



UNNAMALAI INSTITUTE OF TECHNOLOGY

Suba Nagar, Kovilpatti - 628502

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

NACC Accredited, ISO 9001:2015 Certified Institution

Project Viva-Voce Examination

The Viva-Voce Examination of this Project work submitted by

M. A. PADMA KUMAR (953322104035)

D. SUDALAI RAJA (953322104056)

M. PIRAVIN KUMAR (953322104040)

B. MUTHUESWARAN (953322104029)

S.ARIKARAMURTHY (953322104005)

Was held on _____

at

UNNAMALAIINSTITUTEOFTECHNOLOGY

Internal Examiner

External Examiner

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report "**GARAGE MANAGEMENT SYSTEM**" is the bonafide work of M.A.PADMAKUMAR (953322104035), D.SUDALAIRAJA (953322104056), M.PIRAVIN KUMAR (953322104040), B.MUTHU ESWARAN (953322104029), S.ARI KARA MURTHY (953322104005)" who carried out the project work under my supervision

SIGNATURE

Mrs.A.CHITRA, M.E.,

HEAD OF THE DEPARTMENT

Assistant Professor

Computer Science and Engineering

Unnamalai Institute of Technology

Kovilpatti-628 502

SIGNATURE

Mr.J.SETHURAM,M.E,

SUPERVISOR

Assistant Professor

Computer Science and Engineering

Unnamalai Institute of Technology

Kovilpatti-628 502

ACKNOWLEDGEMENT

We express our sincere gratitude to the Management, the Head of the Institution, **Dr.D.RAVINDHRAN M.E.,PHD.**, and Head of the CSE Department **Mrs. A.CHITRA, M.E.**, Assistant Professor, for encouraging and providing all the necessary facilities to carry out this project.

Profound thanks to the Project Coordinator **Mr.J.SETHURAM, M.E**, Assistant Professor, Department of CSE for providing necessary environment and encouragement in completing this project work.

We express our heartfelt thanks to the guide Mr.J.SETHURAM, M.E, Assistant Professor, Department of CSE who motivated us, to carry out this project successfully.

We express our heartfelt thanks to all the staff members who involved directly or indirectly to make this project more purposeful

Garage Management System

Abstraction

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff. The **Garage Management System (GMS)** is a comprehensive software solution designed to streamline and optimize the operations of automotive repair facilities, service centers, and garages. It provides an array of features tailored to meet the needs of mechanics, service advisors, and business owners, ensuring smoother workflows and higher customer satisfaction.

The **Garage Management System** is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff. The Garage Management System (GMS) is a comprehensive software solution designed to streamline and optimize the operations of automotive repair facilities, service centers, and garages. It provides an array of features tailored to meet the needs of mechanics, service advisors, and business owners, ensuring smoother workflows and higher customer satisfaction.

● Appointment Scheduling:

- Simplifies the booking process for customers.
- Enables staff to manage daily schedules efficiently, reducing downtime and improving resource allocation.

● Vehicle Management:

- Maintains detailed records of vehicles, including service history, repairs, and maintenance schedules.
- Tracks vehicle status during servicing for better communication with customers.

● Customer Relationship Management (CRM):

- Stores customer details and preferences.
- Sends service reminders, follow-ups, and promotional offers to build loyalty.

● Inventory and Spare Parts Management:

- Tracks spare parts stock levels, automates reorder processes, and prevents stockouts.
- Ensures that mechanics always have the necessary tools and parts on hand.

● Billing and Invoicing:

- Generates professional invoices quickly and accurately.
- Supports multiple payment methods, discounts, and tax calculations.

● Work Order Management:

- Creates detailed work orders with a list of tasks, estimated costs, and timelines.
- Helps staff prioritize jobs and ensures timely completion.

● Reporting and Analytics:

- Provides insights into key performance indicators like revenue, job completion rates, and customer feedback.
- Helps identify trends and areas for improvement.

Salesforce Introduction:

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

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

Creating Developer Account:

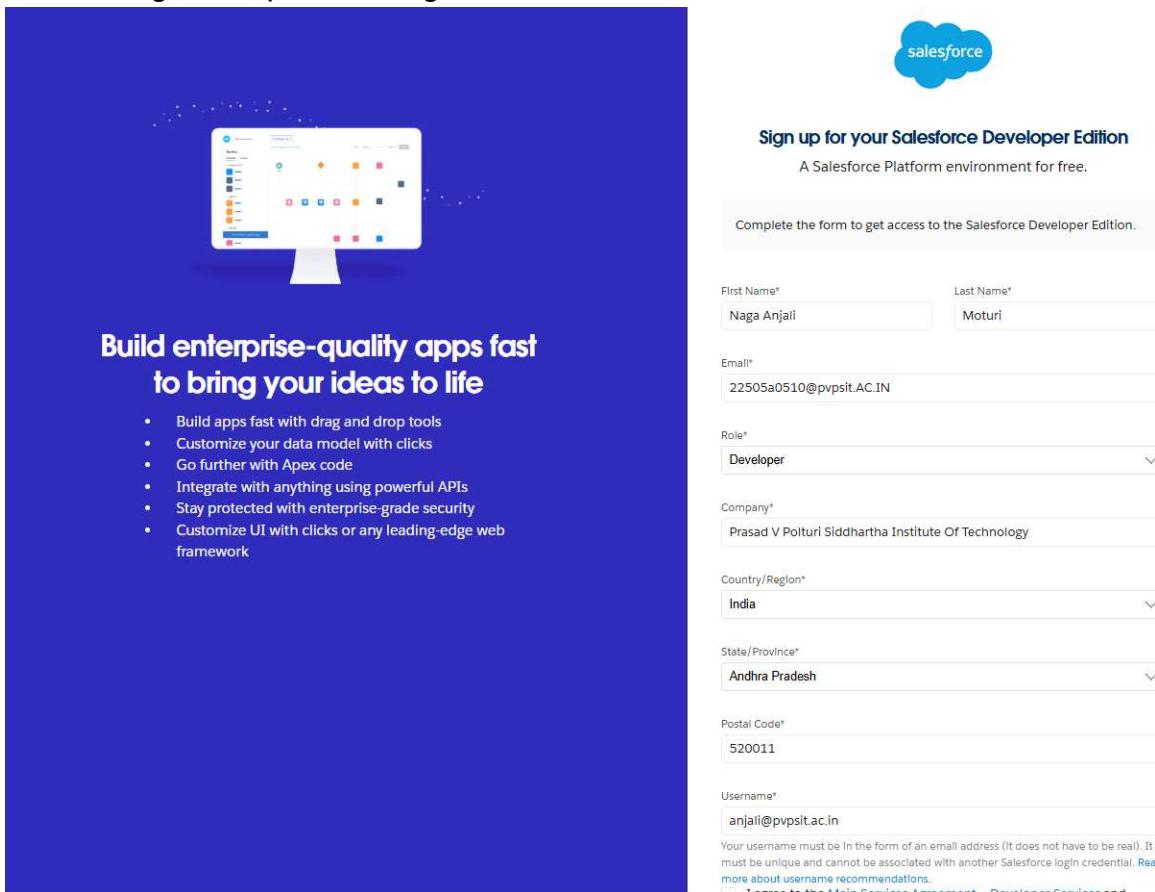
Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :
 - First name & Last name
 - Email
 - Role : Developer

- Company : College Name
- County : India
- Postal Code : pin code
- Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format :
username@organization.com

Click on sign me up after filling these.



Sign up for your Salesforce Developer Edition

A Salesforce Platform environment for free.

Complete the form to get access to the Salesforce Developer Edition.

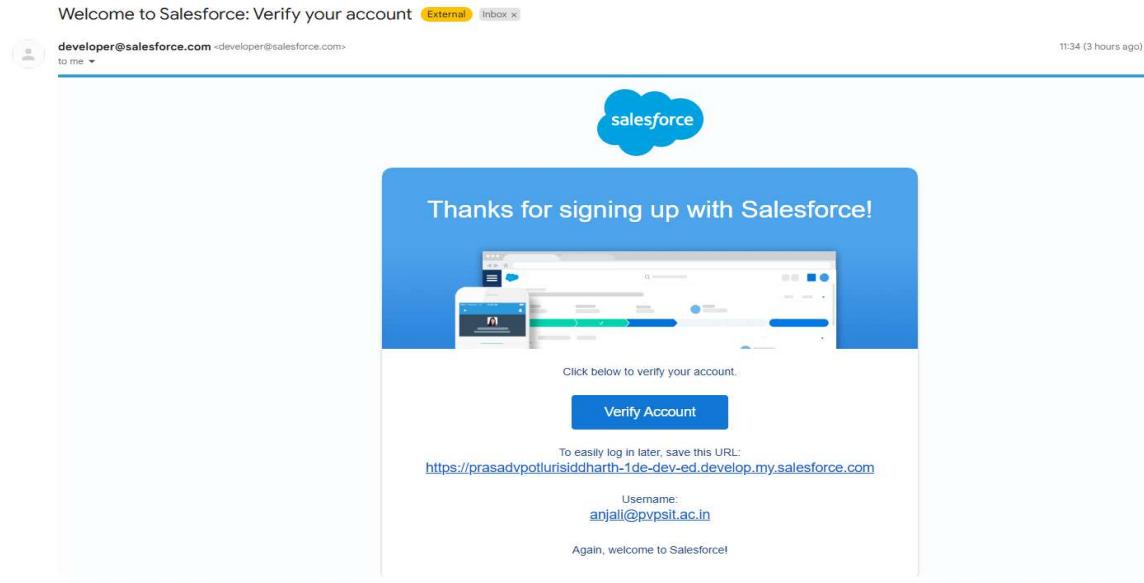
| | |
|------------------------|--|
| First Name* | Last Name* |
| Naga Anjali | Moturi |
| Email* | |
| 22505a0510@pvpst.ac.in | |
| Role* | Developer |
| Company* | Prasad V Poluri Siddhartha Institute Of Technology |
| Country/Region* | India |
| State/Province* | Andhra Pradesh |
| Postal Code* | 520011 |
| Username* | anjali@pvpst.ac.in |

Your username must be in the form of an email address (it does not have to be real). It must be unique and cannot be associated with another Salesforce login credential. [Read more about username recommendations.](#)

[Learn to the *Master Services Agreement*, *Developer Services* and *Platform Services*.](#)

Account Activation

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account.



OBJECT

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects. Salesforce objects are of two types:

- **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
- **Custom Objects:** Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

Create Customer Details Object

To create an object: From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- Enter the label name >> Customer Details
- Plural label name >> Customer Details

- Enter Record Name Label and Format
 - Record Name >> Customer Name
 - Data Type >> Text
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Create Appointment Object

To create an object:

From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- Enter the label name >> Appointment
- Plural label name >> Appointments
- Enter Record Name Label and Format
 - Record Name >> Appointment Name
 - Data Type >> Auto Number
 - Display Format >> app-{000}
 - Starting number >> 1
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Appointment Field

Appointment Name

[Back to Appointment](#)

[Set Field-Level Security](#) [View Field Accessibility](#)

| Field Information | | Field Name |
|---------------------------|------------------|------------|
| Field Label | Appointment Name | |
| Data Type | Auto Number | |
| Description | | |
| Data Owner | | |
| Field Usage | | |
| Data Sensitivity Level | | |
| Compliance Categorization | | |
| Display Format | app-{000} | |

Create Service records Object

To create an object: From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- Enter the label name >> Service records
- Plural label name >> Service records
- Enter Record Name Label and Format
 - Record Name >>Service records Name
 - Data Type >> Auto Number
 - Display Format >> ser-{000}
 - Starting number >> 1
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Service records Field

Service records Name

[Back to Service records](#)

[Set Field-Level Security](#) | [View Field Accessibility](#)

| Field Information | | Field Name |
|---------------------------|----------------------|------------|
| Field Label | Service records Name | N |
| Data Type | Auto Number | |
| Description | | |
| Data Owner | | |
| Field Usage | | |
| Data Sensitivity Level | | |
| Compliance Categorization | | |
| Display Format | ser-{000} | |

Create Billing details and feedback Object

To create an object: From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

- Enter the label name >> Billing details and feedback
- Plural label name >> Billing details and feedback
- Enter Record Name Label and Format
 - Record Name >> Billing details and feedback Name
 - Data Type >> Auto Number
 - Display Format >> bill-{000}
 - Starting number >> 1
- Click on Allow reports and Track Field History,
- Allow search >> Save.

Billing details and feedback Field

Billing details and feedback Name

[Back to Billing details and feedback](#)

[Set Field-Level Security](#) [View Field Accessibility](#)

| Field Information | | Field Name |
|---------------------------|-----------------------------------|------------|
| Field Label | Billing details and feedback Name | |
| Data Type | Auto Number | |
| Description | | |
| Data Owner | | |
| Field Usage | | |
| Data Sensitivity Level | | |
| Compliance Categorization | | |
| Display Format | bill-{000} | |

Tabs

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

Creating a Custom Tab

To create a Tab:(Customer Details)

- Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)
- Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
- Make sure that the Append tab to users' existing personal customizations is checked. Click save.

Creating Remaining Tabs

- Now create the Tabs for the remaining Objects, they are “ Appointments, Service records,Billing details and feedback”.
- Follow the same steps as mentioned in Activity -1 .

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It includes three main sections: 'Custom Object Tabs' (listing Appointments, Billing details and feedback, Customer Details, and Service records with tab styles like Balls, Bell, Apple, and Jewel), 'Web Tabs' (empty), and 'Visualforce Tabs' (empty). Each section has a 'New' button and a 'What Is This?' link.

| Custom Object Tabs | | | |
|--------------------|------------------------------|-----------|-------------|
| Action | Label | Tab Style | Description |
| Edit Del | Appointments | | Balls |
| Edit Del | Billing details and feedback | | Bell |
| Edit Del | Customer Details | | Apple |
| Edit Del | Service records | | Jewel |

| Web Tabs | |
|-------------------------------|--|
| New What Is This? | |
| No Web Tabs have been defined | |

| Visualforce Tabs | |
|---------------------------------------|--|
| New What Is This? | |
| No Visualforce Tabs have been defined | |

The Lightning 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.

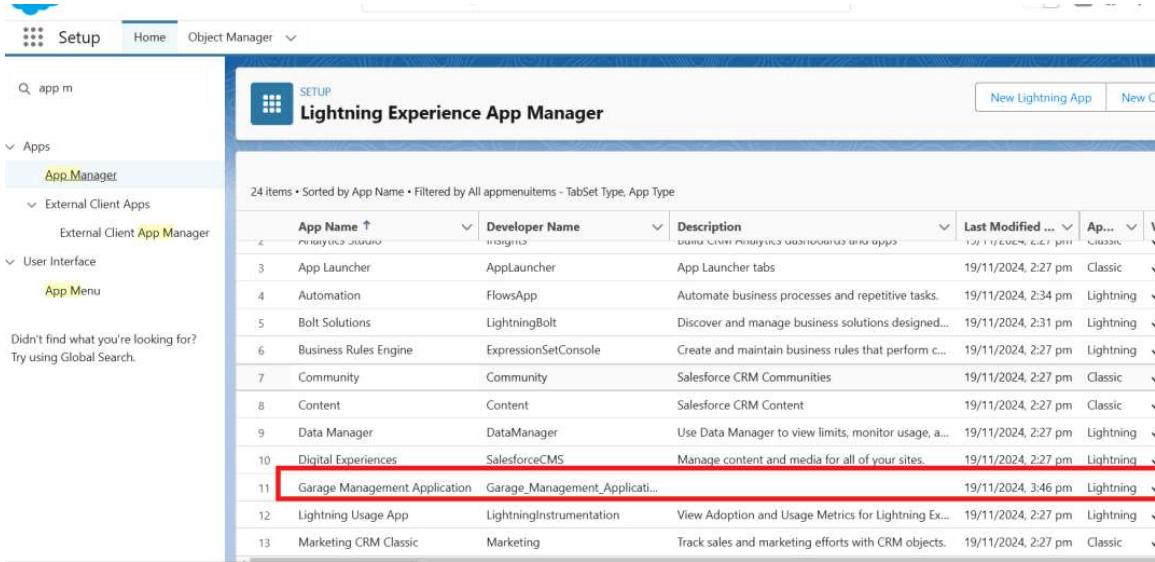
Lightning apps let you brand your apps with a custom colour and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

Create a Lightning App

To create a lightning app page:

- Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
- Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.
- To Add Navigation Items:
- Select the items (Customer Details, Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.

- To Add User Proles: Search proles (System administrator) in the search bar >> click on the arrow button >> save & Finish.



The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. In the search bar, 'app m' is typed. The 'Apps' section is expanded, and 'App Manager' is selected. A list of 24 installed apps is displayed in a table. The columns are: App Name, Developer Name, Description, Last Modified, and App Type. The 'Garage Management Application' (Developer Name: Garage_Management_Applicati...) is highlighted with a red box.

| App Name ↑ | Developer Name | Description | Last Modified ... | Ap... v |
|----------------------------------|--------------------------------|--|---------------------|-----------|
| 3 App Launcher | AppLauncher | App Launcher tabs | 19/11/2024, 2:27 pm | Classic |
| 4 Automation | FlowsApp | Automate business processes and repetitive tasks. | 19/11/2024, 2:34 pm | Lightning |
| 5 Bolt Solutions | LightningBolt | Discover and manage business solutions designed... | 19/11/2024, 2:31 pm | Lightning |
| 6 Business Rules Engine | ExpressionSetConsole | Create and maintain business rules that perform c... | 19/11/2024, 2:27 pm | Lightning |
| 7 Community | Community | Salesforce CRM Communities | 19/11/2024, 2:27 pm | Classic |
| 8 Content | Content | Salesforce CRM Content | 19/11/2024, 2:27 pm | Classic |
| 9 Data Manager | DataManager | Use Data Manager to view limits, monitor usage, a... | 19/11/2024, 2:27 pm | Lightning |
| 10 Digital Experiences | SalesforceCMS | Manage content and media for all of your sites. | 19/11/2024, 2:27 pm | Lightning |
| 11 Garage Management Application | Garage_Management_Applicati... | View Adoption and Usage Metrics for Lightning Ex... | 19/11/2024, 3:46 pm | Lightning |
| 12 Lightning Usage App | LightningInstrumentation | Track sales and marketing efforts with CRM objects. | 19/11/2024, 2:27 pm | Lightning |
| 13 Marketing CRM Classic | Marketing | Marketing | 19/11/2024, 2:27 pm | Classic |

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.

Creation of fields for the Customer Details object

- To create fields in an object:
 - Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
 - Now click on “Fields & Relationships” >> New
 - Select Data Type as a “Phone”
 - Click on next.
 - Fill the Above as following:
 - Field Label: Phone number
 - Field Name : gets auto generated

- Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining elds for the same object.

- To create another elds in an object:
 - Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
 - Now click on “Fields & Relationships” >> New
 - Select Data type as a “Email” and Click on Next
 - Fill the Above as following:
 - Field Label : Gmail
 - Field Name : gets auto generated
 - Click on Next >> Next >> Save and new.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Customer Details'. On the left, a sidebar lists various configuration options under 'Fields & Relationships'. The main content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data in the table is as follows:

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

Creation of Lookup Fields

Creation of Lookup Field on Appointment Object :

- Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select “Look-up relationship” as data type and click Next. 4. Select the related object “ Customer Details” and click next.

- Next >> Next >> Save.

Note: Make sure you complete Activity 4 Before continuing.

Creation of Lookup Field on Service records Object :

- Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
- Now click on “Fields & Relationships” >> New
- Select “Look-up relationship” as data type and click Next.
- Select the related object “ Appointment ” and click next.
- Make it a required field so click on Required.
- Scroll down for Lookup Filter and click on Show Iter settings.
- Now add the Iter criteria.
- Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date
- Filter type should be Required.
- Error Message : Value does not match the criteria.
- Enable the Iter by click on Active.
- Next >> Next >> Save.

SETUP > OBJECT MANAGER

Service records

Fields & Relationships

8 Items, Sorted by Field Label

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

Creation of Lookup Field on Billing details and feedback Object :

- Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
- Now click on “Fields & Relationships” >> New.
- Select “Look-up relationship” as data type and click Next.
- Select the related object “ Service records” and click next.
- Next >> Next >> Save & new.

| Fields & Relationships | | | | | | |
|--------------------------------|-----------------------------------|-----------------------|-----------------------------------|-----------|-------------------|-------|
| 8 Items, Sorted by Field Label | | FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEX |
| | Billing details and feedback Name | Name | Auto Number | | | ✓ |
| | Created By | CreatedById | Lookup(User) | | | |
| | Last Modified By | LastModifiedById | Lookup(User) | | | |
| | Owner | OwnerId | Lookup(User,Group) | | | ✓ |
| | Payment Paid | Payment_Paid__c | Currency(18, 0) | | | |
| | Payment Status | Payment_Status__c | Picklist | | | |
| | Rating for service | Rating_for_service__c | Text(1) (Unique Case Insensitive) | | | ✓ |
| | Service records | Service_records__c | Lookup(Service records) | | | ✓ |

Creation of Checkbox Fields

Creation of CheckboxField on Appointment Object :

- Go to setup >>click on Object Manager >>type object name(Appointment) in search bar >> click on the object.
- Now click on “Fields & Relationships” >>New.
- Select “Check box” as data type and click Next.
- Give the Field Label: Maintenance service
- Field Name : is auto populated
- Default value : unchecked
- Click on next >>next >> save.

Creation of AnotherCheckbox Field on Appointment Object:

- Repeat the steps form 1 to 3.
- Give the Field Label : Repairs
- Field Nme : is auto populated
- Default value : unchecked

- Click on next >>next >> save.
- Follow the same and create another checkbox with given names
- Give the Field Label : Replacement Parts
- Field Nme : is auto populated
- Default value : unchecked
- Click on next >> next >> save.

Creation of CheckboxField on Servicerecords Object :

- Go to setup >>click on Object Manager >>type object name(Service records) in search bar >>click on the object.
- Now click on “Fields & Relationships” >>New.
- Select “Check box” as data type and click Next.
- Give the Field Label : QualityCheck Status
- Field Nme : is auto populated
- Default value : unchecked
- Click on next >>next >> save

Creation of date Fields

Creation of Date Field on Appointment Object:

- Go to setup >>click on Object Manager >>type object name(Appointment) in the search bar >> click on the object.
- Now click on “Fields & Relationships” >>New.
- Select “Date” as data type and click Next.
- Give the Field Label : Appointment Date
- Field Nme : is auto populated
- Make it as a Requiredfield by click on the Required option.
- Click on next >>next >> save.

Creation of CurrencyFields

Creation of CurrencyField on Appointment Object :

- Go to setup >>click on Object Manager >>type object name(Appointment) in the search bar >>click on the object.
- Now click on “Fields& Relationships” >>New.
- Select “Currency” as data type and click Next.
- Give the Field Label: Service Amount
- Field Nme : is auto populated
- Click on next
- Give read only for all the profiles in field level security for profile.
- Click on next >> save.

Creation of CurrencyField on Billingdetails and feedbackObject :

- Follow the same steps as mentioned above in Billing details and feedback Object.
- Change the label name as mentioned.
- Give the Field Label : PaymentPaid
- Field Nme : is auto populated

Creation of Text Fields

- Go to setup >>click on Object Manager >>type object name(Appointment) in the search bar >> click on the object.
- Now click on “Fields & Relationships” >>New.
- Select “Text” as data type and click Next.
- Give the Field Label : Vehicle number plate
- Field Name : is auto populated
- Length : 10
- Make field as Required and Unique.
- Click on next >>next >> save.

Creation of Text Fields in Billingdetails and feedbackobject :

- Go to setup >>click on Object Manager >>type object

name(Billing detailsand feedback) in search bar >> click on the object.

- Now click on “Fields & Relationships” >>New.
- Select “text” as data type and click Next.
- Give the Field Label: Rating for service
- Field Name : is auto populated
- Length : 1
- Make field as Required and Unique.
- Click on next >> next >> save

Creation of PicklistFields

Creation of PicklistFields in Servicerecords object :

- Go to setup >>click on ObjectManager >> type object name(Service records) in search bar >> click on the object.
- Click on fields & relationship >> click on New.
- Select Data type as “Picklist” and click Next.
- Enter Field Label as “Service Status”,under values select“Enter values, with each value separated by a new line” and enter values as shown below.
- The values are: Started, Completed.
- Click Next.
- Next >> Next >>Save.

Creation of Picklist Fieldsin Billing detailsand feedback object:

- Go to setup >>click on Object Manager >>type object name(Billing details and feedback) in search bar >> click on the object.
- Click on fields & relationship >> click on New.
- Select Data type as “Picklist” and click Next.
- Enter Field Label as “Payment Status”,under values select“Enter values,

with each value separated by a new line" and enter values as shown below.

- The values are: Pending, Completed.
- Click Next.
- Next >> Next >>Save.

Creating Formula Fieldin Service recordsObject

- Go to setup >>click on ObjectManager >> type object name(Service records) in search bar >> click on the object.
- Click on fields & relationship >> click on New.
- Select Data type as “Formula” and click Next.
- Give Field Label and Field Name as “servicedate” and select formula return type as “Date” and click next.
- Insert field formulashould be : CreatedDate
- click “Check Syntax” .
- Click next >> next >> Save.

Validation rule

Validation rules are appliedwhen a user tries to save a record and are used to checkif the data meets specified criteria.If the criteria are not met, the validation rule triggers an error message and prevents the user from savingthe record untilthe issues are resolved.

To create a validation rule to an Appointment Object

- Go to the setuppage >> click on objectmanager >> From drop down click edit for Appointment object.
- Click on the validation rule >> clickNew.
- Enter the Rule name as “ Vehicle ”.
- Insert the ErrorCondition Formula as :-
NOT(REGEX(Vehicle_number_plate_c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))

5 . Enter the Error Messageas “Please enter valid number”, select the Error locationas

Field and select the field as "Vehicle number plate", and click Save.

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'Appointment' object. On the left, there's a sidebar with options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions. The main area is titled 'Validation Rules' and shows one item: 'Vehicle'. The table columns are RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The 'Vehicle' rule has 'Vehicle number plate' as the error location and 'Please enter valid number' as the error message. It is marked as ACTIVE and modified by 'Naga Anjali Moturi' on 21/11/2024 at 12:36 pm.

| RULE NAME | ERROR LOCATION | ERROR MESSAGE | ACTIVE | MODIFIED BY |
|-----------|----------------------|---------------------------|--------|--|
| Vehicle | Vehicle number plate | Please enter valid number | ✓ | Naga Anjali Moturi, 21/11/2024, 12:36 pm |

To create a validation rule to an Service records Object

- Go to the setup page >> click on object manager >> From drop down click edit for Service records object.
- Click on the validation rule >> click New.
- Enter the Rule name as "service_status_note".
- Insert the ErrorCondition Formula as : -
- NOT(ISPICKVAL(Service_Status__c , "Completed"))
- Enter the Error Message as "still it is pending", select the Error location as Field and select the field as "Service status", and click Save.

The screenshot shows the Salesforce Setup interface under the Object Manager section for the 'Service records' object. On the left, there's a sidebar with options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, and Field Sets. The main area is titled 'Validation Rules' and shows one item: 'service_status_note'. The table columns are RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The 'service_status_note' rule has 'Service Status' as the error location and 'still it is pending' as the error message. It is marked as ACTIVE and modified by 'Naga Anjali Moturi' on 21/11/2024 at 12:42 pm.

| RULE NAME | ERROR LOCATION | ERROR MESSAGE | ACTIVE | MODIFIED BY |
|---------------------|----------------|---------------------|--------|--|
| service_status_note | Service Status | still it is pending | ✓ | Naga Anjali Moturi, 21/11/2024, 12:42 pm |

To create a validation rule to an Billing details and feedback Object

- Go to the setup page >> click on object manager >> From drop down

clickedit for Billing details and feedback object.

- Click on the validation rule >> click New.
- Enter the Rule name as “ rating_should_be_less_than_5”.
- Insert the ErrorCondition Formula as : - NOT(REGEX(Rating_for_service c , "[1-5]{1}"))

5.Enter the ErrorMessage as “ratingshould be from 1 to 5”, selectthe Error locationas Field and select the field as “Rating for Service”, and click Save.

| Validation Rules | | | | |
|------------------------------|--------------------|------------------------------|--------|--|
| 1 Items, Sorted by Rule Name | | | | |
| RULE NAME | ERROR LOCATION | ERROR MESSAGE | ACTIVE | MODIFIED BY |
| rating_should_be_less_than_5 | Rating for service | rating should be from 1 to 5 | ✓ | Naga Anjali Moturi, 21/11/2024, 12:43 pm |

Duplicate rule

To create a matching rule to an Customer detailsObject

- Go to quick find box in setup and search for matchingRule.
- Click on matchingrule >> click on New Rule.
- Select the object as Customer detailsand click Next.
- Give the Rule name : Matching customer details
- Unique name : is auto populated
- Define the matching criteria as
 - Field : MatchingMethod
 - Gmail:Exact
 - Phone Number:Exact
- Click save.

- After Saving Click on Activate.

The screenshot shows the 'Matching Rules' page in the Salesforce Setup. At the top, there's a blue header bar with the word 'SETUP' and a small 'd' icon. Below it, the page title is 'Matching Rules'. Underneath, there's a sub-header 'Matching Rule' and the specific rule name 'Matching customer details'. On the right side of the sub-header, there are three buttons: 'Delete', 'Clone', and 'Deactivate'. The main content area is titled 'Matching Rule Detail' and contains the following information:

| | Customer Details |
|-------------------|--|
| Object | Customer Details |
| Rule Name | Matching customer details |
| Unique Name | Matching_customer_details |
| Description | (Customer_Details: Gmail EXACT MatchBlank = FALSE) AND (Customer_Details: Phone_number EXACT MatchBlank = FALSE) |
| Matching Criteria | Status: Active |
| Created By | Naga Anjali Moturi, 21/11/2024, 12:46 pm |
| Modified By | Naga Anjali Moturi, 21/11/2024, 12:46 pm |

Below this, there's a large empty table with two columns and two rows, likely for viewing or managing other matching rules.

To create a Duplicate rule to an Customer detailsObject

- Go to quick find box in setup and search for Duplicate rules.
- Click on Duplicate rule >> click on New Rule >>select customer detailsobject.
- Give the Rule name as : CustomerDetail duplicate
- Scroll a little in Matching rule section
- Select the matching rule : Matching customer details
- And Click on save.
- After saving the Duplicate Rule, Click on Activate.

Duplicate Rule Detail

| Rule Name | Customer Detail duplicate | Order | 1 of 1 [Reorder] |
|-----------------------|---|----------------------|--|
| Description | | Operations On Create | <input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report |
| Object | Customer Details | Operations On Edit | <input type="checkbox"/> Alert <input type="checkbox"/> Report |
| Record-Level Security | Enforce sharing rules | | |
| Action On Create | Allow | | |
| Action On Edit | Allow | | |
| Alert Text | Use one of these records? | | |
| Active | <input checked="" type="checkbox"/> | | |
| Matching Rule | <input checked="" type="checkbox"/> Matching customer details <input type="checkbox"/> Mapped | Matching Criteria | (Customer_Details: Gmail_EXACT MatchBlank = FALSE) AND (Customer_Details: Phone_number_EXACT MatchBlank = FALSE) |
| Conditions | | Modified By | Naga Anjali Moturi, 21/11/2024, 12:48 pm |
| Created By | Naga Anjali Moturi, 21/11/2024, 12:48 pm | | |

Role & 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.

Creating Manager Role

- Go to quick find >> Search for Roles >> click on set up roles.
- Click on Expand All and click on add role under whom this role works.
- Give Label as “Manager” and Role name gets auto populated. Then click on Save.

Creating another roles

- Go to quick find >> Search for Roles >> click on set up roles.
- Click plus on CEO role, and click add role under manager.
- Give Label as “sales person” and Role name gets auto populated. Then click on Save.



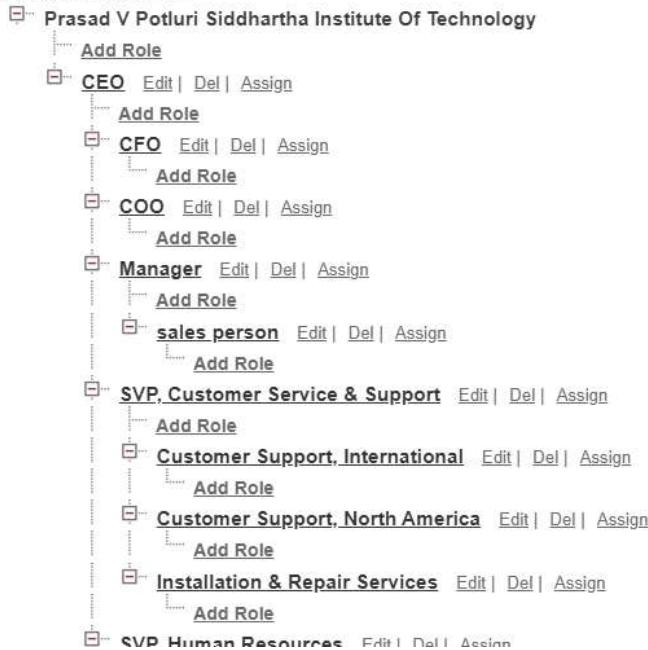
SETUP
Roles

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

[Collapse All](#) [Expand All](#)



Users

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.

Create User

- Go to setup >> type users in quick find box >> select users >> click New user.

- Fill in the fields
 - First Name : Niklaus
 - Last Name : Mikaelson
 - Alias : Give a Alias Name
 - Email id : Give your Personal Emailid
 - Username : Username should be in this form: text@text.text
 - Nick Name : Give a Nickname
 - Role : Manager
 - User licence : Salesforce
 - Profiles : Manager
- Save.

User Niklaus Mikaelson

Permission Set Assignments [0] | Permission Set Assignments Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] | Public Group Membership [1] | Queue Members [0] | Team [0] | Managers in the Role Hierarchy [0] | OAuth Apps [0] | Third-Party Account Links [0] | Installed Mobile Apps [0] | Authentication Settings for External Systems [0] | Login History [0] | User Provisioning Accounts [0]

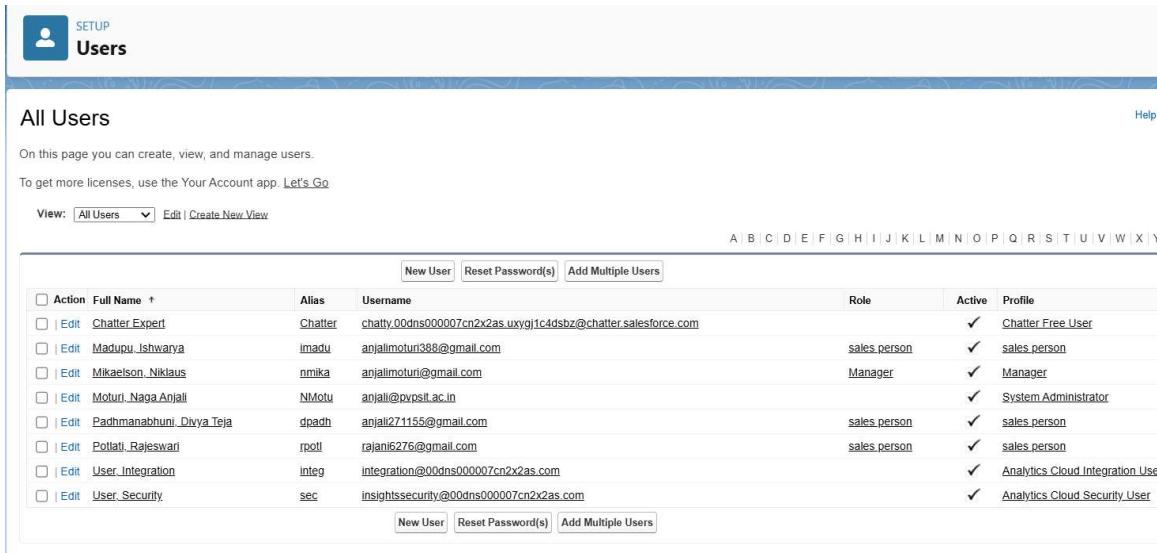
User Detail

| User Detail | | Edit | | Sharing | Reset Password | Freeze | View Summary |
|--------------------|--|-----------------------------------|--|---------|----------------|--------|--------------|
| Name | Niklaus Mikaelson | Role | Manager | | | | |
| Alias | nmika | User License | Salesforce | | | | |
| Email | anjalmoturi@gmail.com [Verify] i | Profile | Manager | | | | |
| Username | anjalmoturi@gmail.com | Active | <input checked="" type="checkbox"/> | | | | |
| Nickname | User17321742697356042344 i | Marketing User | <input type="checkbox"/> | | | | |
| Title | | Offline User | <input type="checkbox"/> | | | | |
| Company | | Knowledge User | <input type="checkbox"/> | | | | |
| Department | | Flow User | <input type="checkbox"/> | | | | |
| Division | | Service Cloud User | <input type="checkbox"/> | | | | |
| Address | | Site.com Contributor User | <input type="checkbox"/> | | | | |
| Time Zone | (GMT+05:30) India Standard Time (Asia/Kolkata) | Site.com Publisher User | <input type="checkbox"/> | | | | |
| Locale | English (India) | WDC User | <input type="checkbox"/> | | | | |
| Language | English | Mobile Push Registrations | View | | | | |
| Delegated Approver | Manager | Data.com User Type | i | | | | |
| | | Accessibility Mode (Classic Only) | <input type="checkbox"/> i | | | | |

creating another users

- Repeat the steps and create another user using
 - Role : sales person
 - User licence : Salesforce Platform
 - Profile :sales person

Note : create atleast3 users with these permissions.



The screenshot shows the Salesforce Setup interface under the 'Users' section. At the top, there's a blue header bar with the 'SETUP' icon and the word 'Users'. Below it, a banner says 'All Users' and 'On this page you can create, view, and manage users.' A link to 'Let's Go' is present. The main area has a toolbar with 'View: All Users' (selected), 'Edit', and 'Create New View'. Above the table, there are buttons for 'New User', 'Reset Password(s)', and 'Add Multiple Users'. The table itself has columns for 'Action', 'Full Name', 'Alias', 'Username', 'Role', 'Active', and 'Profile'. The data includes:

| Action | Full Name | Alias | Username | Role | Active | Profile |
|---------------------------------|----------------------|---------|---|--------------|-------------------------------------|---------------------------------|
| <input type="checkbox"/> Edit | Chatter Expert | Chatter | chatty.00dns000007cn2x2as.uxyqj1c4dsbz@chatter.salesforce.com | | <input checked="" type="checkbox"/> | Chatter Free User |
| <input type="checkbox"/> Edit | Madipu Ishvarya | imadu | anjalmoturi388@gmail.com | sales person | <input checked="" type="checkbox"/> | sales person |
| <input type="checkbox"/> Edit | Mikaelson Niklaus | nmiika | anjalmoturi@gmail.com | Manager | <input checked="" type="checkbox"/> | Manager |
| <input type="checkbox"/> Edit | Moturi Naga Anjali | NMotu | anjali@pyosit.ac.in | | <input checked="" type="checkbox"/> | System Administrator |
| <input type="checkbox"/> Edit | Padmanabhu Diya Teja | dpath | anjali271155@gmail.com | sales person | <input checked="" type="checkbox"/> | sales person |
| <input type="checkbox"/> Edit | Pollati Rajeswari | rpolli | rajan6276@gmail.com | sales person | <input checked="" type="checkbox"/> | sales person |
| <input type="checkbox"/> Edit | User Integration | integ | integration@00dns000007cn2x2as.com | | <input checked="" type="checkbox"/> | Analytics Cloud Integration Use |
| <input type="checkbox"/> Edit | User Security | sec | insightssecurity@00dns000007cn2x2as.com | | <input checked="" type="checkbox"/> | Analytics Cloud Security User |

Public groups

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.

Creating New Public Group

- Go to setup >> type users in quick find box >>select public groups>> click New.
- Give the Label as “sales team”.
- Group name is autopopulated.
- Search for Roles.
- In Available Members select Sales person and click on add it will be moved to selected member.
- Click on save.

The screenshot shows the Salesforce 'Public Groups' page. At the top, there's a blue header bar with a person icon and the word 'SETUP'. Below it, the page title is 'Public Groups'. A sub-header says 'A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.' There are navigation links for 'View: All' (selected), 'Edit', 'Create New View', and 'Help for'. Below this is a table with one row, showing a group named 'sales_team' created by 'Moturi, Naga Anjali' on '21/11/2024, 1:08 pm'. The table has columns for Action, Label +, Group Name, Created By, and Created Date.

| Action | Label + | Group Name | Created By | Created Date |
|---|------------|------------|---------------------|---------------------|
| Edit Delete | sales team | sales_team | Moturi, Naga Anjali | 21/11/2024, 1:08 pm |

Sharing Setting

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.

Creating Sharing settings

- Go to setup >>type users in quick find box >>select Sharing Settings>> click Edit.
- Change the OWD setting of the Service records Object to private as shown in fig.
- Click on save and refresh.
- Scroll down a bit, Click new on Service records sharing Rules.
- Give the Label name as “Sharing setting”
- Rule name is auto populated.
- In step 3 : Select which records to be shared, members of “Roles” >> “Sales person”
- In step 4: sharewith, select “Roles” >> “Manager”
- In step 5 : Change the access level to “Read / write”.
- Click on save.

Flows

Create a Flow

- Go to setup >>type Flow in quick find box >> Click on the Flow and Select the New Flow.
- Select the Record-triggered flow and Click on Create.
- Select the Object as “Billing details and feedback” in the Drop down list.
- Select the Trigger Flow when: “A record is Created or Updated”.
- Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
- Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”. Give the Label Name : Amount Update
- Api name : is auto populated
- Set a filter condition : All Conditions are met(AND)
- Field : Payment_Status_c

- Operator : Equals
- Value : Completed
- And Set Field Values for the Billingdetails and feedbackRecord
- Field : Payment_Paid_c
- Value : {\$Record.Service_records_r.Appointment_r.Service_Amount_c}
- Click On Done.Before creating another Element.Create a New Resource form Toolbox form top left.
- Click on the New Resource,And select Variable.
- Select the resourcetype as text template.
- Enter the API name as “ alert”.
- Change the view as Rich Text ? View to Plain Text.
- In body fieldpaste the syntaxthat given below.

Dear {\$Record.Service_records_r.Appointment_r.Customer_Name_r.Name},

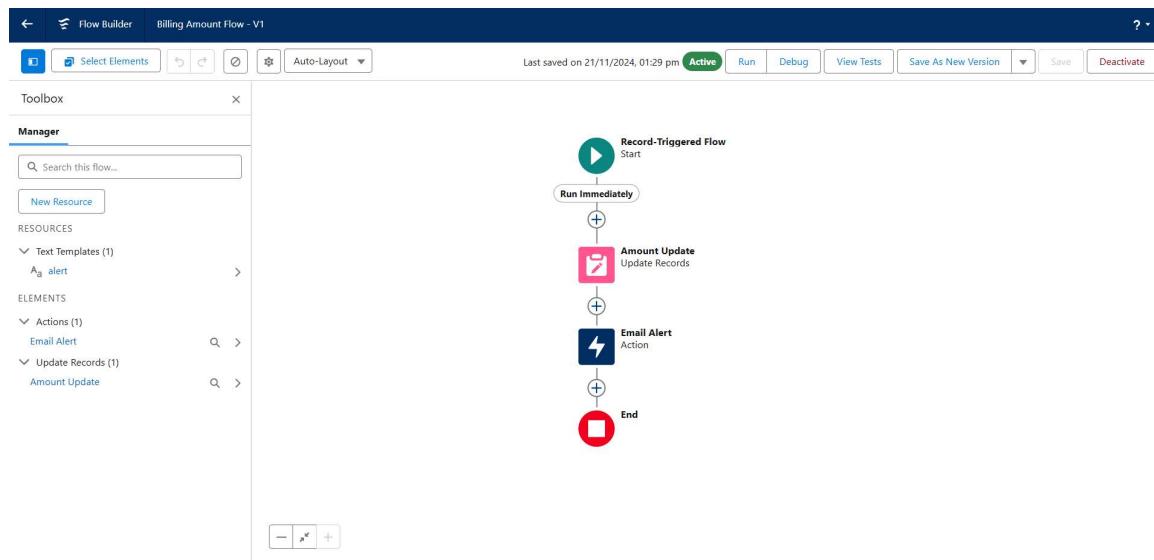
I hope this message findsyou well. I wanted to take a moment to express my sincere gratitude for your recent paymentfor the servicesprovided by our garage management team. Your prompt paymentis greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {\$Record.Payment_Paid_c}

Thank you for Coming .

- Click done.
- Now Click on Add Element, select Action.
- Their action bar will be opened in that searchfor “ send email ” and click on it.
- Give the label name as “ Email Alert”
- API name will be auto populated.
- Enable the body in set input values for the selected action.
- Select the text template that created , Body : {!alert}
- Include recipient address list select the email form the record.

- RecipientAddressList:
`{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail
c}`
- Include subject as “ Thank You for Your Payment - Garage Management”.
- Click done.
- Click on save. Give the Flow label , Flow Api name will be autopopulated.
- And click save, and click on activate.



Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

Apex handler

Use Case : This use case works for Amount Distribution for each Service the customer selected for their Vehicle.

- Login to the respective trailhead account and navigate to the gear icon in the top right corner.
- Click on the Developer console. Now you will see a new console window.
- In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
- Name the class as "AmountDistributionHandler".

Code:

```
public class AmountDistributionHandler {  
    public static void amountDist(list<Appointment__c>
```

```

listApp){ list<Service_records_c> serList = new list
<Service_records_c>(); for(Appointment_c app :
listApp){
    if(app.Maintenance_service_c == true && app.Repairs_c == true &&
app.Replacement_Parts_c == true){

        app.Service_Amount_c = 10000;

    }

    else if(app.Maintenance_service_c == true && app.Repairs_c == true)
    { app.Service_Amount_c = 5000;
    }

    else if(app.Maintenance_service_c == true && app.Replacement_Parts_c
    == true){ app.Service_Amount_c = 8000;
    }

    else if(app.Repairs_c == true && app.Replacement_Parts_c == true)
    { app.Service_Amount_c = 7000;
    }

    else if(app.Maintenance_service_c == true)
    { app.Service_Amount_c = 2000;
    }

    else if(app.Repairs_c ==
    true)
    { app.Service_Amount
    c = 3000;
    }

    else if(app.Replacement_Parts_c == true)
    { app.Service_Amount_c = 5000;
    }

}
}
}
}

```

Trigger Handler :

How to create a new trigger :

- While still in the trailhead account, navigate to the gear icon in the top right

- corner.
- Click on developer console and you will be navigated to a new console window.
 - Click on File menu in the tool bar, and click on new? Trigger.
 - Enter the trigger name and the object to be triggered.
 - Name : AmountDistribution
 - sObject : Appointment_c

Syntax For creating trigger:

The syntax for creating trigger is :

Trigger [trigger name] on [object name](Before/After event)

```
{  
}
```

In this project, trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

Code:

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

Reports

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.

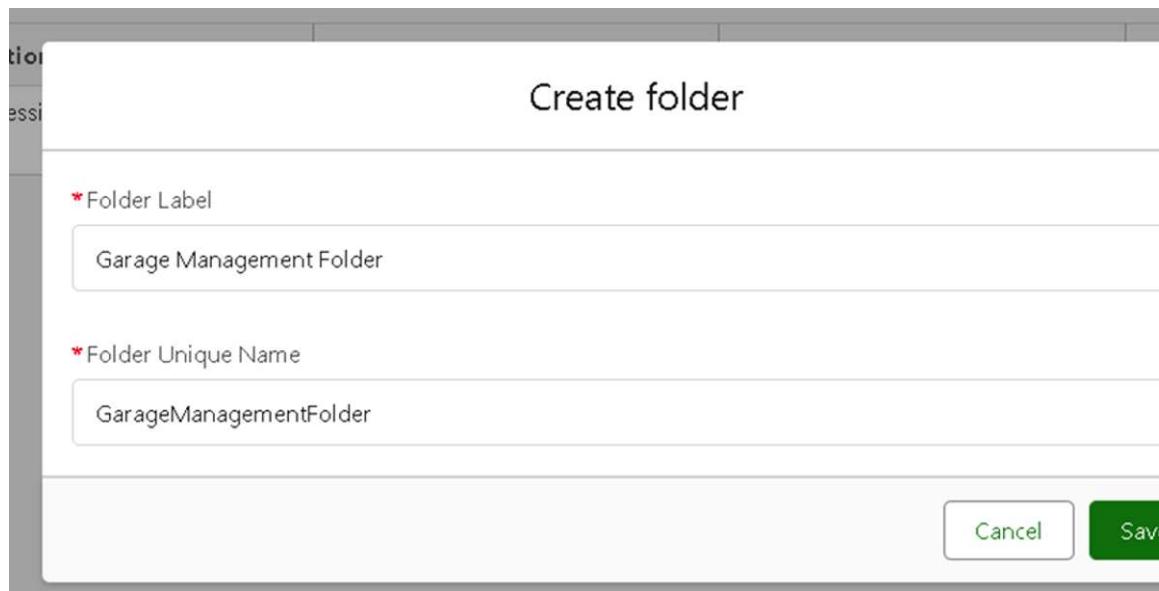
Types of Reports in Salesforce

- Tabular
- Summary

- Matrix
- Joined Reports

create a report folder

- Click on the app launcher and search for reports.
- Click on the report tab, click on new folder.
- Give the Folder label as “Garage Management Folder”, Folder uniquename will be auto populated.
- Click save.



Sharing a report folder

- Go to the app >>click on the reports tab.
- Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
- Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
- Then click share, and click on Done.

Create Report Type

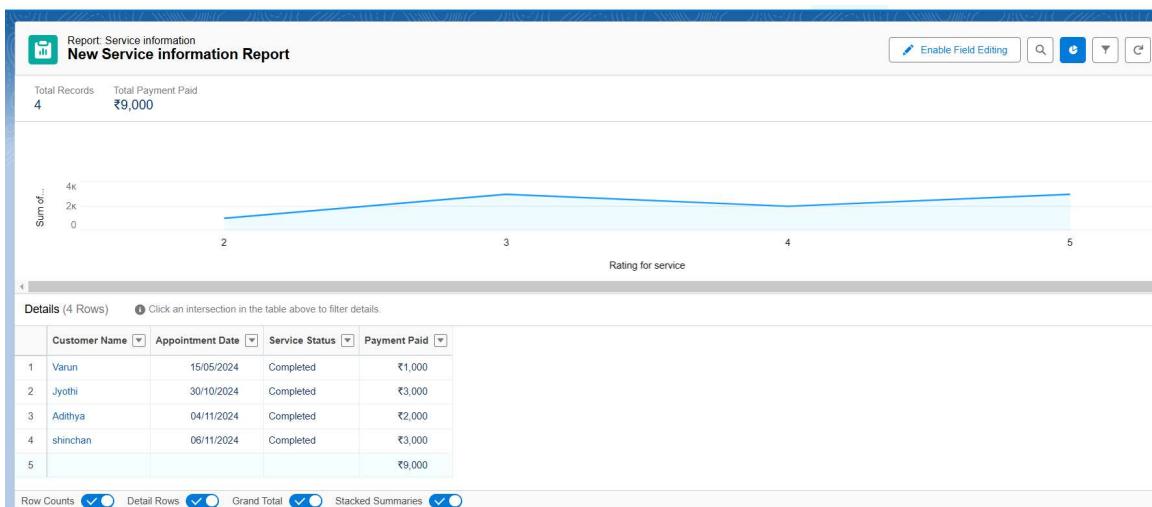
- Go to setup >> type users in quick find box >>select Report Type >> click on Continue.
- Click on new custom reporttype.
- Select the Primary object as “ Customer details” .
- Give the Reporttype Label as “ Serviceinformation ”
- Report type Name is autopopulated.
- Keep the Description as same.
- Select Store in Category as “ otherReports ”
- Select the deployment status as “ Deployed ”, click on Next.
- now , Click on Relatedobject box.
- Click on SelectObject, choose Appointment Object as shownin fig
- Again Click to relate anotherobject.
- And select the related objectas “ service records”.
- Repeat the process and select the related object as “ Billing details and feedback”.
- And click on save.

Create Report

Note : Before creating report,create latest “10” records in every object. Try to fill every field in each record for better experience.

- Go to the app >>click on the reports tab
- Click New Report.
- Select the Category as other reports,search for ServiceInformation, select that report, click on it. And click on start report.
- Their outline pane is opened alredy,select the fields that mentionedbelow in column section.
 - Customer name
 - Appointment Date
 - Service Status

- Payment paid
- Remove the unnecessary fields.
- Select the fields that mentioned below in GROUP ROWS section.
 - Rating for Service
- Select the fields that mentioned below in GROUP ROWS section.
 - Payment Status
- Click on Add Chart , Select the Line Chart.
- Click on save, Give the reportName : New Service information Report
- Report unique Name is auto populated.
- Select the folder the createdand Click on save.



Dashboards

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 identifytrends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboardbasics.

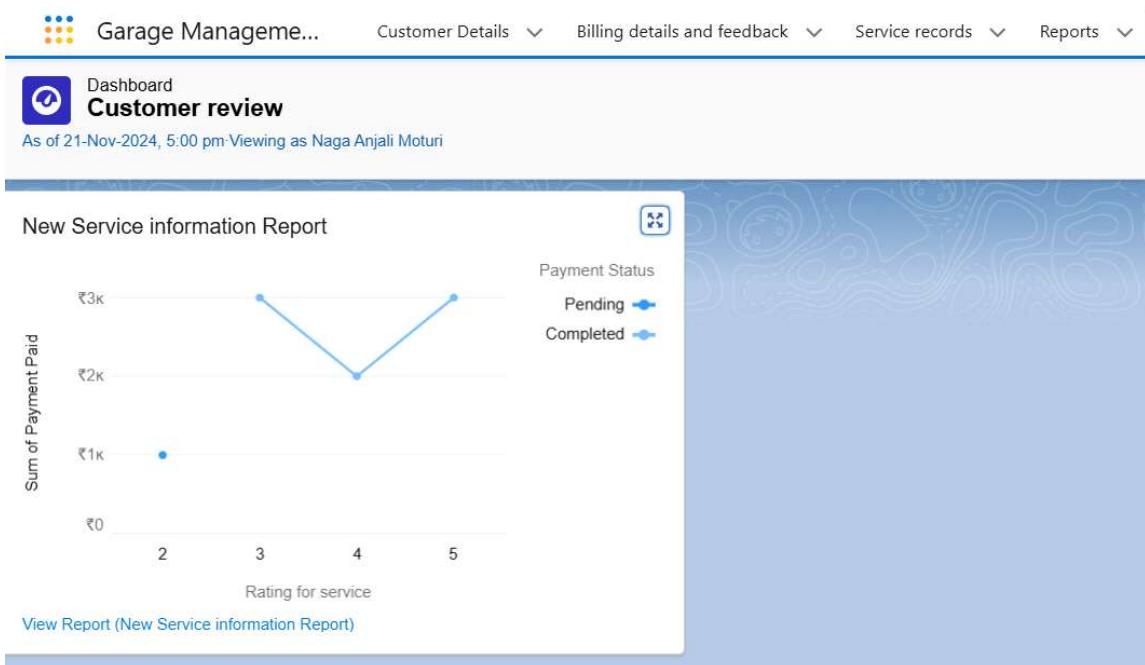
Create Dashboard Folder

- Click on the app launcher and search for dashboard.

- Click on dashboard tab.
- Click new folder,give the folderlabel as “ Service Ratingdashboard”.
- Folder unique name will be auto populated.
- Click save.
- Follow the same steps,form milestone 15, and activity2, and provide the sharing settings for the folder that just created.

Create Dashboard

- Go to the app >>click on the Dashboards tabs.
- Give a Name and select the folder that created, and click on create.
- Select add component.
- Select a Reportand click on select.
- Select the Line Chart. Change the theme.
- Click Add then click on Save and then click on Done.
- Preview is shown below.



Subscription:

- After that Click on Subscribe on top right.
- Set the Frequency as “ weekly”.
- Set a day as monday.
- And Click on save.

Edit Subscription

Schedule dashboard refreshes and subscribe to receive results.

Settings

Frequency

Days

Time

1:00 pm ▾

Recipients

 Recipients see the same report data as the person running the report.

Receive new results by email when dashboard is refreshed. 

Send email to

Me

[Edit Recipients](#)

[Unsubscribe](#)

[Cancel](#)

Conclusion:

The Garage Management System (GMS) implemented on the Salesforce platform successfully achieves its goal of streamlining and automating daily garage operations. By leveraging Salesforce’s cloud-based CRM capabilities, the system provides an integrated solution for managing customers, vehicles, service bookings, inventory, billing, and reporting—all within a single, scalable environment.

Through this project, we demonstrated how Salesforce can be customized using objects, workflows, process builders, and automation tools such as Flow and Apex to meet the unique needs of a garage business. The system improves operational efficiency, data accuracy, and customer satisfaction by ensuring real-time tracking of service requests, automated communication, and transparent billing processes.