

Project Name: Sales Automobile Using Salesforce CRM

Team ID : LTVIP2024TMID11576

Team Size : 5

Team Leader : V SHILPA

Team member : V TEJASREE

Team member : R LIKHITHA

Team member : RAMKUPPAM SURENDRA

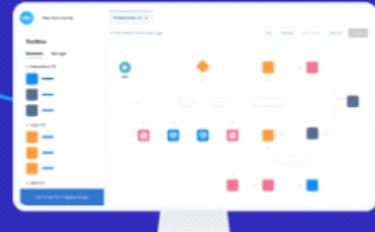
Team member : THAMBA NANDINI

Salesforce

Creating Developer Account

Creating a developer org in salesforce.

- Go to <https://developer.salesforce.com/signup>
- On the sign up form, enter the following details :



Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework

Sign up for your Salesforce Developer Edition

A full-featured copy of the Platform Builder.

Complete the form to start your free trial. Our team will help you make the most of your trial.

First Name*

Last Name

Email*

Role*

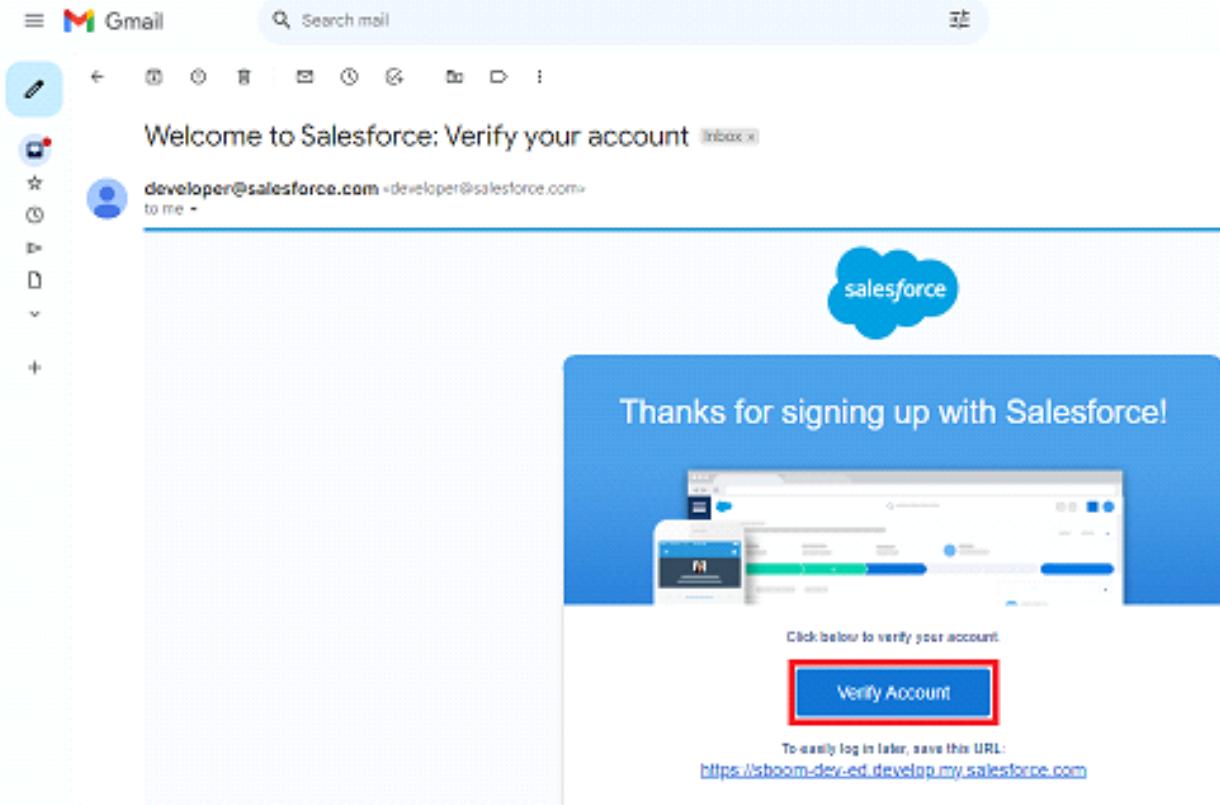
Company*

- 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.

Account Activation

- Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



1. Click on Verify Account
2. Give a password and answer a security question and click on change password.

Change Your Password

Enter a new password for lead@sb.oom.

Make sure to include at least:

- 8 characters
- 1 letter
- 1 number

* New Password

 Good

* Confirm New Password

 Match

Security Question

In what city were you born?

* Answer

 asdfghjkl

Change Password

1. Then you will redirect to your salesforce setup page.



Search Setup



Setup

Home

Object Manager

Quick Find

Setup Home

Service Setup Assistant

Multi-Factor Authentication
Assistant

Release Updates

Lightning Experience
Transition Assistant

Salesforce Mobile App

Lightning Usage

Optimizer

ADMINISTRATION

> Users



SETUP
Home



Get Started with Einstein Bots

Launch an AI-powered bot to automate
your digital connections.

[Get Started](#)



Mobile Publisher

Use the Mobile Publisher to create your
own branded mobile app.

[Learn More](#)



Real-time

Transform pro
docs, spread



Object

Create Automobile Information Object

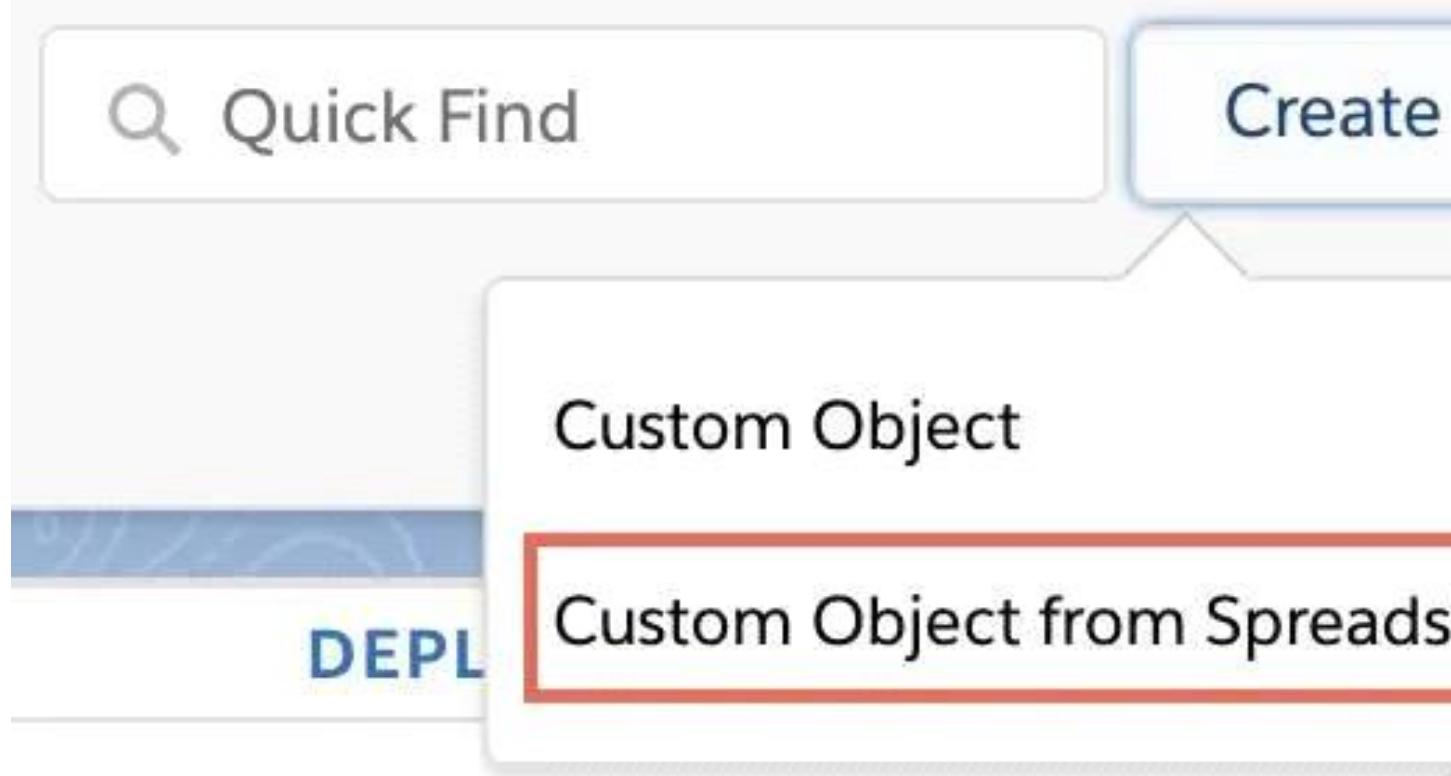
- Download and open [this spreadsheet](#), save it as AutomobileInformation.csv.
- Make sure to download the File into CSV format.
Note : Make sure to have the name of the file as “Automobile Information”.

- Log into your salesforce account, click , then select Setup.
1. Click the Object Manager tab.

2. Click Create.



3. Select Custom Object from Spreadsheet.



4. Click Login With Salesforce.
5. Enter your Salesforce account username and password. (which you have created in the Milestone 1, Activity 1)
6. Click Log In.
7. Click Allow.
8. Click Upload.
9. Navigate to the Automobile Information.csv file you downloaded and upload it. Salesforce automatically detects the fields and populates all its record data. Choose the Record Name field and make sure all fields are with the proper datatypes as below as they are.

Define object and fields
Choose the data source, map fields and their types, and import field data.

| CSV File Details | | Field Label Source | * Field Labels Row | Import 9 rows of Data? |
|---|---------------------------|--|-------------------------------------|--|
| Encoding Format <small>i</small> | Values Separated By | <input type="radio"/> Enter manually <input checked="" type="radio"/> Detect from row | 1 | No, skip import <input checked="" type="radio"/> Yes, Import data |
| Fields 9 of 9 to import <input type="checkbox"/> Hide mapped fields | | | | |
| IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS <small>i</small> | |
| Name Of Manufacturer | Name Of Manufacturer | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Model | Model | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Engine Number | Engine Number | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| VIN | VIN | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Total Number of Cylinders | Total Number of Cylinders | Picklist | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Colour | Colour | Picklist | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Built date | Built date | Phone | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Price | Price | Currency | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Quantity | Quantity | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

[Back](#) Next

1. Click Next and enter the following settings.
2. Click Finish. The Automobile Information object is successfully created and data imported, all within minutes.

Create Invoice Object

Create Invoice object, just as we have created an Automobile Information Object using [this sheet](#)
Make sure to Download the File into CSV Format.

Note: Make sure you do field mapping with proper field type as shown below.

Define object and fields

Choose the data source, map fields and their types, and import field data.

| CSV File Details | | Fields 5 of 5 to import | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-----------------------|--|-------------------------------------|--|------------------------|-----------------------|-----------------------|----------------|--|--------------|------------|------|-------------------------------------|-------------------------------------|---------------|-------------|---------|-------------------------------------|-------------------------------------|------------|----------|---------|-------------------------------------|-------------------------------------|--------------|------------|---------|-------------------------------------|-------------------------------------|-----------------|---------------|------|-------------------------------------|-------------------------------------|
| Encoding Format | Values Separated By | Field Label Source | * Field Labels Row | Import 0 rows of Data? | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Unicode (UTF8) | Comma | <input type="radio"/> Enter manually <input checked="" type="radio"/> Detect from row | 1 | <input checked="" type="radio"/> No, skip import <input type="radio"/> Yes, import data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>IMPORT FILE FIELD NAME</th> <th>SALESFORCE FIELD NAME</th> <th>SALESFORCE FIELD TYPE</th> <th colspan="2">ADD TO LAYOUTS</th> </tr> </thead> <tbody> <tr> <td>✓ Invoice ID</td> <td>Invoice ID</td> <td>Text</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>✓ Total Price</td> <td>Total Price</td> <td>Integer</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>✓ Quantity</td> <td>Quantity</td> <td>Integer</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>✓ Unit Price</td> <td>Unit Price</td> <td>Integer</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>✓ Purchase Date</td> <td>Purchase Date</td> <td>Date</td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table> | | | | | IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS | | ✓ Invoice ID | Invoice ID | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ✓ Total Price | Total Price | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ✓ Quantity | Quantity | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ✓ Unit Price | Unit Price | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | ✓ Purchase Date | Purchase Date | Date | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ✓ Invoice ID | Invoice ID | Text | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ✓ Total Price | Total Price | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ✓ Quantity | Quantity | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ✓ Unit Price | Unit Price | Integer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ✓ Purchase Date | Purchase Date | Date | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Create Automobile Object

The purpose of creating an Automobile custom object is to store and manage information about Invoice.
To create an object:

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

The screenshot shows the Salesforce Setup page. At the top, there is a blue header bar with a cloud icon, the word "SETUP" in white, and a search bar labeled "Search Setup". Below the header, there are navigation tabs: "Setup", "Home", and "Object Manager", with "Object Manager" highlighted by a red box. The main content area has a title "Object Manager" with a subtitle "51+ Items, Sorted by Label". On the right, there is a "Quick Find" search bar. The main table lists objects with columns: "LABEL", "API NAME", "TYPE", "DESCRIPTION", and "LAST ACTIVITY". The first item listed is "Account" with "Account" in all columns.

| LABEL | API NAME | TYPE | DESCRIPTION | LAST ACTIVITY |
|---------|----------|-----------------|-------------|---------------|
| Account | Account | Standard Object | | |

- Enter the label name? Opportunity Automobile
- Plural label name?Opportunity Automobiles

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. The page title is 'New Custom Object'. The 'Custom Object Information' section contains fields for 'Label' (Example: Account) and 'Plural Label' (Example: Accounts), both of which are highlighted with red boxes and arrows pointing to them. Below these are fields for 'Object Name' (Example: Account), 'Description', and 'Content Name'. The 'Enter Record Name Label and Format' section includes a 'Record Name' field (Example: Account Name) and a 'Data Type' dropdown set to 'Text', also highlighted with red boxes and arrows.

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label Example: Account

Plural Label Example: Accounts

Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Content Name -None-

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always case-sensitive.

Record Name Example: Account Name

Data Type Text

1. Enter Record Name Label and Format
 - Record Name >> Opportunity Automobile Id
 - Data Type >> Auto Number
 - Display Format >> OA-{0000}
 - Starting Number >> 1

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Be careful when changing the name or label as it may affect existing integrations and merge templates.

Label

Opportunity Automobile

Example: Account

Plural Label

Opportunity Automobiles

Example: Accounts

Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name

Opportunity_Automobile

Example: Account

Description

Context-Sensitive Help Setting

Open the standard Salesforce.com Help & Training window

Open a window using a Visualforce page

Content Name

—None— ▾

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Acco

Record Name

Opportunity Automobile Id

Example: Account Name

Data Type

Auto Number ▾

Display Format

OA-{0000}

Example: A-{0000} What Is This?

Starting Number

1

1. Click on Allow reports.
2. Allow search
3. >> Save.

Tabs

Creating A Custom Tab

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

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functions.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to your page. You can also use the Lightning Page tab to allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

[New](#)

[What Is This?](#)

No Custom Object Tabs have been defined

Web Tabs

[New](#)

[What Is This?](#)

No Web Tabs have been defined

- Select Object(Opportunity Automobile) >> Select any tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) keep it as default >> Save.

Custom Object Tab Opportunity Automobiles

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail

[Edit](#) [Delete](#)

Tab Style



Splash Page Custom Link

Modified By

Mohammad Sa

Tab Label Opportunity Automobiles

Object [Opportunity Automobile](#)

Description

Created By [Mohammad Sameer](#), 27/11/2023, 2:48 pm

Note: Tabs for Automobile Information & Invoice objects do get created automatically. We do not need to create tabs for those objects.

The Lightning App

Create A Lightning App

- Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

The screenshot shows the Salesforce Lightning Experience App Manager interface. At the top, there is a search bar with the text "app manager". Below the search bar, there is a section titled "Lightning Experience App Manager" with a sub-section "Clone (App,Beta)". There is also a note: "Quickly create new Lightning apps by cloning existing apps. To use the beta feature, indicate that you've read all legal requirements and agree to participate by toggling Enable Release notes." A toggle switch labeled "Enable App Cloning" is set to "Disabled". The main area displays a table of 35 items, sorted by App Name, filtered by All appmenuitems - TabSetType. The table has columns for App Name, Developer Name, and Description. The first few rows of the table are:

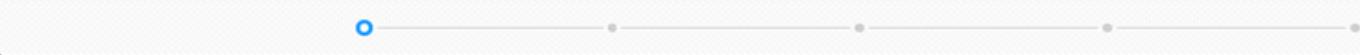
| App Name | Developer Name | Description |
|-------------------------------|------------------------|---|
| All Tabs | AllTabsSet | |
| Analytics Studio | Insights | Build CRM Analytics dashboards and apps |
| App Launcher | AppLauncher | App Launcher tabs |
| Bolt Solutions | LightningBolt | Discover and manage business solutions designed for your industry |
| Chatter Desktop | Chatter/Desktop | Chatter Desktop is an Adobe AIR-based desktop application that lets Chatter users stay connected... |
| Chatter Mobile for BlackBerry | Chatter_for BlackBerry | The Salesforce.com Chatter Mobile app lets you access Chatter data on the go. Use it to view fe... |
| College Management Systems | Nadeem | demo app |
| Community | Community | Salesforce CRM Communities |
| Content | Content | Salesforce CRM Content |
| Data Manager | DataManager | Use Data Manager to view limits, monitor usage, and manage recipes |

- Fill the app name in app details and branding as follow
 - a. App Name :Sales Automobile Using Salesforce CRM
 - b. Developer Name : this will auto populated
 - c. Description : Give a meaningful description
 - d. Image : optional (if you want to give any image you can otherwise not mandatory)
 - e. Primary color hex value : keep this default
- Then click Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

| | |
|---|--|
| App Details | App Branding |
| * App Name <small>i</small> <input type="text" value="Name your app..."/> | Image <small>i</small>  <input type="button" value="Upload"/> |
| * Developer Name <small>i</small> <input type="text" value="Enter a developer name..."/> | Primary Color Hex Value <input type="color" value="#0070D2"/> #0070D2 |
| Description <small>i</small> <input type="text" value="Enter a description..."/> | Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme |
| App Launcher Preview | |
|  | |

- Add Navigation Items:

New Lightning App

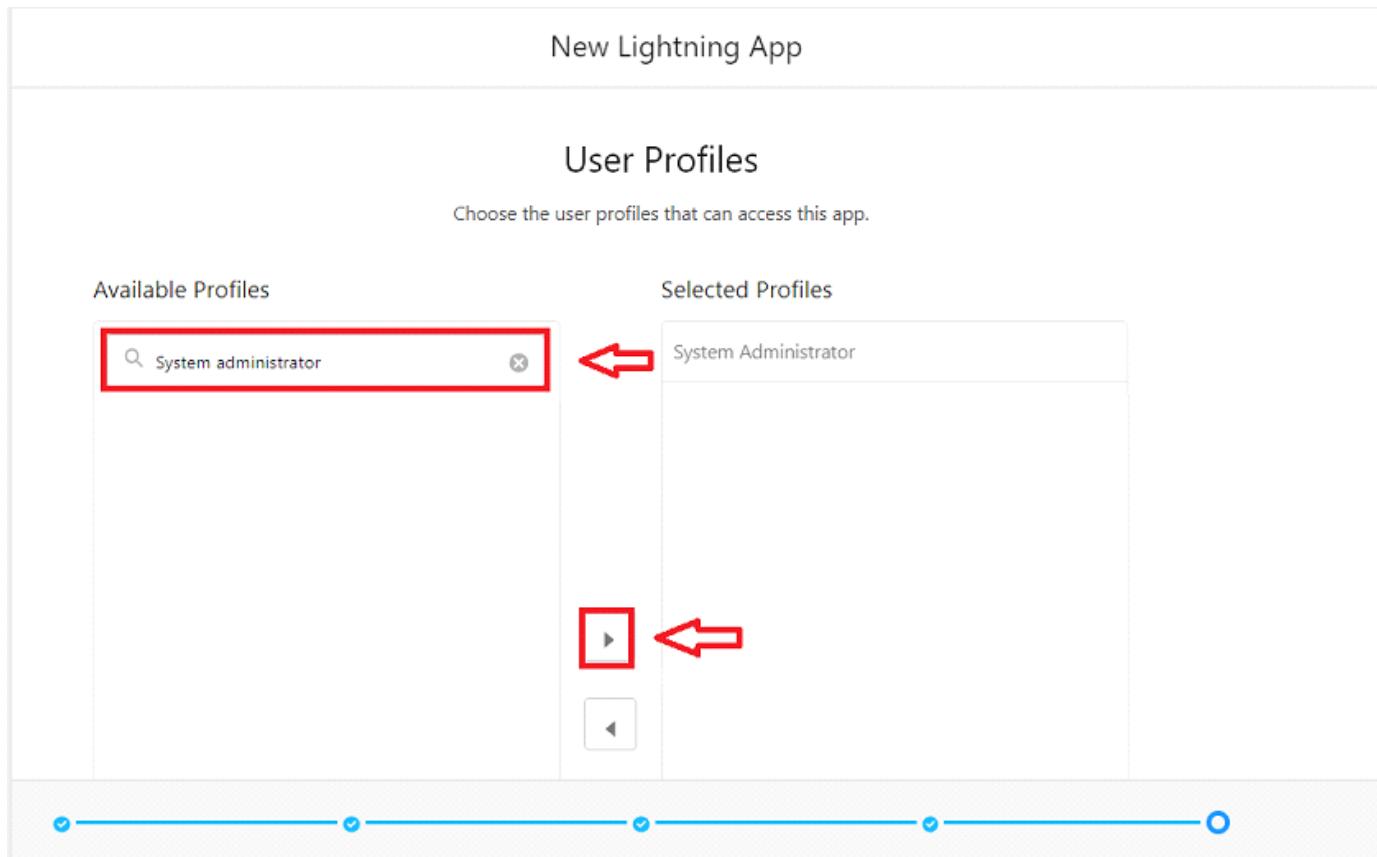
Navigation Items

They appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items. Items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

The screenshot shows the 'Navigation Items' configuration screen. On the left, there's a search bar with 'Dash' typed in, a 'Create' button, and two arrow buttons for moving items between lists. A red box highlights the 'Selected Items' list on the right, which contains the following items:

| Icon | Item Name |
|-----------------------------|------------------------|
| Accounts icon | Accounts |
| Contacts icon | Contacts |
| Opportunities icon | Opportunities |
| Automobile Information icon | Automobile Information |
| Automobiles icon | Automobiles |
| Invoice icon | Invoice |
| Reports icon | Reports |
| Dashboard icon | Dashboards |

- Search the items in the search bar(Account,Contact ,Opportunities,Automobile Information,Opportunity Automobile,Invoice, Reports, Dashboard) from the search bar and move it using the arrow button ? Next. Note: select asset the custom object which we have created in the previous activity.



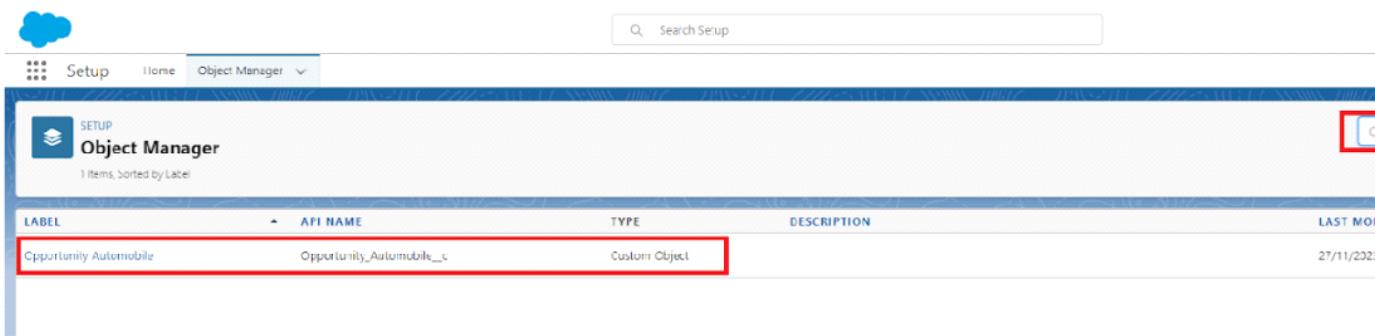
1. Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

Fields & Relationships

Creating Opportunity Master Detail Relationship Field In Opportunity AutoMobile Object

To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar>> click on the object.



The screenshot shows the Salesforce Object Manager interface. At the top, there is a navigation bar with icons for Home, Object Manager, and a search bar labeled "Search Setup". Below the navigation bar, the title "Object Manager" is displayed, followed by a subtitle "1 Items, Sorted by Label". A table lists one item: "Opportunity_Automobile" with the API name "Opportunity_Automobile__c" and the type "Custom Object". The entire row for "Opportunity_Automobile" is highlighted with a red border.

| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED |
|------------------------|---------------------------|---------------|-------------|---------------|
| Opportunity_Automobile | Opportunity_Automobile__c | Custom Object | | 27/11/2022 |

- Now click on “Fields & Relationships” >> New

SETUP > OBJECT MANAGER
Patient

Details

Fields & Relationships **New** Deleted Fields Field D

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Fields & Relationships

12 Items, Sorted by Field Label

| Blood Group | Blood_Group__c | Picklist | |
|-------------|----------------|--------------|--|
| Created By | CreatedById | Lookup(User) | |

1. Select Data type as “Master Details Relationship”.

SETUP > OBJECT MANAGER
Opportunity Automobile

Details

Fields & Relationships **New Custom Field**

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Opportunity Automobile
New Custom Field

Step 1. Choose the field type

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

Data Type

None Selected Select one of the data types below.

Auto Number A system-generated sequence number that uses a display format you define. The number is unique for each record.

Formula A read-only field that derives its value from a formula expression you define. The formula can include calculations, functions, and references to other fields.

Roll-Up Summary A read-only field that displays the sum, minimum, or maximum value of a field in a related master record.

Master-Detail Relationship **Master-Detail Relationship** 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 of related records.

External Lookup Relationship Creates a relationship that links this object to an external object whose data is stored outside of Salesforce.

1. Click on Next

Opportunity Automobile
New Relationship

Step 2. Choose the related object

Select the other object to which this object is related.

Related To

1. Fill the above as following:

- Field Label: gets auto Generated(Opportunity)
- Field Name : gets auto generated(Opportunity)
- Click on Next >> Next >> Save and new.

Opportunity Automobile
New Relationship

Step 3. Enter the label and name for the lookup field

| | |
|--------------------------------|--|
| Field Label | <input type="text" value="Opportunity"/> i |
| Field Name | <input type="text" value="Opportunity"/> i |
| Description | <input type="text"/> |
| Help Text | <input type="text"/> |
| Child Relationship Name | <input type="text" value="Opportunity_Automobiles"/> i |
| Sharing Setting | Select the minimum access level required on the Master record to create, edit, or delete related Detail records: <input type="radio"/> Read Only: Allows users with at least Read access to the Master record to create, edit, or delete related Detail records. <input checked="" type="radio"/> Read/Write: Allows users with at least Read/Write access to the Master record to create, edit, or delete related Detail records. |
| Allow reparenting | <input type="checkbox"/> Child records can be reparented to other parent records after they are created |
| Auto add to custom report type | <input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity i |

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. [Tell me more!](#)

▶ [Show Filter Settings](#)

Creating The AutoMobile Information Lookup Field In Opportunity Automobile Object

To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar>> click on the

object.

The screenshot shows the Salesforce Object Manager interface. At the top, there are navigation links: Setup, Home, and Object Manager. A search bar is at the top right. Below the header, the title 'Object Manager' is displayed above a table. The table has columns: LABEL, API NAME, TYPE, DESCRIPTION, and LAST MODIFIED. A single row is visible, representing a custom object named 'Opportunity Automobile' with the API name 'Opportunity_Automobile__c' and a type of 'Custom Object'. The 'LAST MODIFIED' column shows the date '27/11/2023'. A red box highlights the 'Opportunity Automobile' label in the first column.

- Now click on “Fields & Relationships” >> New

The screenshot shows the 'Fields & Relationships' page for the 'Patient' object. On the left, there's a sidebar with options: Details, Fields & Relationships (which is selected and highlighted with a red box), Page Layouts, Lightning Record Pages, and Buttons, Links, and Actions. The main area is titled 'Fields & Relationships' with a sub-instruction '12 Items, Sorted by Field Label'. It lists two fields: 'Blood Group' (Type: Picklist) and 'Created By' (Type: Lookup(User)). At the top right, there are 'Quick Find' and 'New' buttons, with the 'New' button highlighted by a red box. A red arrow points to the 'New' button. The bottom right corner of the page has a 'Deleted Fields' link.

- Select Data type as “Lookup RelationShip”.

The screenshot shows the 'Fields & Relationships' section of the 'Opportunity Automobile' object setup. On the left, the sidebar includes: Details, Fields & Relationships (selected and highlighted with a red box), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main panel is titled 'Data Type' and asks 'Specify the type of information that the custom field will contain.' It lists several options: 'None Selected' (selected), 'Auto Number', 'Formula', 'Roll-Up Summary', 'Lookup Relationship' (highlighted with a red box), and 'Master-Detail Relationship'. To the right of each option is a detailed description. For 'Lookup Relationship': 'Creates a relationship that links this object to another object. The relationship field allows'. For 'Master-Detail Relationship': 'Creates a special type of parent-child relationship between this object (the child) or "detail record" and another object (the master).'. A red box also highlights the 'Lookup Relationship' description.

- Click on Next

Opportunity Automobile
New Relationship

Step 2. Choose the related object

Select the other object to which this object is related

Related To

- Fill the above as following:
 - Field Label: Automobile
 - Field Name : Automobile
- Click on Next >> Next>> Save and new.

Opportunity Automobile
New Relationship

Step 3. Enter the label and name for the lookup field

Field Label

Field Name

Description

Help Text

Child Relationship Name

Required

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

What to do if the lookup record is deleted?

Clear the value of this field. You can't choose this option if you make this field required.

Don't allow deletion of the lookup record that's part of a lookup relationship.

Auto add to custom report type Add this field to existing custom report types that contain this entity

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. [Tell me more](#)

[Show Filter Settings](#)

Creating The AutoMobile Information Lookup Field In Opportunity Automobile Object

To create fields in an object:

- Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar>> click on the object.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. A single item, 'Opportunity Automobile', is listed in the table. The 'Label' column shows 'Opportunity Automobile', the 'API Name' column shows 'Opportunity_Automobile__c', and the 'Type' column shows 'Custom Object'. The 'Last Modified' column shows '27/11/2023'. A red box highlights the row for 'Opportunity Automobile'.

- Now click on “Fields & Relationships” >> New

The screenshot shows the 'Fields & Relationships' section for the 'Patient' object. It lists fields like 'Blood Group' and 'Created By'. On the right, there are 'Quick Find' and 'New' buttons, with 'New' highlighted by a red box and an arrow pointing to it.

- Select Data type as “Lookup RelationShip”.

The screenshot shows the 'Fields & Relationships' section for the 'Opportunity Automobile' object. Under 'Data Type', the 'Lookup Relationship' option is selected, highlighted by a red box. The description for 'Lookup Relationship' is visible on the right, explaining its function and relationship requirements.

- Click on Next

Opportunity Automobile
New Relationship

Step 2. Choose the related object

Select the other object to which this object is related

Related To

- Fill the above as following:
 - Field Label: Automobile
 - Field Name : Automobile
- Click on Next >> Next>> Save and new.

Opportunity Automobile
New Relationship

Step 3. Enter the label and name for the lookup field

| | |
|---|--|
| Field Label | <input type="text" value="Automobile"/> |
| Field Name | <input type="text" value="Automobile"/> |
| Description | <input type="text"/> |
| Help Text | <input type="text"/> |
| Child Relationship Name | <input type="text" value="Opportunity Automobiles"/> |
| Required | <input type="checkbox"/> Always require a value in this field in order to save a record <input checked="" type="radio"/> Clear the value of this field. You can't choose this option if you make this field required. <input type="radio"/> Don't allow deletion of the lookup record that's part of a lookup relationship. <input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity |
| What to do if the lookup record is deleted? | |
| Auto add to custom report type | |

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. [Tell me more](#)

[Show Filter Settings](#)

Creating Formula Field In Opportunity Automobile Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar >> click on the object.

- Now click on “Fields & Relationships” >> New.
- Select Data type as “Formula” and click Next.

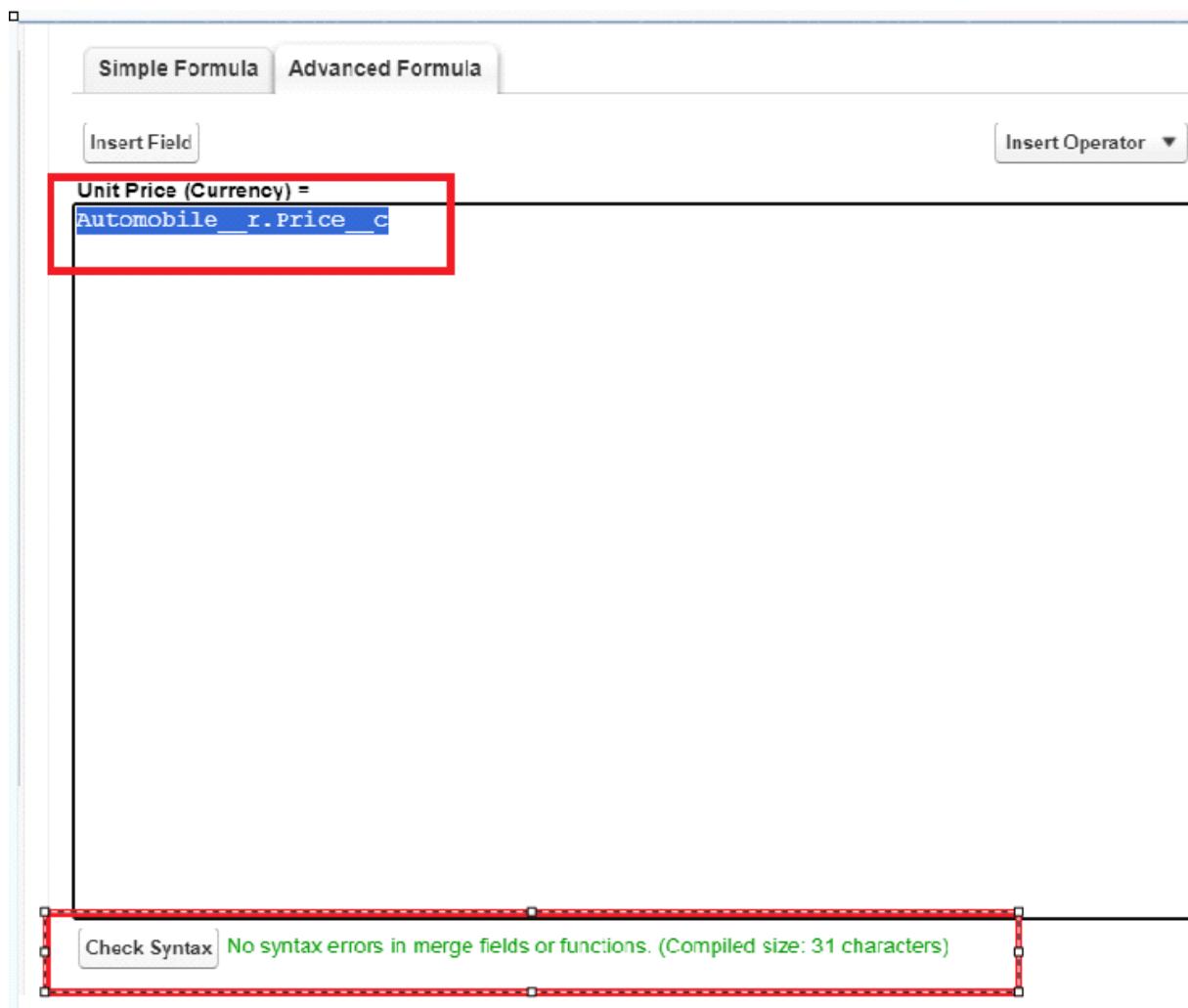
- Give Field Label and Field Name as “Unit Price” and select formula return type as “Currency” and change the decimal values to two and click next.

Opportunity Automobile
New Custom Field

Step 2. Choose output type

| | | | |
|--|---|---|---|
| Field Label | <input type="text" value="Unit Price"/> | Field Name | <input type="text" value="Unit_Price"/> |
| Auto add to custom report type | | <input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity [i] | |
| Formula Return Type <p><input type="radio"/> None Selected Select one of the data types below.</p> <p><input type="radio"/> Checkbox</p> <p><input checked="" type="radio"/> Currency</p> <p><input type="radio"/> Date</p> <p><input type="radio"/> Date/Time</p> <p><input type="radio"/> Number</p> <p><input type="radio"/> Percent</p> <p><input type="radio"/> Text</p> <p><input type="radio"/> Time</p> <p>Options</p> <p>Decimal Places <input type="text" value="2"/> Example: 999.00</p> | | | |

- Under Advanced Formula write down the formula : Automobile__r.Price__c



1. click “Check Syntax” and Next >> Next >> Save & New.

Creating The Formula Field In Opportunity Automobile Object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar >> click on the object.
 - Now click on “Fields & Relationships” >> New.
 - Select Data type as “Formula” and click Next.
 - Give Field Label and Field Name as “Total Price” and select formula return type as “Currency” and change the decimal values to two and click next.

Opportunity Automobile
New Custom Field

Step 2. Choose output type

| | | | |
|-------------|--|------------|--|
| Field Label | <input type="text" value="Total Price"/> | Field Name | <input type="text" value="Total_Price"/> |
|-------------|--|------------|--|

Auto add to custom report type Add this field to existing custom report types that contain this entity [i](#)

Formula Return Type

None Selected Select one of the data types below.

Currency Calculate a boolean value.
Example: `TODAY() > CloseDate`

Date Calculate a Julian or Julian currency amount and automatically format the field as a currency amount.
Example: `Gross Margin = Amount * Cust__c`

Date/Time Calculate a date, for example, by adding or subtracting days to other dates.
Example: `Reminder Date = CloseDate - 7`

Number Calculate a float/time, for example, by adding a number of hours or days to another date/time.
Example: `Next = NOW() + 1`

Percent Calculate a numeric value.
Example: `Fahrenheit = 1.8 * Celsius__c + 32`

Text Calculate a percent and automatically add the percent sign to the number.
Example: `Discount = (Amount - Discounted_Amount__c) / Amount`

Time Create a text string, for example, by concatenating other text fields.
Example: `Full Name = LastName & ", " & FirstName`

Options

Decimal Places Example: 999.00

- Under Advanced Formula write down the formula : Unit_Price__c * Quantity__c

Example: Gross Margin - Amount - Cost_c | [More Examples...](#)

[Simple Formula](#) [Advanced Formula](#)

[Insert Field](#) [Insert Operator ▾](#)

Total Price (Currency) =

`Unit_Price_c * Quantity_c`

[Check Syntax](#) No syntax errors in merge fields or functions. (Compiled size: 75 characters)

The screenshot shows a formula editor interface. At the top, there's a status bar with the text 'Example: Gross Margin - Amount - Cost_c | More Examples...'. Below it are two tabs: 'Simple Formula' and 'Advanced Formula', with 'Simple Formula' being active. There are also 'Insert Field' and 'Insert Operator' buttons. The main area contains a formula field with the placeholder 'Total Price (Currency) =' followed by a red-bordered input box containing the formula 'Unit_Price_c * Quantity_c'. At the bottom, there's a 'Check Syntax' button with a dashed red border and a green message indicating no syntax errors ('No syntax errors in merge fields or functions. (Compiled size: 75 characters)').

1. click "Check Syntax" and Next >> Next >> Save.

Updating Field In Invoice Object

To Update fields in an object:

- Go to setup ? click on Object Manager ? type object name(Invoice) in quick find bar? click on the object.

- Now click on “Fields & Relationships”, Click on the edit of Invoice Id field.

SETUP > OBJECT MANAGER
Invoice

Fields & Relationships
8 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD |
|-------------------|------------------|--------------------|-------------------|
| Created By | CreatedById | Lookup(User) | |
| Last Modified By | LastModifiedById | Lookup(User) | |
| Owner | OwnerId | Lookup(User,Group) | |
| Purchase Date | Purchase_Date__c | Date | |
| Quantity | Quantity__c | Number(18, 0) | |
| Total Price | Total_Price__c | Number(18, 0) | |
| Unit Price | Unit_Price__c | Number(18, 0) | |
| Invoice ID | Name | Text(80) | |

SETUP > OBJECT MANAGER
Invoice

Fields & Relationships

Record Name: **Invoice ID**

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For more information, see [Field Names](#).

| | |
|-----------------|--|
| Record Name | <input type="text" value="Invoice ID"/> Example: Account Name |
| Data Type | Auto Number <input type="button" value="▼"/> |
| Display Format | I-{0000} Example: A-{0000} What Is This? |
| Starting Number | <input type="text" value="1"/> |

Save **Cancel**

- Select Data type as “Auto Number ” and click Next.

- a. Display Format :- I-{0000}
- b. StartingNumber:-

- Click Save.

• Creating Remaining Fields In Objects

- Now create the remaining fields using the data types mentioned.
-

| s.no | Object name | Fields | |
|------|-------------|------------|--|
| 1 | Invoice | Field Name | Opportunity |
| | | Data type | Master Detail relationship Object : Opportunity |

Page Layouts

Edit The Page Layout For Opportunity Object

Step 1: Go to Setup >> Click on Object Manager >> On the search bar, select Opportunity Layout. You can notice Page Layouts on the left panel

Step 2: Click on Page Layouts, Click on ‘Opportunity Layouts’.

The screenshot shows the Salesforce Object Manager interface. The left sidebar has a 'Page Layouts' section selected. The main area displays a table titled 'Page Layouts' with four rows. The first row is 'Opportunity (Marketing) Layout', the second is 'Opportunity (Sales) Layout', the third is 'Opportunity (Support) Layout', and the fourth is 'Opportunity Layout'. The 'Opportunity Layout' row is highlighted with a red box. The table has columns for 'PAGE LAYOUT NAME', 'CREATED BY', and 'MODIFIED BY'. The 'CREATED BY' column shows 'Mohammad Sameer, 22/11/2023, 2:19 pm' for all rows. The 'MODIFIED BY' column shows 'Mohammad Sameer, 28/11/2023, 9:45 am' for the first three rows, and 'Mohammad Sameer, 28/11/2023, 9:45 am' for the 'Opportunity Layout' row.

| PAGE LAYOUT NAME | CREATED BY | MODIFIED BY |
|--------------------------------|--------------------------------------|--------------------------------------|
| Opportunity (Marketing) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity (Sales) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity (Support) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |

Step 3: In the Opportunity Detail Section, you can see various fields. Go on Account And Click on that Properties icon of Account name Field.

Setup Home Object Manager

Opportunity

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Quick Find Field Name

Fields

Buttons
Custom Links
Quick Actions
Mobile & Lightning Actions
Expanded Lookups
Related Lists

| Section | Close Date | Delivery/Meta... | Last Modified By | Opportunity Name | Primary Campaign ... | Stage |
|--------------|----------------------|-------------------|--------------------|-------------------|----------------------|-------|
| Blank Space | Contract | Description | Lead Source | Opportunity Owner | Private | Type |
| Account Name | Created By | Expected Revenue | Main Competitor(e) | Order Number | Probability (%) | |
| Amount | Current Generator(s) | Forecast Category | Next Step | Price Book | Quantity | |

Post File New Task Log a Call New Case New Note New Event Link Poll Question Email

Delete Edit Change Owner Sharing

Opportunity Detail

Standard Buttons

Edit Delete Clone Change Owner Change Record Type Sharing Sharing Hierarchy Printable View Standard Buttons

Add to Cadence Add to an Actionable List Create Summary

Custom Buttons

I Opportunity Information (Header visible on edit only)

| | | | |
|-------------------|-------------|------------------|-----------------|
| Opportunity Owner | Sample Text | Amount | ₹123.45 |
| Private | ✓ | Expected Revenue | ₹123.45 |
| Opportunity Name | Sample Text | Close Date | 29/11/2023 |
| Lead Source | Sample Text | Next Step | Sample Text |
| Type | Sample Text | Properties | Probability (%) |
| Lead Source | Sample Text | Stage | Sample Text |

I Other Information (Header visible on edit only)

I Additional Information (Header visible on edit only)

Hand icon pointing to the 'Required' checkbox in the 'Field Properties' dialog.

New Note New Event Link Poll Question

Field Properties

Standard Edit Add to Cadence Custom Buttons

Read-Only Account Name Required

OK Cancel

Amount ₹123.45

Expected Revenue ₹123.45

Close Date 29/11/2023

Next Step Sample Text

Step 4: check the Required box for Account name and click on Ok.

Step 5: Click on Save.

Edit The Page Layout For Automobiles Information

Step 1: Go to Setup >> Click on Object Manager >> On the search bar, select Automobile Information. You can notice Page Layouts on the left panel

Step 2: Click on Page Layouts. Click on ‘Automobile Information Layout’.

The top screenshot shows the 'Page Layouts' list for the 'Automobile Information' object. It displays one item: 'Automobile Information Layout', created by Mohammad Sameer on 27/11/2023 at 12:49 pm, and modified by him on the same date at 10:32 pm. The bottom screenshot shows the 'Automobile Information Layout' configuration page. The 'Fields' section lists fields such as Section, Blank Space, Created By, Name Of Manufact..., Total Number Of C..., Engine Number, Owner, VIN, Last Modified By, Price, Built date, Colour, Model, and Quantity. The 'Automobile Information Sample' panel contains sections for 'Highlights Panel' (Customize the highlights panel for this page layout...), 'Quick Actions in the Salesforce Classic Publisher' (Actions in this section are currently inherited from the global publisher layout. You can override the global publisher layout to set a customized list of actions for the publisher on pages that use this layout.), and 'Salesforce Mobile and Lightning Experience Actions' (Actions in this section are predefined by Salesforce. You can override the predefined actions to set a customized list of actions on Lightning Experience and mobile app pages that use this layout. If you customize the actions section, and have saved the layout, then this section inherits that set of actions by default when you click to override.). The 'Automobile Information Detail' panel includes standard buttons for Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, Sharing Hierarchy, and Submit for Approval.

Step 3: Just Go for each one field of Automobile Information Object, Click on Gear Icon and mark as Required just as Done for Above Account Object. After required is done it will show the red color as given in below image.

SETUP > OBJECT MANAGER

Automobile Information

Page Layouts

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

Fields

| Section | Created By | Name Of Manufact... | Total Number of C... |
|-------------|------------------|---------------------|----------------------|
| Blank Space | Engine Number | Owner | VIN |
| Built date | Last Modified By | Price | |
| Colour | Model | Quantity | |

Information (Header visible on edit only)

| | | | |
|-----------------------------|----------------|-------|-------------|
| ★ ● Name Of Manufacturer | Sample Text | Owner | Sample Text |
| ★ Model | Sample Text | | |
| ★ Engine Number | Sample Text | | |
| ★ VIN | Sample Text | | |
| ★ Total Number of Cylinders | Sample Text | | |
| ★ Colour | Sample Text | | |
| ★ Built date | 1-415-555-1212 | | |
| ★ Price | ₹123.45 | | |
| ★ Quantity | 95,650 | | |

System Information (Header visible on edit only)

| | | | |
|------------|-------------|------------------|-------------|
| Created By | Sample Text | Last Modified By | Sample Text |
|------------|-------------|------------------|-------------|

Custom Links (Header visible on detail only)

Step 4 : Adjust the Fields as given below for A good looking view.

SETUP > OBJECT MANAGER

Automobile Information

Page Layouts

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

Fields

| Section | Created By | Name Of Manufact... | Total Number of C... |
|-------------|------------------|---------------------|----------------------|
| Blank Space | Engine Number | Owner | VIN |
| Built date | Last Modified By | Price | |
| Colour | Model | Quantity | |

Information (Header visible on edit only)

| | | | |
|--------------------------|----------------|-------|-------------|
| ★ ● Name Of Manufacturer | Sample Text | Owner | Sample Text |
| ★ Model | Sample Text | | |
| ★ Engine Number | Sample Text | | |
| ★ Built date | 1-415-555-1212 | | |
| ★ Price | ₹123.45 | | |

System Information (Header visible on edit only)

| | | | |
|------------|-------------|------------------|-------------|
| Created By | Sample Text | Last Modified By | Sample Text |
|------------|-------------|------------------|-------------|

Custom Links (Header visible on detail only)

Mobile Cards (Salesforce mobile only)

Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards.

Step 5 : Click on Save.

Apex Trigger

Opportunity Automobile Quantity

UseCase : Whenever Opportunity Closed won Then Neglect / Minus the Quantity From Automobile Information on the Bases of Opportunity Automobile quantity.

- 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 “OpportunityHandlerClass ”.

```

public class OpportunityHandlerClass {

    public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity, Map<Id,Opportunity> OldMapOpportunity) {
        set<Id> opportunityIds = new set<Id>();
        for(Opportunity opp : LstOpportunity){
            if(opp.StageName =='Closed Won'){
                opportunityIds.add(opp.Id);
            }
        }

        set<Id> opportunityIds = new set<Id>();
        for(Opportunity opp : LstOpportunity){
            if(opp.StageName =='Closed Won'){
                opportunityIds.add(opp.Id);
            }
        }
        Map<Id,Opportunity_Automobile__c> lstOpportunityAutomobile =new Map<Id,Opportunity_Automobile__c>([SELECT Id, Opportunity__c, Automobile__c, Quantity__c FROM Opportunity_Automobile__c WHERE Opportunity__c IN :opportunityIds]);
        Map<Id,AutoInformation__c> MapAutoInformation = new Map<Id,AutoInformation__c>([SELECT Quantity__c, Price__c, Name, Id FROM AutoInformation__c WHERE Id IN: AutoInformationIds]);

        set<Id> AutoInformationIds = new set<Id>();
        for(Opportunity_Automobile__c OppAuto: lstOpportunityAutomobile.values()){
            if(OppAuto.Automobile__c != null){
                AutoInformationIds.add(OppAuto.Automobile__c);
            }
        }
        List<AutoInformation__c> lstAutoInformation = new List<AutoInformation__c>();
        Map<Id,AutoInformation__c> MapAutoInformation = New Map<Id,AutoInformation__c>([SELECT Quantity__c, Price__c, Name, Id FROM AutoInformation__c WHERE Id IN: AutoInformationIds]);
        For(Opportunity_Automobile__c AutoOpp : lstOpportunityAutomobile.Values()){
            decimal num = 0;
            if(AutoOpp.Automobile__c == MapAutoInformation.get(AutoOpp.Automobile__c).Id && OldMapOpportunity.get(AutoOpp.Opportunity__c).stagename == 'Closed Won'){
                num = MapAutoInformation.get(AutoOpp.Automobile__c).Quantity__c - AutoOpp.Quantity__c;
                MapAutoInformation.get(AutoOpp.Automobile__c).quantity__c = num;
                lstAutoInformation.add(MapAutoInformation.get(AutoOpp.Automobile__c));
            }
        }
        If(!lstAutoInformation.IsEmpty()){
            update lstAutoInformation;
        }
    }
}

```

Code:

```

public class OpportunityHandlerClass {

    public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity,
Map<Id,Opportunity> OldMapOpportunity){
        set<Id> opportunityIds = new set<Id>();
        for(Opportunity opp : LstOpportunity){
            if(opp.StageName =='Closed Won'){
                opportunityIds.add(opp.Id);
            }
        }
    }
}

```

```

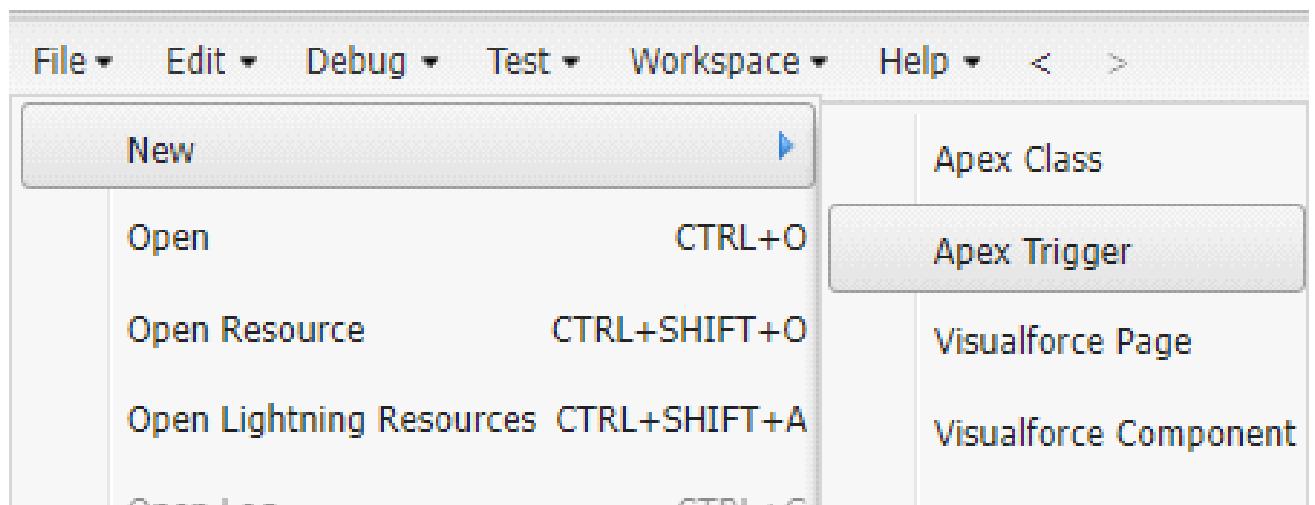
        }
        Map<Id,Opportunity_Automobile__c> lstOpportunityAutomobile =new
        Map<Id,Opportunity_Automobile__c>([SELECT Id, Opportunity__c, Automobile__c, Quantity__c,
        Unit_Price__c, Total_Price__c FROM Opportunity_Automobile__c Where Opportunity__c IN:
        opportunityIds]);
        set<Id> AutoInformationIds = new set<Id>();
        for(Opportunity_Automobile__c OppAuto: lstOpportunityAutomobile.values()){
            if(OppAuto.Automobile__c != null){
                AutoInformationIds.add(OppAuto.Automobile__c);
            }
        }
        List<Automobile_Information__c> lstAutomobileInfomation = new
        List<Automobile_Information__c>();
        Map<Id,Automobile_Information__c> MapAutomobileInformation = New
        Map<Id,Automobile_Information__c>([SELECT Quantity__c, Price__c, Name, Id FROM
        Automobile_Information__c WHERE Id IN: AutoInformationIds]);
        For(Opportunity_Automobile__c AutoOpp : lstOpportunityAutomobile.Values()){
            decimal num = 0;
            if(AutoOpp.Automobile__c == MapAutomobileInformation.get(AutoOpp.Automobile__c).Id &&
            OldMapOpportunity.get(AutoOpp.Opportunity__c).stagename != 'Closed Won'){
                num = MapAutomobileInformation.get(AutoOpp.Automobile__c).Quantity__c-
                AutoOpp.Quantity__c;
                MapAutomobileInformation.get(AutoOpp.Automobile__c).quantity__c = num;
                lstAutomobileInfomation.add(MapAutomobileInformation.get(AutoOpp.Automobile__c));
            }
        }
        If(!lstAutomobileInfomation.IsEmpty()){
            update lstAutomobileInfomation;
        }
    }
}

```

Trigger Handler :

How to create a new trigger :

- While still in the 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 the File menu in the toolbar, and click on new? Trigger.
- Enter the trigger name and the object to be triggered.
- Name : OpportunityTrigger
- sObject : Opportunity



New Apex Trigger

Name:

sObject:

A modal dialog box titled 'New Apex Trigger'. It contains two input fields: 'Name' and 'sObject'. The 'Name' field is empty. The 'sObject' field also contains an empty input field and a dropdown arrow button. At the bottom right of the dialog is a 'Submit' button.

Syntax For creating trigger :

The syntax for creating trigger is :

```
Trigger [trigger name] on [object name]( Before/After event){  
//block of code  
}
```

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.

- Trigger for Opportunity Object.

```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
OpportunityAutomobileHandler.apxc OpportunityHandlerClass.apxc * OpportunityTrigger.apxt *
Code Coverage: None API Version: 59
1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2     if(trigger.isbefore && trigger.isUpdate){
3         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap)
4     }
5 }
6

```

Code:

```

trigger OpportunityTrigger on Opportunity (before update, After Update) {
    if(trigger.isbefore && trigger.isUpdate){
        OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
    }
}

```

Opportunity-Automobile Error

UseCase : If Quantity of Automobile is Zero or Less than The Quantity from The Opportunity-Automobile Than Throw an error .

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 “OpportunityAutomobileHandler”.

```

1 public class OpportunityAutomobileHandler {
2     public static void quantityErrorOnAutomobileInformation(List<Opportunity_Automobile__c> lstOpportunityAutomobile){
3         Set<Id> AutomobileIds = new Set<Id>();
4         For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
5             if(OppAutomobile.Automobile__c != null){
6                 AutomobileIds.add(OppAutomobile.Automobile__c);
7             }
8         }
9         Map<Id, Automobile_Information__c> lstAutomobileInformation = new Map<Id, Automobile_Information__c>([SELECT Id, CreatedById, Quantity__c, Price__c
10                                         FROM Automobile_Information__c WHERE Id IN: AutomobileIds]);
11         For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
12             If(OppAutomobile.Automobile__c == lstAutomobileInformation.get(OppAutomobile.Automobile__c).Id && lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c <= OppAutomobile.Quantity__c)
13                 OppAutomobileaddError('the Number of Automobile u want are not Available !! the Automobile are Available Count is ' + lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c);
14         }
15     }
16 }

```

Code:

```

public class OpportunityAutomobileHandler {
    public static void quantityErrorOnAutomobileInformation(List<Opportunity_Automobile__c>
lstOpportunityAutomobile){

```

```

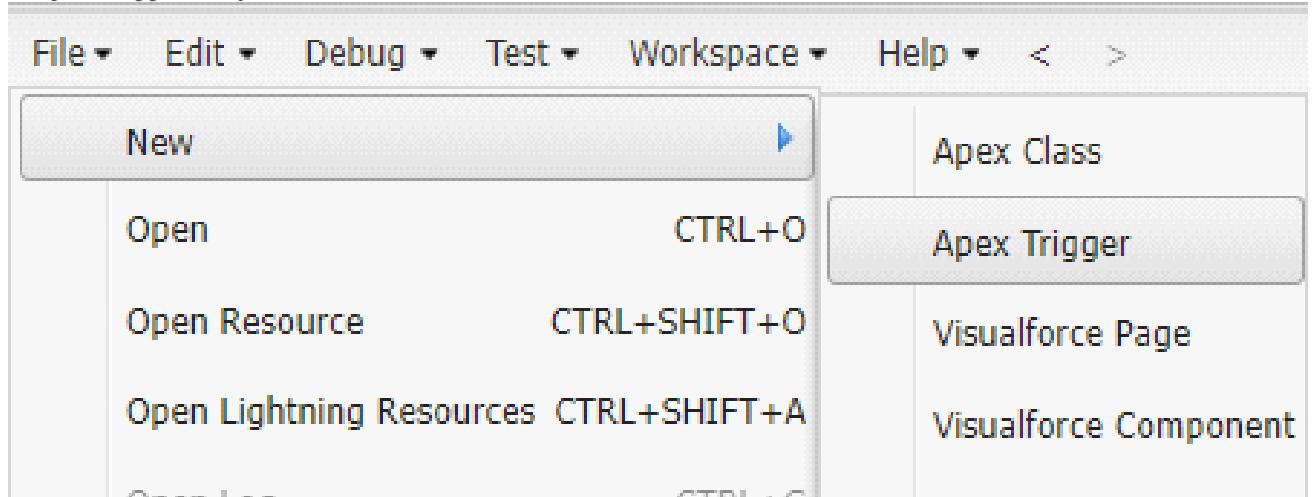
set<Id> AutomobileIds = new Set<Id>();
For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
    if(oppAutomobile.Automobile__c != null){
        AutomobileIds.add(oppAutomobile.Automobile__c);
    }
}
Map<Id, Automobile_Information__c> lstAutomobileInformation = new
map<Id, Automobile_Information__c>([SELECT Id, CreatedById, Quantity__c, Price__c FROM
Automobile_Information__c WHERE Id IN: AutomobileIds]);
For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
    If(OppAutomobile.Automobile__c ==
lstAutomobileInformation.get(OppAutomobile.Automobile__c).Id &&
lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c < OppAutomobile.Quantity__c){
        OppAutomobileaddError('the Number of Automobile u want are not Available !! the Automobile
are Available Count is ' + lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c );
    }
}
}

```

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 the File menu in the toolbar, and click on new? Trigger.
- Enter the trigger name and the object to be triggered.
- Name : OpportunityAutoMobileTrigger
- sObject : Opportunity_Automobile__c



New Apex Trigger

| | |
|----------|--|
| Name: | <input type="text"/> |
| sObject: | <input type="text"/>  |

Submit

Trigger :

Handler for the Opportunity_Automobile__c Object

OpportunityAutomobileHandler.apxc | OpportunityHandlerClass.apxc | OpportunityTrigger.apxt | **OpportunityAutoMobileTrigger.apxt**

Code Coverage: None API Version: 59

```

1 trigger OpportunityAutoMobileTrigger on Opportunity_Automobile__c (before insert,
2     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
3         OpportunityAutomobileHandler.quantityErrorOnAutomobileInformation(trigger.
4     }
5 }
```

Code:

```
trigger OpportunityAutoMobileTrigger on Opportunity_Automobile__c (before insert, before Update) {
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        OpportunityAutomobileHandler.quantityErrorOnAutomobileInformation(trigger.new);
    }
}
```

Invoice Creation Trigger

UseCase : Whenever an opportunity is Closed won then create the Invoice on the Bases of Opportunity Automobile Data.

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 “InvoiceCreation”.

```

1 public class InvoiceCreation {
2     public static void OpportunityClosedwonInvoiceGeneration(List<Opportunity> lstOpportunity, Map<Id,Opportunity>OldMapOpportunity){
3         set<Id> oppIds = new Set<Id>();
4         For(Opportunity opp : lstOpportunity){
5             if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
6                 oppIds.add(opp.Id);
7             }
8         }
9         List<Opportunity_Automobile__c> lstOpportunityAutomobile = [SELECT Unit_Price__c, Total_Price__c, Automobile__c, Quantity__c,Opportunity__c, Id FROM Opportunity_Automobile__c];
10        List<Invoice__c> lstInvoice = new List<Invoice__c>();
11        For(Opportunity_Automobile__c oppAuto : lstOpportunityAutomobile){
12            Invoice__c i = new Invoice__c();
13            i.Quantity__c = oppAuto.Quantity__c;
14            i.Unit_Price__c = oppAuto.Unit_Price__c;
15            i.Total_Price__c = oppAuto.Total_Price__c;
16            i.Purchase_Date__c = date.today();
17            i.Opportunity__c = oppAuto.Opportunity__c;
18            lstInvoice.add(i);
19        }
20        if(!lstInvoice.isEmpty()){
21            insert lstInvoice;
22        }
23    }
24 }
```

Code:

```

public class InvoiceCreation {
    public static void OpportunityClosedwonInvoiceGeneration(List<Opportunity> lstOpportunity,
Map<Id,Opportunity>OldMapOpportunity){
        set<Id> oppIds = new Set<Id>();
        For(Opportunity opp : lstOpportunity){
            if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
                oppIds.add(opp.Id);
            }
        }
        List<Opportunity_Automobile__c> lstOpportunityAutomobile = [SELECT Unit_Price__c,
Total_Price__c, Automobile__c, Quantity__c,Opportunity__c, Id FROM Opportunity_Automobile__c
WHERE Opportunity__c IN: oppIds];
        List<Invoice__c> lstInvoice = new List<Invoice__c>();
        For(Opportunity_Automobile__c oppAuto : lstOpportunityAutomobile){
            Invoice__c i = new Invoice__c();
            i.Quantity__c = oppAuto.Quantity__c;
            i.Unit_Price__c = oppAuto.Unit_Price__c;
            i.Total_Price__c = oppAuto.Total_Price__c;
            i.Purchase_Date__c = date.today();
            i.Opportunity__c = oppAuto.Opportunity__c;
            lstInvoice.add(i);
        }
        if(!lstInvoice.isEmpty()){
            insert lstInvoice;
        }
    }
}
```

Trigger Handler :

For this class we don't need to create any trigger, we will call this Code in “Opportunity Trigger”.

- Go on files and click on open.
- Click on triggers.
- Double click on OpportunityTrigger.

Open

| Entity Type | Entities | Related |
|------------------|--------------------------|---------------------------|
| Entity Type | Name | Name |
| Classes | OpportunityTrigger | ← Opportunity... ApexC... |
| Triggers | OpportunityAutoMobile... | ← Opportunity SObj... |
| Pages | | |
| Page Components | | |
| Objects | | |
| Static Resources | | |
| Packages | | |

Code Coverage: None | API Version: 59

```

1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2     if(trigger.isbefore && trigger.isUpdate){
3         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
4     }
5 }
6 IF(trigger.isafter && trigger.isupdate){
7     InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);
8 }
9 }
```

Trigger:

```
trigger OpportunityTrigger on Opportunity (before update, After Update) {
    if(trigger.isbefore && trigger.isUpdate){
        OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
    }
}
```

```

        IF(trigger.isafter && trigger.isupdate){
            InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);
        }
    }
}

```

Check Contact Role

UseCase : Whenever an opportunity is Going to Closed won then check it has the contact role or Not.

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 “ContactRoleCheck”.

```

public class ContactRoleCheck {
    public static void CheckcontactRoleonOpportunity(List<Opportunity> lstOpportunity, Map<Id,Opportunity>OldMapOpportunity)
    {
        List<OpportunityContactRole> lstContactRole = [SELECT Id From OpportunityContactRole WHERE OpportunityId IN: OldMapOpportunity.keySet()];
        For(Opportunity opp : lstOpportunity)
        {
            if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
                If(lstContactRole.isEmpty()){
                    opp.adderror('Please add contact Role on opportunity whenever Opportunity is Going to Closed Won.');
                }
            }
        }
    }
}

```

Trigger:

```

public class ContactRoleCheck {
    public static void CheckcontactRoleonOpportunity(List<Opportunity> lstOpportunity,
    Map<Id,Opportunity>OldMapOpportunity){
        List<OpportunityContactRole> lstContactRole = [SELECT Id From OpportunityContactRole WHERE
        OpportunityId IN: OldMapOpportunity.keySet()];
        For(Opportunity opp : lstOpportunity){
            if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
                If(lstContactRole.isEmpty()){
                    opp.adderror('Please add contact Role on opportunity whenever Opportunity is Going to Closed
        Won.');
                }
            }
        }
    }
}

```

Trigger Handler :

For this class we don't need to create any trigger, we will call this Code in "Opportunity Trigger".

- Go on files and click on open.
- Click on triggers.
- Double click on OpportunityTrigger.

| Open | | |
|------------------|--------------------------|------------------|
| Entity Type | Entities | Related |
| Entity Type | | |
| Classes | OpportunityTrigger | ← Opportunity... |
| Triggers | OpportunityAutoMobile... | ← Opportunity |
| Pages | | |
| Page Components | | |
| Objects | | |
| Static Resources | | |
| Packages | | |

Trigger Code :

```
Code Coverage: None API Version: 59
1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2     if(trigger.isbefore && trigger.isUpdate){
3         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap)
4         ContactRoleCheck.CheckcontactRoleonOpportunity(trigger.new, trigger.oldMap)
5     }
6     IF(trigger.isafter && trigger.isupdate){
7         InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap)
8     }
9 }
```

Trigger:

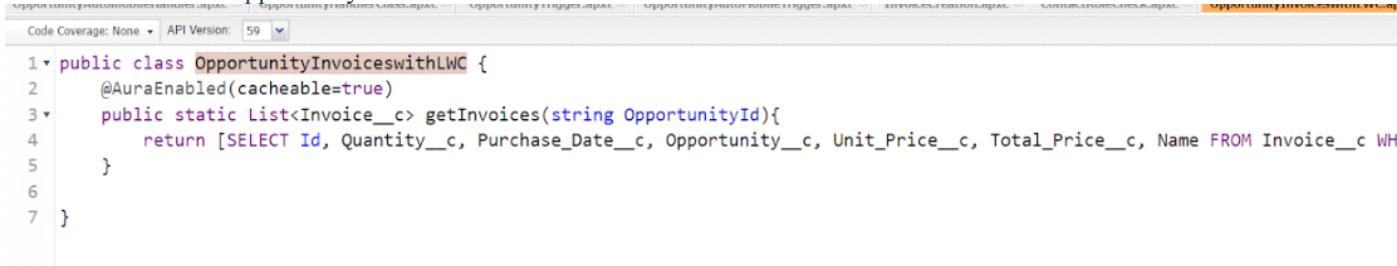
```
trigger OpportunityTrigger on Opportunity (before update, After Update) {  
    if(trigger.isbefore && trigger.isUpdate){  
        OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);  
        ContactRoleCheck.CheckcontactRoleonOpportunity(trigger.new, trigger.oldMap);  
    }  
    IF(trigger.isafter && trigger.isupdate){  
        InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);  
    }  
}
```

LWC Component:

Create Apex Class To Get Invoices

- Login to the respective 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 “OpportunityInvoiceswithLWC”.



```

Code Coverage: None API Version: 59
1 public class OpportunityInvoiceswithLWC {
2     @AuraEnabled(cacheable=true)
3     public static List<Invoice__c> getInvoices(string OpportunityId){
4         return [SELECT Id, Quantity__c, Purchase_Date__c, Opportunity__c, Unit_Price__c, Total_Price__c, Name FROM Invoice__c WHERE Opportunity__c =: OpportunityId];
5     }
6 }
7 }
```

Code:

```

public class OpportunityInvoiceswithLWC {
    @AuraEnabled(cacheable=true)
    public static List<Invoice__c> getInvoices(string OpportunityId){
        return [SELECT Id, Quantity__c, Purchase_Date__c, Opportunity__c, Unit_Price__c, Total_Price__c,
Name FROM Invoice__c WHERE Opportunity__c =: OpportunityId];
    }
}
```

Install Salesforce CLI

The Salesforce CLI is a powerful command line interface that simplifies development and build automation when working with your Salesforce org.

[Download and install Salesforce CLI](#)

To confirm that the Salesforce CLI is installed and working correctly, you can open a command prompt and type sfdx. This will display the version number of the Salesforce CLI that is currently installed on your system.

```
C:\Users\navee>sfdx  
Salesforce CLI
```

VERSION

```
sfdx-cli/7.182.1 win32-x64 node-v18.12.1
```

USAGE

```
$ sfdx [COMMAND]
```

TOPICS

| | |
|---------|--|
| alias | manage username aliases |
| auth | authorize an org for use with the Sale |
| config | configure the Salesforce CLI |
| force | tools for the Salesforce developer |
| info | access cli info from the command line |
| plugins | add/remove/create CLI plug-ins |
| version | |

Install Microsoft VS Code

VS Code, or Visual Studio Code, is a free, open-source code editor developed by Microsoft. It is a lightweight, cross-platform code editor that provides features such as debugging, Git integration, and support for a wide range of programming languages.

[Download the version of the software](#) that is compatible with your operating system and install it.

The following instructions are for Windows OS. Other operating systems may have slightly different steps.

Install The Salesforce Extension Pack

In the VS Code,

1. go to extensions (1) as shown in the image below.
2. Search with the Salesforce extension pack (2) as shown in the image below.
3. select Salesforce Extension Pack from the list (3) as shown in the image below.
4. Click the Install button (4) as shown in the image below.

The screenshot shows the VS Code Extensions Marketplace. On the left, there's a sidebar with icons for file operations, search, and extensions. The 'Extensions' icon is highlighted with a red box and the number '1'. The main area shows a search bar with 'salesforce extension pack' and a magnifying glass icon, also highlighted with a red box and the number '2'. Below the search bar, a list of extensions is displayed. The first item in the list is the 'Salesforce Extension ...' by Salesforce, which is highlighted with a red box and the number '3'. It has an 'Install' button. To the right of the list, there's a detailed view of the 'Salesforce Extension Pack' page, which includes a large blue cloud icon, the extension's name, its popularity (947K), its developer (Salesforce), and its description ('Extensions for developing on the Sa...'). At the bottom of this page, there's another 'Install' button highlighted with a red box and the number '4'.

The extension pack is installed successfully

Extension: Salesforce Extension Pack X



Salesforce Extension Pack

 Salesforce |  9,47,031 | 

Extensions for developing on the Salesforce Platform.

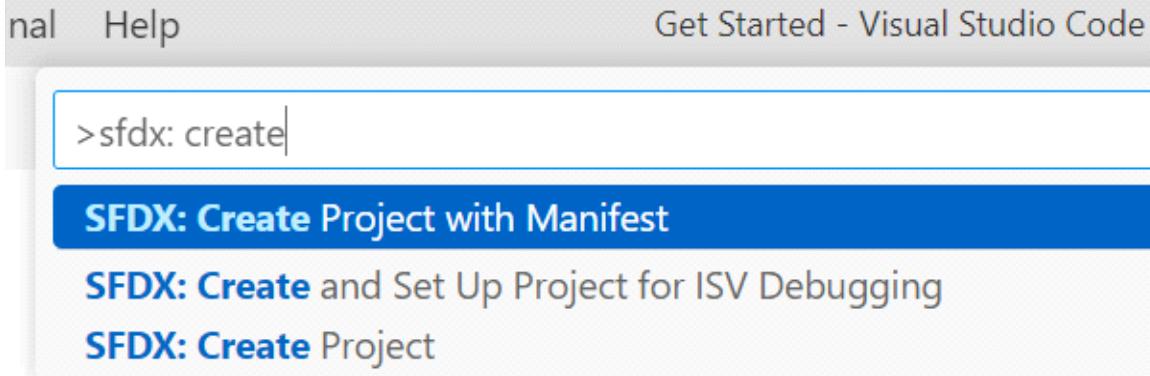
[Disable](#) [Uninstall](#) 

This extension is enabled globally.

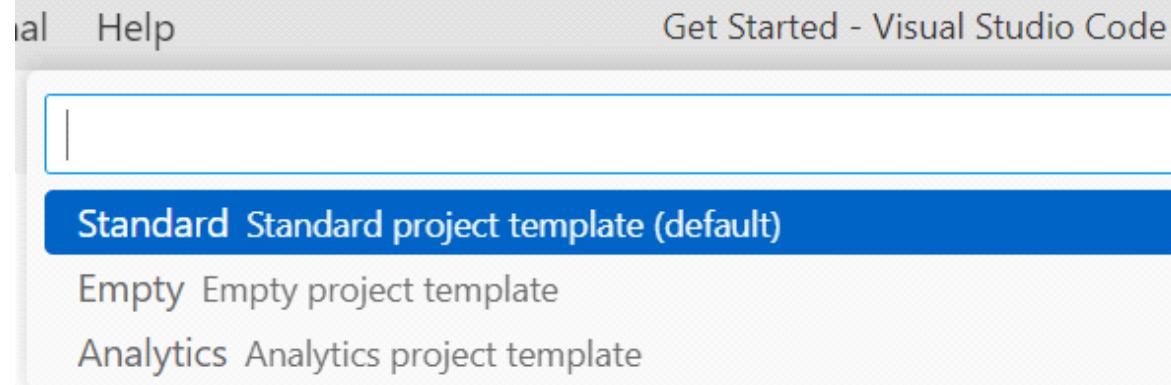
Install the Salesforce Extension Pack

Create A Project In VS Code

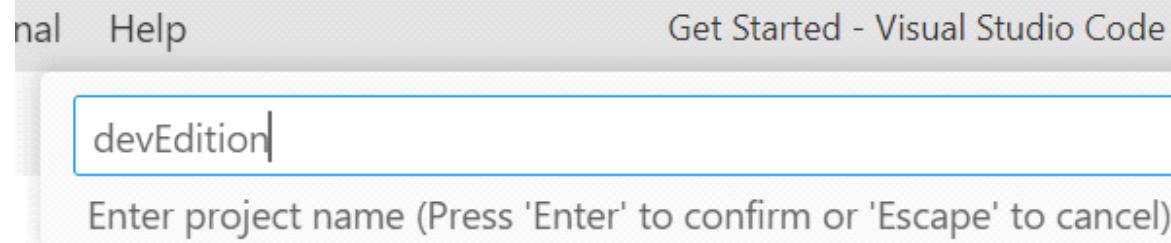
- Press CTRL + SHIFT + P, type sfdx: create
- select SFDX: Create Project with Manifest



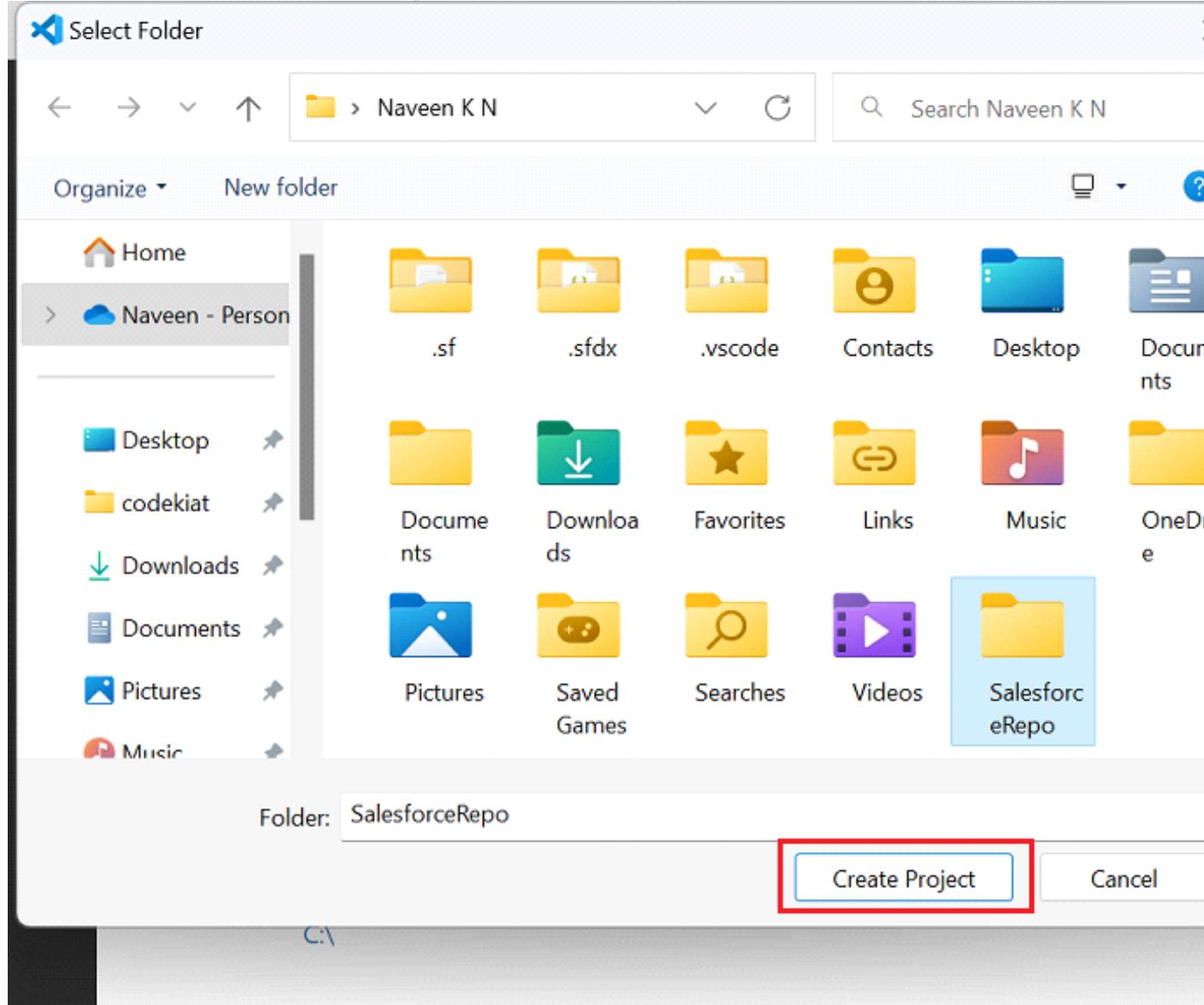
- Select the Standard project template



- Type a project name and Click Enter.



- Select the folder (create a new folder if required) and click Create Project



- The new project is created with package.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ApexClass</name>
    </types>
    <types>
        <members>*</members>
        <name>ApexComponent</name>
    </types>
    <types>
        <members>*</members>
        <name>ApexPage</name>
    </types>
    <types>
        <members>*</members>
        <name>ApexTestSuite</name>
    </types>
    <types>
        <members>*</members>
    </types>
</Package>

```

Default Package.xml contains various metadata types. I have updated Package.xml as shown below as we deal only with LWC in this article.

```

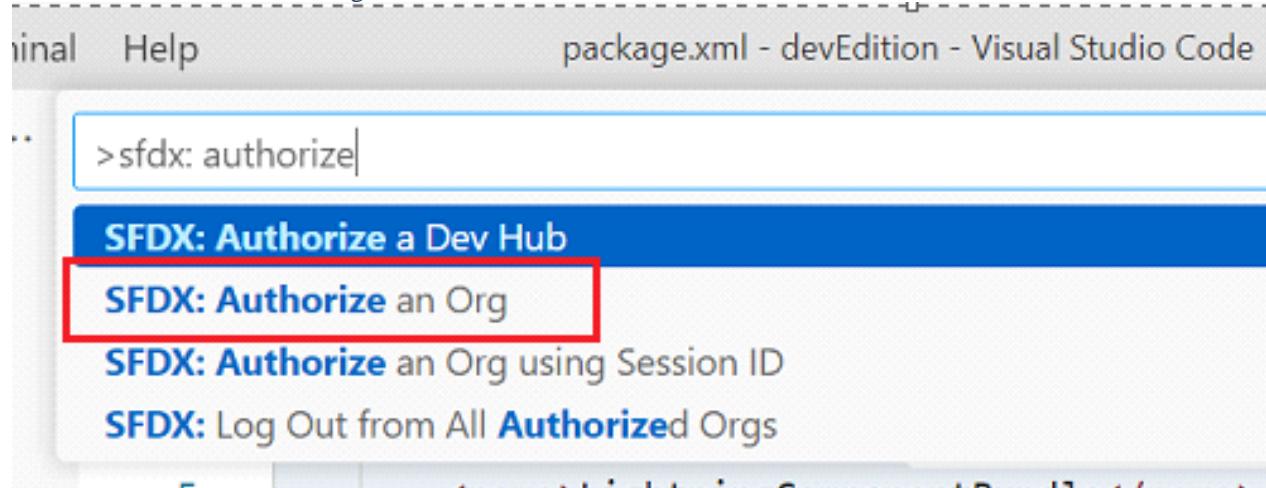
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>LightningComponentBundle</name>
    </types>
    <version>55.0</version>
</Package>

```

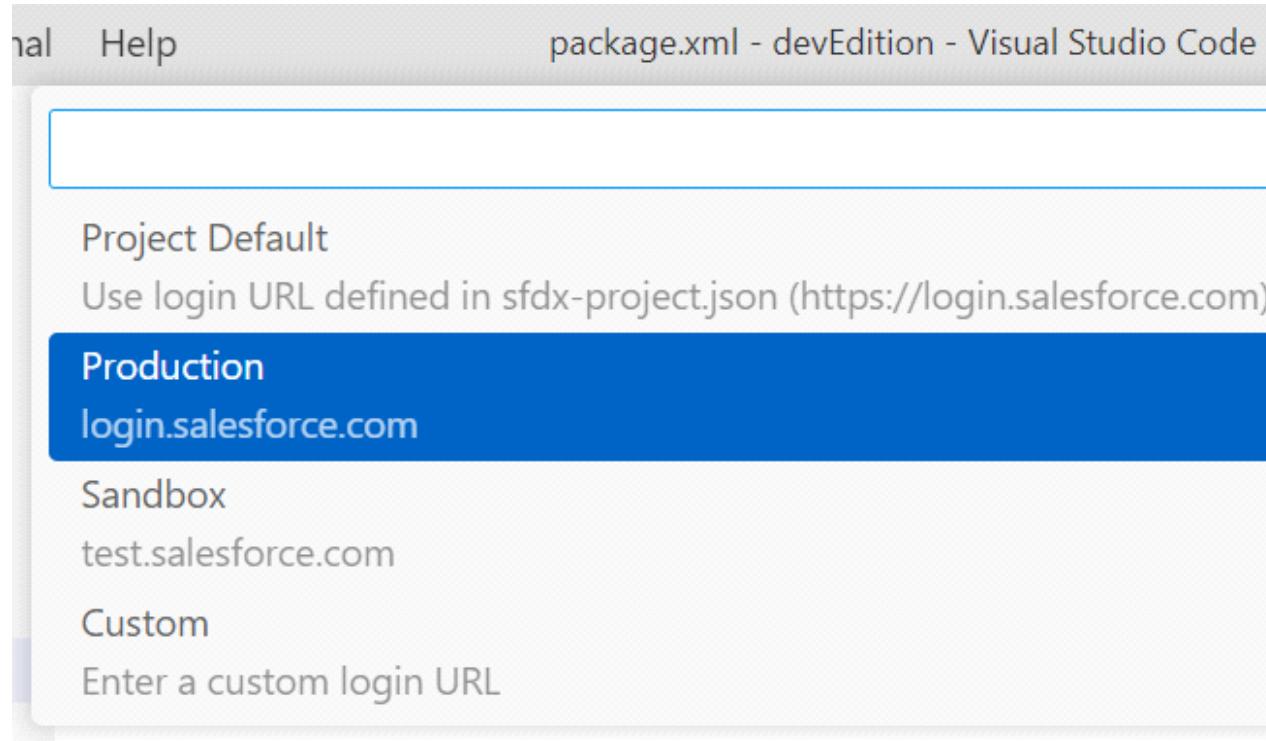
Authorize An Org

Establish a connection between the local project and the Salesforce instance to retrieve and deploy the components.

- Press CTRL + SHIFT + P, type sfdx: authorize.
- select SFDX: Authorize an Org from the list



- Choose your Salesforce instance.
For developer edition and production instances select Production.



- For this demonstration, I used the developer edition, hence it is Production.

- Give a project name and press Enter

nal Help

package.xml - devEdition - Visual Studio Code

devEdition

Enter an org alias or use the default alias (Press 'Enter' to confirm or 'Escape' to

- The Salesforce login page opens in the browser.

- Enter the credentials and click Log In

 login.salesforce.com/?startURL=%2Fsetup%2Fsecur%2FRe

codekia



Username

1 Saved Username

Password

Log In



Remember me

Forgot Your Password?

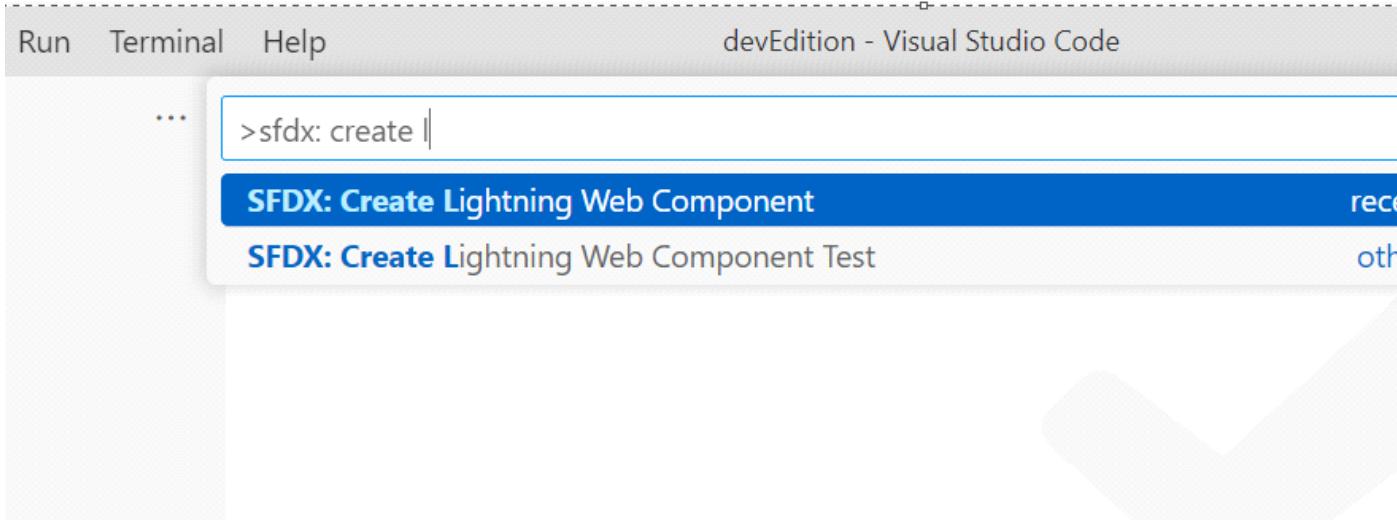
Use Custom Domain

1. It will be successfully authorized.

Create Lightning Web Component

XML File :

- In the VS Code, press CTRL + SHIFT + P, type sfdx: create lightning in the search bar, and select SFDX: Create Lightning Web Component



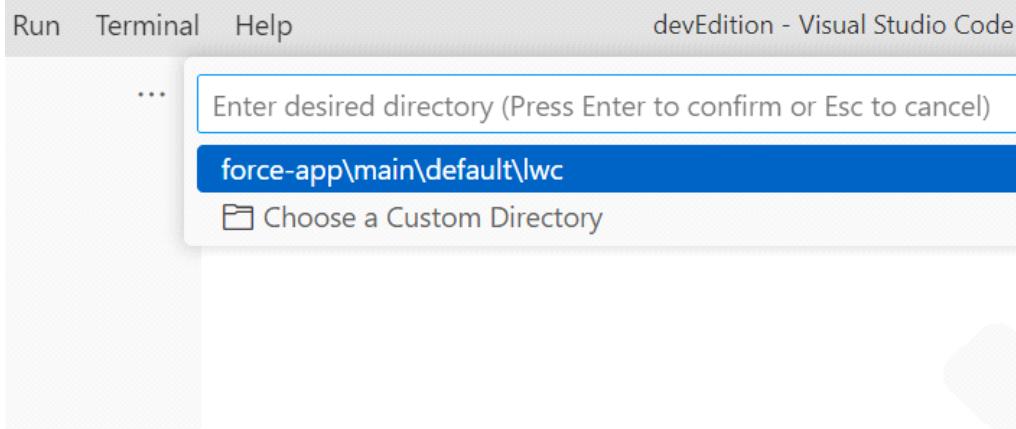
- Give the name “InvoiceOpportunity” and press Enter.

A screenshot of the Visual Studio Code code editor. The title bar says 'InvoiceOpportunity'. The editor shows a file with the following XML content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/
3   <apiVersion>58.0</apiVersion>
4   <isExposed>true</isExposed>
5   <targets>
6     <target>lightning__RecordAction</target>
7     <target>lightning__RecordPage</target>
8   </targets>
9 </LightningComponentBundle>
```

The code editor has syntax highlighting and a status bar at the bottom.

- Choose the directory.



- LWC is created successfully.

The screenshot shows the Visual Studio Code interface for an LWC project named 'LWC PROJECT PRACTISE'. The Explorer sidebar lists various components like 'customValidationInIWC', 'dataBaseWizardButton', etc. The code editor displays the contents of 'invoiceOpportunity.js-meta.xml'.

```

<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>58.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordAction</target>
    <target>lightning__RecordPage</target>
  </targets>
</LightningComponentBundle>

```

- Copy and paste the below-mentioned code in the InvoiceOpportunity.js-meta.xml and update the apiVersion tag with the latest API version.

XML File Code:

```

<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>58.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordAction</target>
    <target>lightning__RecordPage</target>
  </targets>
</LightningComponentBundle>

```

JS File :

- Copy and paste the below-mentioned code in the InvoiceOpportunity.js and update the apiVersion tag with the latest API version.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** LWC Project Practise
- Explorer:** Shows the project structure under "LWC PROJECT PRACTISE". The "invoiceOpportunity" folder is expanded, showing "invoiceOpportunity.html", "invoiceOpportunity.js" (which is selected), and "invoiceOpportunity.js-meta....". Other folders like "customValidationInIWC", "dataBaseWizardButton", etc., are also listed.
- Editor:** The "invoiceOpportunity.js" file is open in the editor. The code is as follows:

```
force-app > main > default > lwc > invoiceOpportunity > invoiceOpportunity.js
1 import { LightningElement, api, track, wire } from 'lwc';
2 import getInvoices from '@salesforce/apex/OpportunityInvoiceswithLWC.getInvoices';
3 export default class InvoiceOpportunity extends LightningElement {
4     @api recordId;
5     @track invoiceCollection
6     cols = [
7         {label:"ID" , fieldName:'Name'},
8         {label:"Opportunity Id" , fieldName:'Opportunity__c'},
9         {label:"Quantity" , fieldName:'Quantity__c'},
10        {label:"Unit Price" , fieldName:'Unit_Price__c'},
11        {label:"Total Price" , fieldName:'Total_Price__c'},
12        {label:"Purchase Date" , fieldName:'Purchase_Date__c'}
13    ]
14    @wire(getInvoices,{OpportunityId:$recordId})
15    invoicefunction({data,error}){
16        console.log(this.recordId +'this is record Id');
17        if(data){
18            console.log(data);
19            this.invoiceCollection = data
20        }
21        if(error){
22            console.log('this is error')
23            console.log(error);
24        }
25    }
26 }
```

JS File Code :

```
import { LightningElement, api, track, wire } from 'lwc';
import getInvoices from '@salesforce/apex/OpportunityInvoiceswithLWC.getInvoices';
export default class InvoiceOpportunity extends LightningElement {
    @api recordId;
    @track invoiceCollection
    cols = [
        {label:"ID" , fieldName:'Name'},
        {label:"Opportunity Id" , fieldName:'Opportunity__c'},
        {label:"Quantity" , fieldName:'Quantity__c'},
        {label:"Unit Price" , fieldName:'Unit_Price__c'},
        {label:"Total Price" , fieldName:'Total_Price__c'},
        {label:"Purchase Date" , fieldName:'Purchase_Date__c'}
    ]
    @wire(getInvoices,{OpportunityId:$recordId})
```

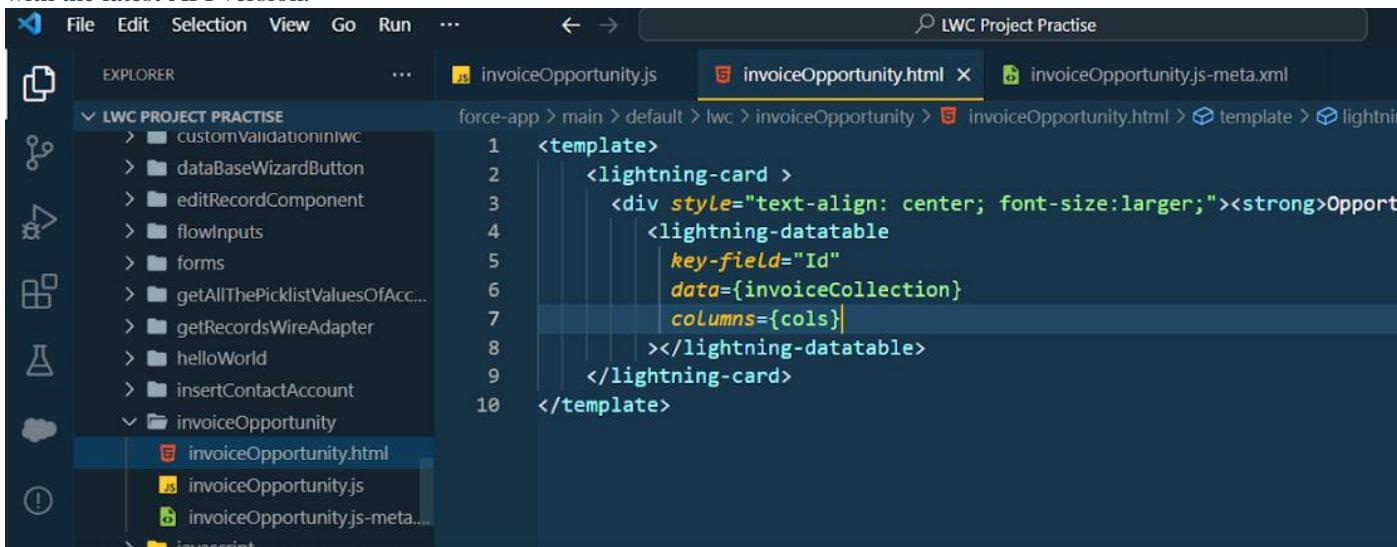
```

invoicefunction({data,error}){
    console.log(this.recordId +'this is record Id');
    if(data){
        console.log(data);
        this.invoiceCollection = data
    }if(error){
        console.log('this is error')
        console.log(error);
    }
}
}

```

HTML File :

- Copy and paste the below-mentioned code in the InvoiceOpportunity.html and update the apiVersion tag with the latest API version.



The screenshot shows a code editor interface for a Lightning Web Component project named "LWC PROJECT PRACTISE". The left sidebar displays the project structure with various components and utilities. The main editor area shows the "invoiceOpportunity.html" file content:

```

<template>
  <lightning-card>
    <div style="text-align: center; font-size:larger;"><strong>Opportunity Ivoices</strong></div>
    <lightning-datatable
      key-field="Id"
      data={invoiceCollection}
      columns={cols}
    ></lightning-datatable>
  </lightning-card>
</template>

```

HTML File Code:

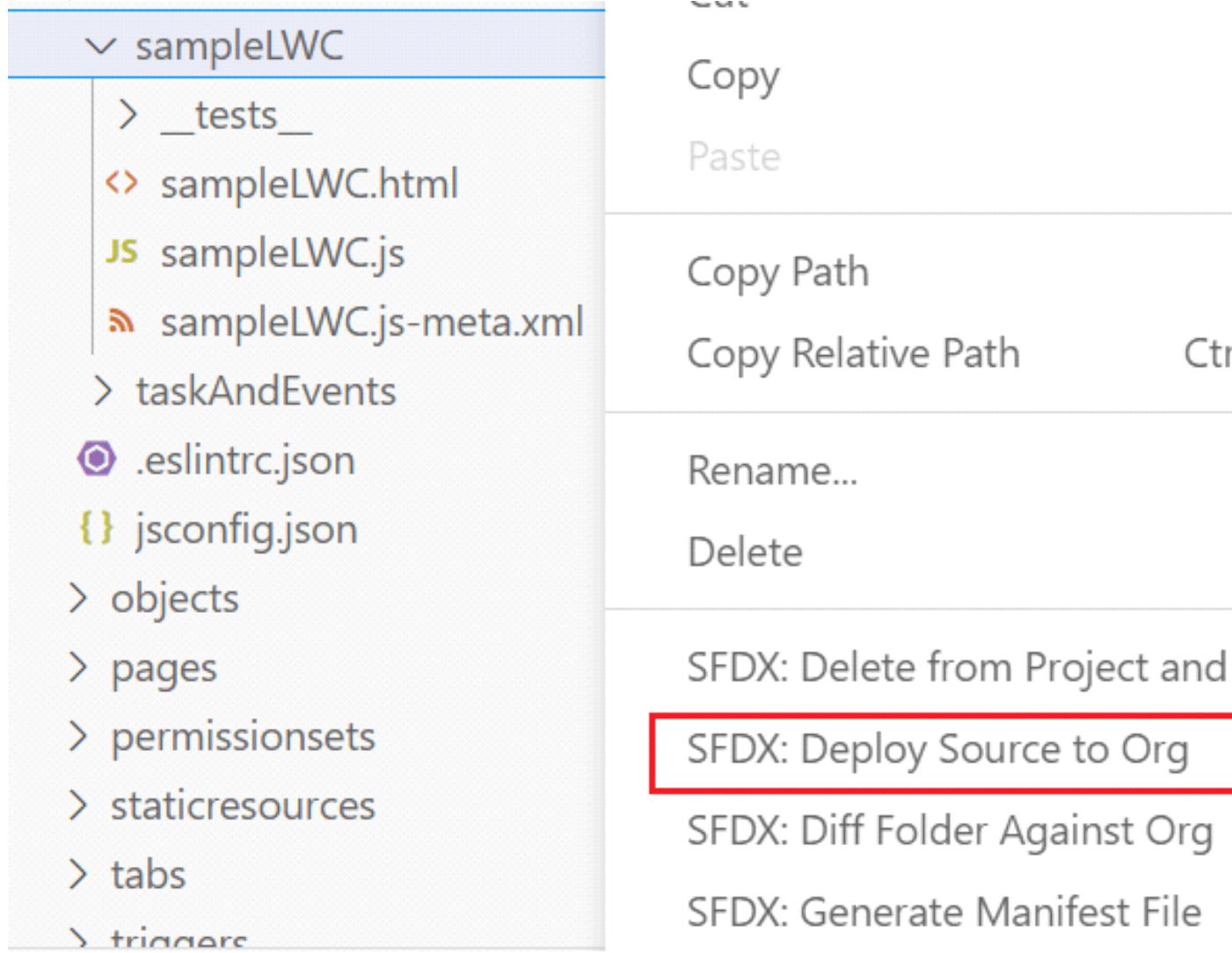
```

<template>
  <lightning-card>
    <div style="text-align: center; font-size:larger;"><strong>Opportunity Ivoices</strong></div>
    <lightning-datatable
      key-field="Id"
      data={invoiceCollection}
      columns={cols}
    ></lightning-datatable>
  </lightning-card>
</template>

```

Deploy Component:

- Right-click on the component folder, and select SFDX: Deploy Source to Org to deploy the component to the org.



- Once the deployment is complete, you will see the below-highlighted message in the output tab

| STATE | FULL NAME | TYPE | PROJECT PATH |
|---------|--------------------|--------------------------|--|
| Changed | invoiceOpportunity | LightningComponentBundle | force-app\main\default\lwc\invoiceOpportunity\ |
| Changed | invoiceOpportunity | LightningComponentBundle | force-app\main\default\lwc\invoiceOpportunity\ |
| Changed | invoiceOpportunity | LightningComponentBundle | force-app\main\default\lwc\invoiceOpportunity\ |

10:05:04.331 ended SFDX: Deploy Source to Org

Create Button To Add On Opportunity

- To add the newly created component to the view, Go to Salesforce Setup
- Click on Object Manager
- Search Opportunity and Click on it .
- click on Button Links and Action.
- click on the New Action.

SETUP > OBJECT MANAGER
Opportunity

Buttons, Links, and Actions
14 Items, Sorted by Label

| LABEL | NAME | DESCRIPTION | TYPE | CONTENT SOURCE |
|--------------------|----------------|-------------|------------------|----------------|
| Add to Campaign | AddInfluence | | | Standard page |
| Clone | Clone | | | Standard page |
| Clone with Related | DeepClone | | | Standard page |
| Delete | Delete | | | Standard page |
| Delivery Status | DeliveryStatus | | Detail Page Link | URL |
| Edit | Edit | | | Standard page |

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters

- Select Action type as Lightning Web Component
- Select the InvoiceOpportunity component
 - a. Label :- Invoices
 - b. Name :- Invoices
- As given on below image

SETUP > OBJECT MANAGER
Opportunity

Opportunity Actions
New Action

Enter Action Information

| | |
|-------------------------|-------------------------|
| Object Name | Opportunity |
| Action Type | Lightning Web Component |
| Lightning Web Component | c:invoiceOpportunity |
| Subtype | Screen Action |
| Standard Label Type | --None-- |
| Label | Invoices |
| Name | Invoices |
| Description | |
| Icon | |

Save Cancel

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

- Click on Save and your action Button is Ready.

Add InvoiceOpportunity Into Opportunity Record Page

- On Opportunity Object Manager Click on Page layout.
- Click on OpportunityLayout.

The screenshot shows the Salesforce Object Manager interface for the Opportunity object. The left sidebar has a 'Page Layouts' tab highlighted with a red box. The main area displays a table of page layouts:

| PAGE LAYOUT NAME | CREATED BY | MODIFIED BY |
|--------------------------------|--------------------------------------|---------------------------------------|
| Opportunity (Marketing) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity (Sales) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity (Support) Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 28/11/2023, 9:45 am |
| Opportunity Layout | Mohammad Sameer, 22/11/2023, 2:19 pm | Mohammad Sameer, 01/12/2023, 10:57 am |

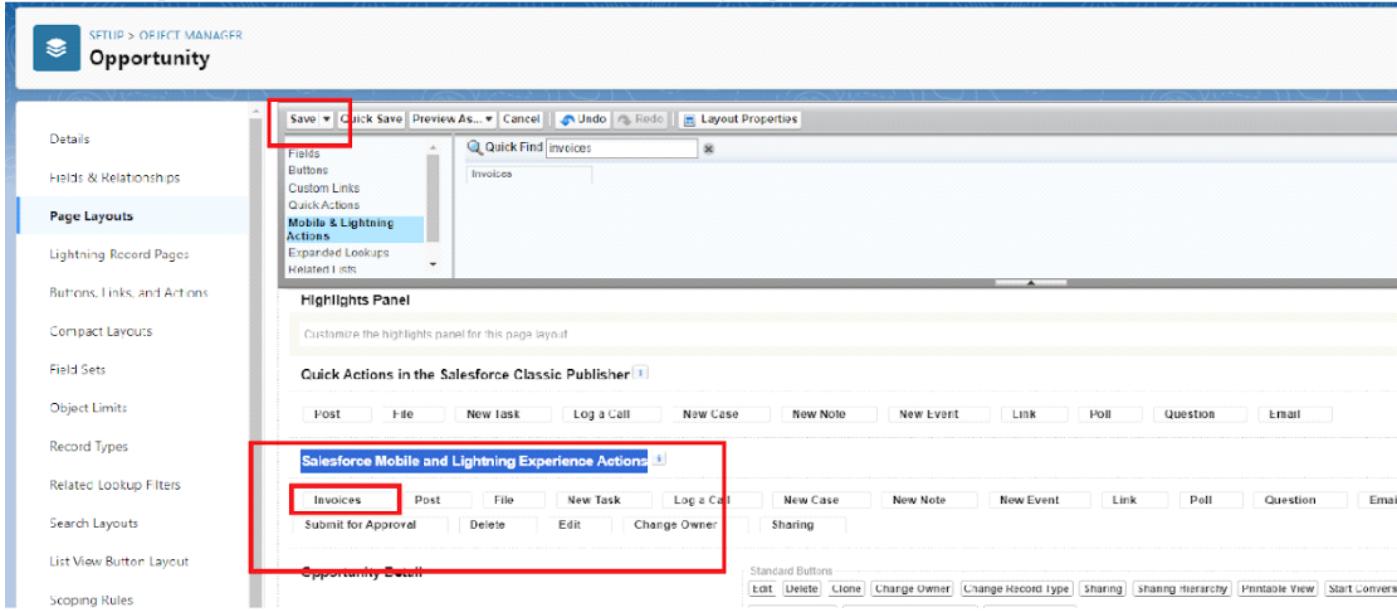
- Click on Mobile And Lightning Action as show on below Image

The screenshot shows the 'Opportunity Layout' configuration screen. The left sidebar has a 'Page Layouts' tab selected. The main area shows the 'Mobile & Lightning Actions' section highlighted with a red box. The actions listed include:

- Add to an Action...
- Clone
- Edit
- Link
- New Account
- New Case
- New Group
- New Lead
- New Note
- New Opportunity
- New Task
- Post
- File
- New Task
- Log a Call
- New Case
- New Note
- New Event
- Link
- Poll
- Question
- Email
- Delete
- Edit
- Change Owner
- Sharing

- Search for invoice on Quick Find.

- Drag and Drop the Invoice into Salesforce Mobile and Lightning Experience Actions.



Click on Save.

Apex Schedulers

Delete Opportunity Schedule Class

Objective :

- Through this schedulable class, we can see all the Closed Lost Opportunities.
- We can delete all the Closed lost Opportunities by this Scheduled method on every monday as weekly.

- Login to the respective 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 “DeleteClosedLostOpportunities”

CODE SNIPPET :

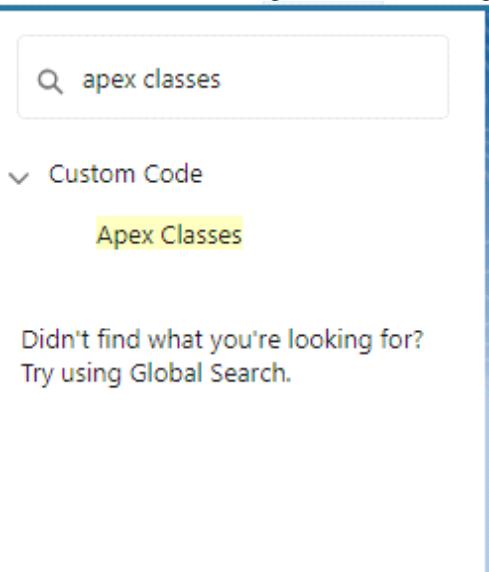
The screenshot shows the Salesforce code editor with the tab bar at the top containing several tabs: OpportunityAutomobileHandler.apxc, OpportunityHandlerClass.apxc, OpportunityTrigger.apxt, OpportunityAutoMobileTrigger.apxt, InvoiceCreation.apxc, ContactRoleCheck.apxc, and OpportunityInvoiceswithl. Below the tabs, there is a dropdown for 'Code Coverage' set to 'None' and a dropdown for 'API Version' set to '59'. The main area displays the Apex code for the DeleteClosedLostOpportunities class:

```
1 public class DeleteClosedLostOpportunities implements Schedulable{
2     public static void execute(SchedulableContext sc){
3         List<Opportunity> getLostOpportunities = [SELECT Id, Name From Opportunity Where StageName =: 'Closed Lost' LIMIT 5000];
4         if(!getLostOpportunities.IsEmpty()){
5             Delete getLostOpportunities;
6         }
7     }
8 }
```

```
public class DeleteClosedLostOpportunities implements Schedulable{
    public static void execute(SchedulableContext sc){
        List<Opportunity> getLostOpportunities = [SELECT Id, Name From Opportunity Where StageName =: 'Closed Lost' LIMIT 50000];
        if(!getLostOpportunities.IsEmpty()){
            Delete getLostOpportunities;
        }
    }
}
```

Schedule the Apex class:

- Go to the Home page in your salesforce account.
- In the search bar, enter Apex and click on Apex Classes.



The screenshot shows the Salesforce Setup interface with the 'Apex Classes' tab selected. The main content area is titled 'Apex Classes' and contains a brief introduction about Apex Code. A green callout box displays 'Percent of Apex Used: 0.14%' and notes that 8,427 characters are used out of a limit of 6,000,000. Below this are links for 'Estimate your organization's code coverage' and 'Compile all classes'. The table below lists six Apex classes with columns for Action, Name, Namespace Prefix, API Version, Status, Size Without Comments, and Last Modified By.

| Action | Name ↑ | Namespace Prefix | API Version | Status | Size Without Comments | Last Modified By |
|-----------------------|-------------------------------|------------------|-------------|--------|-----------------------|----------------------------|
| Edit Del Security | ContactRoleCheck | | 59.0 | Active | 672 | Mohammad Sameer 29/11/2023 |
| Edit Del Security | DeleteClosedLostOpportunities | | 59.0 | Active | 356 | Mohammad Sameer 01/12/2023 |
| Edit Del Security | InvoiceCreation | | 59.0 | Active | 1,243 | Mohammad Sameer 29/11/2023 |
| Edit Del Security | OpportunityAutomobileHandler | | 59.0 | Active | 1,085 | Mohammad Sameer 29/11/2023 |
| Edit Del Security | OpportunityHandlerClass | | 59.0 | Active | 4,041 | Mohammad Sameer 29/11/2023 |
| Edit Del Security | OpportunityInvoiceswithLWC | | 59.0 | Active | 319 | Mohammad Sameer 29/11/2023 |

- Click on Schedule Apex and enter the Job name.

- Job Name : DeleteOpportunitySchedule

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

| | | <input type="button" value="Save"/> | <input type="button" value="Cancel"/> |
|---|--|--|--|
| Job Name | <input type="text" value="DeleteOpportunitySchedule"/> | | |
| Apex Class | <input type="text" value="DeleteClosedLostOpportuni"/>  | | |
| Schedule Apex Execution | | Frequency | <input checked="" type="radio"/> Weekly <input type="radio"/> Monthly |
| | | <input type="checkbox"/> Sunday <input checked="" type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input type="checkbox"/> Friday <input type="checkbox"/> Saturday | |
| Start | <input type="text" value="01/12/2023"/> [01/12/2023] | | |
| End | <input type="text" value="01/01/2024"/> [01/12/2023] | | |
| Preferred Start Time | <input type="text" value="10:00 am"/>  | | |
| Exact start time will depend on job queue | | | |

- Now click on the search icon present near the Apex class : Goto the Lookup icon beside ? click on it ? select DeleteClosedLostOpportunities.

Apex Classes ~ Salesforce - Developer Edition - Google Chrome

smartbridge328-dev-ed.develop.my.salesforce.com/_ui/common/data/LookupPage?lkfm=edit... 

Lookup

DeleteClosedLostOpportuni

You can use "*" as a wildcard next to other characters to improve your search results.

< [Clear Search Results](#)

Search Results

| Name | Namespace Prefix | Api Version |
|---|------------------|-------------|
| DeleteClosedLostOpportunities | | 59 |

Copyright © 2000-2023 salesforce.com, inc. All rights reserved.

1. In the Schedule Apex section , select weekly and select Monday mentioned and preferred time as 10:00 AM.

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

Save Cancel

| | |
|---|--|
| Job Name | <input type="text" value="DeleteOpportunitySchedule"/> |
| Apex Class | <input type="text" value="DeleteClosedLostOpportuni"/>  |
| Schedule Apex Execution | |
| Frequency | |
| <input checked="" type="radio"/> Weekly <input type="radio"/> Monthly | |
| Recurs every week on | |
| <input type="checkbox"/> Sunday | |
| <input checked="" type="checkbox"/> Monday | |
| <input type="checkbox"/> Tuesday | |
| <input type="checkbox"/> Wednesday | |
| <input type="checkbox"/> Thursday | |
| <input type="checkbox"/> Friday | |
| <input type="checkbox"/> Saturday | |
| Start | <input type="text" value="01/12/2023"/> [01/12/2023] |
| End | <input type="text" value="01/01/2024"/> [01/12/2023] |
| Preferred Start Time | <input type="text" value="10:00 am ▾"/> |
| Exact start time will depend on job queue activity. | |

1. Click on Save. Now enter Apex in the search box and select Apex jobs.

The screenshot shows the Salesforce Setup interface with the following navigation bar:

- Setup
- Home
- Object Manager

The main content area displays the "Scheduled Jobs" page under the "SETUP" tab. The page title is "Scheduled Jobs". A sidebar on the left lists various setup categories:

- Setup Home
- Service Setup Assistant
- Commerce Setup Center
- Multi-Factor Authentication Assistant
- Hyperforce Assistant
- Release Updates
- Lightning Experience Transition Assistant
- Salesforce Mobile App
- Lightning Usage

The main content area includes a "View:" dropdown set to "All Scheduled Jobs" and a "Create New View" button. Below this is a table listing scheduled jobs:

| Action | Job Name | Submitted By | Submitted | Started | Next Scheduled |
|--|--|------------------|----------------------|----------------------|----------------------|
| Manage Del | DeleteOpportunitySchedule | Sameer. Mohammad | 01/12/2023, 12:02 pm | | 04/12/2023, 10:00 am |
| Del | Metalytics Data Loader Job for Org : 00D5 00000DhWhY | User.Integration | 22/11/2023, 2:21 pm | 30/11/2023, 10:32 pm | 01/12/2023, 10:00 am |

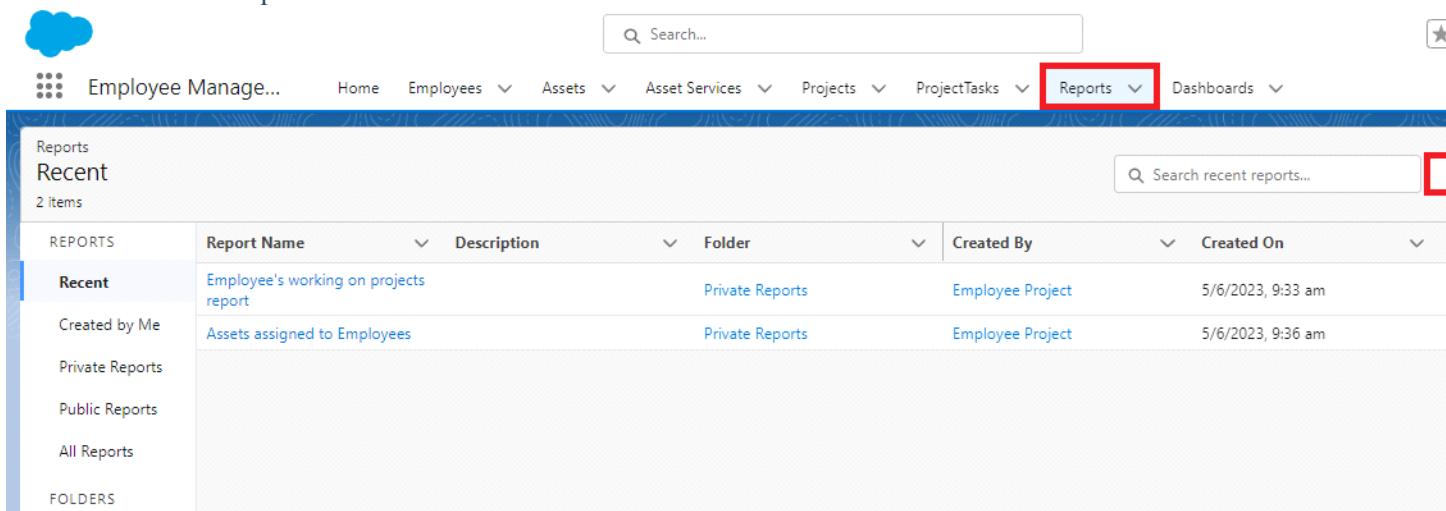
At the bottom right of the table, there are links for navigating between views: A | B | C | D | E | F | G | H | I | J | K | L.

You can see that the batch job is in queue and will run whenever the day mentioned comes.

Reports

Create Report On Opportunity

- Go to the app >> click on the reports tab
- Click New Report.



The screenshot shows a software application interface with a top navigation bar and a central report list area.

Top Navigation Bar:

- Employee Manage...
- Home
- Employees
- Assets
- Asset Services
- Projects
- ProjectTasks
- Reports** (highlighted with a red box)
- Dashboards

Search and Filter:

- Search bar: Search...
- Recent reports search bar: Search recent reports...

Left Sidebar:

- Reports
- Recent (selected)
- 2 items
- REPORTS
- Recent: Employee's working on projects report (Private Reports, Employee Project, 5/6/2023, 9:33 am)
- Created by Me: Assets assigned to Employees (Private Reports, Employee Project, 5/6/2023, 9:36 am)
- Private Reports
- Public Reports
- All Reports
- FOLDERS

1. Select report type from category or from report type panel or from search panel >> click on start report.

Create Report

| Category | Select a Report Type | | Details |
|----------------------------------|---|----------|---|
| Recently Used | <input type="text" value="Opportunities"/> X | | <p>Created No Reports</p> <p>Objects</p> <ul style="list-style-type: none"> █ Accounts █ Products █ Campaigns █ Users |
| All | Report Type Name | Category | |
| Accounts & Contacts | Opportunities | Standard | |
| Opportunities | Opportunities with Products | Standard | |
| Customer Support Reports | Opportunities with Contact Roles | Standard | |
| Leads | Opportunities with Partners | Standard | |
| Campaigns | Opportunities with Competitors | Standard | |
| Activities | Opportunities with Contact Roles and Products | Standard | |
| Contracts and Orders | Opportunities with Opportunity Automobiles | Standard | |
| Price Books, Products and Assets | Opportunities with Opportunity Automobiles and Automobile | Standard | |
| Administrative Reports | Opportunities with Invoice | Standard | |
| File and Content Reports | Campaigns with Opportunities | Standard | |
| Individuals | Campaigns with Influenced Opportunities | Standard | |
| | Activities with Opportunities | Standard | |
| | Products with Opportunities | Standard | |

1. Customize your report

- Add fields from left pane as shown below

Employee Management Home Employees Assets Asset Services Projects Project Tasks Reports Dashboards

REPORT New Employees Report Employees

Outline Filters 1 Previewing a limited number of records. Run the report to see everything.

| | Employee: Employee Name | Employee: ID | Reports to | Login Time | Logout Time | Mode of Work | LinkedIn Profile |
|---|-------------------------|----------------|------------|------------|-------------|--------------|-------------------------|
| 1 | Employee | #005000000HYtD | - | - | - | - | Http://Pentest/LinkedIn |
| 2 | Erica Jordan test | #005000000HygV | - | 8:00 am | 5:00 pm | - | Http://LinkedIn |

Groups GROUP ROWS Add group...

Columns Add column...

Employee: Employee Name

Employee: ID

Reports to

Login Time

Logout Time

Mode of Work

LinkedIn Profile

REPORT Opportunity Closed Won Report Opportunities

Previewing a limited number of records. Run the report to see everything.

| Account Name | Opportunity Name | Owner Role | Opportunity Owner | Stage | Next Step |
|---|---|------------|-------------------|---------------------|-----------|
| - (1) | Test | - | Mohammad Sameer | Closed Won | - |
| Subtotal | | | | | |
| Burlington Textiles Corp o' America (1) | Burlington Textiles Weaving Plant Generator | - | Mohammad Sameer | Closed Won | - |
| Subtotal | | | | | |
| Dickenson plc (1) | Dickenson Mobile Generators | - | Mohammad Sameer | Qualification | - |
| Subtotal | | | | | |
| Edge Communications (4) | Edge Emergency Generator | - | Mohammad Sameer | Closed Won | - |
| | Edge Installation | - | Mohammad Sameer | Closed Won | - |
| | Edge SLA | - | Mohammad Sameer | Closed Won | - |
| | Edge Emergency Generator | - | Mohammad Sameer | Id. Decision Makers | - |
| Subtotal | | | | | |
| Grand Hotels & Resorts Ltd (5) | Grand Hotels Kitchen Generator | - | Mohammad Sameer | Id. Decision Makers | - |
| | Grand Hotels Guest Portable Generators | - | Mohammad Sameer | Value Proposition | - |
| | Grand Hotels Generator Installations | - | Mohammad Sameer | Closed Won | - |
| | Grand Hotels SLA | - | Mohammad Sameer | Closed Won | - |
| | Grand Hotels Emergency Generators | - | Mohammad Sameer | Closed Won | - |
| Subtotal | | | | | |

Filters Add filter...

Show Me All opportunities

Close Date All Time

Opportunity Status Any

Probability All

Add the Above Filter as well.

1. Save or run it.

Note: Reports may get varied from the above pictures as the data might be different.

Create Report On Automobile Information

- Create a report with a report type: “Automobile Information”.

The screenshot shows a CRM application interface with a navigation bar at the top. The main area is titled "Automobile Information Report". On the left, there's a sidebar labeled "REPORT" with sections for "Outline" and "Filters". The "Filters" section is highlighted with a red box. It contains a "Groups" section with "Add group..." and a "Columns" section with "Add column...". Below these are several filter criteria: "Automobile Information: Name Of Manufacturer", "Model", "Built date", "Total Number of Cylinders", "Colour", "# Quantity", "# Price", and "VIN", each with an "x" icon to remove it. To the right of the sidebar is a table titled "Automobile Information" with the following columns: Name Of Manufacturer, Model, Built date, Total Number of Cylinders, Colour, Quantity, Price, and VIN. The table lists various car models and their details.

| Automobile Information: Name Of Manufacturer | Model | Built date | Total Number of Cylinders | Colour | Quantity | Price | VIN |
|--|----------|------------|---------------------------|--------|----------|---------|-------------------|
| Toyota | Corolla | 15-05-2022 | 4 | Red | 12 | ₹20.00 | 1IICCM02633A00435 |
| Ford | Mustang | 10-01-2023 | 8 | Blue | 54 | ₹35.00 | 2C5CDZA94KH12345 |
| Subaru | Outback | 14-10-2022 | 6 | Green | 55 | ₹30.00 | 5JGTF2H51EL123456 |
| Hyundai | Sonata | 08-06-2022 | 4 | Red | 78 | ₹26.00 | 1G1YY5ZG555123456 |
| Nissan | Altima | 25-07-2023 | 4 | Silver | 77 | ₹24.00 | 2T2HA31U45C123456 |
| Audi | A4 | 12-08-2022 | 4 | Blue | 9 | ₹33.00 | 1GNEK13R7XJ12345 |
| MERCEDES-BENZ | C-Class | 18-09-2023 | 4 | Grey | 24 | ₹38.00 | IN1B11CPXHW12345 |
| BMW | 3 Series | 05-04-2023 | 6 | White | 115 | ₹42.00 | WA1VAAF74KD123456 |
| Chevrolet | Malibu | 30-11-2022 | 6 | Black | 33 | ₹28.00 | 5VJ3E1E3KF123456 |
| | | | | | 450 | ₹276.00 | |

Filters:-

The screenshot shows a Microsoft Dynamics 365 Sales report interface. At the top, there's a navigation bar with links for Accounts, Contacts, Opportunities, Automobile Information, Opportunity Automobiles, Invoice, Reports, and Dashboards. Below the navigation is a search bar. The main area is titled 'REPORT ▾' and 'Automobile Information Report'. A sub-section title 'Automobile Information' is visible. On the left, there's a sidebar with 'Outline' and 'Filters' tabs, and a 'Filters' dropdown menu open, showing options like 'Add filter...', 'Show Me', 'All automobile information', and 'Automobile Information: Created Date' (with 'All Time' selected). The main content area displays a table of automobile data with 10 rows. The columns are: 'Automobile Information: Name Of Manufacturer', 'Model', 'Built date', 'Total Number of Cylinders', 'Colour', 'Quantity', 'Price', and 'VIN'. The data includes entries for Toyota, Ford, Subaru, Hyundai, Nissan, Audi, Mercedes-Benz, BMW, and Chevrolet.

| | Automobile Information: Name Of Manufacturer | Model | Built date | Total Number of Cylinders | Colour | Quantity | Price | VIN |
|----|--|----------|------------|---------------------------|--------|----------|---------|-----------------|
| 1 | Toyota | Corolla | 15-05-2022 | 4 | Red | 12 | ₹20.00 | 1HGCM82633A00 |
| 2 | Ford | Mustang | 10-01-2023 | 8 | Blue | 54 | ₹35.00 | 2C3CDZAG4KH1234 |
| 3 | Subaru | Outback | 14-10-2023 | 6 | Green | 56 | ₹30.00 | 5J6TF2H51EL1234 |
| 4 | Hyundai | Sonata | 08-06-2022 | 4 | Red | 78 | ₹26.00 | 1G1YY32G551234 |
| 5 | Nissan | Altima | 25-02-2023 | 4 | Silver | 77 | ₹24.00 | 2T2HA2U45C1234 |
| 6 | Audi | A4 | 12-08-2022 | 4 | Blue | 9 | ₹33.00 | 1GNEK13R7XJ1234 |
| 7 | Mercedes-Benz | C-Class | 18-09-2023 | 4 | Gray | 24 | ₹38.00 | JN1BJ1CPXHW1234 |
| 8 | BMW | 3 Series | 05-04-2023 | 6 | White | 116 | ₹42.00 | WA1VAAF74KD1234 |
| 9 | Chevrolet | Malibu | 30-11-2022 | 6 | Black | 33 | ₹28.00 | 5VJ3E1EA3KF1234 |
| 10 | | | | | | 459 | ₹276.00 | |

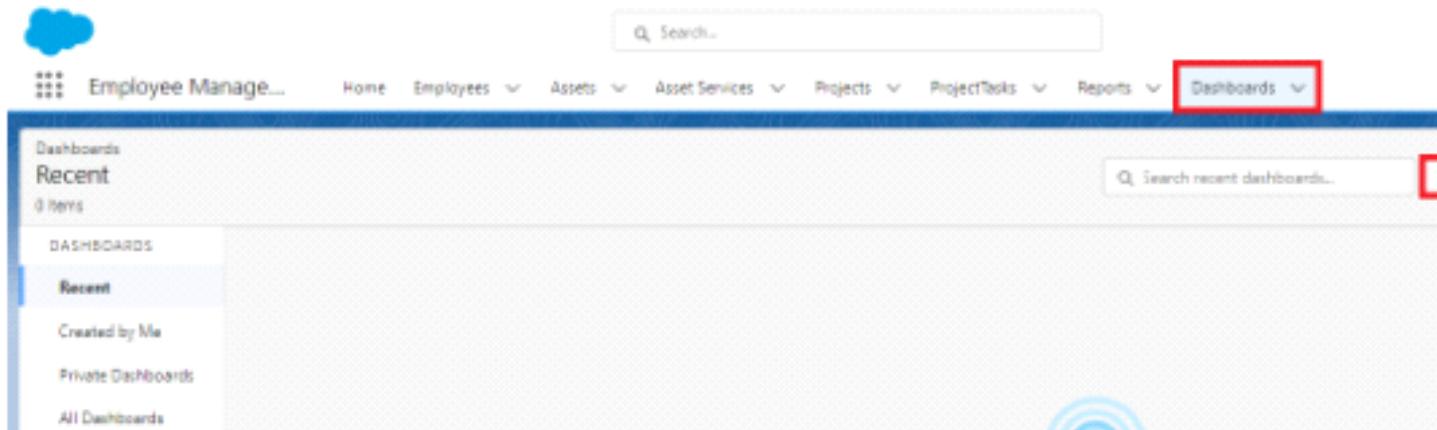
- Create a Report by using “Opportunities with Opportunity Automobiles and Automobile” Report Type.

Dashboard

Sales Dashboard

Create Dashboard

- Go to the app ? click on the Dashboards tabs.



1. Give a Name and click on Create.

New Dashboard

* Name

Description

Folder

Name : Automobile Sales

1. Select add component.

- Select a Report and click on select.

- Click Add then click on Save and then click on Done.

The Created Dashboard will look like this.

| Account Name | Sum of Quantity |
|--------------|-----------------|
| Burlin... | 2 |
| Edge ... | 34 |
| Unit... | 126 |
| Total | 160 |

| Account Name | Record Count |
|------------------|--------------|
| Burlington T... | 1 |
| Dickenson plc | 1 |
| Edge Comm... | 4 |
| Express Log... | 3 |
| GenePoint | 3 |
| Grand Hotel... | 5 |
| Pyramid Co... | 1 |
| United Oil &... | 10 |
| University of... | 3 |

| Manufacturer |
|--------------|
| Audi |
| BMW |
| Chevrolet |
| Ford |

*****END*****