETI NIKITHA, 7/17/2025, 8:05 AM

Last Modified By: PANDETI NIKITHA, 7/17/2025, 8:05 AM

Cancel Save & New Save

Final Output:



# CONCEPTS UTILIZED

(FOR OUR PROJECT IMPLEMENTATION)

- **What is SALESFORCE?**

Salesforce is like a digital hub that helps businesses stay connected with their customers. Imagine having a central place where a company can keep track of every conversation, email, and sale, so no detail slips through the cracks. It's built to bring together sales, customer support, and marketing efforts, so everyone is on the same page.

With Salesforce, businesses can customize tools to fit their needs, automate repetitive tasks, and use built-in AI to get insights on what customers want. It's essentially a smart assistant for companies to build stronger, more personalized customer relationships.

- **Objects**

Objects are database tables that permit you to store data that is specific to an organization.

## Types:-

1. Standard Objects
  - Provided by salesforce
2. Custom Objects
  - Created by users as per their requirements and flexibilities.

### - **Tabs**

Tabs are navigation links that give users quick access to different objects, features, or data, like accounts, contacts, or dashboards, within the platform.

### - **Lightning App**

A Lightning App in Salesforce is a tool that lets users build easy-to-use, interactive apps to streamline work and improve how they interact with data.

### - **Fields**

Fields in Salesforce are like labelled blanks in a form where you fill in specific details like a person's name, phone number, or Gmail, so that all relevant information is organized and easy to find.

### - **Validation & Duplication Rules**

- Validation rules in Salesforce are conditions that ensure data entered meets specific criteria before being saved, helping maintain data quality and accuracy.
- Duplication rules in Salesforce are settings that help prevent or manage duplicate records by identifying and blocking or alerting users about potential duplicates during data entry.

### - **Profiles**

Profiles define what a user can see and do within the platform, acting like "permission sets" for different roles.

- **Role & Role Hierarchy**

Roles define a user's level of access to data and functions, creating a hierarchy that helps organizations manage who can see and edit information, ensuring everyone has the right permissions while still promoting collaboration across teams.

- **Users**

Users are people who can log in to the platform to manage customer information, track sales, and work with their teammates, helping them do their jobs better.

- **Public Groups**

Public groups in Salesforce are like team huddles that bring users together, making it easier to share information and collaborate on projects seamlessly.

- **Sharing Settings**

Sharing settings determine how and with whom you share data within your organization, ensuring the right people have access to the right information to collaborate effectively.

- **Flows**

Flows are powerful tools that allow users to automate complex business processes by visually designing step-by-step workflows that guide users or update records without needing to write code.

- **Apex Triggers**

APEX Triggers are custom pieces of code that automatically execute before or after specific events occur in the database,

helping businesses automate tasks and enforce rules without needing manual input.

- **Reports**

Reports are tools that help businesses visualize and analyse their data, making it easier to understand trends and track performance at a glance.

- **Dashboard**

Dashboards are visual displays that summarize key metrics and data, helping businesses quickly see their performance and make informed decisions at a glance.

## TEST AND VALIDATION

- **Approach to Testing:**

For the Garage Management System, our testing strategy ensures everything works smoothly and meets user needs. We'll combine both automated and manual testing methods, including:

1. Functional Testing
2. Integration Testing
3. User Acceptance Testing (UAT)
4. Performance Testing

- **Unit Testing (Apex Classes & Triggers):**

Unit testing will focus on individual components, like Apex classes and triggers, to ensure they function correctly. Key elements include:

1. **Positive Tests:** To confirm expected functionality under normal conditions.
2. **Negative Tests:** To check how the system handles errors or unexpected inputs.
3. **Bulk Tests:** To ensure the system can handle large amounts of data without issues.

## CONCLUSION

In conclusion, the Salesforce Garage Management System is a powerful tool that makes running a service centre easier and more efficient. It combines features like online appointment scheduling, real-time updates on car repairs, and smart inventory management, all in one place. This means customers get faster service and clear communication, which helps keep them happy. For the staff, it reduces confusion and makes their jobs smoother. Plus, the system gives managers valuable insights into performance, allowing them to spot trends and improve the business. Overall, this system helps garages provide excellent service and build strong relationships with their customers.

Thank You