



# NAAN MUDHALVAN- SALESFORCE REPORT

## **A CRM Application To Manage The Services**

### **Offered by an Institution**

Department of Artificial Intelligence And Data Science

Project created by

Arunkumar V - 421622243003

Keerthi K - 421622243026

Yashwanth Kumar A - 421622243057

Pugazhendhi N - 421622243041

Hariharan S -421622243020

## **PROJECT ABSTRACT**

The Salesforce-based Garage Management System (GMS) is a sophisticated solution tailored for automotive repair facilities to elevate service quality, streamline operations, and cultivate enduring customer relationships. Leveraging Salesforce's powerful CRM capabilities, GMS offers an intuitive interface and comprehensive features, allowing garages to thrive in a competitive marketplace. The system automates critical processes such as appointment scheduling, inventory management, billing, and customer communications. By integrating these functionalities within Salesforce, GMS ensures a seamless and satisfying experience for both customers and staff. This advanced solution empowers automotive repair businesses to deliver top-notch service, optimize workflows, and maintain a competitive edge.

## **INTRODUCTION**

Running an automotive repair shop involves juggling many tasks, from scheduling appointments to managing inventory and billing customers. The Salesforce-based Garage Management System (GMS) is designed to make these tasks easier and more efficient.

This system uses Salesforce's powerful tools to help garages deliver better service and keep operations running smoothly. With GMS, repair shops can automate important tasks like booking appointments, tracking inventory, sending bills, and communicating with customers.

The user-friendly design of GMS ensures that both staff and customers have a pleasant experience. By using this system, garages can stay ahead of the competition, provide excellent service, and manage their business more effectively. GMS not only simplifies

daily operations but also provides valuable insights to help make better business decisions.

#### INDEX PAGE

TOPICS	PAGE NO
Task 1	4
Task 2	4 - 7
Task 3	8 - 11
Task 4	11 - 12
Task 5	12 - 18
Task 6	18 - 21

Task 7	21 - 22
Task 8	22 - 24
Task 9	24 - 25
Task 10	26 - 27
Task 11	27 - 28
Task 12	28 - 29
Task 13	30 - 31
Task 14	32 - 33
Task 15	33 - 35
Task 16	35 - 36

## **TASK 1**

### **1.1 Creating the Developer Account**

1) To create a developer account in Salesforce, follow these steps:

2) Go to Salesforce Developer Signup.

- Fill in the signup form with the following details:
- First name & Last name
- Email
- Role: Developer
- Company: College Name
- Country: India

- Postal Code: Pin code
- Username: A combination of your name and company (format: username@organization.com, not an actual email id)

3)Click Sign me up.

## 1.2 Activating the Account

- Check the inbox of the email you used for signup.
- Click on the verification link to activate your account (the email may take 5-10 minutes to arrive).
- Click on **Verify Account**.
- Set a password and answer a security question.
- Click on **Change Password**.
- You will be redirected to your Salesforce setup page.

## TASK 2

### Creating the Custom Objects:

This report outlines the steps to create various custom objects in Salesforce, specifically for a Garage Management System. The objects include Customer Details, Appointments, Service Records, and Billing Details and Feedback. These objects will help streamline operations and improve data management within the system.

### 2.1 Customer Details Object

- Navigation Path: From the setup page, click on Object Manager || Create || Custom Object.
- Details:
 

Label Name: Customer Details

Plural Label Name: Customer Details

Record Name: Customer Name

Data Type: Text

Options: Allow reports, Track Field History, Allow search
- Save: Click Save to create the object.

## Details

Description

API Name

Customer\_Details\_c

Custom

✓

Singular Label

Customer Details

Plural Label

Customer Details

Enable Reports

Track Activities

Track Field History

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

## 2.2 Appointment Object

- Navigation Path: From the setup page, click on Object Manager || Create || Custom Object.
- Details:
 

Label Name: Appointment

Plural Label Name: Appointments

Record Name: Appointment Name

Data Type: Auto Number

Display Format: app - {000}

Starting Number: 1

Options: Allow reports, Track Field History, Allow search

- Save: Click Save to create the object.

## Details

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](#)

## 2.3 Service Records Object

- Navigation Path: From the setup page, click on Object Manager || Create || Custom Object.

- Details:

Label Name: Service Records

Plural Label Name: Service Records

Record Name: Service Records Name

Data Type: Auto Number

Display Format: ser - {000}

Starting Number: 1

Options: Allow reports, Track Field History, Allow search

- Save: Click Save to create the object.

## Details

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

## 2.4 Billing Details and Feedback Object

- Navigation Path: From the setup page, click on Object Manager || Create || Custom Object.
- Details:
  - Label Name: Billing Details and Feedback
  - Plural Label Name: Billing Details and Feedback
  - Record Name: Billing Details and Feedback Name
  - Data Type: Auto Number
  - Display Format: bill - {000}
  - Starting Number: 1
  - Options: Allow reports, Track Field History, Allow search
- Save: Click Save to create the object.



## Details

Description

API Name

Billing\_details\_and\_feedback\_\_c

Custom

✓

Singular Label

Billing details and feedback

Plural Label

Billing details and feedbacks

Enable Reports

✓

Track Activities

Track Field History

✓

Deployment Status

Deployed

Help Settings

[Standard salesforce.com Help Window](#)

## TASK 3

### 3.1 Creating a Custom Tab

- Creating a Custom Tab for Customer Details

- Navigation Path:

Go to the setup page.

Type 'Tabs' in the Quick Find bar.

Click on 'Tabs'.

Click on 'New' under the Custom Object Tabs.

- Details:

Select Object: Customer Details

Select Tab Style: Choose a style that represents Customer Details.

- Click Next.

- Add to Profiles Page: Keep the default settings.

- Click Next.

- Add to Custom App: Uncheck 'Include Tab'.

- Ensure 'Append tab to users' existing personal customizations' is checked.

- Click Save.

### 3.2 Creating Remaining Tabs (Appointments, Service Records, Billing Details and Feedback)

- Repeat the above steps for each remaining object: Appointments, Service Records, and Billing Details and Feedback.

- Navigation Path:

Go to the setup page.

Type 'Tabs' in the Quick Find bar.

Click on 'Tabs'.

Click on 'New' under the Custom Object Tabs.

- Details for Each Object:

Select Object: Choose the respective object (Appointments, Service Records, Billing Details and Feedback).

Select Tab Style: Choose a suitable style for each object.

- Click Next.
- Add to Profiles Page: Keep the default settings.
- Click Next.
- Add to Custom App: Uncheck 'Include Tab'.
- Ensure 'Append tab to users' existing personal customizations' is checked.
- Click Save.



Edit Custom Object Tab

## Customer Details

Fill in the fields below to define the custom tab.

**Custom Tab Definition Edit**

**Custom Object Tab Information**

Tab Label	Customer Details
Object	<a href="#">Customer Details</a>
Tab Style	 Alarm clock 

(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



Edit Custom Object Tab

## Appointments

Fill in the fields below to define the custom tab.

**Custom Tab Definition Edit**

**Custom Object Tab Information**

Tab Label	Appointments
Object	<a href="#">Appointment</a>
Tab Style	 Airplane 

(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



Edit Custom Object Tab

## Service records

Fill in the fields below to define the custom tab.

**Custom Tab Definition Edit**

**Custom Object Tab Information**

Tab Label	Service records
Object	<u>Service records</u>
Tab Style	 <b>Bottle</b> 

(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 --None-- ▾

Enter a short description.

Description

Save Cancel



Edit Custom Object Tab

## Billing details and feedbacks

Fill in the fields below to define the custom tab.

**Custom Tab Definition Edit**

**Custom Object Tab Information**

Tab Label	Billing details and feedbacks
Object	<u>Billing details and feedback</u>
Tab Style	 <b>Fan</b> 

(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 --None-- ▾

Enter a short description.

Description

Save Cancel

## Task 4

### Building Lightning App

Creating a Lightning App in Salesforce for managing garage operations allows for streamlined processes and efficient management of various business aspects. This guide outlines the steps to create the Garage Management Application, including adding essential navigation items and user profiles.

#### 4.1 Creating a Lightning App

##### Steps to Create a Lightning App

- Accessing App Manager:

Begin by navigating to the setup page. In the Quick Find bar, search for 'App Manager' and select it. Click on 'New Lightning App' to start the creation process.

- App Details:

Enter the name 'Garage Management Application' in the app details. Proceed to the next steps, keeping all settings on the App Options and Utility Items pages as default.

- Adding Navigation Items:

To enhance navigation within the app, select essential items such as Customer Details, Appointments, Service Records, Billing Details and Feedback, Reports, and Dashboards from the search bar. Move these items using the arrow button to include them in the app's navigation menu.

- Assigning User Profiles:

To ensure appropriate access and functionality, search for the ‘System Administrator’ profile in the search bar. Add this profile by clicking the add button. Complete the setup by saving and finishing the configuration.

The screenshot displays the 'App Configuration' interface, divided into two main sections: 'App Details' and 'App Branding'.

**App Details:**

- \* App Name:** A text input field containing 'Garage Management Application'.
- \* Developer Name:** A text input field containing 'Garage\_Management\_Application'.
- Description:** A text input field with the placeholder text 'Enter a description...'.

**App Branding:**

- Image:** A large square area for the app icon, with an 'Upload' button (indicated by an upward arrow icon) at the bottom.
- Primary Color Hex Value:** A color selection interface showing a pink color swatch and the hex value '#E41197'.
- Org Theme Options:** A checkbox labeled 'Use the app's image and color instead of the org's custom theme', which is currently unchecked.
- App Launcher Preview:** A preview of the app launcher showing a pink square icon with the letters 'GM' and a label 'Garage Management Appli...'.

## Task 5

### Creating Fields

#### 5.1 Customer Details Object Fields

- Phone Field:

Created a phone field labeled ‘Phone Number’ to store customer contact numbers.

- Email Field:

Added an email field labeled ‘Gmail’ to capture customer email addresses.

#### 5.2 Lookup Fields

- Appointment Object:

Established a lookup relationship field to link appointments to customer details, ensuring appointments are associated with the correct customer.

- **Service Records Object:**

Created a lookup relationship field to link service records to appointments. Added a filter to ensure the appointment date is less than the created date, making it a required field with error validation.

- **Billing Details and Feedback Object:**

Added a lookup relationship field to connect billing details and feedback with service records.

### **5.3 Checkbox Fields**

- **Appointment Object:**

Added multiple checkbox fields to indicate different services, including 'Maintenance Service,' 'Repairs,' and 'Replacement Parts,' all with default values set to unchecked.

- **Service Records Object:**

Created a checkbox field labeled 'Quality Check Status' to track the quality check status of the service records.

### **5.4 Date Fields**

- **Appointment Object:**

Added a date field labeled 'Appointment Date' and marked it as required to ensure each appointment has a specified date.

### **5.5 Currency Fields**

- **Appointment Object:**

Created a currency field labeled 'Service Amount' to record the cost of services provided, with read-only access for all profiles.

- Billing Details and Feedback Object:

Added a currency field labeled 'Payment Paid' to track payments made by customers.

## 5.6 Text Fields

- Appointment Object:

Created a text field labeled 'Vehicle Number Plate' with a length of 10 characters, marked as required and unique for vehicle identification.

- Billing Details and Feedback Object:

Added a text field labeled 'Rating for Service' with a length of 1 character, required and unique to capture customer ratings.

## 5.7 Picklist Fields

- Service Records Object:

Created a picklist field labeled 'Service Status' with values 'Started' and 'Completed' to track the progress of services.

- Billing Details and Feedback Object:

Added a picklist field labeled 'Payment Status' with values 'Pending' and 'Completed' to monitor payment statuses.

## 5.8 Formula Fields

- Service Records Object:



Created a formula field labeled 'Service Date' to automatically populate with the created date of the record.

Edit Customer Details Custom Field  
Phone number

---

Custom Field Definition Edit Change Field Type Save Cancel

Field Information

Field Label  Data Type Phone

Field Name

Description

Help Text

Data Owner

Field Usage

Data Sensitivity Level

Compliance Categorization

Available		Chosen
PII		
HIPAA		
GDPR		
PCI		

General Options

Required ☐ Always require a value in this field in order to save a record

Default Value

Use formula syntax: Enclose text and picklist value API names in double quotes: ("the\_text"), include numbers without quotes: (26), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type\_\_mdt:RecordAPIName.Field\_\_c

Change Field Type Save Cancel

Edit Appointment Custom Field  
Maintenance service

Custom Field Definition Edit

Change Field TypeSaveCancel

Field Information

Field Label

Maintenance service

Field Name

Maintenance\_service

Description

Help Text

Data Owner

User

Field Usage

--None--

Data Sensitivity Level

--None--

Compliance Categorization

Available

PII  
HIPAA  
GDPR  
PCI

Chosen

General Options

Default Value

☐ Checked  
☒ Unchecked

Change Field Type

Save

Cancel

Edit Appointment Custom Field  
Appointment Date

Custom Field Definition Edit

Change Field TypeSaveCancel

Field Information

Field Label

Appointment Date

Field Name

Appointment\_Date

Description

Help Text

Data Owner

User

Field Usage

--None--

Data Sensitivity Level

--None--

Compliance Categorization

Available

PII  
HIPAA  
GDPR  
PCI

Chosen

General Options

Required

☒ Always require a value in this field in order to save a record

Default Value

Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the\_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type\_\_mdt.RecordAPIName.Field\_\_c

Change Field Type

Save

Cancel

Edit Appointment Custom Field  
Service Amount

Custom Field Definition Edit

Change Field TypeSaveCancel

Field Information

Field Label

Service Amount

Field Name

Service\_Amount

Description

Help Text

Data Owner

User

Field Usage

--None--

Data Sensitivity Level

--None--

Compliance Categorization

Available

PII  
HIPAA  
GDPR  
PCI

Chosen

General Options

Required

☐ Always require a value in this field in order to save a record

Default Value

Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the\_text"), include numbers without quotes : (25), show percentages as decimals : (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type\_\_mdt.RecordAPIName.Field\_\_c

Currency Options

Length

18

Decimal Places

0

Change Field TypeSaveCancel

Edit Appointment Custom Field  
Vehicle number plate

Custom Field Definition Edit

Change Field TypeSaveCancel

Field Information

Field Label

Vehicle number plate

Field Name

Vehicle\_number\_plate

Description

Help Text

Data Owner

User

Field Usage

--None--

Data Sensitivity Level

--None--

Compliance Categorization

Available

PII  
HIPAA  
GDPR  
PCI

Chosen

General Options

Required

☒ Always require a value in this field in order to save a record

Unique

☒ Do not allow duplicate values

☒ Treat "ABC" and "abc" as duplicate values (case insensitive)

☐ Treat "ABC" and "abc" as different values (case sensitive)

External ID

☐ Set this field as the unique record identifier from an external system

Default Value

Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the\_text"), include numbers without quotes : (25), show percentages as decimals : (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type\_\_mdt.RecordAPIName.Field\_\_c

Text Options

Length

10

Change Field TypeSaveCancel

Edit Service records Custom Field  
Service Status

Custom Field Definition Edit

Change Field TypePromote to Global Value SetSaveCancel

Field Information

Field LabelService Status

Field NameService\_Status

Description

Help Text

Data OwnerUser

Field Usage--None--

Data Sensitivity Level--None--

Compliance Categorization

AvailablePIIHIPAAGDPRPCI

Chosen

General Options

Required

☐ Always require a value in this field in order to save a record

Default Value

Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the\_text"), include numbers without quotes : (20), show percentages as decimals : (0.10), and express date calculations in the standard format : (Today() + 7). To reference a field from a Custom Metadata type use: SCustomMetadata.Type\_\_mdt.RecordAPIName Field\_\_c

Picklist Options

☒ Restrict picklist to the values defined in the value set

Change Field TypePromote to Global Value SetSaveCancel

Custom Field Definition Edit

SaveQuick SaveCancel

Field Information

Field Labelservice date

Field Nameservice\_date

Description

Help Text

Data OwnerUser

Field Usage--None--

Data Sensitivity Level--None--

Compliance Categorization

AvailablePIIHIPAAGDPRPCI

Chosen

Formula Options

Formula Return TypeDate

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.  
Example: Reminder Date = CloseDate - 7 [More Examples](#)

Simple FormulaAdvanced Formula

Insert Field

Insert Operator

service date (Date) =  
CreatedDate

## Task 6

### Validation Rules

#### 6.1 Validation Rule for Appointment Object

- Rule Name: Vehicle
- Purpose: Ensure that the vehicle number plate follows a specific format.
- Error Condition Formula:

`NOT(REGEX(Vehicle_number_plate__c , '[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}'))`

- Error Message: 'Please enter a valid number'
- Error Location: Field - Vehicle number plate

Validation Rule Edit
Save Save & New Cancel

Rule Name

Active ☒

Description

Error Condition Formula

Example:  [More Examples...](#)

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

Insert Field
Insert Operator ▼

Check Syntax

Functions
-- All Function Categories - ▼
ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

Insert Selected Function

ABS(number)  
Returns the absolute value of a number, a number without its sign

[Help on this function](#)

Error Message

Example:

This message will appear when Error Condition formula is **true**

Error Message

This error message can either appear at the top of the page or below a specific field on the page

Error Location
☐ Top of Page
☒ Field

## 6.2 Validation Rule for Service Records Object

- Rule Name: service\_status\_note
- Purpose: Ensure that the service status is marked as 'Completed' before the record can be saved.
- Error Condition Formula:

**NOT(ISPICKVAL(Service\_Status\_\_c, 'Completed'))**

- Error Message: ‘Still it is pending’
- Error Location: Field - Service status

Validation Rule Edit
Save
Save & New
Cancel

Rule Name

Active
☐

Description

Error Condition Formula

Example: 
[More Examples...](#)  
Display an error if Discount is more than 30%  
If this formula expression is true, display the text defined in the Error Message area

Insert Field
Insert Operator

Check Syntax

Functions
-- All Function Categories -
ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN
Insert Selected Function
ABS(number)  
Returns the absolute value of a number, a number without its sign  
[Help on this function](#)

Error Message

Example:

This message will appear when Error Condition formula is true

Error Message

This error message can either appear at the top of the page or below a specific field on the page

Error Location
☐ Top of Page
☒ Field

### 6.3 Validation Rule for Billing Details and Feedback Object

- Rule Name: rating\_should\_be\_less\_than\_5
- Purpose: Ensure that the rating for service is between 1 and 5.
- Error Condition Formula:  
  
NOT(REGEX(Rating\_for\_service\_\_c, '[1-5]{1}'))
- Error Message: ‘Rating should be from 1 to 5’

## ● Error Location: Field - Rating for Service

The screenshot shows the 'Validation Rule Edit' window in Salesforce. The 'Rule Name' field contains 'rating\_should\_be\_less\_than\_5'. The 'Active' checkbox is checked. The 'Description' field is empty. Below the description is the 'Error Condition Formula' section. It includes an example: 'Discount\_Percent\_\_c > 0.30' with a note 'Display an error if Discount is more than 30%'. It also states: 'If this formula expression is true, display the text defined in the Error Message area'. There are buttons for 'Insert Field' and 'Insert Operator'. The formula field contains: 'NOT( REGEX( Rating\_for\_service\_\_c , "[1-5]{1}") )'. To the right of the formula is a 'Functions' list with a dropdown for 'All Function Categories'. The list includes: ABS, ACOS, ADDMONTHS, AND, ASCII, and ASIN. There is an 'Insert Selected Function' button at the bottom. A 'Quick Tips' box on the right says 'Operators & Functions'. A legend at the bottom right indicates that a red exclamation mark means 'Required Information'.

## Task 7

### Duplicate Rules

#### 7.1 Creating a Matching Rule for Customer Details Object

##### 1. Rule Name: Matching Customer Details

- **Object:** Customer Details
- **Matching Criteria:**
  - **Gmail:** Exact match
  - **Phone Number:** Exact match
- **Activation:** Once the rule is saved, it is activated to start matching customer details based on the specified criteria.



## Matching customer details

## Matching Rule Detail

[Delete](#) [Clone](#) [Deactivate](#)

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	Active		
Created By	Ruchitha Labisetty, 06/08/2024, 7:49 pm		Modified By Ruchitha Labisetty, 06/08/2024, 7:49 pm

## 7.2 Creating a Duplicate Rule for Customer Details Object

## 1. Rule Name: Customer Detail Duplicate

- **Object:** Customer Details
- **Matching Rule:** Utilizes the previously created ‘Matching Customer Details’ rule.
- **Activation:** After saving the duplicate rule, it is activated to begin identifying duplicate customer details based on the matching criteria.

## Duplicate Rule Detail

[Edit](#) [Delete](#) [Clone](#) [Deactivate](#)

Rule Name	Customer Detail duplicate		Order	1 of 1 [ <a href="#">Reorder</a> ] <a href="#">i</a>	
Description					
Object	Customer Details				
Record-Level Security	Enforce sharing rules				
Action On Create	Allow		Operations On Create	<input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report	
Action On Edit	Allow		Operations On Edit	<input type="checkbox"/> Alert <input type="checkbox"/> Report	
Alert Text	Use one of these records?				
Active	<input checked="" type="checkbox"/>				
Matching Rule	<input checked="" type="checkbox"/> <a href="#">Matching_customer_details</a>	<input checked="" type="checkbox"/> <a href="#">Mapped</a>	Matching Criteria	(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_number EXACT MatchBlank = FALSE)	
Conditions					
Created By	<a href="#">Ruchitha Labisetty</a> , 06/08/2024, 7:50 pm		Modified By	<a href="#">Ruchitha Labisetty</a> , 06/08/2024, 7:50 pm	
<div><a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Clone</a> <a href="#">Deactivate</a></div>					

[Edit](#) [Delete](#) [Clone](#) [Deactivate](#)

## Task 8

## Profiles

## 8.1 Creating the Manager Profile

### 1. Profile Cloning:

- Clone the Standard User profile to create a new profile named 'Manager.'

### 2. Custom App Settings:

- Set the Garage Management Application as the default custom app for the Manager profile.

### 3. Custom Object Permissions:

- Grant access permissions for the following objects:

- ☐ Appointments
- ☐ Billing Details and Feedback
- ☐ Service Records
- ☐ Customer Details

### 4. Session Timeout:

- Configure the session timeout to occur after 8 hours of inactivity.

### 5. Password Policies:

- Set user passwords to never expire.
- Establish a minimum password length of 8 characters.

## Manager

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges \[0\]](#) | [Enabled Apex Class Access \[0\]](#) | [Enabled Visualforce Page Access \[0\]](#) | [Enabled External Data Source Access \[0\]](#) | [Enabled Named Credential Access \[0\]](#) | [Enabled External Credential Principal Access \[0\]](#) | [Enabled Custom Metadata Type Access \[0\]](#) | [Enabled Custom Setting Definitions Access \[0\]](#) | [Enabled Flow Access \[0\]](#) | [Enabled Service Presence Status Access \[0\]](#) | [Enabled Custom Permissions \[0\]](#)

### Profile Detail

[Edit](#) [Clone](#) [Delete](#) [View Users](#)

Name	Manager	
User License	Salesforce	Custom Profile <input checked="" type="checkbox"/>
Description		
Created By	Ruchitha Labisetty, 06/08/2024, 7:53 pm	
Modified By	Ruchitha Labisetty, 06/08/2024, 7:56 pm	

## 8.2 Creating the Sales Person Profile

### 1. Profile Cloning:

- Clone the Salesforce Platform User profile to create a new profile named 'Sales Person.'

### 2. Custom App Settings:

- Set the Garage Management Application as the default custom app for the Sales Person profile.

### 3. Custom Object Permissions:

- Grant access permissions for the following objects:
  - ☐ Appointments
  - ☐ Billing Details and Feedback
  - ☐ Service Records
  - ☐ Customer Details

## sales person

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges \[0\]](#) | [Enabled Apex Class Access \[0\]](#) | [Enabled Visualforce Page Access \[0\]](#) | [Enabled External Data Source Access \[0\]](#) | [Enabled Named Credential Access \[0\]](#) | [Enabled External Credential Principal Access \[0\]](#) | [Enabled Custom Metadata Type Access \[0\]](#) | [Enabled Custom Setting Definitions Access \[0\]](#) | [Enabled Flow Access \[0\]](#) | [Enabled Service Presence Status Access \[0\]](#) | [Enabled Custom Permissions \[0\]](#)

### Profile Detail

[Edit](#) [Clone](#) [Delete](#) [View Users](#)

Name	sales person		
User License	Salesforce Platform	Custom Profile	✓
Description			
Created By	<a href="#">Ruchitha Labisetty</a> , 06/08/2024, 7:57 pm		Modified By <a href="#">Ruchitha Labisetty</a> , 06/08/2024, 7:58 pm

## Task 9

### Role And Role Hierarchy

#### 9.1 Creating the Manager Role

##### 1. Access Roles Setup:

- Navigate to the roles setup section in Salesforce.

##### 2. Expand Role Hierarchy:



- Expand all roles and add a new role under the appropriate superior role.

##### 3. Define Role Details:

- Label the new role as 'Manager' and save the configuration.

## Role Edit Manager

**Role Edit**

Label	<input type="text" value="Manager"/>	
Role Name	<input type="text" value="Manager"/>	
This role reports to	<input type="text" value="CEO"/>	
Role Name as displayed on reports	<input type="text"/>	

### 9.2 Creating Sales Person Role

#### 1. Access Roles Setup:

- Navigate to the roles setup section in Salesforce.

#### 2. Add Role Under Manager:

- Expand the CEO role and add a new role under the Manager role.

#### 3. Define Role Details:

- Label the new role as 'Sales Person' and save the configuration.

Role Edit  
**sales person**

Role Edit	
Label	<input type="text" value="sales person"/>
Role Name	<input type="text" value="sales_person"/> 
This role reports to	<input type="text" value="Manager"/> 
Role Name as displayed on reports	<input type="text"/>
<div>Save Save &amp; New Cancel</div>	

## Task 10

### Users

#### 10.1 Creating a Manager User

1. Access User Setup:

- Navigate to the users section in Salesforce setup.

2. New User Details:

- Fill in the required fields with the user's details, such as first name, last name, alias, email, username, and nickname.
- Assign the role of 'Manager.'
- Set the user license to 'Salesforce.'
- Assign the profile of 'Manager.'

3. Save the Configuration:

- Save the user details to create the new Manager user.

The screenshot shows the 'Users' page in the Salesforce Setup interface. The 'General Information' tab is selected. The page is divided into two columns of fields. The left column contains personal and organizational details, while the right column contains role and license information. Required fields are marked with a red exclamation mark icon. The user 'Niklaus Mikaelson' is being created with the role of 'Manager' and the 'Salesforce' license. The 'Active' checkbox is checked, and the 'Data.com User Type' is set to '--None--'.

Field	Value	Required
First Name	Niklaus	No
Last Name	Mikaelson	Yes
Alias	nmika	No
Email	abc@niklaus.com	Yes
Username	abc@niklaus.com	Yes
Nickname	nik	No
Title		No
Company		No
Department		No
Division		No
Role	Manager	Yes
User License	Salesforce	Yes
Profile	Manager	Yes
Active	<input checked="" type="checkbox"/>	No
Marketing User	<input type="checkbox"/>	No
Offline User	<input type="checkbox"/>	No
Knowledge User	<input type="checkbox"/>	No
Flow User	<input type="checkbox"/>	No
Service Cloud User	<input type="checkbox"/>	No
Site.com Contributor User	<input type="checkbox"/>	No
Site.com Publisher User	<input type="checkbox"/>	No
WDC User	<input type="checkbox"/>	No
Data.com User Type	--None--	No

## 10.2 Creating Sales Person Users

### 1. Access User Setup:

- Navigate to the users section in Salesforce setup.

### 2. New User Details:

- Fill in the required fields with the user's details.
- Assign the role of 'Sales Person.'
- Set the user license to 'Salesforce Platform.'
- Assign the profile of 'Sales Person.'

### 3. Save and Repeat:

- Save the user details and repeat the process to create at least three users with the Sales Person role and profile.

**SETUP Users**

**General Information** \* = Required Information

First Name	seoul	Role	sales person
Last Name	south	User License	Salesforce Platform
Alias	ssout	Profile	sales person
Email	ss1@g.com	Active	<input checked="" type="checkbox"/>
Username	ss1@g.com	Marketing User	<input type="checkbox"/>
Nickname	seso	Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	--None--

## Task 11

### Public Groups

#### 11.1 Creating the Sales Team Public Group

##### 1. Access Public Groups Setup:

- Navigate to the public groups section in Salesforce setup.

##### 2. Define Group Details:

- Provide a label for the new group, such as 'Sales Team.' The group name is automatically generated based on the label.

##### 3. Add Members:

- Search for roles in the available member section.
- Select the 'Sales Person' role and add it to the selected members list.

##### 4. Save the Group:

- Save the new public group configuration.



Group Membership  
Group: sales team

The screenshot shows the 'Edit Public Group' interface in Salesforce. At the top, there's a 'Group Information' header with 'Save' and 'Cancel' buttons. Below it, the 'Edit Public Group' section contains the following fields:

- Label:** A text field containing 'sales team'.
- Group Name:** A text field containing 'sales\_team' with an information icon (i) to its right.
- Grant Access Using Hierarchies:** A checkbox that is checked, with an information icon (i) to its right.
- Search:** A dropdown menu set to 'Public Groups' and a text field for 'for:' followed by a 'Find' button.
- Available Members:** A list box currently showing '--None--'.
- Selected Members:** A list box currently showing 'Role: sales person'.
- Buttons:** Between the two list boxes are 'Add' (a right-pointing arrow) and 'Remove' (a left-pointing arrow) buttons.

At the bottom right of the interface, there are 'Save' and 'Cancel' buttons.

## Task 12

### Sharing Settings

#### 12.1 Configuring Sharing Settings for Service Records

1. **Access Sharing Settings:**

- Navigate to the Sharing Settings section in Salesforce setup.

2. **Set Object-Wide Default (OWD):**

- Change the OWD setting for the Service Records object to 'Private' to restrict access to records by default.

### 3. Create Sharing Rules:


- Define a new sharing rule to grant access to specific roles.

### 4. Define Sharing Rule Details:

- Label the sharing rule appropriately, such as 'Sharing Setting.'
- Specify the members of the role to be shared, selecting 'Sales Person.'
- Determine who will receive the access, selecting 'Manager.'
- Set the access level to 'Read/Write.'

### 5. Save and Refresh:

- Save the sharing rule and refresh the settings to apply changes.


**SETUP**  
**Sharing Settings**

---

Setup

Service records Sharing Rule

Help for this Page ?

Use sharing rules to make automatic exceptions to your organization-wide sharing settings for defined sets of users.


Note: "Roles and subordinates" includes all users in a role, and the roles below that role.

You can use sharing rules only to grant wider access to data, not to restrict access.

Label

Sharing setting

Rule Name

Sharing\_setting 

Description


Service records: owned by members of

Role: sales person

Share with

Role: Manager

Access Level

Read/Write 

Modified By

Ruchitha Labisetty, 06/08/2024, 9:39 pm

Created By

Ruchitha Labisetty, 06/08/2024, 9:39 pm

## Task 13

### Flows

#### 13.1 Creating the Flow

1. **Initiate Flow Creation:**

- Access the Flow setup page and start a new Record-Triggered Flow.

2. **Configure Flow Trigger:**

- Select 'Billing Details and Feedback' as the object.
- Set the flow to trigger when a record is created or updated.
- Choose 'Actions and Related Records' to optimize the flow.

3. **Add Update Records Element:**

- Label the update element as "Amount Update."
- Configure it to update records where the field `Payment_Status__c` equals 'Completed.'
- Set the field value for `Payment_Paid__c` to the service amount from the related Appointment record.

4. **Create and Configure Email Alert:**

- Define a new resource as a text template to construct the email body.
- Use rich text format for clarity and include placeholders for dynamic content.
- The email body should express gratitude for the payment and include the payment amount.

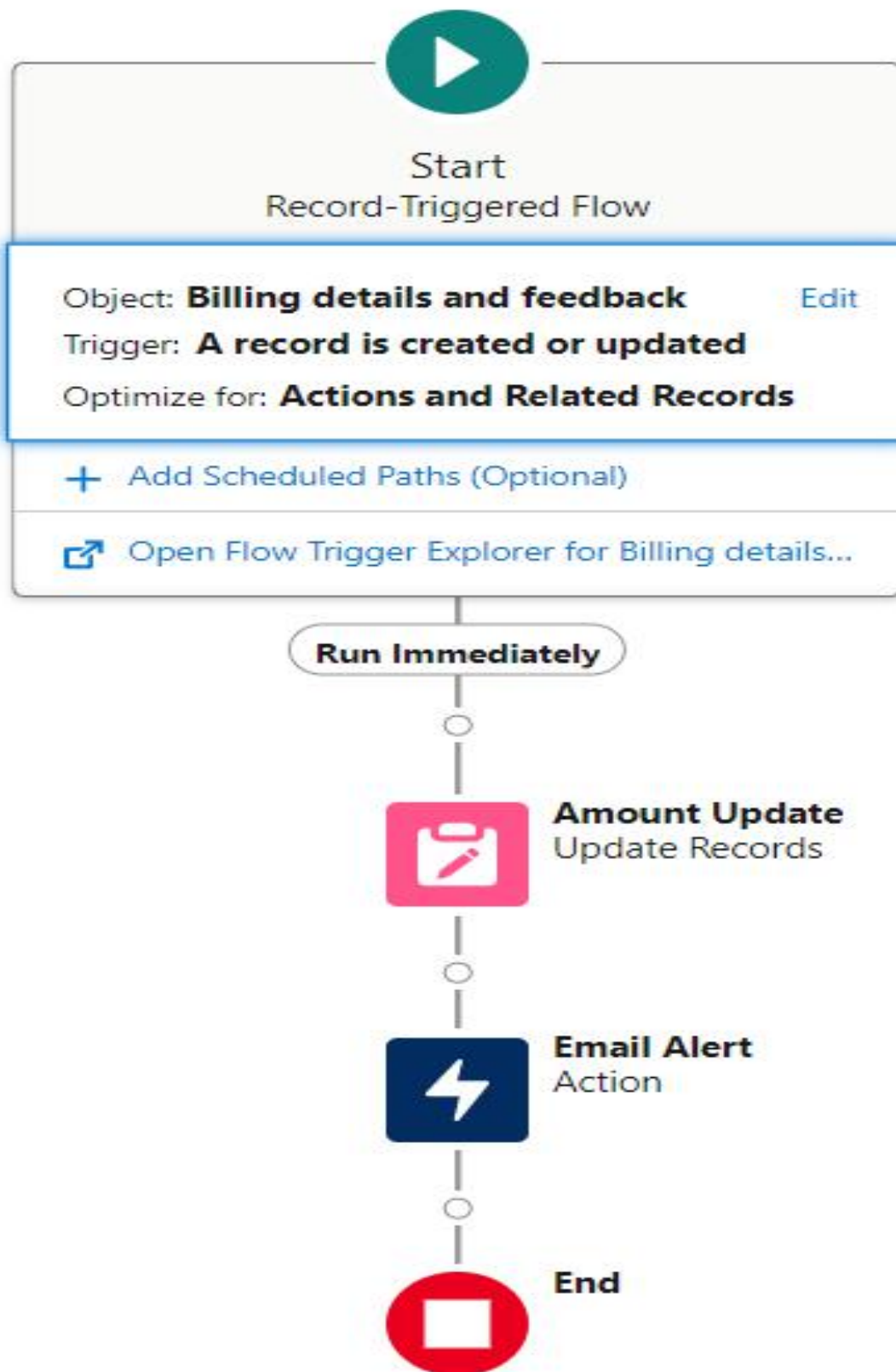
5. **Add Email Action:**

- Insert an 'Action' element to send an email.

- Utilize the previously created text template for the email body.
- Set the recipient address dynamically from the related record.
- Define the subject of the email as 'Thank You for Your Payment - Garage Management.'

**6. Save and Activate the Flow:**

- Save the flow with an appropriate label and auto-populated API name.
- Activate the flow to make it operational.



## Task 14

### Apex Triggers

#### 14.1 Apex Handler: AmountDistributionHandler

**Purpose:** This Apex class calculates and updates the service amount based on selected services in the Appointment\_\_c object.

#### Class Definition:

- **Class Name:** AmountDistributionHandler
- **Method:** amountDist
- **Parameters:** List<Appointment\_\_c> – A list of appointment records to process.

#### Functionality:

- Iterates through the list of appointments.
- Updates the Service\_Amount\_\_c field based on the combination of services selected:
  - All three services: 10000
  - Maintenance and Repairs: 5000
  - Maintenance and Replacement Parts: 8000
  - Repairs and Replacement Parts: 7000
  - Maintenance only: 2000
  - Repairs only: 3000
  - Replacement Parts only: 5000

#### Trigger: AmountDistribution

**Purpose:** This trigger invokes the AmountDistributionHandler to update the Service\_Amount\_\_c field whenever an appointment record is inserted or updated.

### Trigger Definition:

- **Trigger Name:** AmountDistribution
- **sObject:** Appointment\_\_c
- **Events:** before insert, before update

### Trigger Logic:

- Checks if the trigger event is before insert or before update.
- Calls the amountDist method from AmountDistributionHandler to perform the required updates on the Appointment\_\_c records.

### Code:

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

## Task 15

### Reports

#### 15.1 Creating a Report Folder

1. **Objective:** Organize reports by creating a dedicated folder.
2. **Process:**
  - Access the Reports tab through the app launcher.

- Click on **New Folder** and name it '**Garage Management Folder**'. The folder's unique name is automatically generated.
- Save the new folder to store and manage related reports.

### 15.2 Sharing the Report Folder

1. **Objective:** Control access to the report folder.
2. **Process:**
  - Navigate to the Reports tab and locate the '**Garage Management Folder**'.
  - Click on the dropdown menu next to the folder name and select **Share**.
  - Choose **Roles** as the sharing option, search for the '**Manager**' role, and assign **View** access.
  - Confirm the sharing settings and finalize by clicking **Done**.

### 15.3 Creating a Custom Report Type

1. **Objective:** Define a report type to include multiple related objects.
2. **Process:**
  - Access **Setup** and search for **Report Types**.
  - Click on **New Custom Report Type**.
  - Set the **Primary Object** to **Customer Details**.
  - Name the report type '**Service Information**' and confirm the auto-populated name.
  - Categorize under **Other Reports** and set the deployment status to **Deployed**.
  - Click **Next** to define related objects:
    - **Appointment Object**
    - **Service Records**



- **Billing Details and Feedback**

- Save the new report type.

#### 15.4 Creating a Report

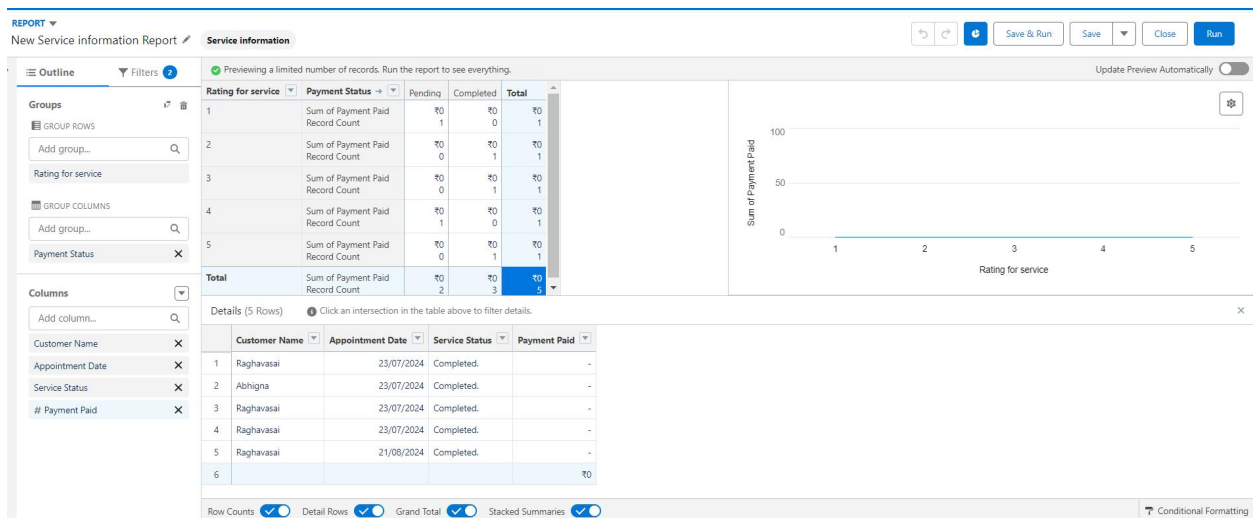
1. **Objective:** Generate a detailed report based on the created report type.

2. **Preparation:**

- Ensure to create at least 10 records for each object to ensure comprehensive data representation.

3. **Process:**

- Access the Reports tab and click on **New Report**.
- Select the category **Other Reports** and choose **Service Information** as the report type.
- Start the report builder and configure the fields:
  - **Columns:** Customer Name, Appointment Date, Service Status, Payment Paid.
  - **Group Rows:** Rating for Service, Payment Status.
- Add a Line Chart to visualize the data.
- Save the report with the name '**New Service Information Report**', ensuring the unique name is auto-populated and the report is saved in the '**Garage Management Folder**'.



## Task 16

### Dashboards

#### 16.1 Creating a Dashboard Folder

1. **Objective:** Organize dashboards by creating a specific folder.
2. **Process:**
  - Navigate to the **Dashboards** tab via the app launcher.
  - Click **New Folder** and name it '**Service Rating Dashboard**'. The unique name is auto-generated.
  - Save the folder to group and manage related dashboards efficiently.

#### Sharing the Dashboard Folder

1. **Objective:** Set access permissions for the dashboard folder.
2. **Process:**
  - Locate the '**Service Rating Dashboard**' folder in the Dashboards tab.
  - Click on the dropdown menu for the folder and select **Share**.

- Assign appropriate permissions based on user roles, ensuring that the relevant team members have access to view or edit the dashboards as needed.
- Confirm the sharing settings to finalize.

## 16.2 Creating a Dashboard

1. **Objective:** Build a dashboard to visualize data and insights.
2. **Process:**
  - Access the **Dashboards** tab and click **New Dashboard**.
  - Enter a name for the dashboard and select the '**Service Rating Dashboard**' folder created earlier.
  - Click **Create** to start building the dashboard.
  - Add a new component by selecting **Add Component**.
  - Choose a report to base the dashboard on, and select a **Line Chart** for visualization.
  - Customize the chart theme as desired.
  - Click **Add** to include the component in the dashboard, then click **Save** and **Done**.

## Subscribing to the Dashboard

1. **Objective:** Automate the delivery of dashboard updates.
2. **Process:**
  - After saving the dashboard, click **Subscribe** at the top right.
  - Set the subscription frequency to **Weekly**.
  - Choose **Monday** as the day for the report to be sent.
  - Click **Save** to activate the subscription and ensure regular updates.

## New Service information Report



[View Report \(New Service information Report\)](#)