

GARAGE MANAGEMENT APPLICATION

The Garage Management System (GMS) is an essential tool for automotive repair facilities, enabling them to deliver exceptional service, enhance operational efficiency, and cultivate strong customer relationships. With its intuitive interface and robust features, GMS equips garages to excel in a competitive market while ensuring a seamless, satisfying experience for both customers and staff. This comprehensive software solution is specifically designed to streamline and optimize the operations of garages, service centers, and repair facilities. It offers a wide range of features tailored to the needs of mechanics, service advisors, and business owners, promoting smoother workflows and greater customer satisfaction.

Appointment Scheduling:

- Streamlines the booking process for customers, making it simple and hassle-free.
- Allows staff to efficiently manage daily schedules, minimizing downtime and optimizing resource use.

Vehicle Management:

- Keeps detailed records of vehicles, including service history, repairs, and maintenance schedules.
- Monitors vehicle status during servicing, improving communication with customers.

Customer Relationship Management (CRM):

- Stores comprehensive customer information and preferences in one place.
- Sends automated service reminders, follow-ups, and promotional offers to encourage repeat business.

Inventory and Spare Parts Management:

- Tracks stock levels of spare parts, automates reordering, and prevents shortages.
- Ensures mechanics always have access to the necessary tools and parts.

Billing and Invoicing:

- Quickly generates accurate and professional invoices for customers.
- Offers support for multiple payment options, discounts, and tax calculations.

Reporting and Analytics:

- Delivers insights into key metrics like revenue, job completion rates, etc.

Salesforce

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".

What Is Salesforce?

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

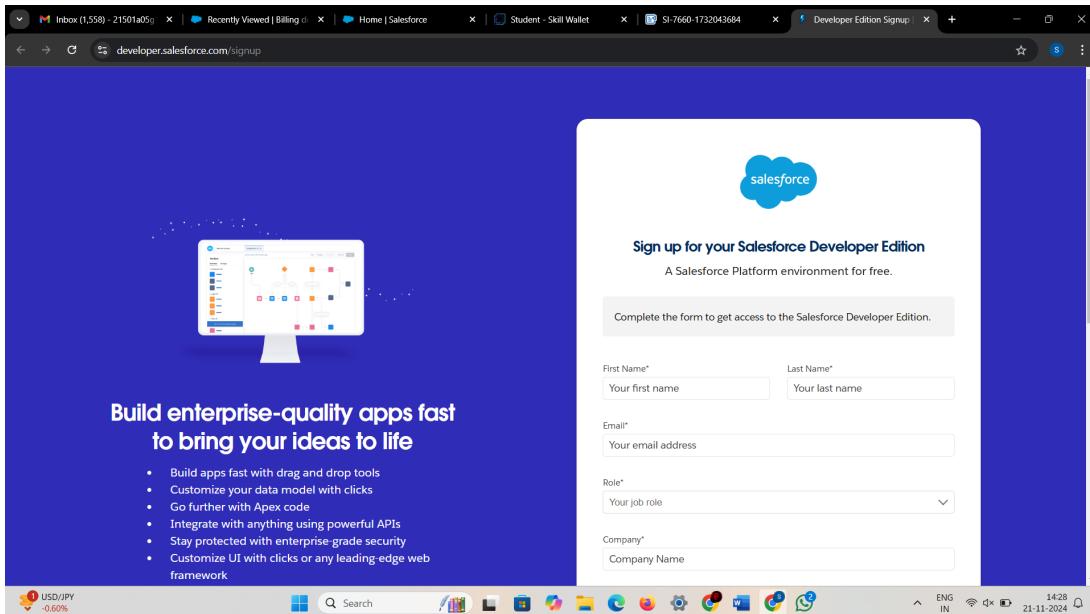
Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:



Creating Developer Account:

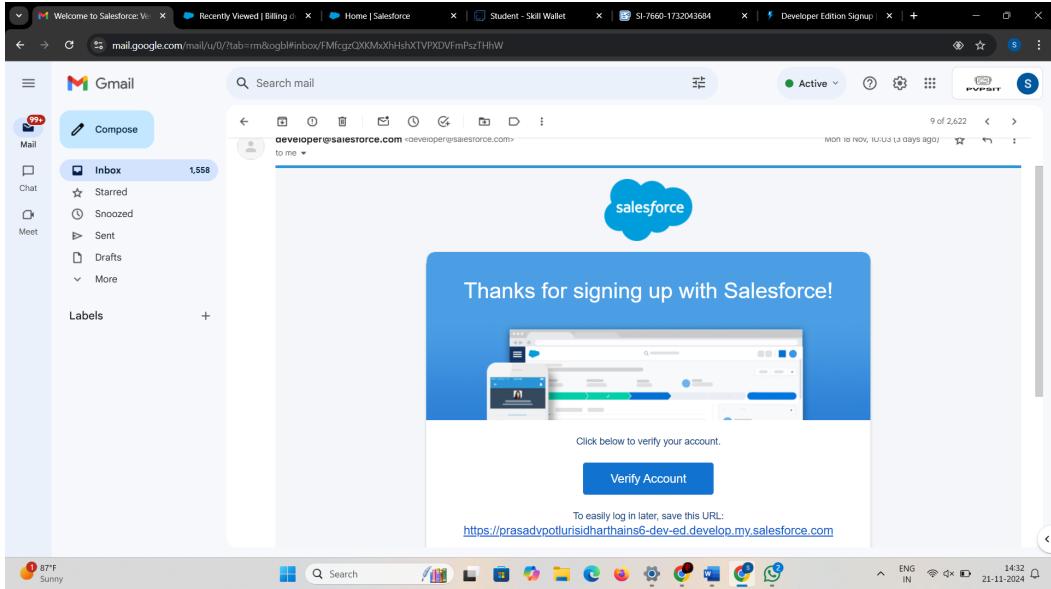
Creating a developer org in salesforce.

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
 - Country : 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.



Account Activation:

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



OBJECT

What Is an 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

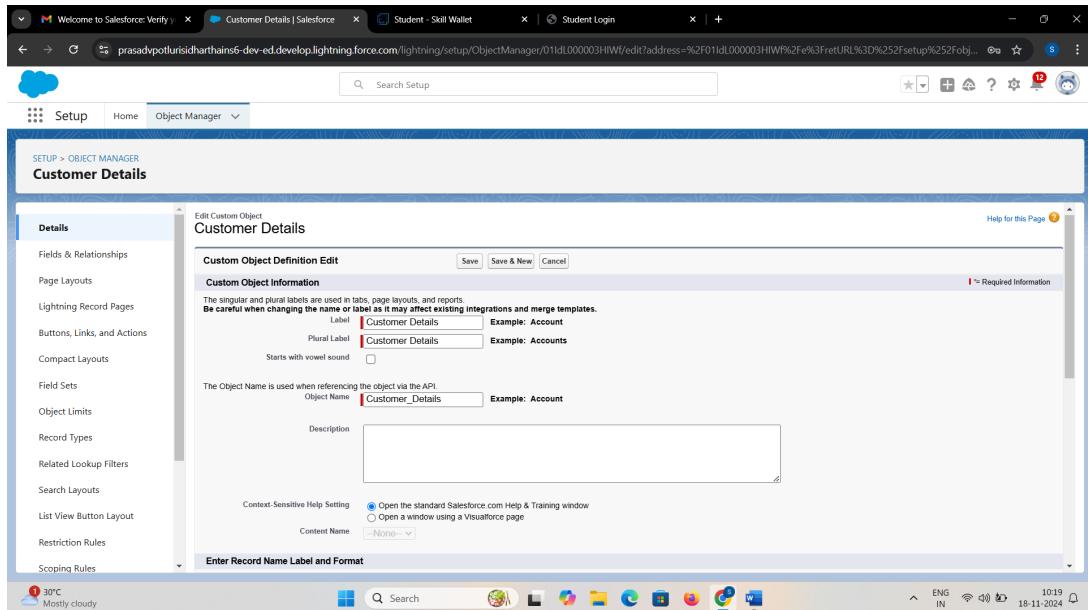
Salesforce objects are of two types:

1. **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **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.

The screenshot shows the Salesforce Object Manager page. At the top, there are tabs for 'Setup', 'Home', and 'Object Manager'. The main area displays a table of objects with columns: Label, API Name, Type, Description, Last Modified, and Deployed. A 'Create' button is located at the top right of the table. A 'Custom Object' button is highlighted with a red box. The table lists various standard objects such as Account, Activity, Address, Alternative Payment Method, API Anomaly Event Store, Appointment Category, Appointment Invitation, Appointment Invitee, Appointment Topic Time Slot, Asset, Asset Action, Asset Action Source, and Asset Relationship. The URL in the browser is <https://prasadypoturisidharthain6-dev-ed.lightning.force.com/lightning/setup/ObjectManager/home>.

Create Customer Details

1. Plural label name >>Customer Details
2. Enter Record Name Label and Format
 1. Record Name >> Customer Name
 2. DataType >> Text
1. Click on Allow reportsand Track Field History,
2. Allow search>> Save.

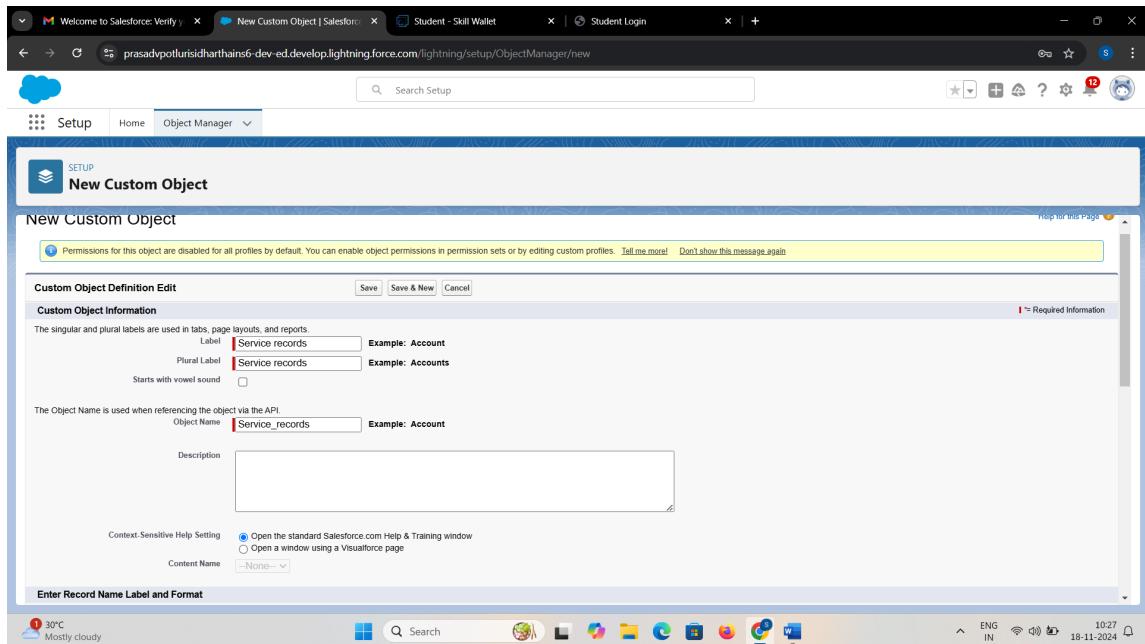


Create Service records Object

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on CustomObject.
1. Enter the label name >> Service records
2. Plural label name >> Service records
3. Enter Record Name Label and Format
1. Record Name >> Service records Name
2. Data Type >> Auto Number
3. Display Format >> ser-{000}
4. Starting number >> 1

1. Click on Allow reports and Track Field History,
2. Allow search >> Save.

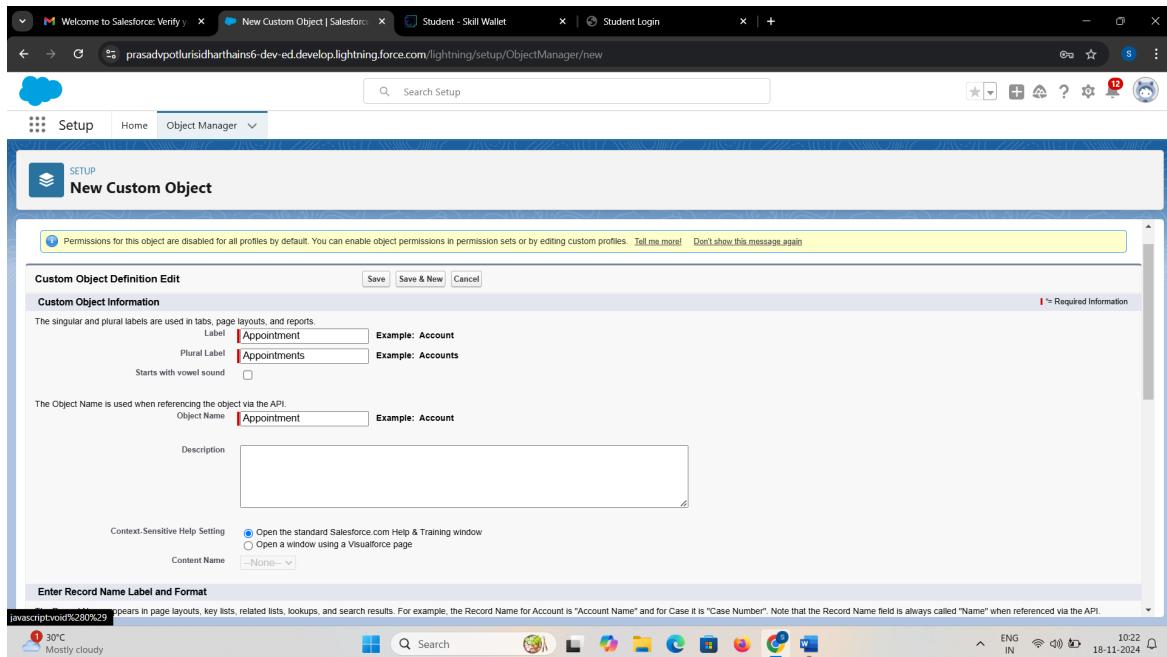


Create Appointment Object

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format
 1. Record Name >> Appointment Name
 2. Data Type >> Auto Number

3. Display Format >> app-{000}
4. Starting number>> 1
1. Click on Allow reports and Track Field History,
2. Allow search >> Save.

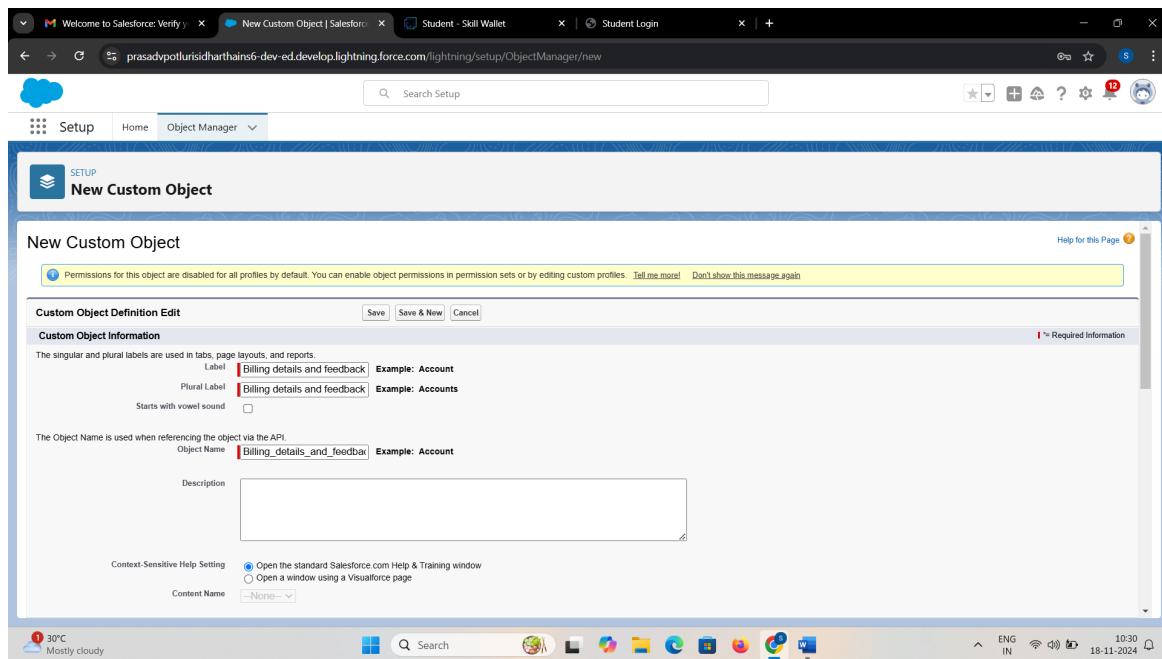


Create Billing details and feedback Object

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >>Click on Custom Object.
1. Enter the label name >> Billing details and feedback
2. Plural label name >>Billing details and feedback
3. Enter Record Name Label and Format

1. Record Name >> Billing details and feedback Name
 2. DataType >> Auto Number
 3. Display Format >> bill-{000}
 4. Starting number >> 1
1. Click on Allow reports and Track Field History,
 2. Allow search>> Save.



Tabs

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

Types of Tabs:

1. Custom Tabs

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

2. WebTabs

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

3. Visualforce Tabs

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

4. Lightning Component Tabs

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

5. Lightning Page Tabs

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

Creating a Custom Tab

To create a Tab:(Customer Details)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)
2. 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 .

3. Make sure that the Append tab to users' existing personal customizations is checked.

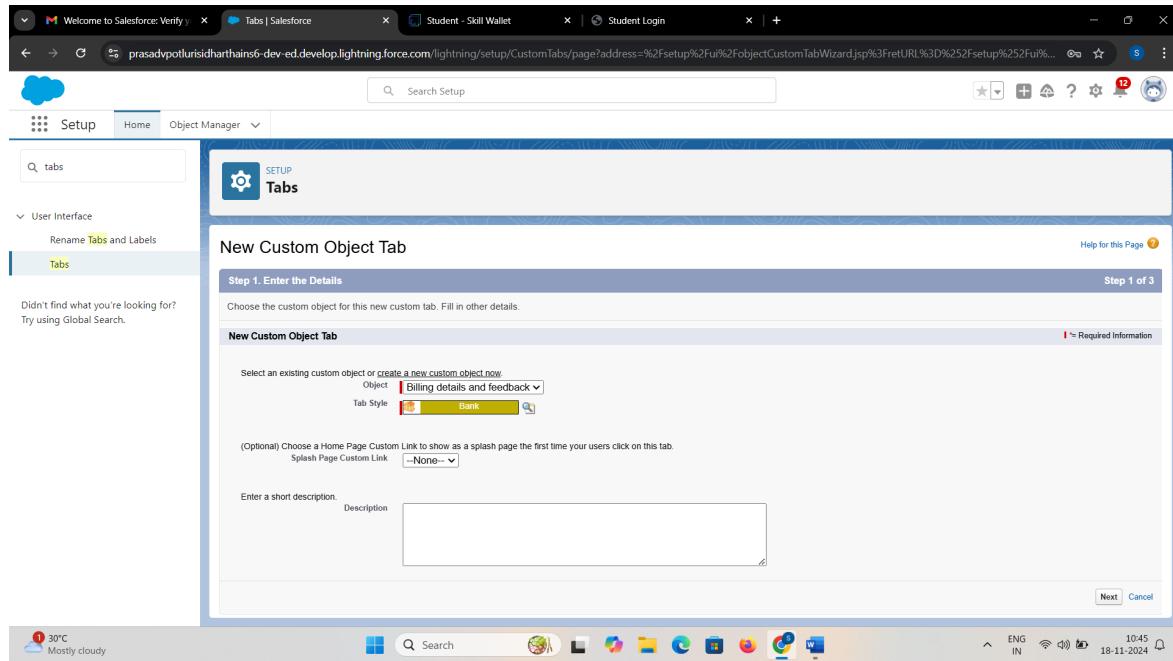
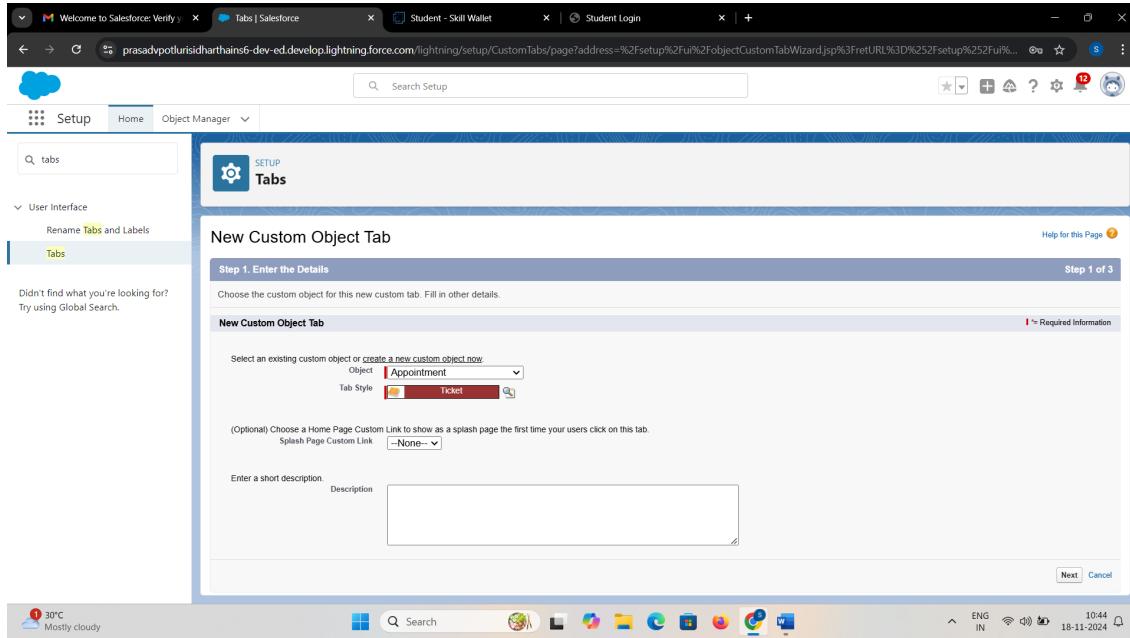
4. Click save.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected under 'User Interface'. The page title is 'Custom Tabs'. It displays sections for 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', 'Lightning Component Tabs', and 'Lightning Page Tabs', each with 'New' and 'What Is This?' buttons. A search bar at the top left and a help link at the top right are also visible.

The screenshot shows the 'New Custom Object Tab' wizard, Step 1: Enter the Details. It asks to choose a custom object or create a new one. 'Customer Details' is selected as the object. Other fields include 'Tab Style' (set to 'People'), a 'Splash Page Custom Link' dropdown (set to 'None'), and a 'Description' text area. A 'Next' button is at the bottom right.

Creating Remaining Tabs

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



The screenshot shows the Salesforce Setup interface with the 'Tabs' tab selected under 'User Interface'. A search bar at the top has 'tabs' typed into it. The main content area is titled 'New Custom Object Tab' and is divided into 'Step 1. Enter the Details' and 'Step 2. Define the Tab'. In Step 1, there is a note: 'Choose the custom object for this new custom tab. Fill in other details.' Below this, a section titled 'New Custom Object Tab' asks to 'Select an existing custom object or create a new custom object now.' An 'Object' dropdown is set to 'Service records'. A 'Tab Style' dropdown is set to 'Square'. A note below says '(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.' A 'Splash Page Custom Link' dropdown is set to 'None'. There is also a 'Description' field with a placeholder 'Enter a short description.' At the bottom right are 'Next' and 'Cancel' buttons.

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected under 'User Interface'. A search bar at the top has 'tabs' typed into it. The main content area is titled 'Custom Tabs' and includes a note: 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' It describes four types of tabs: Custom Object tabs, Web tabs, Visualforce tabs, and Lightning Component tabs. Under 'Custom Object Tabs', there is a table:

Action	Label	Tab Style	Description
Edit Del	Appointments	Ticket	
Edit Del	Billing details and feedback	Bank	
Edit Del	Customer Details	People	
Edit Del	Service records	Square	

Below this, sections for 'Web Tabs' and 'Visualforce Tabs' both state 'No [Type] Tabs have been defined.'

The Lightning App

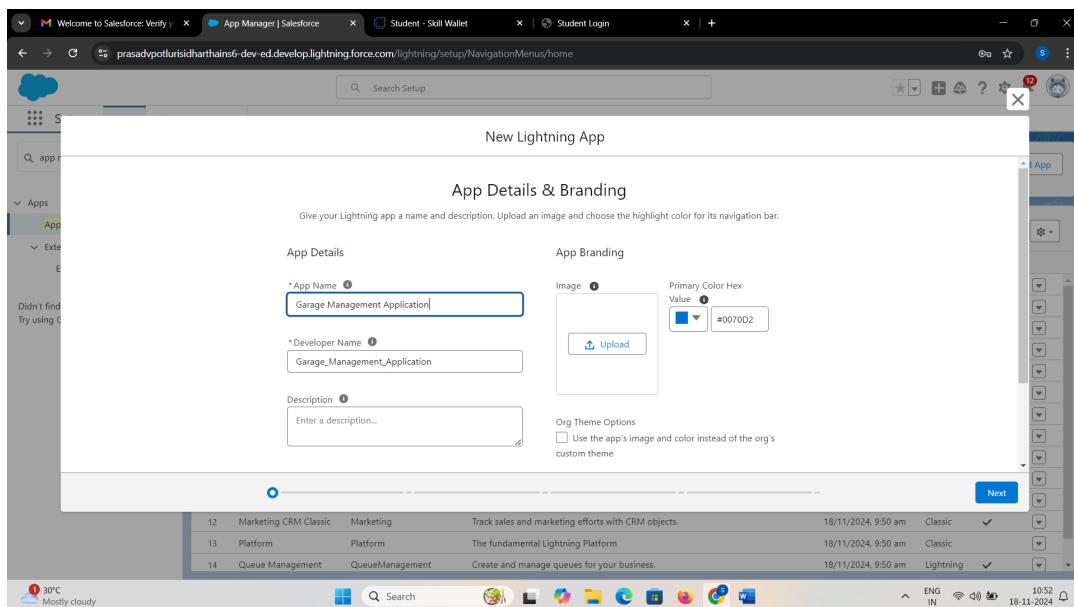
An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

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:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
2. 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.
3. To Add Navigation Items:
4. 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.
5. To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields

1. Standard Fields
2. Custom Fields

Creation of fields for the Customer Details object

1. To create fields in an object:

- a. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.
- b. Now click on "Fields & Relationships" >> New
- c. Select Data Type as a "Phone"
- d. Click on next.
- e. Fill the Above as following:

1. Field Label: Phone number
2. Field Name : gets auto generated
3. Click on Next >> Next >> Save and new.

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

2. To create another fields in an object:

- a. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.
- b. Now click on "Fields & Relationships" >> New
- c. Select Data type as a "Email" and Click on Next
- d. Fill the Above as following:
- e. Field Label : Gmail
- f. Field Name : gets auto generated
- g. Click on Next >> Next >> Save and new.

The screenshot shows the Salesforce Object Manager interface. The left sidebar has 'Fields & Relationships' selected under 'Customer Details'. The main area displays a table titled 'Fields & Relationships' with four items. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		✓
Owner	OwnerId	Lookup(User,Group)		✓

Creation of Lookup Fields

Creation of Lookup Field on Appointment Object :

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

Note: Make sure you complete Activity4 Before continuing.

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

- Setup** tab selected.
- Object Manager** selected under the **Appointment** object.
- Fields & Relationships** section selected.
- Fields & Relationships** table:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

Creation of Lookup Field on Service records Object :

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

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Service records Name	Name	Auto Number		✓

Creation of Lookup Field on Billing details and feedback Object :

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

Step 1. Choose the field type

Specify the type of information that the custom field will contain.

Data Type

Lookup Relationship

Select one of the data types below:

- None Selected
- Auto Number
- Formula
- Roll-Up Summary
- Master-Detail Relationship

A system-generated sequence number that uses a lookup format you define. The number is automatically incremented for each new record.

A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a pop-up list. The other object is the source of the values in the list.

The relationship field is required on all detail records of the master object. The values shown in a detail list are determined by the master record.

When a user deletes the master record, all detail records are deleted.

You can create rollup summary fields on the master record to summarize the detail records.

The relationship field allows users to click on a lookup icon to select a value from a pop-up list. The master object is the source of the values in the list.

Creation of Checkbox Fields

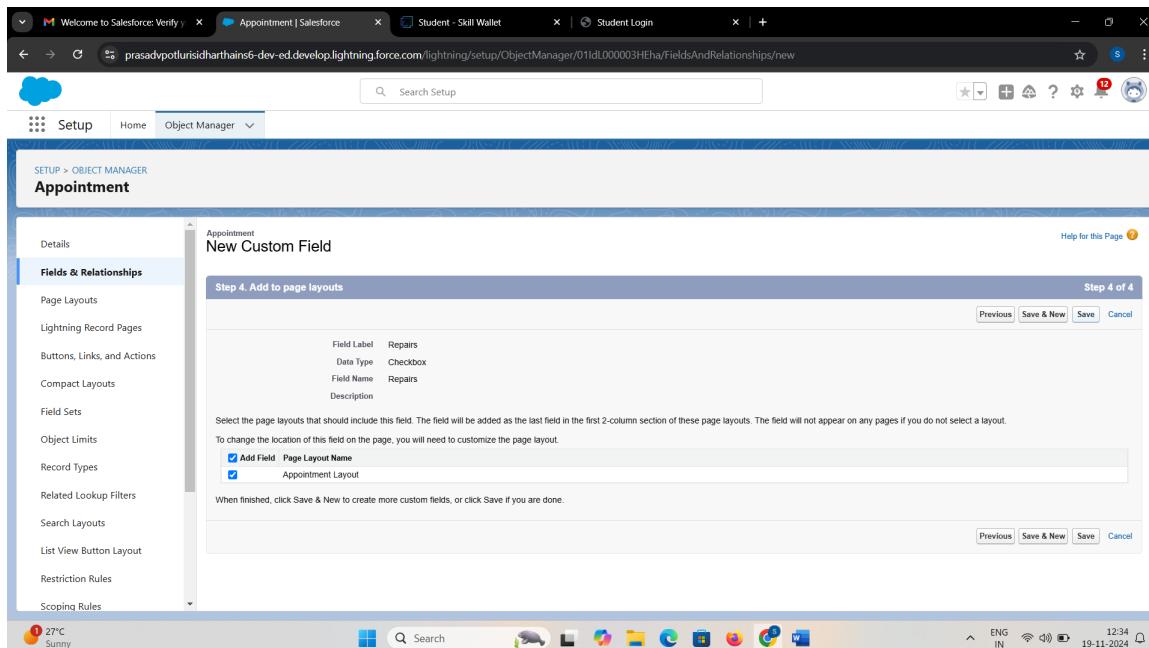
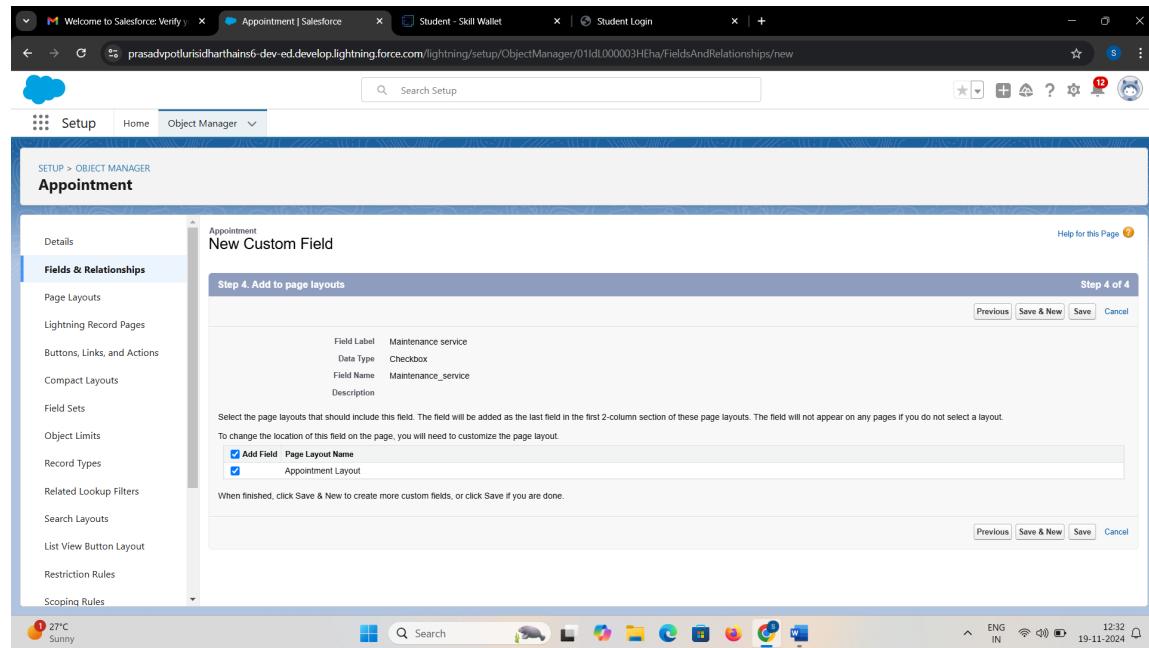
Creation of Checkbox Field on Appointment Object :

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

Creation of Another Checkbox Field on Appointment Object :

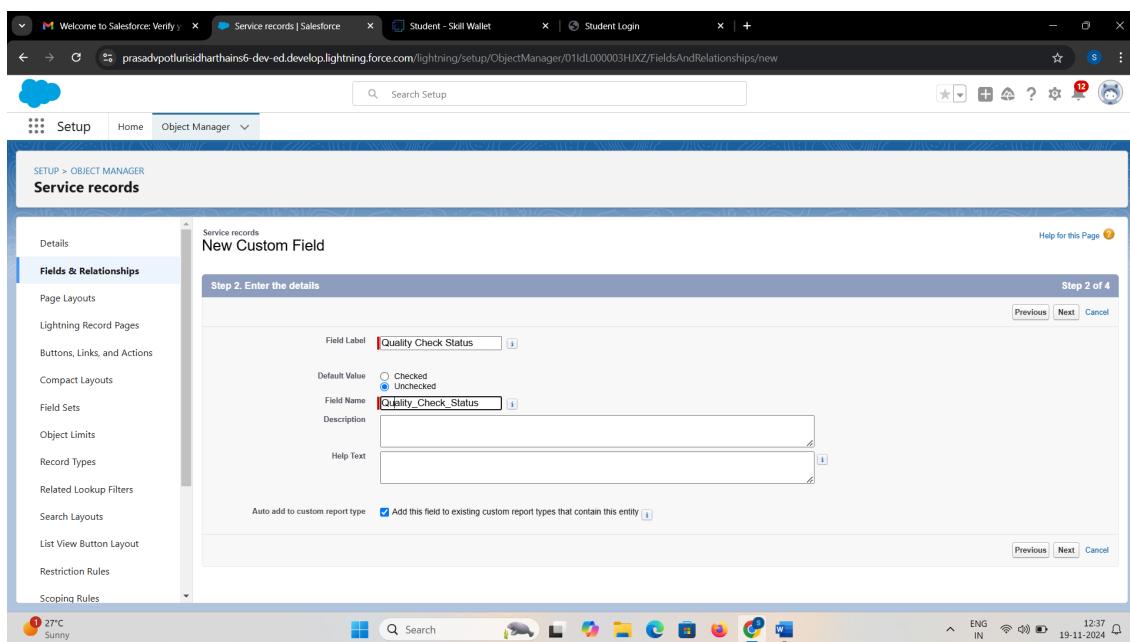
1. Repeat the steps from 1 to 3.
2. Give the Field Label : Repairs
3. Field Name : is auto populated
4. Default value : unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkbox with given names.
7. Give the Field Label : Replacement Parts
8. Field Name : is auto populated
9. Default value : unchecked

10. Click on next >>next >> save.



Creation of Checkbox Field on Service records Object :

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



Creation of date Fields

Creation of Date Field on Appointment Object :

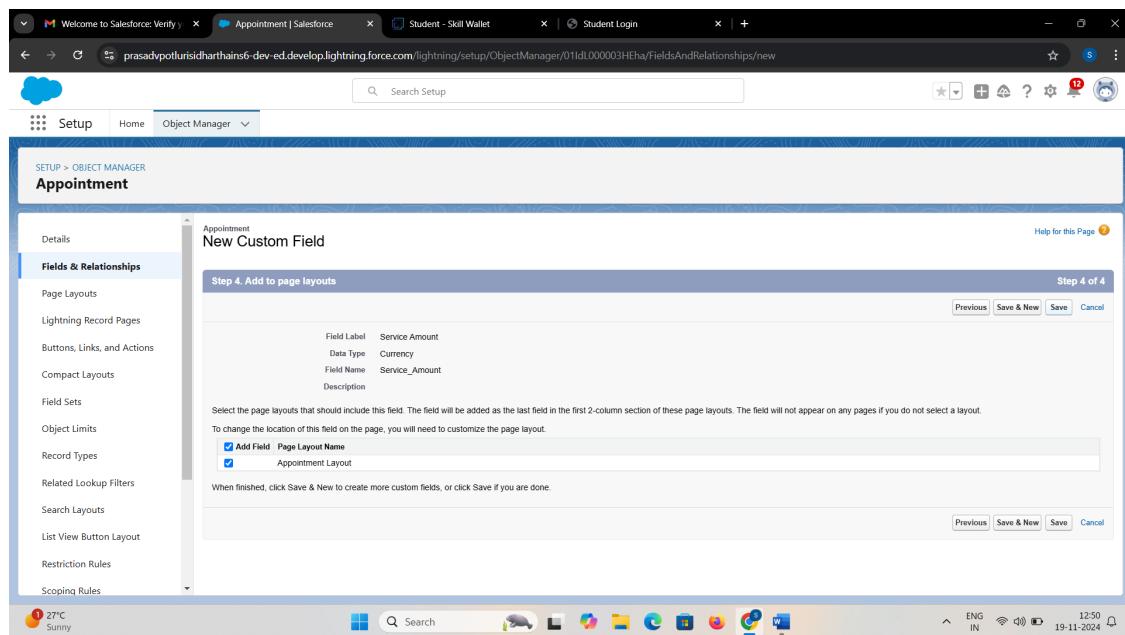
1. Go to setup >>click on Object Manager >>type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields& Relationships” >>New.
3. Select “Date” as data type and click Next.

4. Give the Field Label : Appointment Date
5. Field Name : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >>next >> save.

Creation of Currency Fields

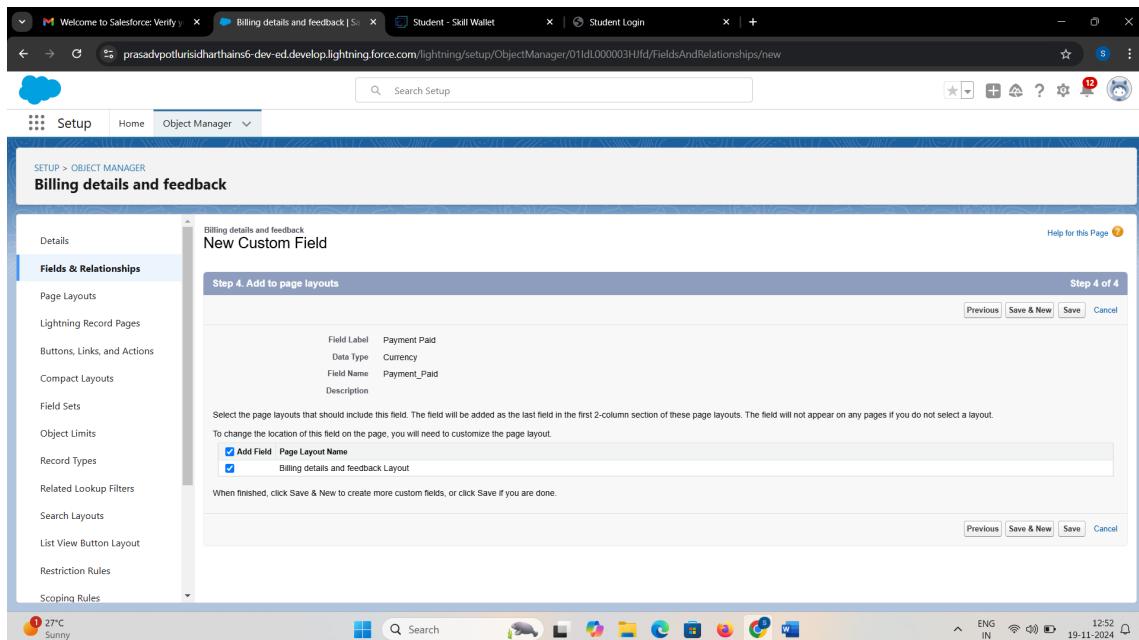
Creation of Currency Field on Appointment Object :

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



Creation of Currency Field on Billing details and feedback Object :

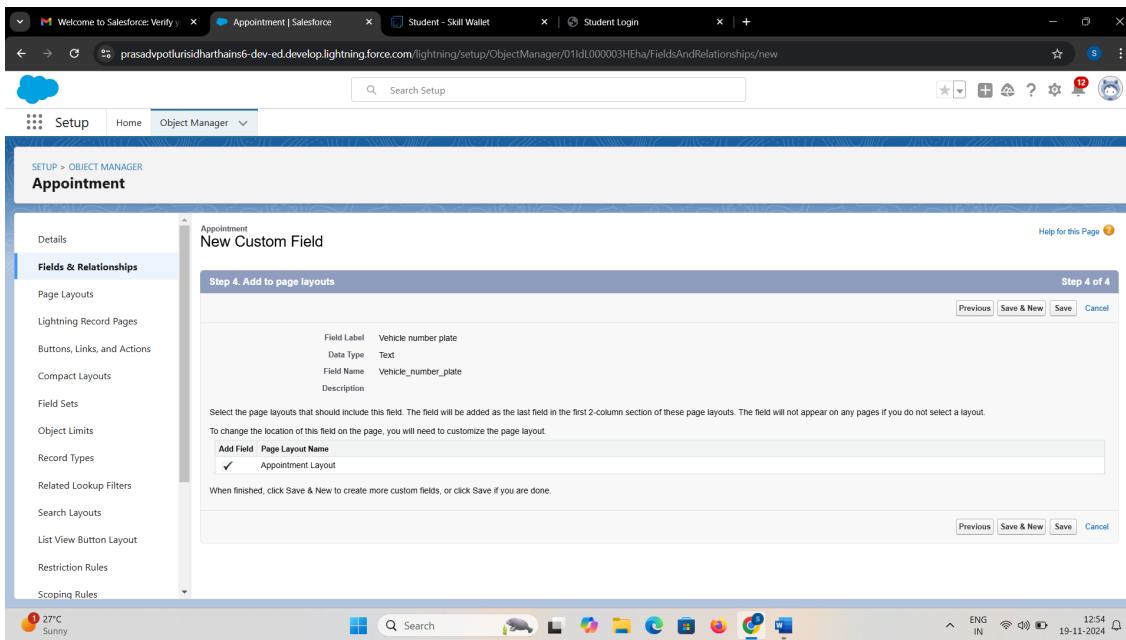
1. Follow the same steps as mentioned above in Billing details and feedback Object.
2. Change the label name as mentioned.
3. Give the Field Label : PaymentPaid.
4. Field Name : is auto populated.



Creation of Text Fields

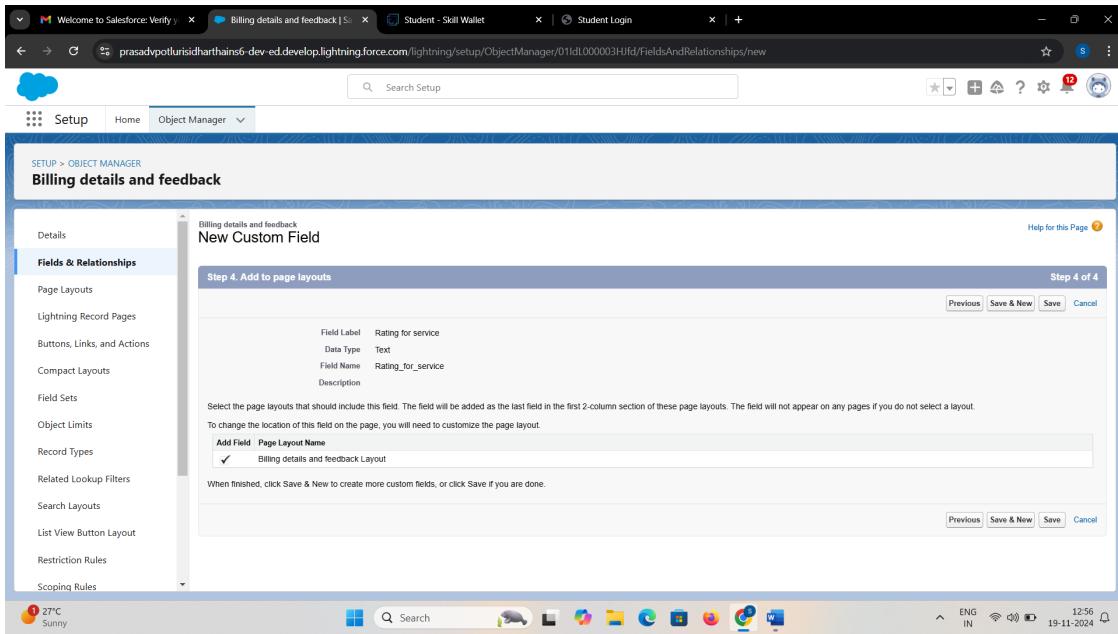
1. Go to setup >>click on Object Manager >>type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >>New.
3. Select “Text” as data type and click Next.
4. Give the Field Label : Vehicle number plate.
5. Field Name : is auto populated.
6. Length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

8. Click on next >>next >> save.



Creation of Text Fields in Billing details and feedback object :

1. Go to setup >> click on Object Manager >>type object name(Billing details and feedback) in searchbar >> click on the object.
2. Now click on “Fields& Relationships” >>New.
3. Select “text” as data type and click Next.
4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required and Unique.
8. Click on next >> next >> save.



Creation of Picklist Fields

Creation of Picklist Fields in Service records object :

1. Go to setup >>click on ObjectManager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Picklist" and click Next.
4. 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.
5. The values are: Started , Completed.
6. Click Next.
7. Next >>Next >> Save.

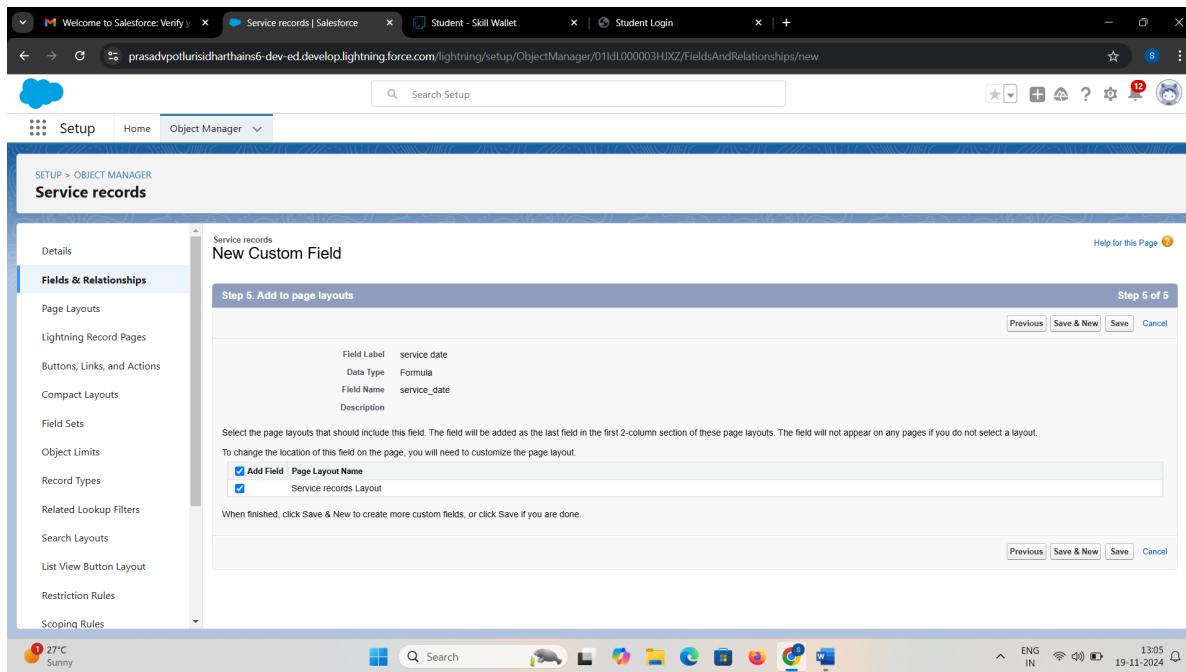
Creation of Picklist Fields in Billing details and feedback object :

1. Go to setup >>click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Picklist" and click Next.

4. 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.
 5. The values are: Pending, Completed.
 6. Click Next.
- Next >>Next >> Save.

Creating Formula Field in Service records Object

1. Go to setup >>click on ObjectManager >> type object name(Service records) in search bar >> clickon the object.
2. Click on fields & relationship >>click on New.
3. Select Data type as "Formula" and click Next.
4. Give Field Label and Field Name as "service date" and select formula return type as "Date"and click next.
5. Insert field formula should be : CreatedDate.
6. click "Check Syntax" .
7. Click next >> next >> Save.

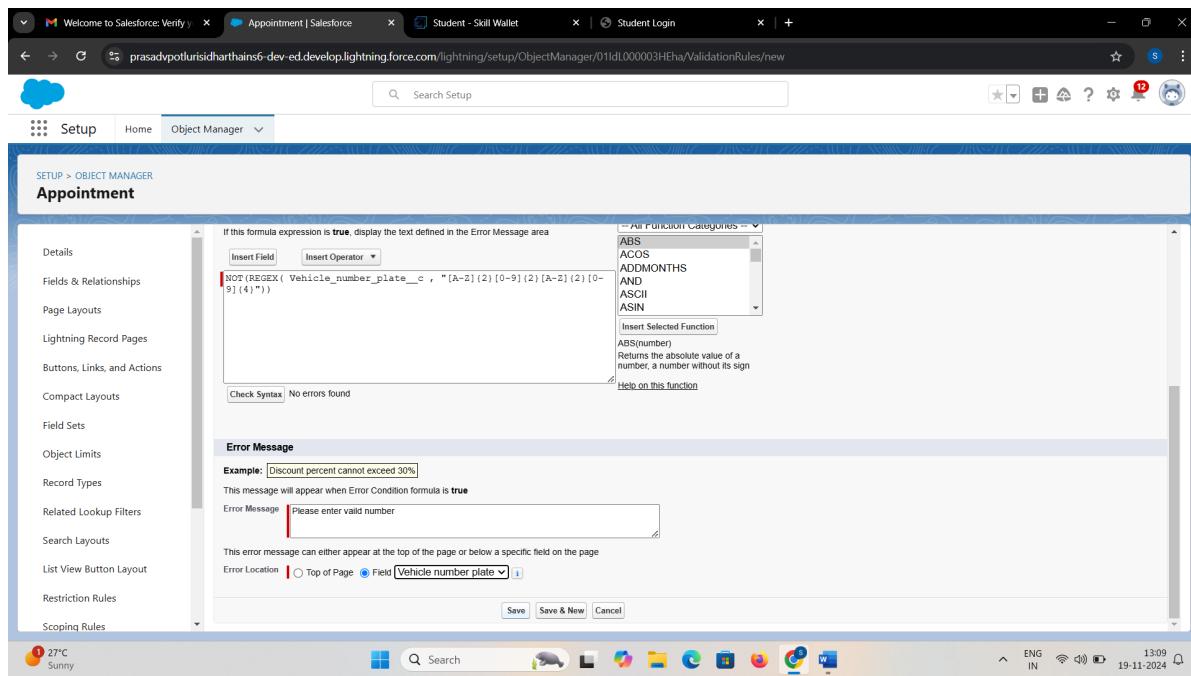


Validation rule

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

To create a validation rule to an Appointment Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as " Vehicle ".
4. Insert the Error Condition Formula as :-
5. NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))
6. Enter the Error Message as "Please enter valid number ", select the Error location as Field and select the field as "Vehicle number plate", and click Save.

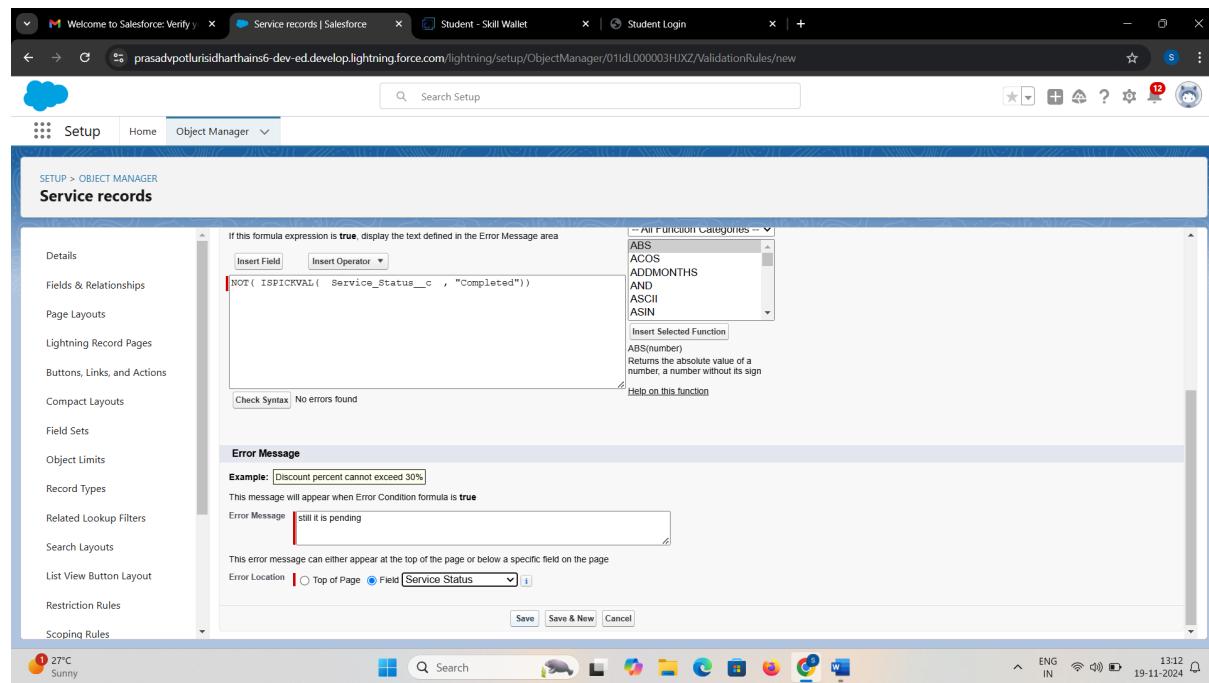


To create a validation rule to an Service records Object

1. Go to the setup page >> click on object manager >> From drop down click edit forService records object.
2. Click on the validation rule >> clickNew.
3. Enter the Rule name as “ service_status_note ”.
4. Insert the Error Condition 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.

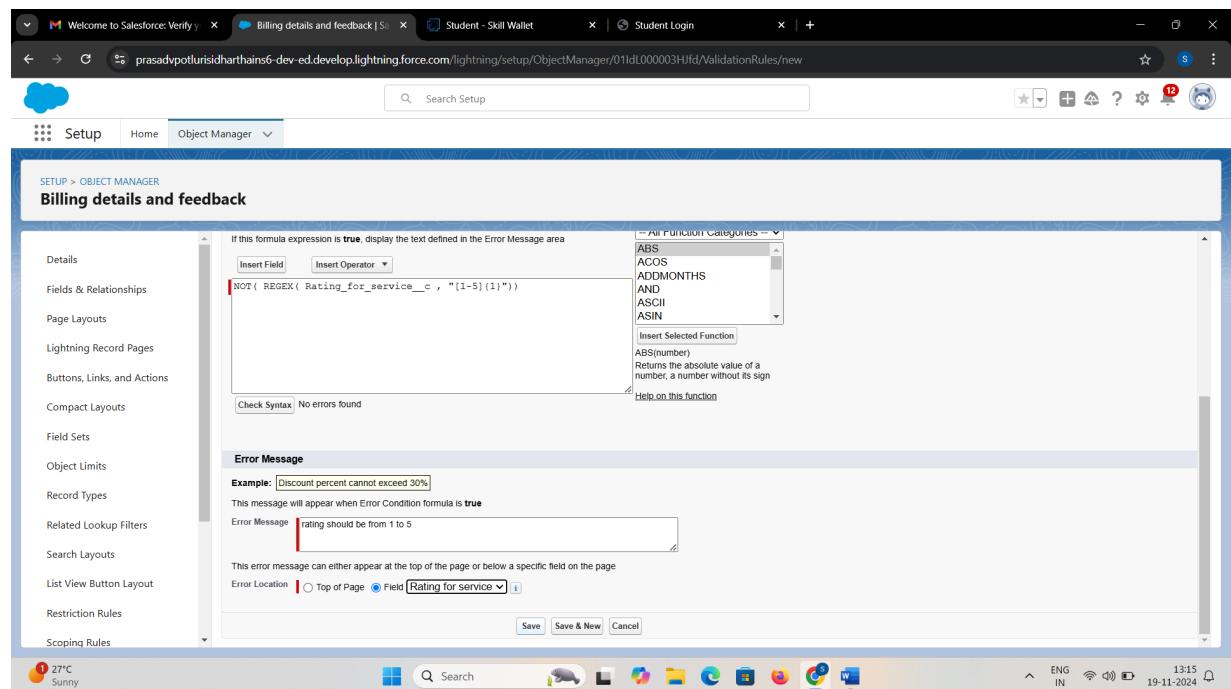


To create a validation rule to an Billing details and feedback Object

1. Go to the setup page >> click on object manager>> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating_should_be_less_than_5”.
4. Insert the Error Condition Formula as :-

`NOT(REGEX(Rating_for_service_c, "[1-5]{1}"))`

Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”, and click Save.



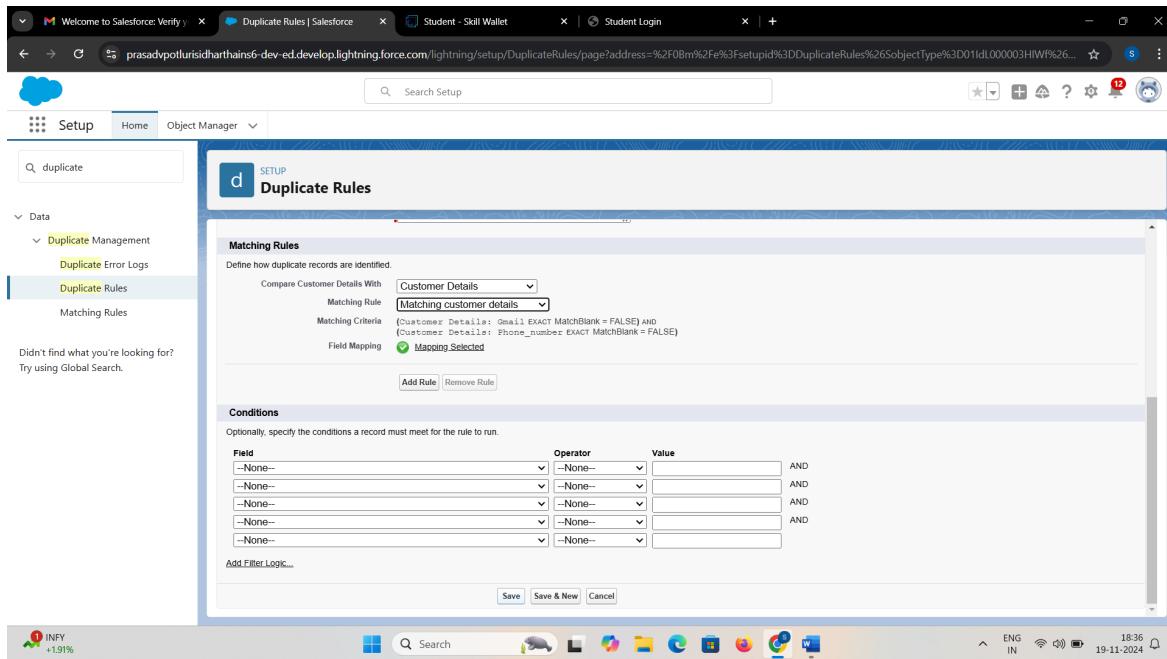
Duplicate rule

To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
 2. Click on matching rule >> click on New Rule.
 3. Select the object as Customer details and click Next.
 4. Give the Rule name : Matching customer details
 5. Unique name : is auto populated
 6. Define the matching criteria as
 7. Field Matching Method
 - a. Gmail Exact
 - b. Phone Number Exact
 8. Click save.
 9. After Saving Click on Activate.

To create a Duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
 2. Click on Duplicate rule >> click on New Rule >> select customer details object.
 3. Give the Rule name as : Customer Detail duplicate
 4. Scroll a little in Matching rule section
 5. Select the matching rule : Matching customer details
 6. And Click on save.
 7. After saving the Duplicate Rule, Click on Activate.



Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls "Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example SystemAdministrator, Developer, SalesRepresentative.

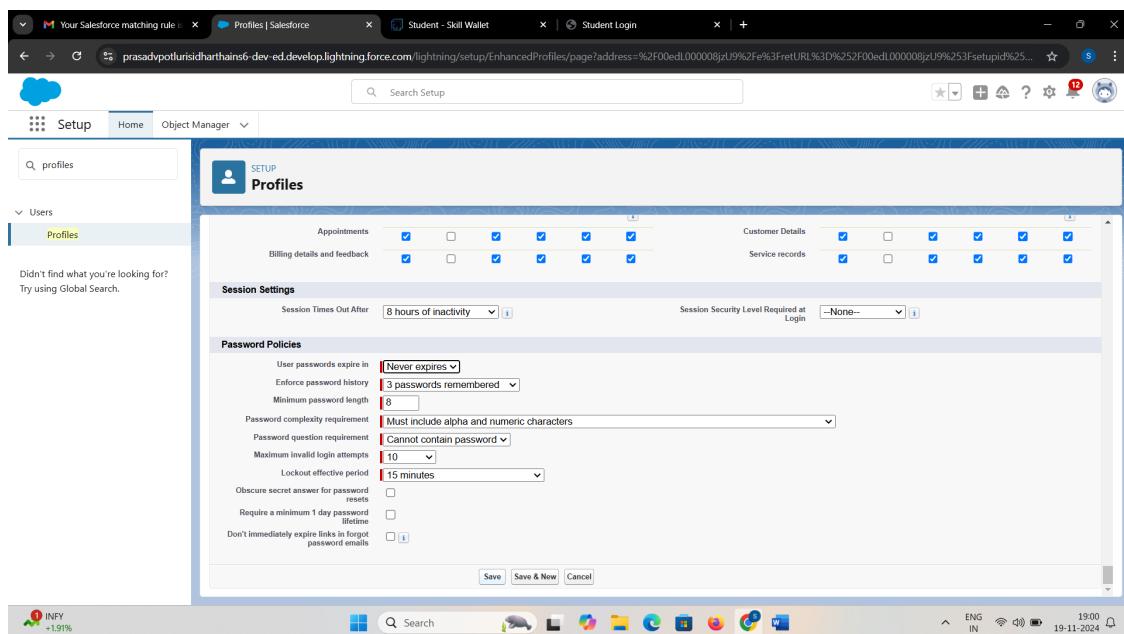
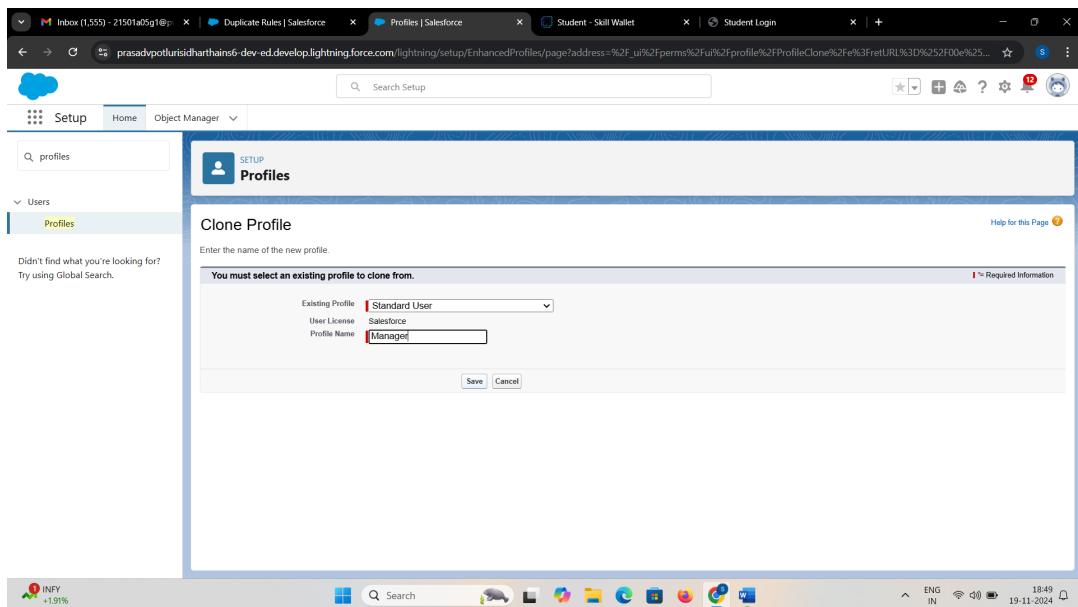
Manager Profile

To create a new profile:

- Goto setup >>type profiles in quick find box >>click on profiles>> clone thedesired profile (Standard User) >> enter profile name (Manager) >>Save.
- While still on the profile page, then click Edit.
- Select the CustomApp settings as default for the Garage management.
- Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing detailsand feedback , service

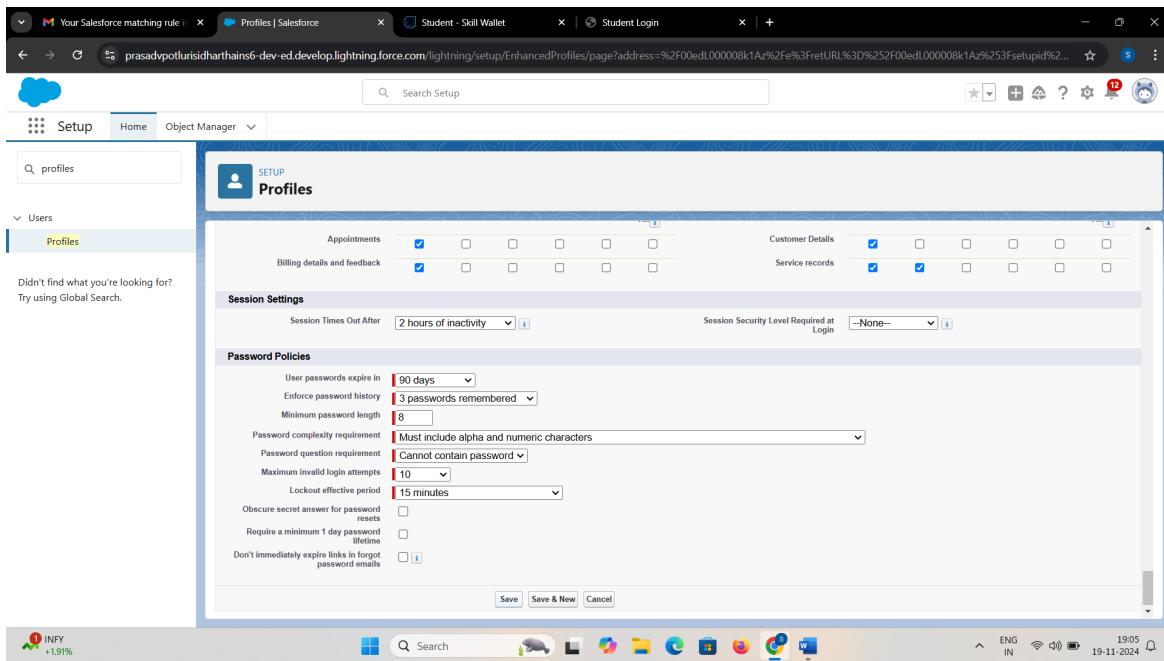
records and customer details objects as mentioned in the below diagram.

- e. Changing the session times out after should be “ 8 hours of inactivity”.
- f. Change the password policies as mentioned :
- g. User passwords expire in should be “ never expires”.
- h. Minimum password length should be “ 8 ”, and click save.



Sales person Profile

1. Goto setup >>type profiles in quick find box >>click on profiles>>clone the desired profile (Salesforce Platform User) >> enter profile name (sales person)>> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Garage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments ,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
5. And click save.

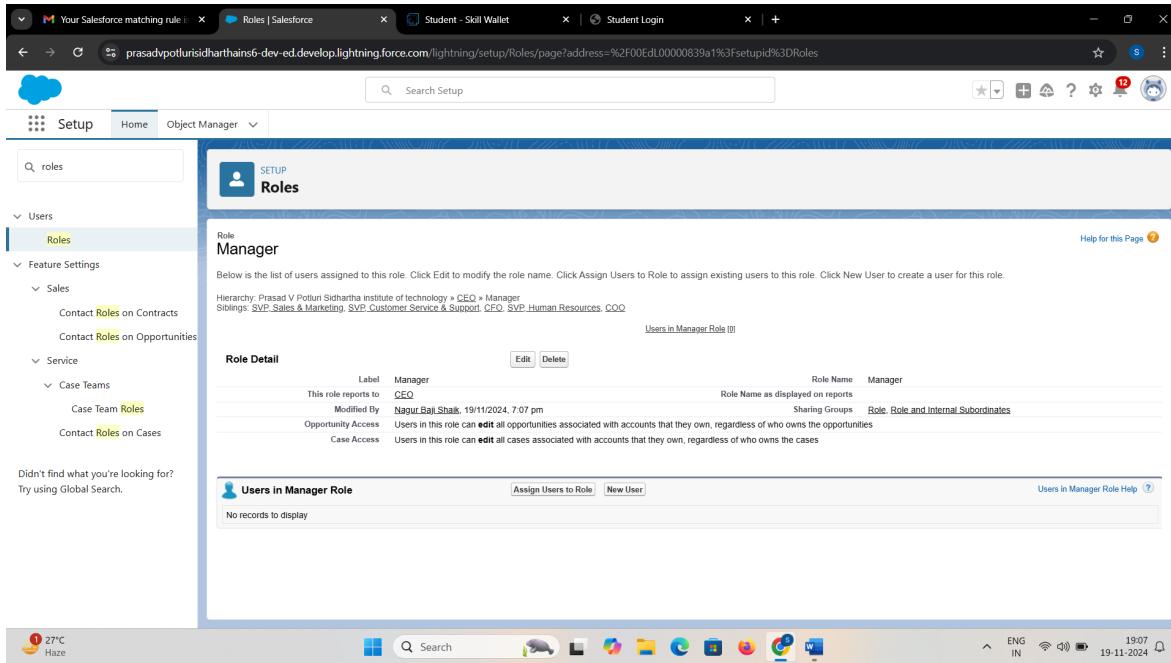


Role & Role Hierarchy

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

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



Creating another roles

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

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with various categories like Users, Roles, Sales, Service, etc. The 'Roles' section is currently selected. In the main content area, a page titled 'Roles' displays a role named 'sales person'. The 'Role Detail' section shows the following information:

	Label	Role Name
This role reports to	sales person	Role Name as displayed on reports
Modified By	Manager	Sharing Groups
Opportunity Access	Nagur Bai Shakk	Role, Role and Internal Subordinates
Case Access	Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities	
	Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases	

Below the table, there's a section for 'Users in sales person Role' with a button to 'Assign Users to Role' and a link to 'New User'. The status bar at the bottom shows the date and time as 19-11-2024 19:08.

Users

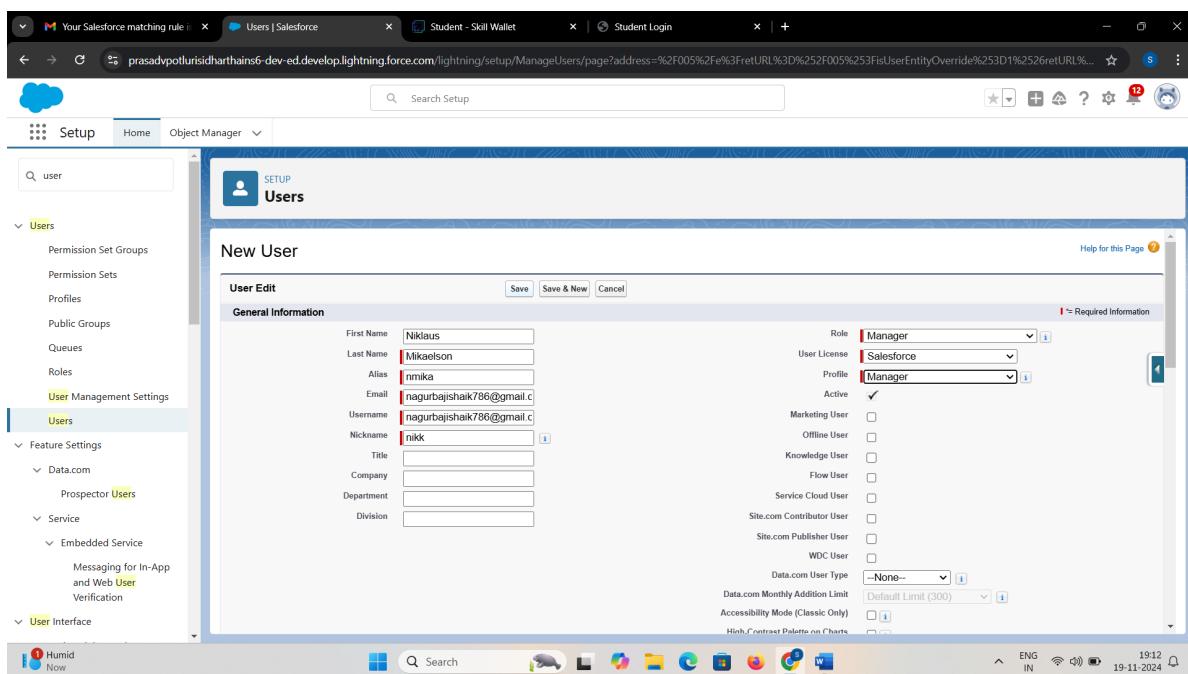
A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Create User

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
 - a. First Name : Niklaus

- b. Last Name : Mikaelson
- c. Alias : Give a AliasName
- d. Email id : Give your Personal Emailid
- e. Username : Username should be in this form:text@text.text
- f. Nick Name : Give a Nickname
- g. Role : Manager
- h. User licence : Salesforce
- i. Profiles : Manager

Save.



creating another users

1. Repeat the steps and create another user using

- a. Role : sales person
- b. User licence : Salesforce Platform
- c. Profile : sales person

Note : create atleast 3 userswith these permissions.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter_Expert	Chatter	chatty@00dd000000fdbuav.hdt53dgh4vau3@chatter.salesforce.com	salesperson	<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	kumar_sai	skuma	sai@kumar.in	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Mikaelson_Niklaus	nmika	nagurnaishak786@gmail.com	salesperson	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	shak_fasi	fshai	fasi@shak.in	salesperson	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	Shaik_Nagur_Bail	NShai	bail@povsit.in	System Administrator	<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Integration	integ	integration@00dd000000fdbuav.com	salesperson	<input checked="" type="checkbox"/>	Analytics Cloud Security User
<input type="checkbox"/>	User_Security	sec	insightsecurity@00dd000000fdbuav.com	salesperson	<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	xb_Shankar	xbts	shankari@xbts.in	salesperson	<input checked="" type="checkbox"/>	sales person

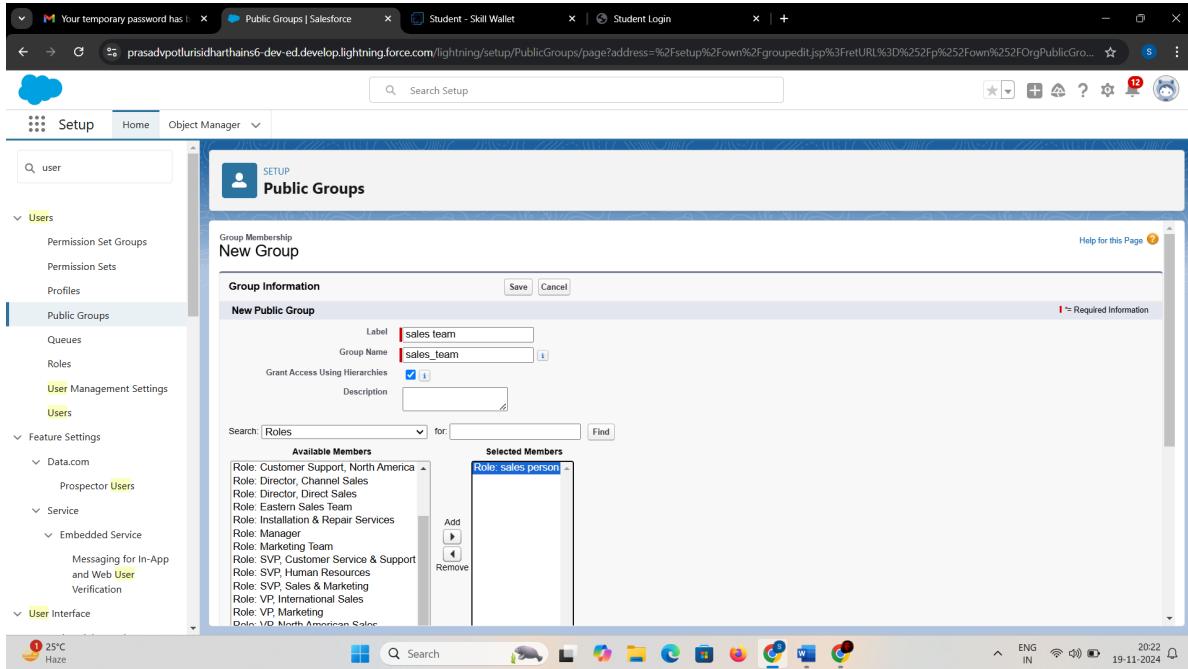
Public groups

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

1. Go to setup >>type users in quick find box >>select public groups>> click New.
2. Give the Label as "sales team".
3. Group name is auto populated.

4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.



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.

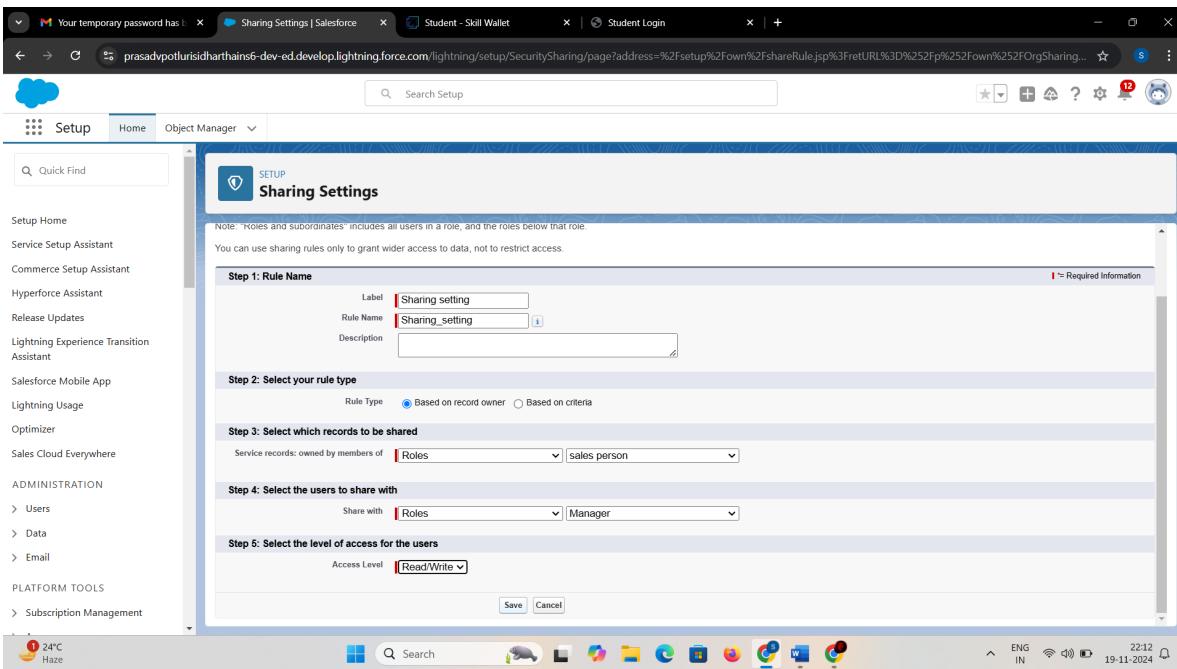
Profiles and Permission Sets:

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

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



Flows

Create a Flow:

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the Record-triggered flow and Click on Create.
3. Select the Object as "Billing details and feedback" in the Drop down list.
4. Select the Trigger Flow when: "A record is Created or Updated".
5. Select the Optimizer the flow for: "Actions and Related Records" and Click on Done.
6. 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
7. API name : is auto populated
8. Set a filter condition : All Conditions are met(AND)
9. Field : Payment_Status_c
10. Operator : Equals
11. Value : Completed
12. And Set FieldValues for the Billing details and feedback Record
13. Field : Payment_Paid_c
14. Value : {!\$Record.Service_records_r.Appointment_r.Service_Amount_c}
15. Click On Done. Before creating another Element. Create a New Resource

formToolbox form top left.

16. Click on the New Resource, And select Variable.
17. Select the resourcetype as text template.
18. Enter the API name as " alert".
19. Change the view as Rich Text ? View to Plain Text.
20. In body field paste the syntax that given below.

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

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is 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}

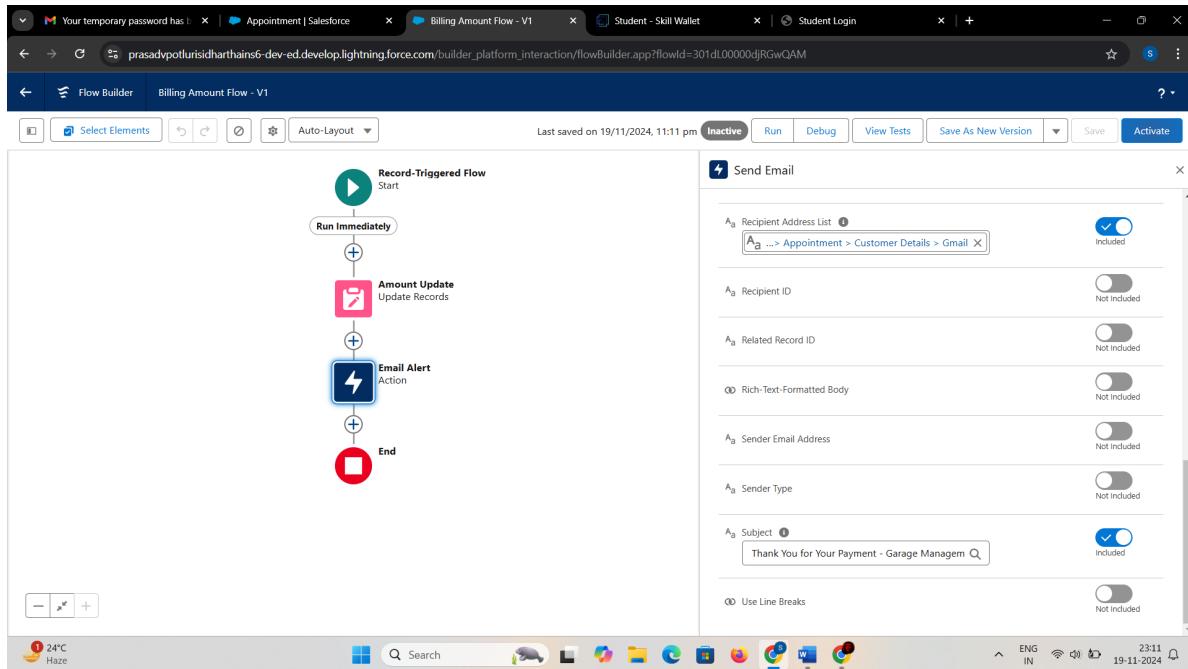
Thank you for Coming .

1. Click done.
2. Now Click on Add Element, select Action.
3. Their action bar will be opened in that search for " send email " and click on it.
4. Give the label name as " Email Alert"
5. API name will be auto populated.
6. Enable the body in set input values for the selected action.
7. Select the text template that created , Body : {!alert}
8. Include recipient address list select the email from the record.
9. RecipientAddressList:

{\$Record.Service_records_r.Appointment_r.Customer_Name_r.Gmail_c}

10. Include subject as " Thank You for Your Payment - Garage Management".
11. Click done.
12. Click on save. Give the Flow label , Flow API name will be autopopulated.

13. 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:

1. insert
2. update
3. delete
4. merge
5. upsert
6. 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.

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

The screenshot shows the Salesforce Developer Console interface. The code editor displays the following Apex class:

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

The logs section shows the following entry:

User	Application	Operation	Time	Status	Read	Size
Nagar Baj Shai	Browser	/aura	11/23/2024, 2:50:46 PM	Success	Unread	272 bytes

The status bar at the bottom indicates the following information: ENG IN, 88°F, Partly sunny, 15:51, 23-11-2024.

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes links to Welcome to Salesforce, Report Types, Customer review, Developer Console, Student - Skill Wallet, and a session ID. The main area displays the code for `AmountDistributionHandler.apxc`. The code implements a series of nested `else if` statements to calculate `app.Service_Amount_c` based on the values of `app.Maintenance_service_c`, `app.Repairs_c`, and `app.Replacement_Parts_c`. The developer console also shows a log entry for a successful operation by user Nagur Baji Shaik.

```
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:

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment_c

Syntax For creating trigger :

The syntax for creating trigger is :

Trigger [triggername] on [objectname](Before/After event)

{

```
}
```

In this project , trigger is called when ever 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);  
    }  
}
```

The screenshot shows the Salesforce Developer Console interface. At the top, there are several tabs: 'Welcome to Salesforce', 'Report Types | Salesforce', 'Customer review | Salesforce', 'Developer Console', 'Student - Skill Wallet', and a session ID 'SI-7660-1732043684'. Below the tabs, the main area displays the Apex trigger code for 'AmountDistribution' on 'Appointment_c'. The code is as follows:

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

Below the code, there is a table titled 'Logs' showing the execution details:

User	Application	Operation	Time	Status	Read	Size
Nagur Baji Shak	Browser	/aura	11/23/2024, 2:50:46 PM	Success	Unread	272 bytes

At the bottom of the screenshot, the Windows taskbar is visible, showing the date '23-11-2024' and time '15:54'.

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

1. Tabular

2. Summary

3. Matrix

4. Joined Reports

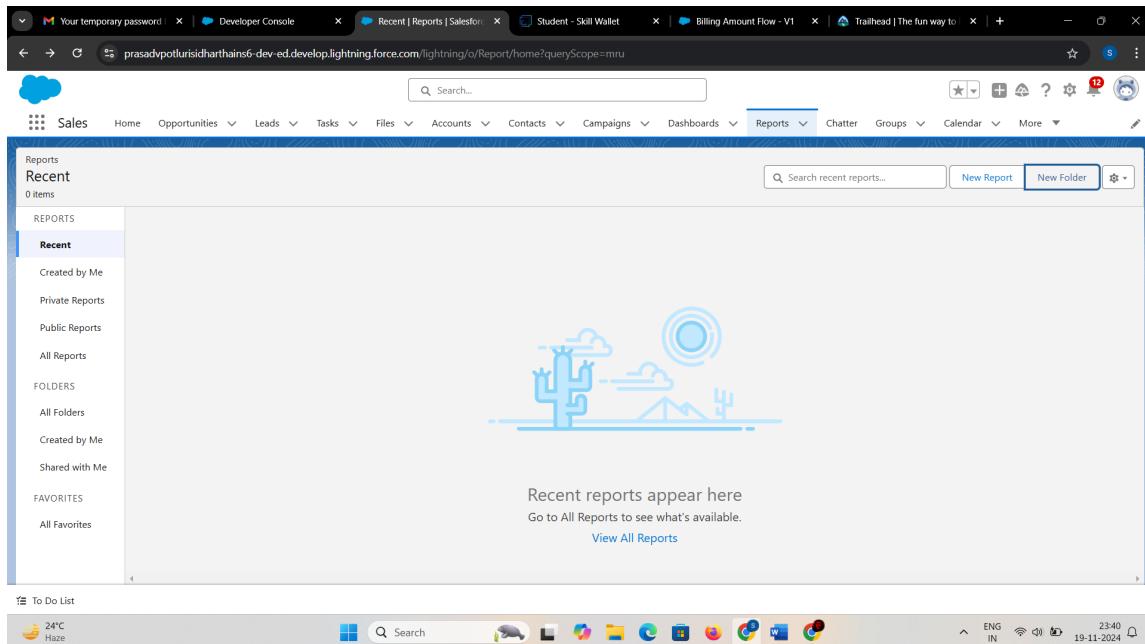
create a report folder

1. Click on the app launcher and search for reports.

2. Click on the report tab, click on new folder.

3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.

4. Click save.



Sharing a report folder

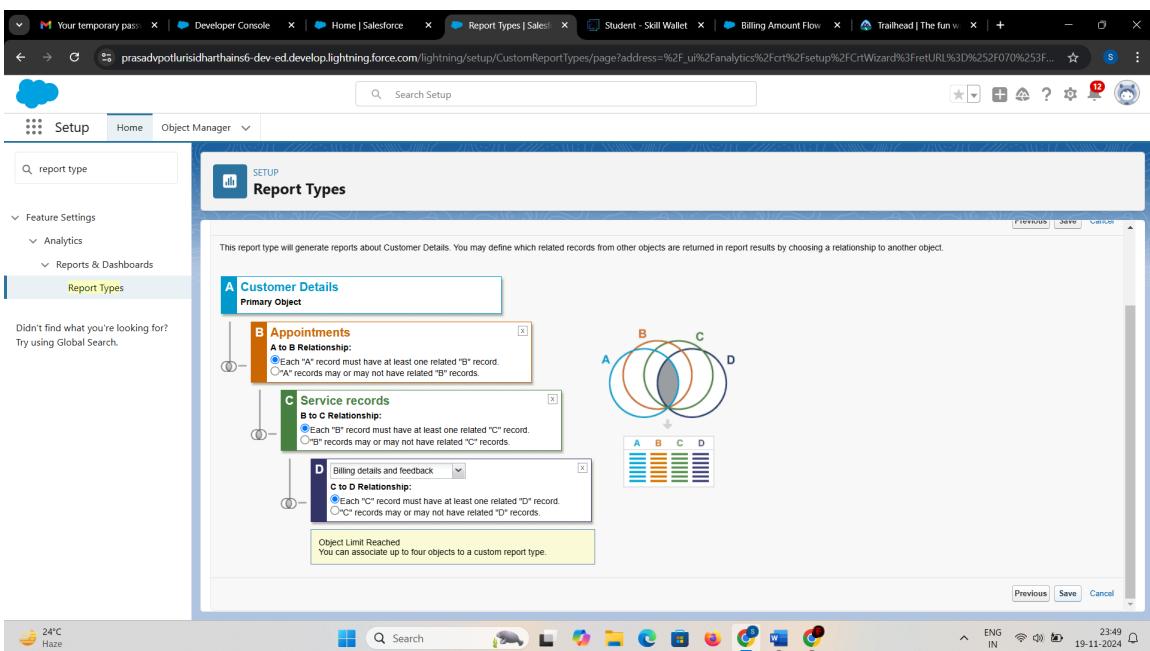
1. Go to the app >> click on the reports tab.

2. Click on the All folder, click on the Drop down arrow for Garage Management folder, and Click on share.

3. Select the share with as "roles", in name field search for "manager", give "view" as access for that role.
4. Then click share, and click on Done.

Create Report Type

1. Go to setup>> type users in quickfind box >>select Report Type >> click on Continue.
2. Click on new custom report type.
3. Select the Primary object as "Customer details".
4. Give the Report type Label as "Serviceinformation"
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as "other Reports"
8. Select the deployment status as "Deployed", click on Next.
9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object as shown in fig
11. Again Click to relate another object.
12. And select the related object as "service records".
13. Repeat the process and select the related object as "Billing details and feedback".
14. And click on save.

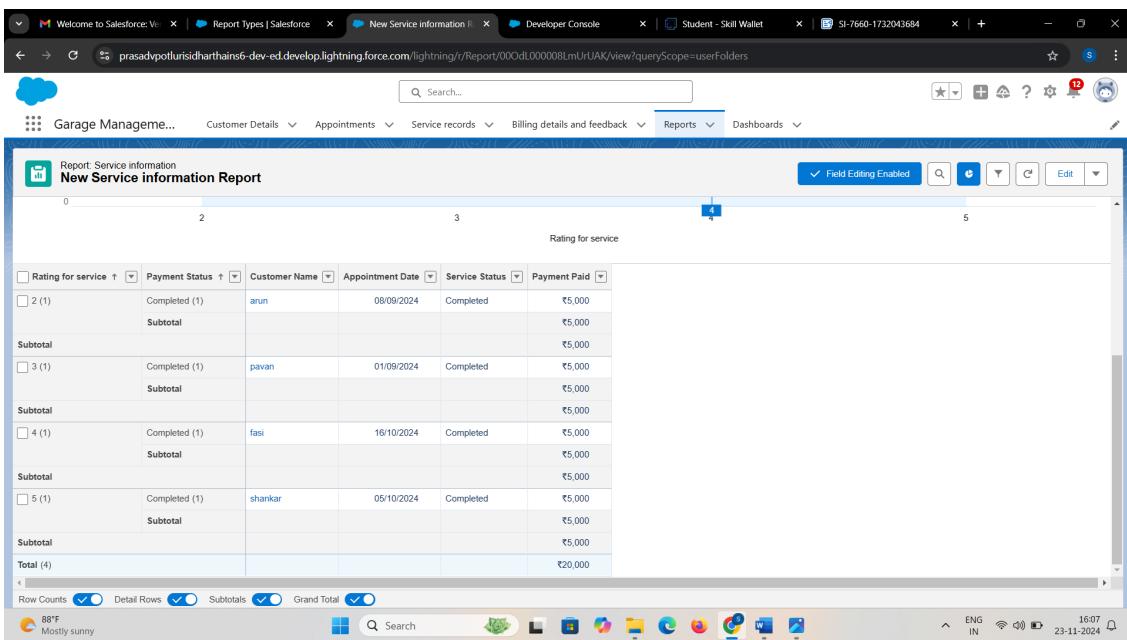
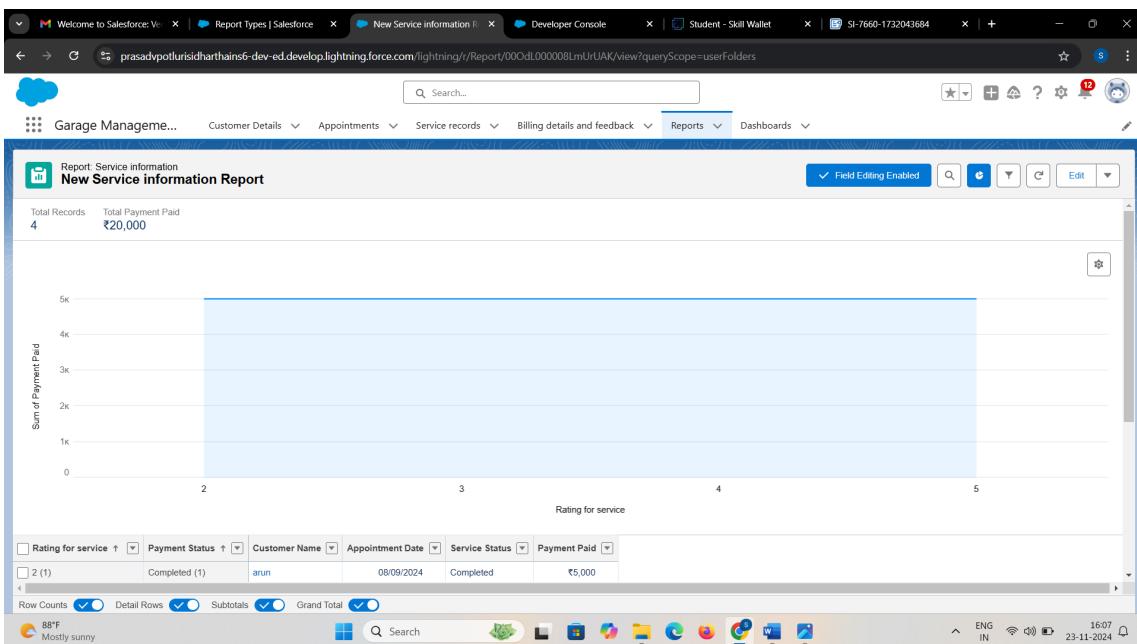


Create Report

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

1. Go to the app >> click on the reports tab
2. Click New Report.
3. Select the Category as other reports, search for ServiceInformation, select that report, click on it. And click on start report.
4. Their outline pane is opened already, select the fields that mentioned below in column section.
 - a. Customer name
 - b. Appointment Date
 - c. Service Status
 - d. Payment paid
 - e. Remove the unnecessary fields.
 - f. Select the fields that mentioned below in GROUP ROWS section.
 - i. Rating for Service
 - g. Select the fields that mentioned below in GROUP ROWS section.

- i. Payment Status
- h. Click on Add Chart , Select the Line Chart.
- i. Click on save, Give the reportName : New Service information Report
- j. Report unique Name is auto populated.
- k. Select the folder the created and Click on save.

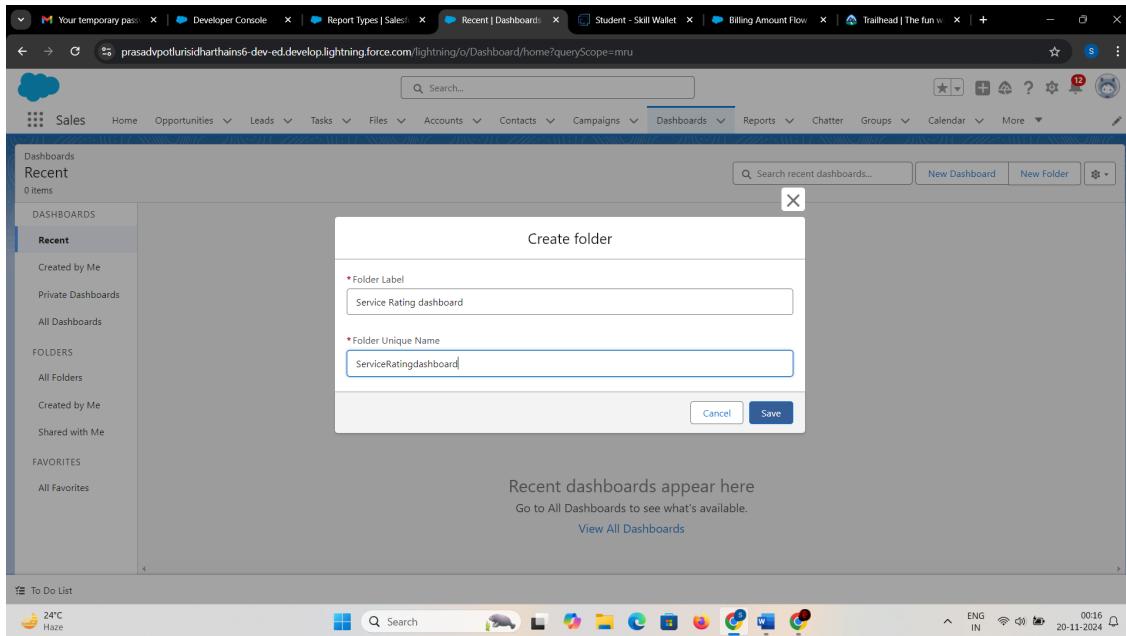


Dashboards

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 identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

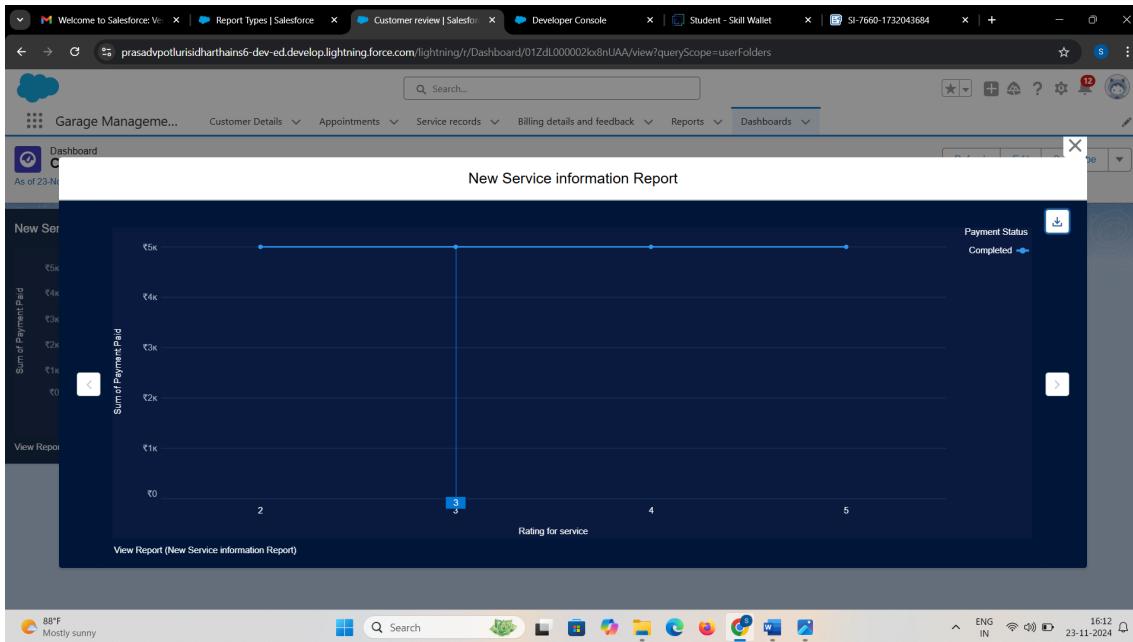
Create Dashboard Folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as "Service Rating dashboard".
4. Folder unique name will be auto populated.
5. Click save.
6. Follow the same steps, from milestone 15, and activity2, and provide the sharing settings for the folder that just created.



Create Dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.
3. Select add component.
4. Select a Report and click on select.
5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.



Subscription:

1. After that Click on Subscribe on top right.
2. Set the Frequency as " weekly".
3. Set a day as monday.
4. And Click on save.

