Garage Management System

By

Dema Keerthana

keerthanaguptha123@gmail.com

Username: dema@salesforce.com

# 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.

# INDEX PAGE

|  |  |
| --- | --- |
| Topics | Page no |
| Task 1 | 5 |
| Task 2 | 6 - 8 |
| Task 3 | 9 - 12 |
| Task 4 | 13 |
| Task 5 | 14 - 19 |
| Task 6 | 19 - 21 |
| Task 7 | 21 - 22 |
| Task 8 | 22 - 23 |
| Task 9 | 24 |
| Task 10 | 26 - 27 |
| Task 11 | 27 |

|  |  |
| --- | --- |
| Task 12 | 28 |
| Task 13 | 29 - 30 |
| Task 14 | 31 - 32 |
| Task 15 | 33 - 35 |
| Task 16 | 35 - 37 |

**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 exent service, and manage their business more eﬀectively. GMS not only simpliﬁes daily operations but also provides valuable insights to help make better business decisions.

# TASK 1:

* 1. **Creating The Developer Account:**

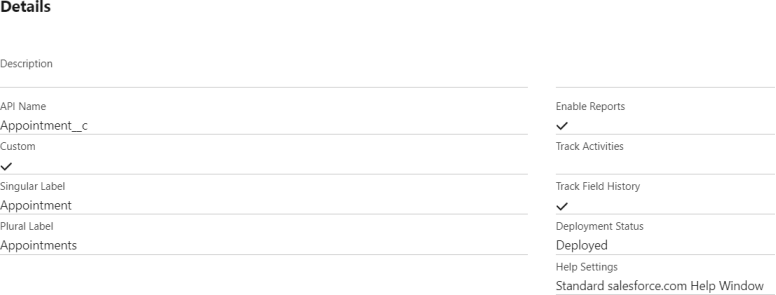
1. To create a developeraccount in Salesforce, follow these steps:

1. Go to SalesforceDeveloper 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,](mailto:username@organization.com) not an actual email id)

Click Sign me up.

# Activatingthe Account:

* + 1. Check the inbox of the email you used for signup.

* + 1. Click on the veriﬁcation link to activate your account (the email may take 5-10 minutes to arrive).

* + 1. Click on **Verify Account**.

* + 1. Set a passwordand answer a security question.

* + 1. Click on **Change Password**.

* + 1. You will be redirectedto your Salesforce setup page.

# TASK 2:

**Creating the 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.

# CreatingCustom Objects

Customer Details Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object. Details:Label Name: CustomerDetails

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.

# 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} StartingNumber: 1

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

# ServiceRecords Object

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

Label Name: Service Records

Plural Label Name: ServiceRecords 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.

# Billing Detailsand Feedback Object

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

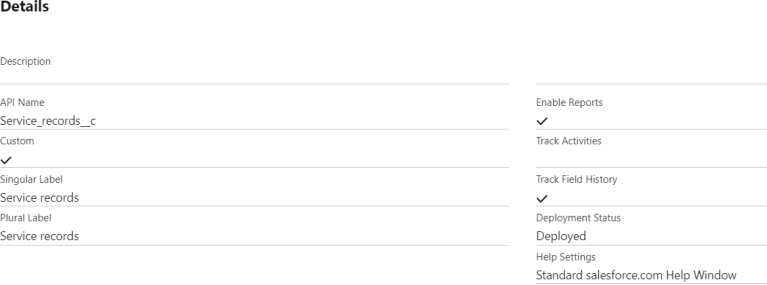
Label Name: BillingDetails and Feedback

Plural Label Name: Billing Details and Feedback Record Name: BillingDetails 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.



|  |  |
| --- | --- |
|  |  |
|  |  |

# TASK 1:

* 1. **Creating The Developer Account:**

1. To create a developeraccount in Salesforce, follow these steps:

1. Go to SalesforceDeveloper 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,](mailto:username@organization.com) not an actual email id)

Click Sign me up.

# Activatingthe Account:

* + 1. Check the inbox of the email you used for signup.

* + 1. Click on the veriﬁcation link to activate your account (the email may take 5-10 minutes to arrive).

* + 1. Click on **Verify Account**.

* + 1. Set a passwordand answer a security question.

* + 1. Click on **Change Password**.

* + 1. You will be redirectedto your Salesforce setup page.

# TASK 2:

**Creating the 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.

# CreatingCustom Objects

Customer Details Object

Navigation Path: From the setup page, click on Object Manager >> Create >> Custom Object. Details:Label Name: CustomerDetails

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.

# 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} StartingNumber: 1

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

# ServiceRecords Object

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

Label Name: Service Records

Plural Label Name: ServiceRecords 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.

# Billing Detailsand Feedback Object

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

Label Name: BillingDetails and Feedback

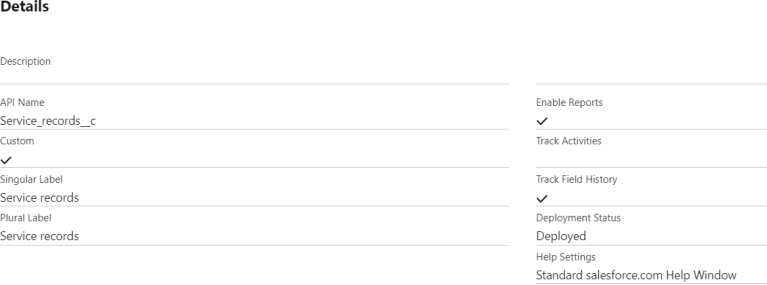
Plural Label Name: Billing Details and Feedback Record Name: BillingDetails and Feedback Name Data Type: Auto Number Display Format:bill-

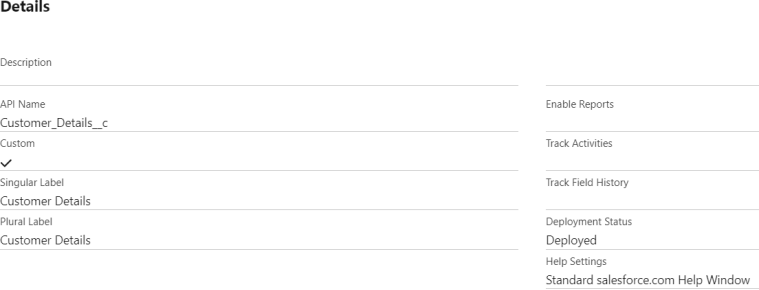
{000} Starting

Number: 1

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







# Task 3:

**Creating Tabs**

1. Creatinga Custom Tab for Customer Details Navigation Path:

Go to the setup page.

Type "Tabs" in the Quick Find bar. Clickon "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 defaultsettings. ClickNext.

Add to Custom App: Uncheck "Include Tab".

Ensure "Append tab to users' existing personal customizations" is checked. Click Save.

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

Repeat the above steps for each remainingobject: Appointments, Service Records, and Billing Detailsand Feedback.

Navigation Path:

Go to the setup page.

Type "Tabs" in the Quick Find bar. Clickon "Tabs".

Click on "New" under the Custom Object Tabs . Detailsfor Each Object:

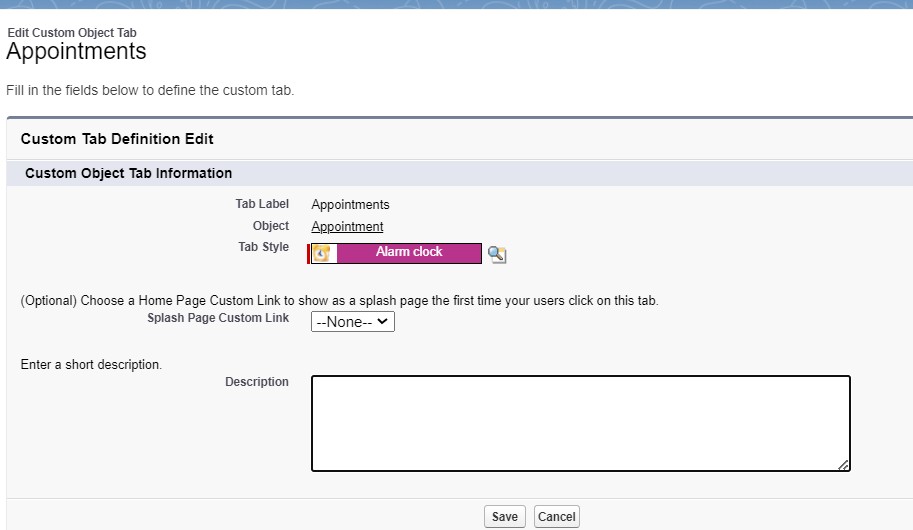
SelectObject: Choose the respective object (Appointments, ServiceRecords, Billing Detailsand Feedback).

SelectTab Style: Choose a sui style for each object. ClickNext.

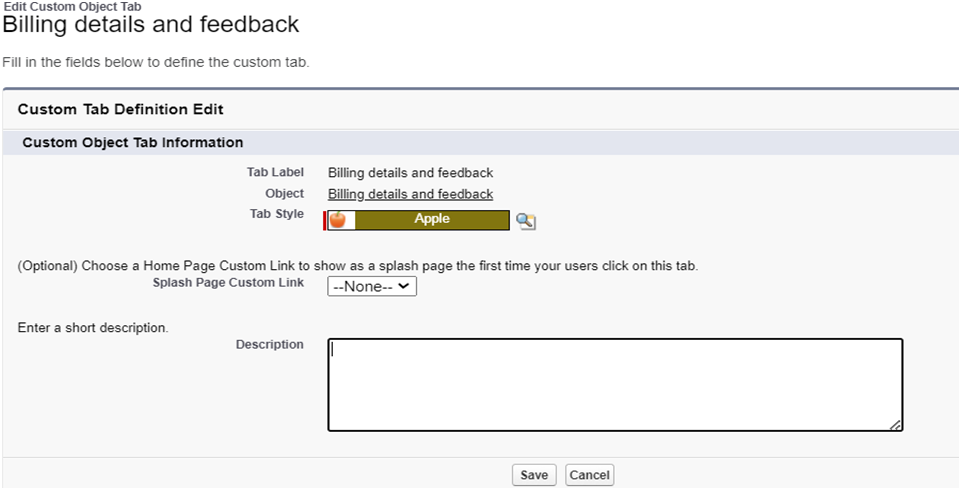
Add to Proﬁles Page: Keep the defaultsettings. ClickNext.

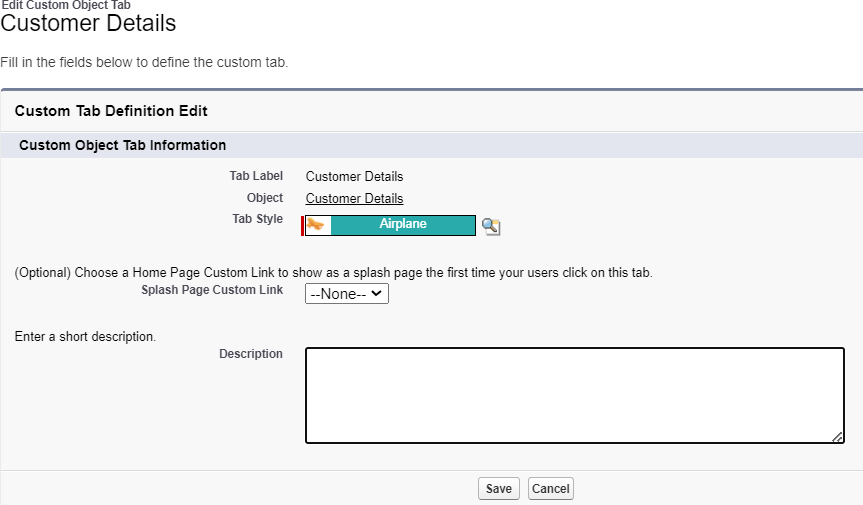
Add to CustomApp: Uncheck "Include Tab".

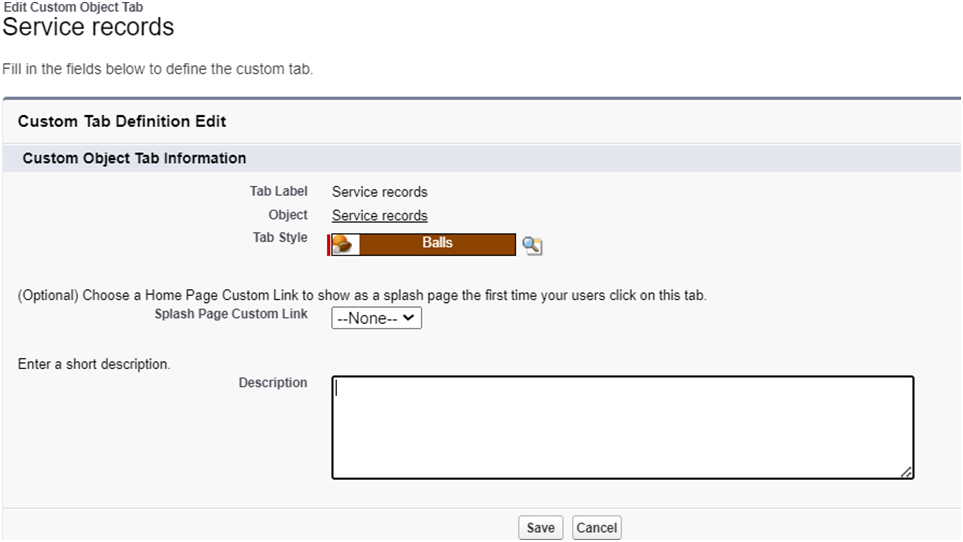
Ensure "Append tab to users' existing personal customizations" is checked. Click Save.



|  |  |
| --- | --- |
|  |  |
|  |  |







# Task 4:

**Building Lignting App**

Creating a Lightning App in Salesforce for managing garageoperations allows for streamlined processes and eﬃcient management of various business aspects. This guide outlinesthe steps to create the Garage Management Application, including addingessential navigation itemsand user proﬁles.

Steps to Create a LightningApp 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 LightningApp" to start the creationprocess.

App Details:

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

Adding Navigation Items:

To enhance navigationwithin the app, select essential items such as Customer Details, Appointments,

ServiceRecords, Billing Details and Feedback, Reports, and Dashboards from the search bar. Move these items using the ar button to includethem 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 ar button. Complete the setup by saving and ﬁnishing the configuration.

# Task 5:

**Creating Fields**

* 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 customeremail addresses. 2)Lookup Fields

\*Appointment Object:

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

\*Service RecordsObject:

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

\*Billing Detailsand Feedback Object:

Added a lookup relationship ﬁeld to connect billing details and feedback with service records.3)Checkbox Fields

\*Appointment Object:

Added multiplecheckbox ﬁelds to indicate diﬀerent services, including "Maintenance Service," "Repairs," and "Replacement Parts," all with default valuesset to unchecked.

\*Service RecordsObject:

Createda checkbox ﬁeld labeled "Quality Check Status" to track the quality checkstatus of the service records.

1. Date Fields

\*Appointment Object:

Added a date ﬁeld labeled "Appointment Date" and marked it as requiredto ensure each appointment has a speciﬁeddate.

1. Currency Fields

\*Appointment Object:

Created a currencyﬁeld labeled "Service Amount" to recordthe cost of services provided,with read- only access for all proﬁles.

\*Billing Detailsand Feedback Object:

Added a currencyﬁeld labeled "Payment Paid" to track payments made by customers.

1. 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 Detailsand Feedback Object:

Added a text ﬁeld labeled"Rating for Service" with a length of 1 character, required and uniqueto capture customerratings.

1. PicklistFields

Service Records Object:

Createda picklist ﬁeld labeled "Service Status" with values "Started" and "Completed" to track the progress of services.

\*Billing Detailsand Feedback Object:

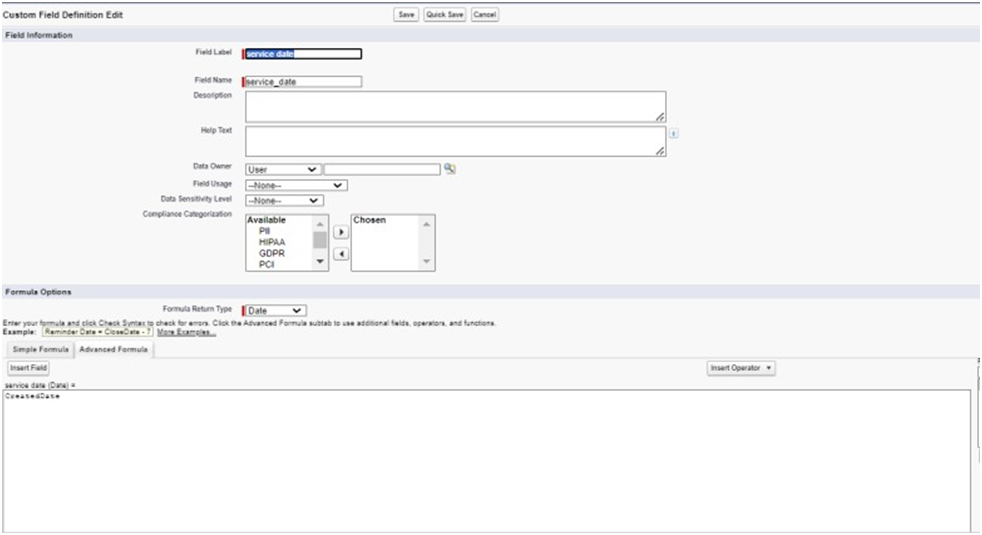
Added a picklistﬁeld labeled "Payment Status" with values "Pending" and "Completed" to monitor payment statuses.

1. Formula Fields

\*Service RecordsObject:

Createda formula ﬁeld labeled "Service Date" to automatically populatewith the createddate of the record.

record.



# Task 6:

**Validation Rules:**

# 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 **2)Validation Rule for Service RecordsObject** Rule Name: service\_status\_note

Purpose: Ensure that the service status is marked as "Completed" before the record can be saved.

Error ConditionFormula:

# NOT(ISPICKVAL(Service\_Status c, "Completed"))

Error Message:"Still it is pending"

Error Location:Field - Service status

1. ValidationRule for Billing Details and Feedback Object RuleName: rating\_should\_be\_less\_than\_5

Purpose:Ensure that the rating for service is between 1 and 5. Error ConditionFormula:

**NOT(REGEX(Rating\_for\_service c, "[1-5]{1}"))** Error Message: "Rating should be from 1 to 5" Error Location: Field- Rating for Service

|  |  |
| --- | --- |
|  |  |
|  |  |

# Task 7:

**Duplicate Rules:**

# Creating a Matching Rule for Customer Details Object

* 1. **Rule Name**: Matching Customer Details **Object**: Customer Details **Matching Criteria:**
     1. **Gmail**: Exact match

* + 1. **Phone Number**: Exact match

1. **Activation**: Once the rule is saved, it is activatedto start matching customer details based on the speciﬁedcriteria.

# Creating a Duplicate Rule for Customer Details Object

1. **Rule Name**: CustomerDetail Duplicate

* 1. **Object**: CustomerDetails

* 1. **Matching Rule**: Utilizesthe previously created "Matching Customer Details" rule.

* 1. **Activation**: After savingthe duplicate rule, it is activated to begin identifying duplicate customer detailsbased on the matching criteria.

|  |  |
| --- | --- |
|  |  |
|  |  |

# Task 8:

**Profiles:**

# Creating the Manager Profile

1. **Proﬁle Cloning**:

* 1. Clone the Standard User proﬁle to create a new proﬁle named "Manager."

# Custom App Settings:

* 1. Set the Garage Management Application as the defaultcustom app for the Managerprofile.

# Custom Object Permissions:

* 1. Grant accesspermissions for the following objects:

* + 1. Appointments

* + 1. BillingDetails and Feedback

* + 1. ServiceRecords

* + 1. CustomerDetails

# Session Timeout:

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

# Password Policies:

* 1. Set user passwords to never expire.

* 1. Establish a minimum password length of 8 characters.

# Creating the Sales Person Profile

1. **Proﬁle Cloning:**

* 1. Clone the Salesforce Platform User proﬁle to create a new proﬁle named "Sales Person."

# Custom App Settings:

* 1. Set the Garage Management Application as the defaultcustom app for the SalesPerson profile.

# Custom Object Permissions:

* 1. Grant accesspermissions for the following objects:

* + 1. Appointments

* + 1. BillingDetails and Feedback

* + 1. ServiceRecords

# Task 9:

1. Customer Details

# Role And Role Hierarch Creating the ManagerRole

1. **Access Roles Setup**:

* 1. Navigate to the roles setup section in Salesforce.

# Expand Role Hierarchy:

* 1. Expand all roles and add a new role under the appropriate superiorrole.

# Deﬁne Role Details:

* 1. Label the new role as "Manager" and save the configuration.

# Creating Sales Person Role

1. **Access Roles Setup**:

* 1. Navigate to the roles setup section in Salesforce.

# Add Role Under Manager:

* 1. Expand the CEO role and add a new role under the Manager role.

# Deﬁne Role Details:

* 1. Label the new role as "Sales Person" and save the configuration.

|  |  |
| --- | --- |
|  |  |
|  |  |

# Task 10:

**Users:**

# Creating a Manager User

1. **AccessUser Setup:**

* 1. Navigate to the users section in Salesforce setup.

# New User Details:

* 1. Fill in the required ﬁeldswith the user'sdetails, such as ﬁrst name, last name, alias, email,username, and nickname.

* 1. Assign the role of "Manager."

* 1. Set the user licenseto "Salesforce."

* 1. Assign the proﬁle of "Manager."

# Save the Configuration:

* 1. Save the user detailsto create the new Manager user.

# Creating Sales Person Users

1. **AccessUser Setup**:

* 1. Navigate to the users section in Salesforce setup.

# New User Details:

* 1. Fill in the requiredﬁelds with the user's details.

* 1. Assign the role of "Sales Person."

* 1. Set the user license to "Salesforce Platform."

* 1. Assign the proﬁleof "Sales Person."

# Save and Repeat:

* 1. Save the user details and repeat the processto create at least three users with the Sales Person role and proﬁle.

# Task 11:

**Public Groups:**

# Creating the Sales Team Public Group

1. **Access Public Groups Setup**:

* 1. Navigate to the public groups section in Salesforce setup.

# Deﬁne Group Details:

* 1. Provide a label for the new group, such as "Sales Team." The group nameis automatically generatedbased on the label.

# Add Members:

* 1. Search for roles in the available members section.

* 1. Select the "Sales Person" role and add it to the selected memberslist.

# Save the Group:

* 1. Save the new public group configuration.

# Task 12:

**Sharing Settings**

# Conﬁguring Sharing Settings for Service Records

1. **AccessSharing Settings**:

* 1. Navigateto the Sharing Settings section in Salesforce setup.

# Set Object-Wide Default (OWD):

* 1. Change the OWD setting for the Service Records object to "Private" to restrictaccess to recordsby default.

# Create Sharing Rules:

* 1. Deﬁne a new sharingrule to grant access to speciﬁc roles.

# Deﬁne Sharing Rule Details:

* 1. Label the sharing rule appropriately, such as "Sharing Setting."

* 1. Specify the members of the role to be shared, selecting"Sales Person."

* 1. Determinewho will receive the access, selecting "Manager."

* 1. Set the access level to "Read/Write."

# Save and Refresh:

* 1. Save the sharing rule and refresh the settings to apply changes.

|  |  |
| --- | --- |
|  |  |
|  |  |

# Task 13:

**Flows:**

# Creating the Flow

1. **InitiateFlow Creation**:

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

# ConﬁgureFlow Trigger:

* 1. Select "Billing Details and Feedback" as the object.

* 1. Set the ﬂow to trigger when a record is createdor updated.

* 1. Choose "Actions and Related Records" to optimize the flow.

# Add Update Records Element:

* 1. Label the updateelement as “Amount Update.”

* 1. Conﬁgureit to update records where the ﬁeld Payment\_Status c equals "Completed."

* 1. Set the ﬁeld valuefor Payment\_Paid c to the service amount from the related Appointment record.

# Create and Conﬁgure Email Alert:

* 1. Deﬁne a new resourceas a text template to construct the email body.

* 1. Use rich text format for clarity and includeplaceholders for dynamic content.

* 1. The email body should express gratitude for the payment and includethe payment amount.

# Add Email Action:

* 1. Insert an "Action" element to send an email.

* 1. Utilizethe previously created text template for the email body.

* 1. Set the recipientaddress dynamically from the related record.

* 1. Deﬁne the subjectof the email as "Thank You for Your Payment - Garage Management."

# Save and Activate the Flow:

* 1. Save the ﬂow with an appropriate label and auto-populated API name.

* 1. Activatethe ﬂow to make it operational.

# Task 14:

**Apex Triggers**

# Apex Handler: AmountDistributionHandler

**Purpose**: This Apex class calculates and updates the service amount based on selected servicesin the Appointment c object.

# Class Definition:

* 1. **Class Name**: AmountDistributionHandler

* 1. **Method**: amountDist

* 1. **Parameters**: List<Appointment c> – A list of appointment records to process.

# Functionality:

* 1. Iteratesthrough the list of appointments.

* 1. Updatesthe Service\_Amount c ﬁeld based on the combination of services selected:

* + 1. All three services:10000

* + 1. Maintenance and Repairs: 5000

* + 1. Maintenance and Replacement Parts: 8000

* + 1. Repairs and Replacement Parts: 7000

* + 1. Maintenance only: 2000

* + 1. Repairsonly: 3000

* + 1. Replacement Parts only: 5000

# Trigger: AmountDistribution

**Purpose**: This trigger invokes the AmountDistributionHandler to update the Service\_Amount c ﬁeldwhenever an appointment record is insertedor updated.

# Trigger Definition:

* 1. **Trigger Name**: AmountDistribution

* 1. **sObject**: Appointment c
  2. **Events**: before insert, before update

# Trigger Logic:

* 1. Checksif the trigger event is before insert or before update.

* 1. 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:**

# Creating a Report Folder

1. **Objective**: Organizereports by creating a dedicated folder.

# Process:

* 1. Access the Reportstab through the app launcher.

* 1. Click on **New Folder**and name it **"Garage Management Folder"**. The folder's uniquename is automatically generated.

* 1. Save the new folder to store and manage relatedreports.

# Sharing the Report Folder

1. **Objective**: Control access to the report folder.

# Process:

* 1. Navigateto the Reports tab and locate the **"Garage Management Folder"**.

* 1. Click on the dropdownmenu next to the foldername and select**Share**.

* 1. Choose **Roles** as the sharing option, search for the **"Manager"** role, and assign **VieW**

access.

* 1. Conﬁrmthe sharing settings and ﬁnalize by clicking

# Done. Creating a Custom ReportType

1. **Objective**: Deﬁne a reporttype to include multiple related objects.

# Process:

* 1. Access **Setup** and search for **Report Types**.

* 1. Click on **New CustomReport Type**.

* 1. Set the **Primary Object** to**Customer Details**.

* 1. Name the report type **"Service Information"** and conﬁrm the auto-populated name.

* 1. Categorize under **Other Reports** and set the deployment status to **Deployed**.

* 1. Click **Next** to deﬁne related objects:

# Appointment Object

* + 1. **Service Records**

# Billing Details and Feedback

* 1. Save the new report type.

# Creating a Report

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

# Preparation:

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

# Process:

* 1. Accessthe Reports tab and click on **New Report**.
  2. Select the category **Other Reports**and choose **ServiceInformation** asthe report type.

* 1. Start the reportbuilder and conﬁgure the fields:

* + 1. **Columns**: CustomerName, Appointment Date, Service Status, Payment Paid.

* + 1. **Group Rows**: Rating for Service,Payment Status.

* 1. Add a Line Chart to visualize the data.
  2. Save the report with the name **"New ServiceInformation Report"**, ensuring the unique name is auto-populated and the report is saved in the **"Garage Management Folder"**.

# Task 16:

**Dashboards:**

# Creating a Dashboard Folder

1. **Objective**: Organizedashboards by creating a speciﬁc folder.

# Process:

* 1. Navigate to the **Dashboards**tab via the app launcher.

* 1. Click **New Folder** andname it **"Service RatingDashboard"**. The uniquename is auto-generated.

* 1. Save the folder to group and manage related dashboards efficiently.

# Sharing the Dashboard Folder

1. **Objective**: Set access permissions for the dashboard folder.

# Process:

* 1. Locate the **"Service Rating Dashboard"** folder in the Dashboards tab.

* 1. Click on the dropdownmenu for the folder and select **Share**.
  2. Assign appropriate permissions based on user roles, ensuring that the relevantteam members have access to view or edit the dashboards as needed.

* 1. Conﬁrm the sharing settings to finalize.

# Creating a Dashboard

1. **Objective**: Build a dashboardto visualize data and insights.

# Process:

* 1. Accessthe **Dashboards** tab and click **New Dashboard**.

* 1. Enter a name for the dashboardand select the **"Service Rating Dashboard"**

folder createdearlier.

* 1. Click **Create** to start buildingthe dashboard.

* 1. Add a new componentby selecting **Add Component**.

* 1. Choose a report to base the dashboard on, and select a **Line Chart** forvisualization.
  2. Customizethe chart theme as desired.

* 1. Click **Add** to includethe component in the dashboard, then click **Save** and

# Done. Subscribing to the Dashboard

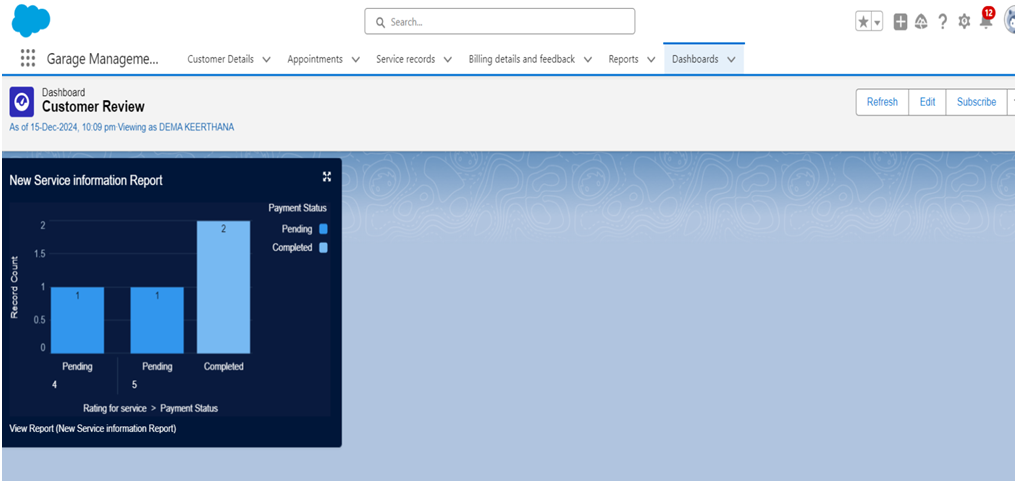
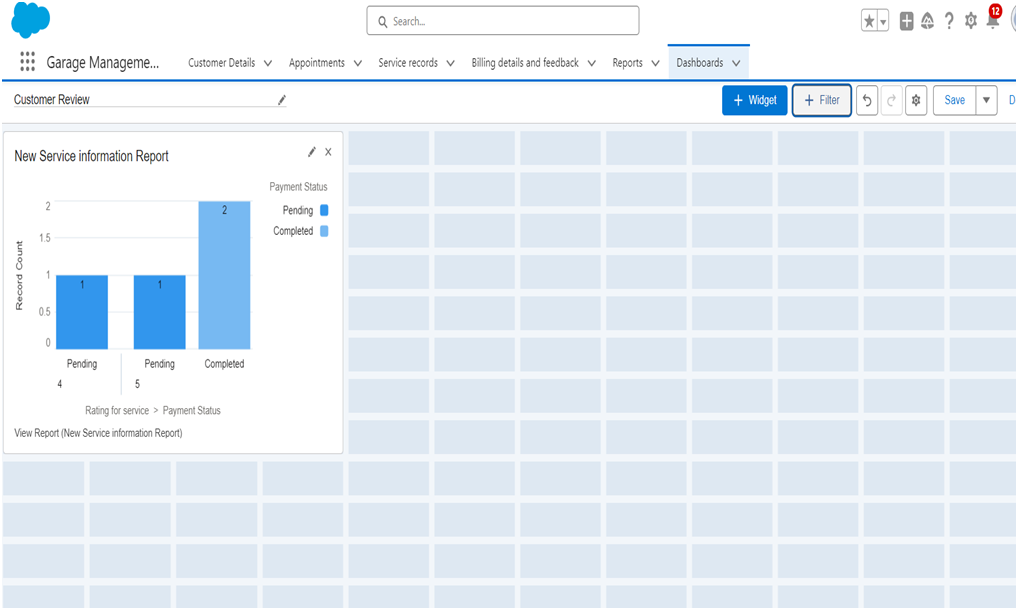
**1. Objective**: Automatethe delivery of dashboard updates.

**2 Process:**

After saving the dashboard, click **Subscribe**at the top right.

Set the subscription frequency to **Weekly**.

Choose **Monday** asthe day for the reportto be sent.

Click **Save** to activate the subscription and ensure regular updates.

|  |  |
| --- | --- |
|  |  |
|  |  |