NAAN MUDHAVALVAN- SALESFORCE REPORT

GARAGE MANAGEMENT SYSTEM

(GMS)

PROJECT CREATED BY

BE –V SEMESTER

M GOPINATH	-	421622104038
A ARAVIND KUMAR	-	421622104011
A ARAVIND	-	421622104008
R JAYAKUMAR	-	421622104054

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAILAM ENGINEERING COLLEGE
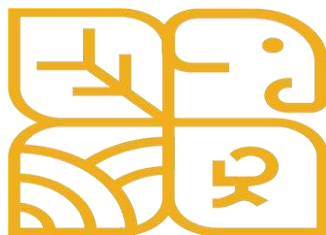
TABLE OF CONTENTS:

1.PROJECT OVERVIEW:

The Garage Management System (GMS) Salesforce project is designed to streamline the management and operation of an automotive repair service center. It enables seamless tracking of customer vehicles, service requests, inventory, billing, appointments, and customer feedback. By leveraging Salesforce's cloud platform, this system provides a unified, scalable solution for managing garage operations while offering real-time insights into business performance. The primary goal is to automate and optimize the service processes, improve customer experience, and increase operational efficiency.

2.OBJECTIVES:

The main objectives of the Garage Management System Salesforce implementation project are:

1. **Enhancing Customer Relationship Management (CRM):**
    a. **Goal:** Improve customer satisfaction and retention by automating and personalizing interactions.
    b. **Description:** Utilize Salesforce's CRM tools to track customer data, service histories, and communication preferences. Automating appointment reminders, follow-ups, and service recommendations to enhance the overall customer experience.

2. **Streamlining Service Operations and Workflow Automation:**
    a. **Goal:** Automate the garage's internal processes to improve operational efficiency.
    b. **Description:** Design automated workflows for handling service requests, assigning tasks to technicians, tracking service progress, and ensuring timely job completion. This includes reducing manual data entry and optimizing scheduling.

3. **Optimizing Inventory Management:**

   a. **Goal:** Improve the management of spare parts and tools to avoid shortages or overstocking.

   b. **Description:** Use Salesforce to track the availability of spare parts, tools, and consumables, and implement automated alerts for low stocklevels.

4. **Improving Billing and Invoicing Accuracy:**

   a. **Goal:** Automate the billing process to ensure accurate and timely invoicing for services rendered.

   b. **Description:** Implement automated billing workflows to generate invoices based on predefined pricing models, labor rates, and parts used. This will minimize errors and reduce manual intervention.

5. **Providing Real-Time Business Insights:**

   a. **Goal:** Enable data-driven decision-making through real-time reports and dashboards.

   b. **Description:** Leverage Salesforce's powerful reporting and analytics tools to track key performance indicators (KPIs) such as service completion time, revenue, customer satisfaction, parts usage, and technician performance.
   These insights will help management make informed business decisions.

6. **Enhancing Communication and Collaboration:**

   a. **Goal:** Improve communication between garage staff and customers.

   b. **Description:** Use Salesforce's case management and messaging tools to ensure that customers receive timely updates on the status of their vehicles. Collaboration features within Salesforce will also allow the service team to work more effectively on complex service orders and share information seamlessly.

7. **Scalability and Flexibility for Future Growth:**

a. **Goal:** Build a system that can scale with the business as it grows.

b. **Description:** Design the solution to be flexible and scalable, so it can easily accommodate future business requirements such as expanding the number of services offered, integrating with new systems, or increasing the number of customers and employees.

8. **Seamless Integration with Existing Systems:**

a. **Goal:** Ensure smooth integration with existing third-party tools and software used by the garage.

b. **Description:** Integrate the Salesforce platform with other business systems (e.g., vehicle diagnostic tools, third-party parts suppliers) to ensure data consistency and streamline workflows across platforms.

9. **Ensuring Compliance and Data Security:**

a. **Goal:** Adhere to industry regulations and maintain the security of customer and business data.

b. **Description:** Implement strong data governance practices and ensure that the system complies with data protection regulations (e.g., GDPR, HIPAA) while providing secure access to sensitive customer and service information.

3.SALESFORCE KEY FEATURES AND CONCEPTS UTILIZED

- **Salesforce Service Cloud:** Utilized to manage customer service processes, track cases (service requests), and schedule appointments.

- **Salesforce Objects:** Custom objects were created for managing vehicles, services, and inventory. Standard objects like Accounts, Contacts, Opportunities, and Cases were also leveraged.

- **Record Types and Page Layouts:** Custom record types were created for vehicles, service orders, and parts, with tailored page layouts for each record type to ensure efficient data entry and management.

- **Process Builder and Flow:** These tools were used to automate workflows such as service order creation, appointment reminders, and stock alerts.

- **Reports and Dashboards:** Built to track the performance of the garage, including KPIs like average service time, customer satisfaction, and parts usage.

- **Apex Programming:** Custom logic was implemented in Apex for more advanced automation needs, such as calculating labor costs or handling complex pricing rules.

- **Salesforce Mobile:** Mobile functionality was incorporated for on-the-go management of customer appointments, service updates, and inventory checks.

- **Integration with External Systems:** The system integrated with third-party services, such as vehicle diagnostic tools and suppliers for parts ordering.

4.DETAILED STEPS TO SOLUTION :

STEP 1: CREATE A OBJECT Object: objects are database tables that permit you to store data that is specific to an organization.

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.
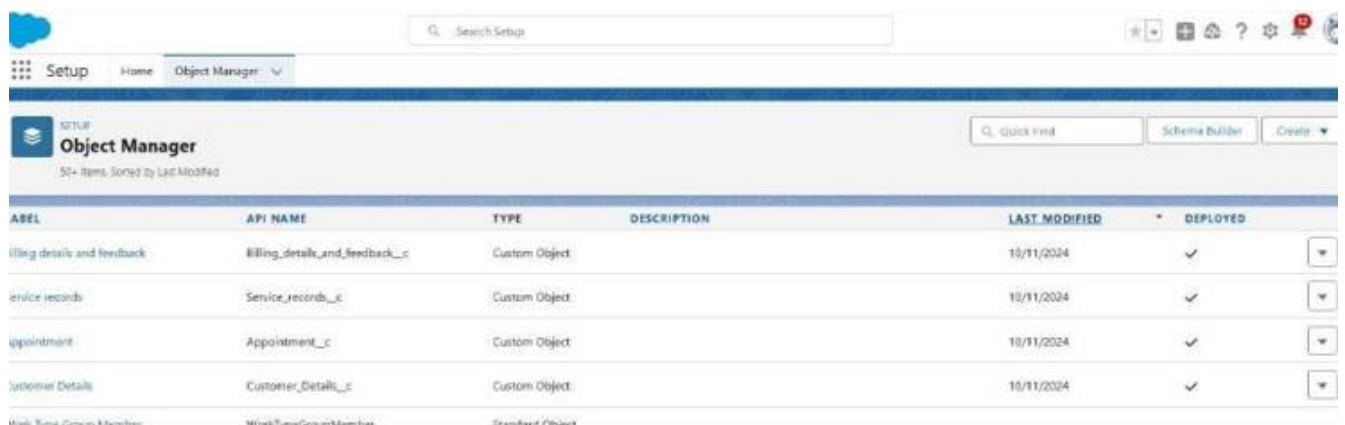
Custom Objects:

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.

Procedure To Create Custom Objects in Salesforce:

1. **Navigate to Setup**: From the setup page, go to **Object Manager**.

2. **Create Custom Object**: Click on **Create**, then **Custom Object**.

3. **Define Object Details**:

   a. **Label**: Enter the object name (e.g., "Customer Details", "Appointment", "Billing Details and Feedback").

   b. **Plural Label**: Enter the plural form (e.g., "Customer Details", "Appointments", "Billing Details and Feedback").

   c. **Record Name**: Define the record name (e.g., "Customer Name", "Appointment Name", "Billing Details Name").

      i. **Data Type**: Choose **Text** for Customer Details, **Auto Number** for others.

      ii. **Auto Number Format**: For Auto Number objects, set a display format (e.g., "app-{000}", "bill-{000}") and starting number (e.g., 1).

4. **Select Additional Options**: Check **Allow Reports**, **Track Field History**, and **Allow Search**.

5. **Save**: Click **Save** to create the object.

FIGURE 1:

Smart
Internz

SMARTBRIDGE
Let's Bridge the Gap

salesforce
PARTNER

STEP 2 : CREATE A TABS

Tab

A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1.Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Web Tabs

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

3. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

4. Lightning Component Tabs

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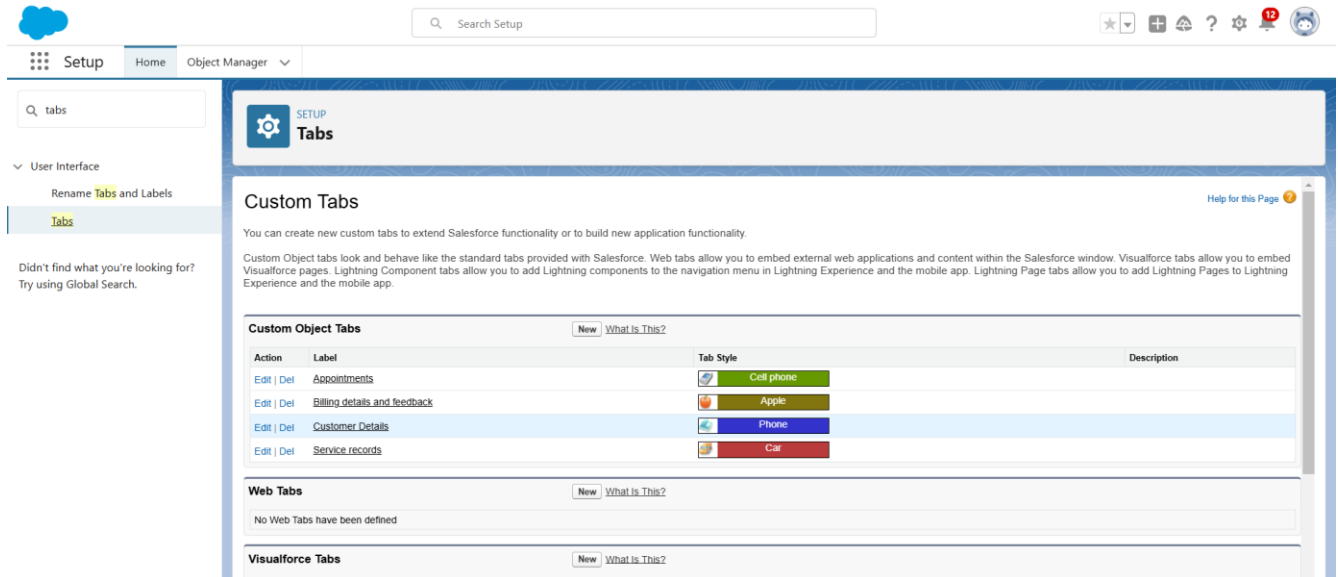
5. **Lightning Page Tabs**

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu. Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customise the tabs for your apps.

Procedure To Create a Tabs:

- Go to **Setup**, search for **Tabs**, and select **Tabs**.
- Click **New** under **Custom Object Tabs**.
- Select **Customer Details**, **Appointments, Service records,Billing details and feedback.** choose a tab style, and click **Next**.
- Keep default profile settings and click **Next**.
- Uncheck **Include Tab** in the Custom App section, check **Append tab**.
- Click **Save**.

FIGURE 2:

STEP 3 : CREATE A LIGHTNING APP Lightening 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.

Procedure To Create a Lightning App:

- Go to **Setup**, search for **App Manager**, and select it.

- Click **New Lightning App** and enter **Garage Management Application** as the app name, then click **Next**.

- Keep default settings for **App Options**, **Utility Items**, and click **Next** each time.

- In **Navigation Items**, select and add **Customer Details**, **Appointments**, **Service Records**, **Billing Details and Feedback**, **Reports and Dashboards**.

- For **User Profiles**, search for **System Administrator**, add it, and click **Save & Finish**.

FIGURE 3 :
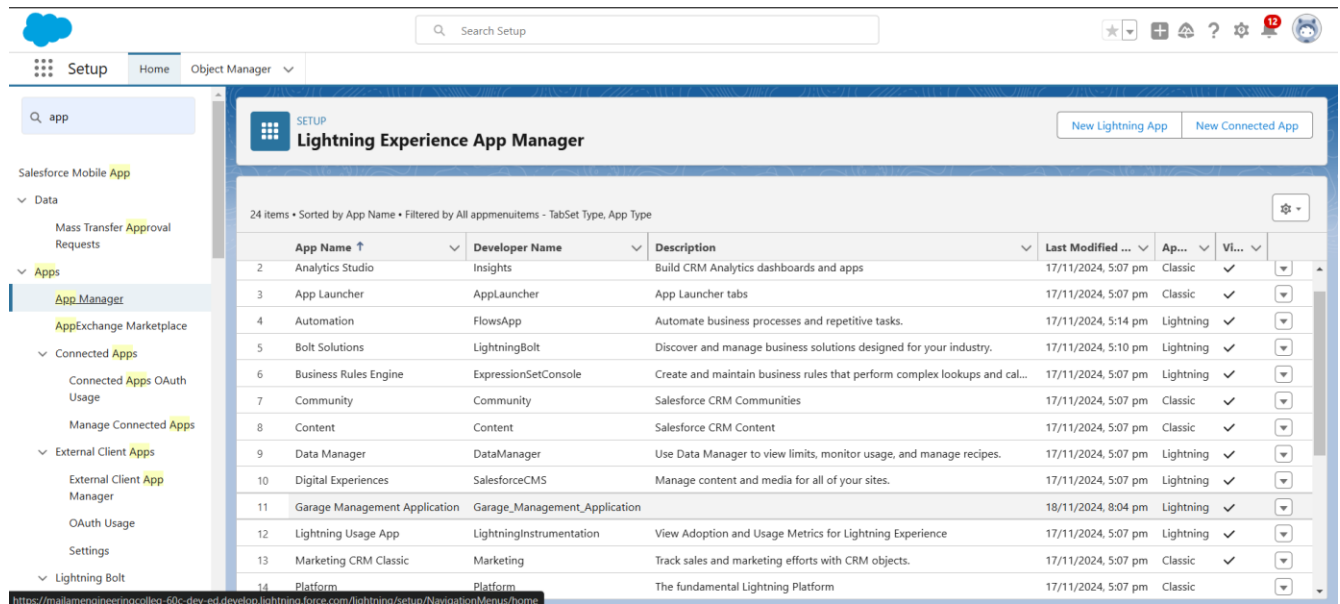
STEP 4: CREATE A FIELDS

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 nonrequired 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.

Procedure to Create a Fields:

1. **Create Phone Field (Customer Details)**:
   a. Go to **Fields & Relationships** > **New** > Select **Phone** > Label: Phone Number > **Save & New**.
2. **Create Email Field (Customer Details)**:
   a. Go to **Fields & Relationships** > **New** > Select **Email** > Label: Gmail > **Save & New**.

3. **Create Lookup (Appointment)**:

    a. Go to **Fields & Relationships** > **New** > Select **Lookup** > Choose **Customer Details** > **Save**.

4. **Create Lookup (Service Records)**:

    a. Go to **Fields & Relationships** > **New** > Select **Lookup** > Choose **Appointment** > **Save**.

5. **Create Lookup (Billing Details)**:

    a. Go to **Fields & Relationships** > **New** > Select **Lookup** > Choose **Service Records** > **Save & New**.

6. **Create Checkbox Fields (Appointment)**:

    a. Go to **Fields & Relationships** > **New** > Select **Checkbox** > Label: Maintenance Service, Repairs, Replacement Parts > **Save**.

7. **Create Checkbox Field (Service Records)**:

    a. Go to **Fields & Relationships** > **New** > Select **Checkbox** > Label: Quality Check Status > **Save**.

8. **Create Date Field (Appointment)**:

    a. Go to **Fields & Relationships** > **New** > Select **Date** > Label: Appointment Date > **Save**.

9. **Create Currency Field (Appointment)**:

    a. Go to **Fields & Relationships** > **New** > Select **Currency** > Label: Service Amount > **Save**.

10. **Create Currency Field (Billing Details)**:

    a. Go to **Fields & Relationships** > **New** > Select **Currency** > Label: Payment Paid > **Save**.

11. **Create Text Field (Appointment)**:

    b. Go to **Fields & Relationships** > **New** > Select **Text** > Label: Vehicle Number Plate > **Save**.
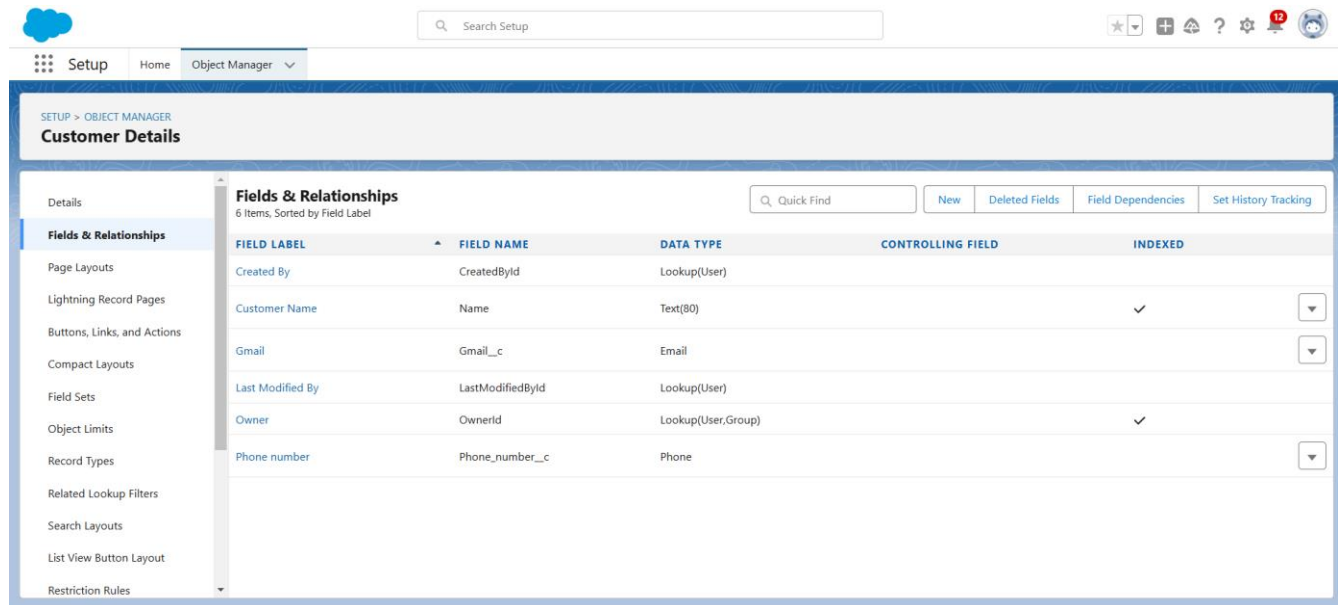
12.Create Picklist (Service Records/Billing Details):

    c.   Go to **Fields & Relationships** > **New** > Select **Picklist** > Add values > **Save**.

13.Create Formula Field (Service Records):

    d.   Go to **Fields & Relationships** > **New** > Select **Formula** > Label: Service Date > **Save**.

FIGURE 4:

Smart
Internz

SMARTBRIDGE
Let's Bridge the Gap

salesforce
PARTNER

STEP 5: CREATE A VALIDATION RULE

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.

Procedure To Create a Validation Rule :

1. **Create Validation Rule (Appointment)**:

    a. Go to **Object Manager** > **Appointment** > **Validation Rules** > **New**.

    b. Rule: "Vehicle"

    c. Formula: NOT(REGEX(Vehicle_number_plate__c, "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))

    d. Error: "Please enter valid number"

    e. **Save**.

2. **Create Validation Rule (Service Records)**:

    a. Go to **Object Manager** > **Service Records** > **Validation Rules** > **New**.

    b. Rule: "service_status_note"

    c. Formula: NOT(ISPICKVAL(Service_Status__c, "Completed"))

    d. Error: "Still it is pending"

    e. **Save**.

3. **Create Validation Rule (Billing Details)**:

    a. Go to **Object Manager** > **Billing Details** > **Validation Rules** > **New**.

b. Rule: "rating_should_be_less_than_5"

c. Formula: `NOT(REGEX(Rating_for_service__c, "[1-5]{1}"))`

d. Error: "Rating should be from 1 to 5"

e. **Save**.

FIGURE 5:

FIGURE 6:

SETUP > OBJECT MANAGER
Service records

Validation Rules
1 Items, Sorted by Rule Name

New

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY	
service_status_note	Service Status	still it is pending	✓	M GOPINATH, 19/11/2024, 7:43 pm	▾

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules

FIGURE 7:

SETUP > OBJECT MANAGER
Billing details and feedback

Validation Rules
1 Items, Sorted by Rule Name

New

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	✓	M GOPINATH, 19/11/2024, 7:44 pm	▾

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules

STEP 6: CREATING A DUPLICATE RULES <u>Procedure To</u> <u>Create</u>

<u>a Duplicate Rules:</u>

1.Create Matching Rule:

- Go to **Setup** > Search **Matching Rules** > Click **New Rule**.

- Select **Customer Details** > **Next**.

- Rule Name: "Matching customer details".

- Add matching criteria: ○ **Gmail**: Exact ○ **Phone Number**: Exact

- **Save**.

 2.Create Duplicate Rule:

- Go to **Setup** > Search **Duplicate Rules** > Click **New Rule**.

- Select **Customer Details** > **Rule Name**: "Customer Detail Duplicate".

- In **Matching Rule Section**, select **Matching customer details**.

- **Save** and **Activate**.

FIGURE 8:

FIGURE 9:

STEP 7: CREATING A PROFILES **What is profile**

in salesforce?

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.

Types of profiles in salesforce:

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.

Custom Profiles:

Custom ones defined by us.

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

Procedure To Create a Profiles:

1. **Create Profile (Manager)**:

    a. Go to **Setup** > Search **Profiles** > Click **Profiles** > Clone **Standard User**.

    b. Name the profile **Manager** > **Save**.

c. Click **Edit**, set **Garage Management** as default app.

d. In **Custom Object Permissions**, grant access for **Appointments**, **Billing Details**, **Service Records**, and **Customer Details**.

e. Set **Session Timeout** to **8 hours** and password policies as specified. f. **Save**.

2. **Create Profile (Sales Person)**:

a. Go to **Setup** > Search **Profiles** > Click **Profiles** > Clone **Salesforce Platform User**.

b. Name the profile **Sales Person** > **Save**.

c. Click **Edit**, set **Garage Management** as default app.

d. In **Custom Object Permissions**, grant access for **Appointments**, **Billing Details**, **Service Records**, and **Customer Details**.

e. **Save**.

FIGURE 10:
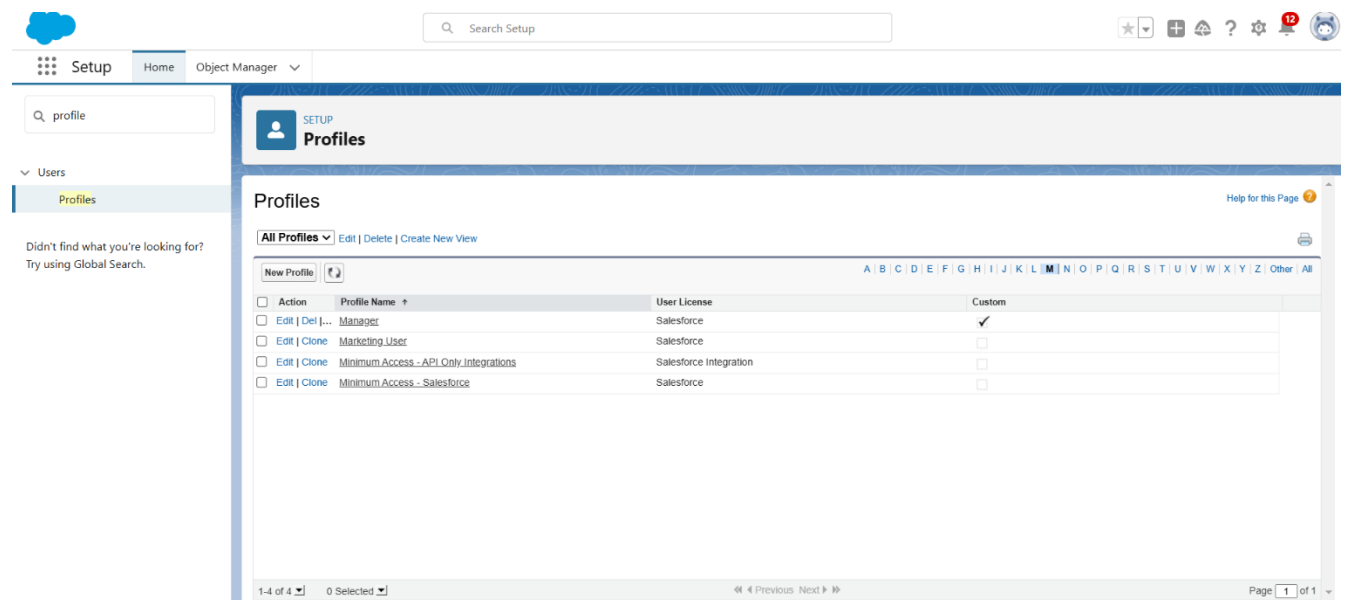
STEP 8: CREATE A ROLE AND ROLE HIERARCHY

Role and 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.

Procedure To Create a Role and Role Hierarchy:

1. **Create Manager Role**:

    a. Go to **Setup** > Search **Roles** > Click **Setup Roles**.

    b. Click **Expand All** > **Add Role** under Manager.

    c. Label: "Manager" (Role Name auto-populates) > **Save**.

2. **Create Sales Person Role under Manager**:

    a. Go to **Setup** > Search **Roles** > Click **Setup Roles**.

    b. Click **+** on CEO role > **Add Role** under Manager.

    c. Label: "Sales Person" (Role Name auto-populates) > **Save**.

FIGURE11:   **11:**

Setup | Home | Object Manager

Q roles

SETUP
Roles

Role
Manager

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: Mailam Engineering College » CEO » Manager
Siblings: SVP, Sales & Marketing, SVP, Customer Service & Support, CFO, SVP, Human Resources, COO

Users in Manager Role [1]

Role Detail

Label	Manager	Role Name	Manager
This role reports to	CEO	Role Name as displayed on reports	
Modified By	M GOPINATH, 19/11/2024, 8:07 pm	Sharing Groups	Role, Role and Internal Subordinates
Opportunity Access	Users in this role can **edit** all opportunities associated with accounts that they own, regardless of who owns the opportunities		
Case Access	Users in this role can **edit** all cases associated with accounts that they own, regardless of who owns the cases		

Users in Manager Role   Assign Users to Role | New User   Users in Manager Role Help

Action	Full Name	Alias	Username	Active
Edit	Niklaus Mikaelson	nmika	gopi@2004.com	✔

FIGURE 12:

Setup | Home | Object Manager

Q roles

SETUP
Roles

Role
sales person

Below is the list of users assigned to this role. Click Edit to modify the role name. Click Assign Users to Role to assign existing users to this role. Click New User to create a user for this role.

Hierarchy: Mailam Engineering College » CEO » Manager » sales person

Users in sales person Role [3]

Role Detail

Label	sales person	Role Name	sales_person
This role reports to	Manager	Role Name as displayed on reports	
Modified By	M GOPINATH, 19/11/2024, 8:07 pm	Sharing Groups	Role, Role and Internal Subordinates
Opportunity Access	Users in this role can **edit** all opportunities associated with accounts that they own, regardless of who owns the opportunities		
Case Access	Users in this role can **edit** all cases associated with accounts that they own, regardless of who owns the cases		

Users in sales person Role   Assign Users to Role | New User   Users in sales person Role Help

Action	Full Name	Alias	Username	Active
Edit	aravind kumar	akuma	aravind@anandhan.com	✔
Edit	Jaya kumar	jkumar	jai@2004.com	✔
Edit	low budget	aravind	low@budget.com	✔

STEP 9: CREATE A USERS

Users in salesforce

Smart
Internz

SMARTBRIDGE
Let's Bridge the Gap

salesforce
PARTNER

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.
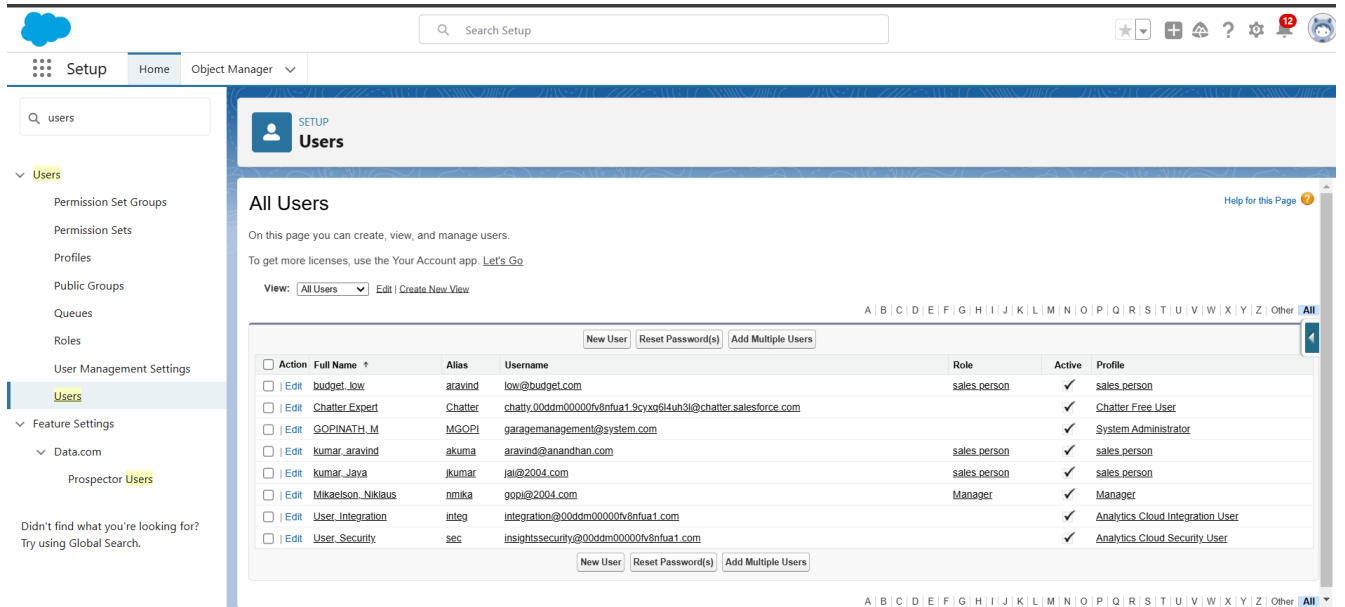
Procedure To Create a Users :

1.Create User (Manager):

- Go to **Setup** > Search **Users** > Click **Users** > **New User**.

- Fill in fields:

    o First Name: Niklaus o Last Name: Mikaelson o Alias:

    (your alias) o Email: (your personal email) o Username:

    text@text.text o

    Nickname: (your nickname) o Role:

    Manager o User License: Salesforce

    o Profile: Manager

- **Save**.

2.Create User (Sales Person):

- Repeat the steps, but use:

    o Role: Sales Person o User

    License: Salesforce Platform o

    Profile: Sales Person

- **Save**.

FIGURE 13:

STEP 10: CREATE A PUBLIC GROUPS

Public Groups in salesforce

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.

Procedure To Create a Public Groups:

- Go to **Setup** > Search **Users** > Select **Public Groups** > **New**.

- Label: "Sales Team" (Group Name auto-populates).

- Search for **Roles** > Select **Sales Person** > Add to **Selected Members**.

- **Save**.

FIGURE 14:

STEP 11: CREATE A SHARING SETTINGS

Sharing Settings in salesforce

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.

Procedure To Create a Sharing Settings:

1. **Set OWD for Service Records**:

    a. Go to **Setup** > Search **Users** > Select **Sharing Settings** > **Edit**.

    b. Set **OWD** for **Service Records** to **Private** > **Save** and **Refresh**.

2. **Create Sharing Rule**:

    a. Scroll down > Click **New** under **Service Records Sharing Rules**.

    b. Label: "Sharing Setting" (Rule Name auto-populates).

    c. In **Step 3**, select **Roles** > **Sales Person**.

    d. In **Step 4**, share with **Roles** > **Manager**.

    e. In **Step 5**, set **Access Level** to **Read/Write**.

    f. **Save**.

FIGURE 15:
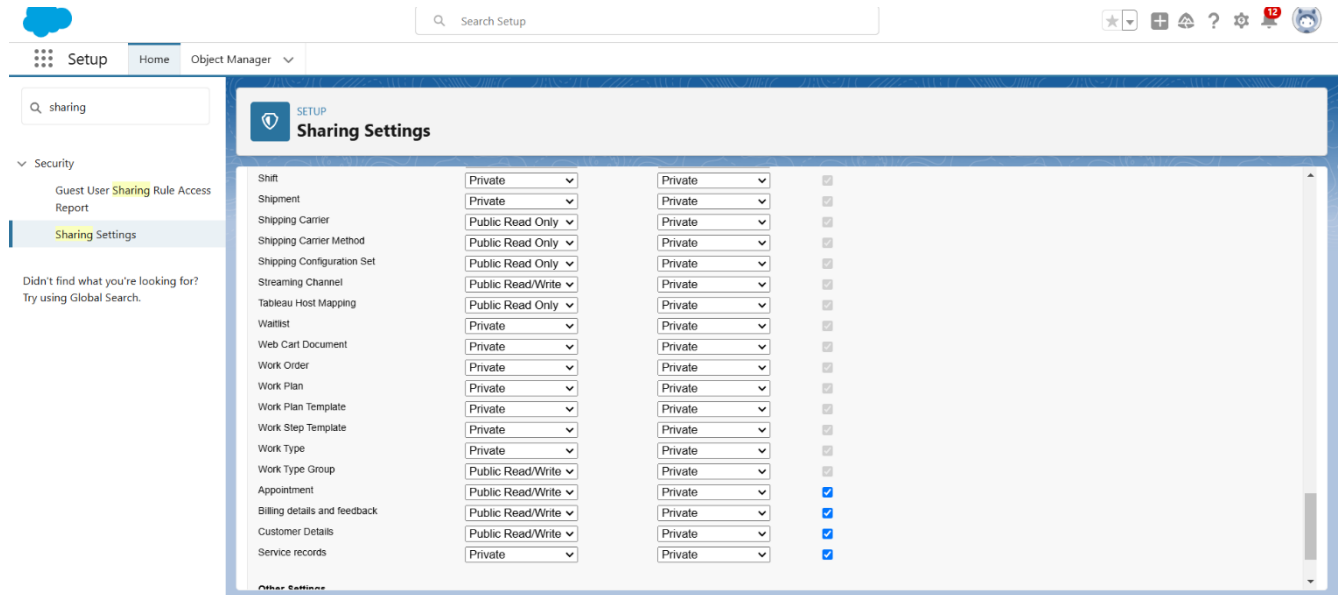
STEP 12 : CREATE A FLOWS

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.

Procedure To Create a Flows:

1. **Create a Record-Triggered Flow**:

    a. Go to **Setup** > Search **Flow** > Click **Flows** > **New Flow**.

    b. Select **Record-Triggered Flow** > **Create**.

    c. Select **Object**: "Billing details and feedback".

    d. Set **Trigger Flow When**: "A record is Created or Updated".

    e. Optimize for: "Actions and Related Records" > **Done**.

2. **Add Update Records Element**:

   a. Click **+** > Select **Update Records**.

   b. Label: "Amount Update" (API name auto-populates).

   c. Set filter condition: **Payment_Status__c = Completed**.

   d. Set field values: **Payment_Paid__c =**

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

   Done.

3. **Create Resource (Text Template)**:

   a. Click **New Resource** > Select **Variable** > **Text Template**.

   b. API Name: "alert" > Change view to **Plain Text**.

   c. Body: Paste the provided message with dynamic fields.

   d. **Done**.

4. **Add Action (Send Email)**:

   a. Click **Add Element** > **Action** > Search "send email" > Select it.

   b. Label: "Email Alert" (API name auto-populates).

   c. Select the **Text Template**: {!alert}.

   d. Recipient: **Gmail** field from the record.

   e. Subject: "Thank You for Your Payment - Garage Management".

   f. **Done**.

5. **Save and Activate Flow**:

   a. **Save** the flow, give a label, and API name will auto-populate.

   b. **Activate**.

FIGURE 16:
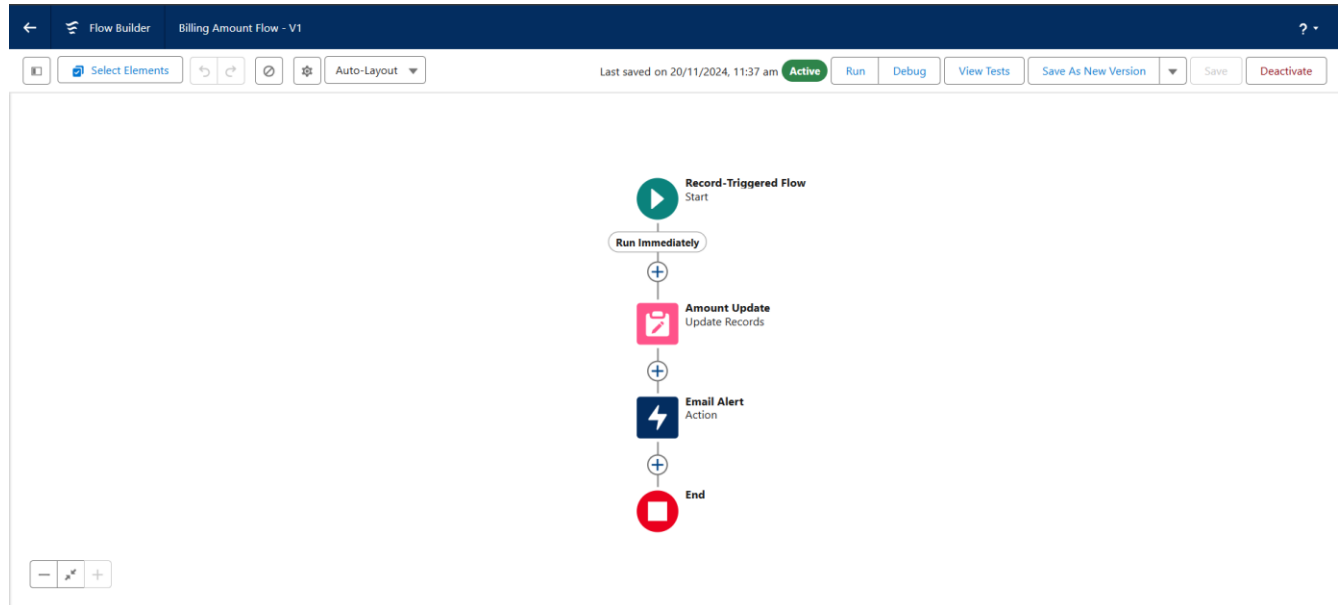
STEP 13: CREATE A APEX TRIGGER

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

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.

Procedure To Create a Apex Trigger:

- Click **Developer Console** to open a new window.

- In the **toolbar**, click **File** > **New** > **Apex Class**.

- Name the class **AmountDistributionHandler**.

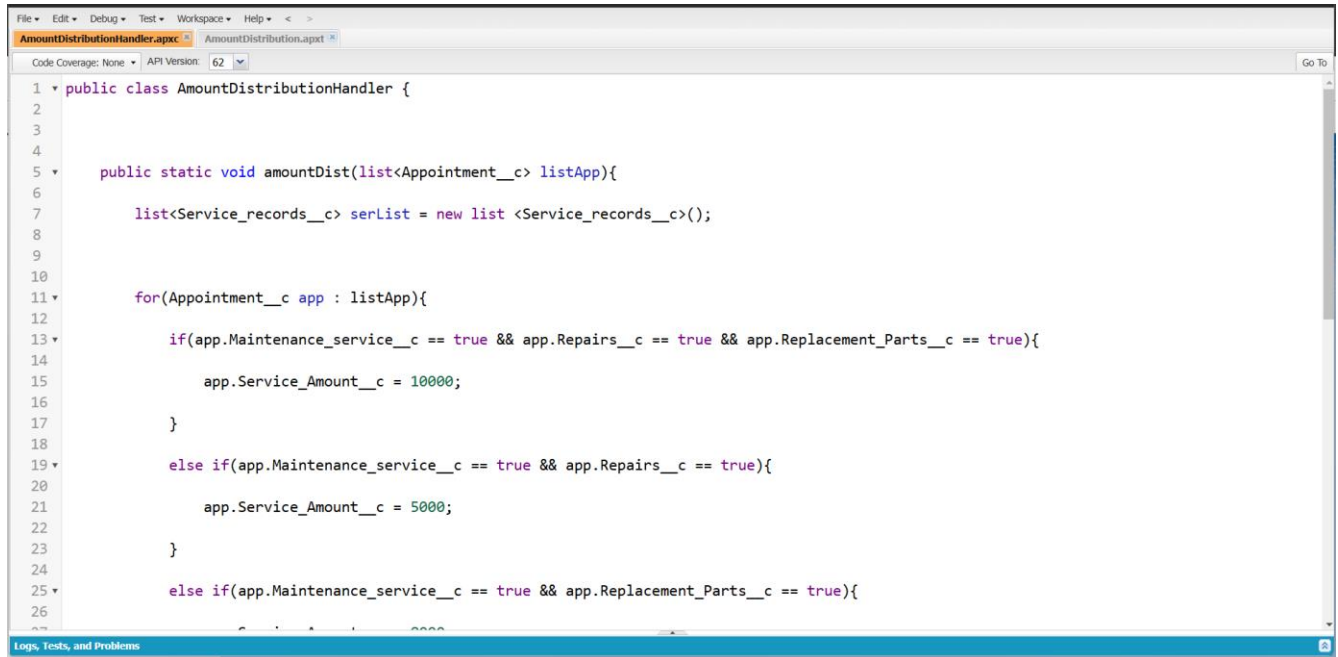- Name the trigger **AmountDistribution** and select the object   **Appointment__c**.

Creating a apex class:

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;
        }

    }
    }
}
```
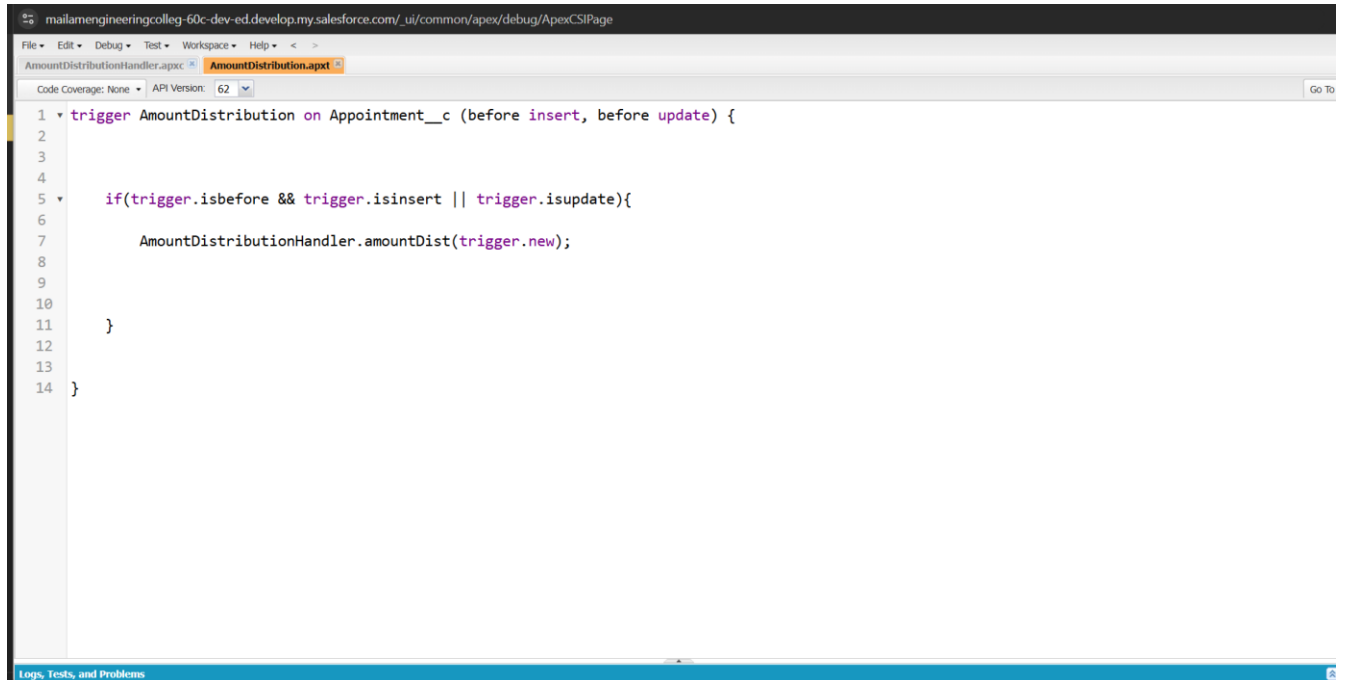
FIGURE 17:

Creating a Apex trirgger:

Code:

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

FIGURE 18:

```
mailamengineeringcolleg-60c-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File ▾   Edit ▾   Debug ▾   Test ▾   Workspace ▾   Help ▾   <   >
AmountDistributionHandler.apxc ✕    AmountDistribution.apxt ✕
Code Coverage: None ▾   API Version: 62 ▾                                            Go To

 1 ▾ trigger AmountDistribution on Appointment__c (before insert, before update) {
 2
 3
 4
 5 ▾     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
 6
 7          AmountDistributionHandler.amountDist(trigger.new);
 8
 9
10
11      }
12
13
14 }

Logs, Tests, and Problems                                                            ▲
```

STEP 14: CREATE A REPORT

Report in salesforce

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce :

- Tabular
- Summary
- MatriX

Procedure To Create a Report:

Create Report Folder:

1. Click the **App Launcher**, search for **Reports**.
2. Click the **Reports Tab**, then **New Folder**.

3. Name the folder **Garage Management Folder** and click **Save**.

Share Report Folder with Manager Role:

1. **Go to the Reports Tab, click on the Garage Management Folder dropdown, and   click Share.**

2. Share with **Roles**, search for **Manager**, and select **View** access.

3. Click **Share**, then **Done**.

Create Custom Report Type:

1. Go to **Setup**, search for **Report Types** and click **Continue**.

2. Click **New Custom Report Type**.

3. Select **Customer Details** as the **Primary Object** and name the report **Service Information**.

4. Select **Other Reports** for the category, and **Deployed** for deployment status.

5. Add **Appointment**, **Service Records**, and **Billing Details and Feedback** as related objects.

6. Click **Save**.

Create Service Information Report:

1. Go to **Reports Tab**, click **New Report**, and select **Service Information**.

2. Select **Customer Name**, **Appointment Date**, **Service Status**, **Payment Paid** for the **Columns**.

3. Add **Rating for Service** and **Payment Status** to **Group Rows**.

4. Add a **Line Chart** and click **Save**.

5. Name the report **New Service Information Report**, save it in the **Garage Management Folder**.

FIGURE 19:
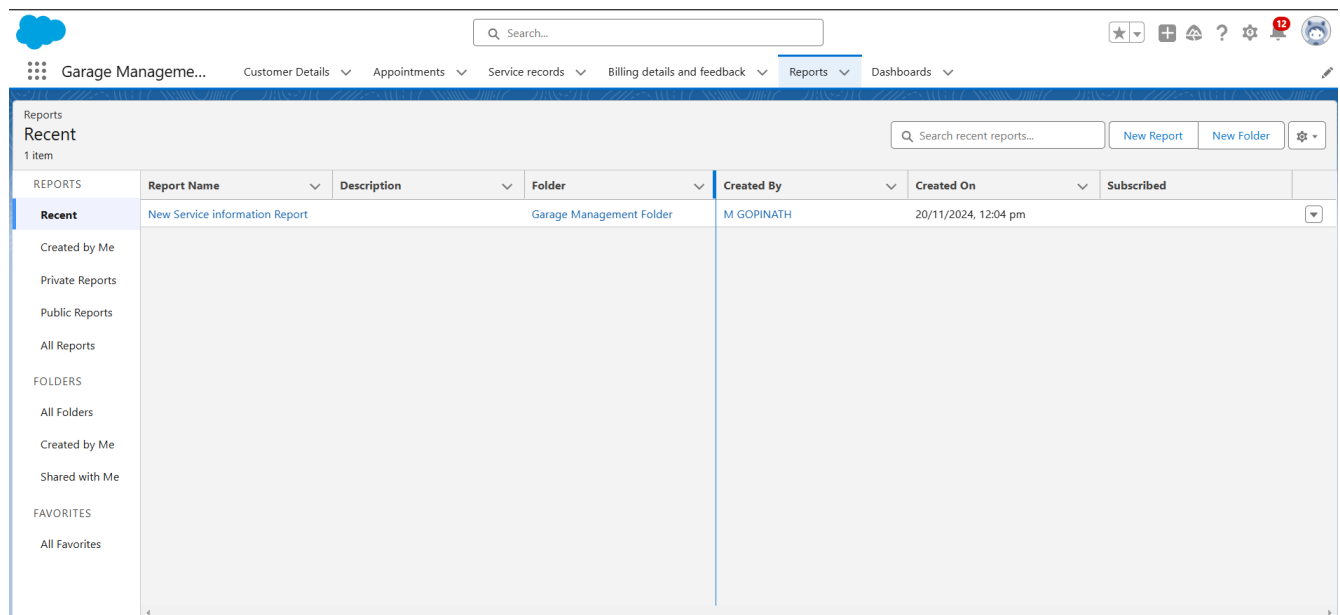
STEP 15: CREATE A DASHBOARDS

Dashboards in salesforce

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.

Procedure To Create A Dashboards:

Create Dashboard Folder:

1. Click the **App Launcher** and search for **Dashboards**.

2. Go to the **Dashboards Tab**, click **New Folder**.

3. Name the folder **Service Rating Dashboard** and click **Save**.

4. **Share the Dashboard Folder:**

5. Follow the same steps as in Milestone 15, Activity 2 to share the **Service Rating Dashboard** folder.

Create Dashboard:

1. Go to the **Dashboards Tab**, click **New Dashboard**.

2. Name the dashboard, select the **Service Rating Dashboard** folder, and click **Create**.

Add Report to Dashboard:

1. Click **Add Component**, select the **Report**, and choose the **Line Chart**.

2. Customize the theme, click **Add**, then **Save** and **Done**.

Subscribe to Dashboard:

1. Click **Subscribe** at the top right.

2. Set frequency to **Weekly** and day to **Monday**, then click **Save**.

FIGURE 20:

5.TESTING AND VALIDATION:

Testing and validation played a critical role in ensuring the Garage Management System met business requirements. The following testing methods were applied:

- **Unit Testing:** Validated individual components, such as custom Apex classes or flows, to ensure they performed correctly in isolation.

- **System Testing:** Tested the entire system in a sandbox environment to ensure that all modules worked together as expected, including integrations with external tools.

- **User Acceptance Testing (UAT):** Conducted UAT sessions with actual garage staff to confirm that the system met their daily operational needs. This phase also provided feedback on usability.

- **Performance Testing:** Ensured that the system could handle multiple users simultaneously, with the ability to scale as the garage grows.

- **Regression Testing:** Conducted regression tests after system updates to ensure new functionality didn't break existing processes.

6.KEY SCENARIOS ADDRESSED BY SALESFORCE IN THE IMPLEMENTATION PROJECT:

Several critical business scenarios were addressed using Salesforce in this implementation:

1. **Service Request Management:**
   a. Automatically captured customer service requests and assigned them to technicians.
   b. Managed the lifecycle of each request, including service scheduling, technician updates, and customer notifications.

2. **Appointment Scheduling:**
   a. The system allowed customers to schedule appointments through an integrated portal, and automatically generated calendar events and reminders for garage staff.

3. **Inventory Tracking and Management:**

    a. Tracked the availability of spare parts and tools, with automatic alerts for low stock levels.

    b. Integrated with suppliers to automate part ordering.

4. **Customer History and Vehicle Records:**

    a. Maintained detailed records of past services, repairs, and part replacements for each vehicle.

    b. Enabled quick access to customer data for service recommendations and improving customer interaction.

5. **Billing and Invoicing:**

    a. Automated the calculation of service costs based on predefined pricing models, labor rates, and parts used.

    b. Generated detailed invoices, ensuring accuracy and consistency.

6. **Real-Time Business Insights:**

    a. Dashboards provided real-time data for key metrics such as service turnaround time, customer satisfaction, and parts consumption.

    b. Enabled managers to identify issues early and optimize processes.

7.CONCLUSION:

The Salesforce-based Garage Management System successfully transformed the garage operations by streamlining service requests, improving customer interaction, optimizing inventory management, and automating billing. The integration of Salesforce's CRM capabilities

with custom objects, automation tools, and third-party system integrations allowed for significant operational efficiencies.

Key benefits realized by the implementation include:

- Enhanced customer satisfaction through timely service and personalized interactions.

- Improved garage efficiency with automated workflows, reducing the risk of errors and manual processes.

- Real-time insights for management to make data-driven decisions and track garage performance.

- Seamless scalability to support business growth and adapt to new service offerings.

Overall, the project has positioned the garage to provide higher-quality services while reducing operational overhead, making it a success in terms of both functionality and business impact.