CSI COLLEGE OF ENGINEERING

NAAN MUDHALVAN PROJECT

COLLEGE CODE - 7106

Sales Automobile Using Salesforce CRM

TEAM MEMBERS

AJITH M - 70C1EE6EE627DEF8CDE4FD7E830C2B43

DEEPAVAHINI M - 828022806F18D1C84254A19EF846F39E

PRANESH R - BB6789E387A0BB788394E5FA6C90ABE8

SANJEEDHA S - fe9288d6c9033fabc324c52492cd1287

Sales Automobile Using Salesforce CRM

User Story:

Salesforce CRM for automobile sales enhances efficiency and customer satisfaction by streamlining lead management, automating follow-ups, and enabling personalized marketing through customer profiling. It integrates inventory management for real-time updates on vehicles and marketing tools for seamless communication and lead nurturing. With insightful analytics, it identifies sales trends, forecasts demand, and supports data-driven decisions, optimizing operations, fostering customer relationships, and boosting revenue growth in the automotive industry.

Objective:

To enhance the sales operations of the automobile industry by implementing Salesforce CRM, enabling improved lead management, customer engagement, and data-driven decision-making.

Project Flow:

Milestone 1 : Salesforce developer account creation

Milestone 2 : Object

Milestone 3 : Tabs

Milestone 4: The Lightning app

Milestone 5 : Fields and Relationships

Milestone 6 : Page Layout

Milestone 7 : Apex Trigger

Milestone 8 : LWC Component

Milestone 9: Apex Schedulers

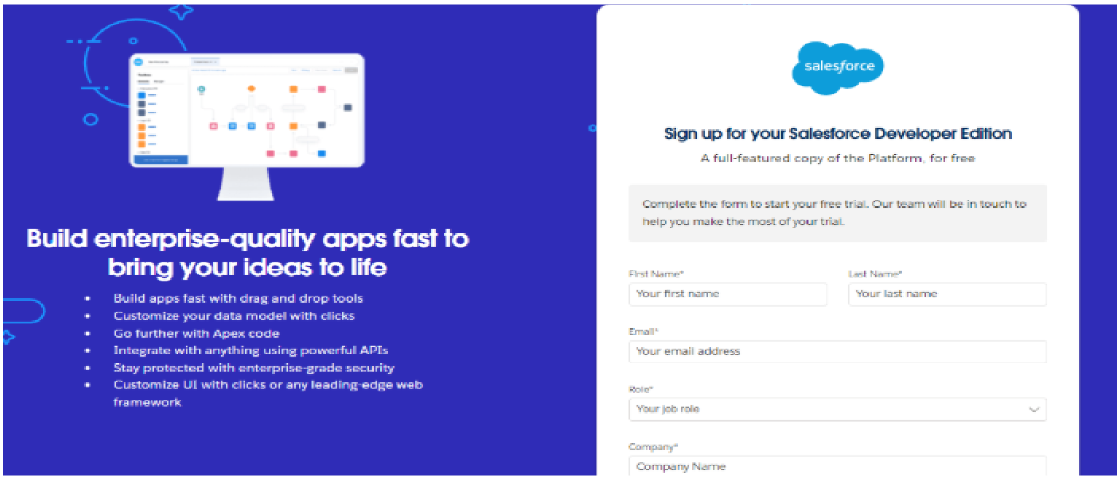
Milestone 10: Reports

Milestone 11: Dashboards

Milestone 1- Salesforce developer account creation:

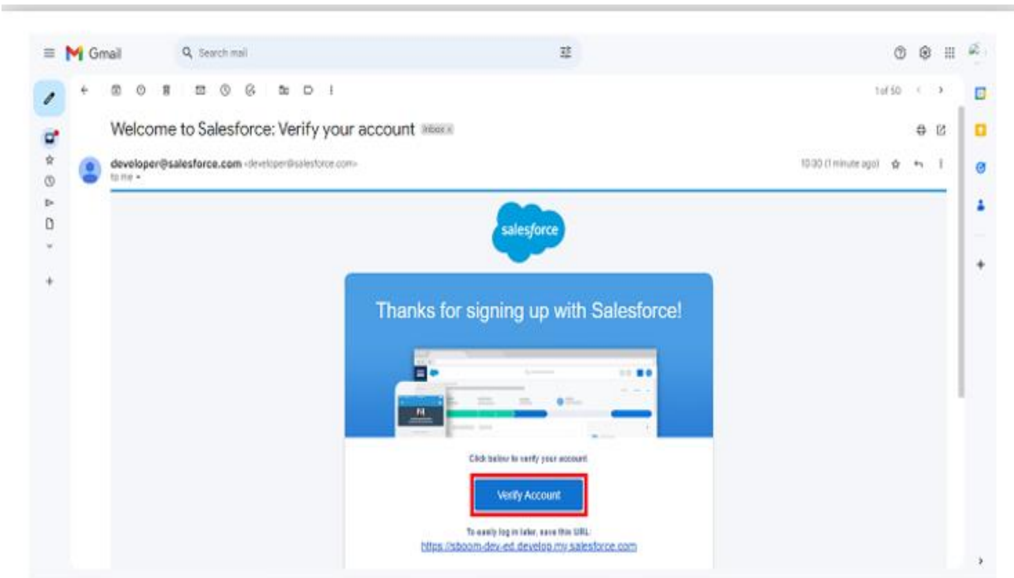
Activity 1: Creating Developers Account:

- Open Salesforce Developer Signup and provide your first name, last name, email, and select Role: Developer.
- Enter details such as your Company (College Name), Country: India, Postal Code, and create a Username in the format username@organization.com (e.g., yourname@collegename.com).
- Click Sign Me Up to finish the signup process.



Activity 2- Account Activation:

1. 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.
2. Click on Verify Account
3. Give a password and answer a security question and click on change password.
4. Then you will redirect to your salesforce setup page.

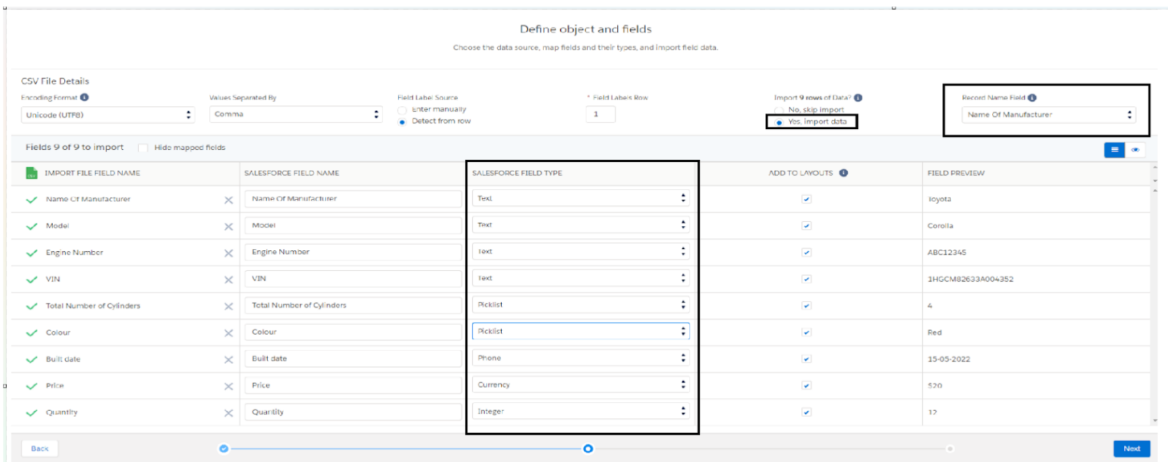


Milestone 2- Create Object :

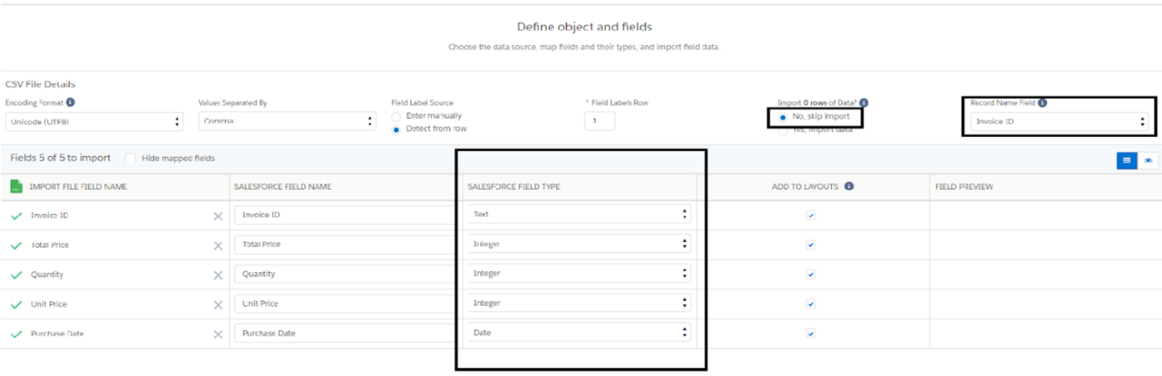
Activity 1: Create Automobile Information Object

1. Download the spreadsheet, save it as AutomobileInformation.csv, and ensure the file name is "Automobile Information" in CSV format.
2. Log into your Salesforce account, click Setup, and navigate to the Object Manager tab.

3. Click Create, select Custom Object from Spreadsheet, and log in using your Salesforce credentials created earlier.
4. Click Upload, choose the AutomobileInformation.csv file, and let Salesforce auto-detect the fields and populate the record data. Verify the field types and select the appropriate Record Name field.
5. Click Next, adjust the settings as required, and then click Finish to create the Automobile Information object and import the data successfully.

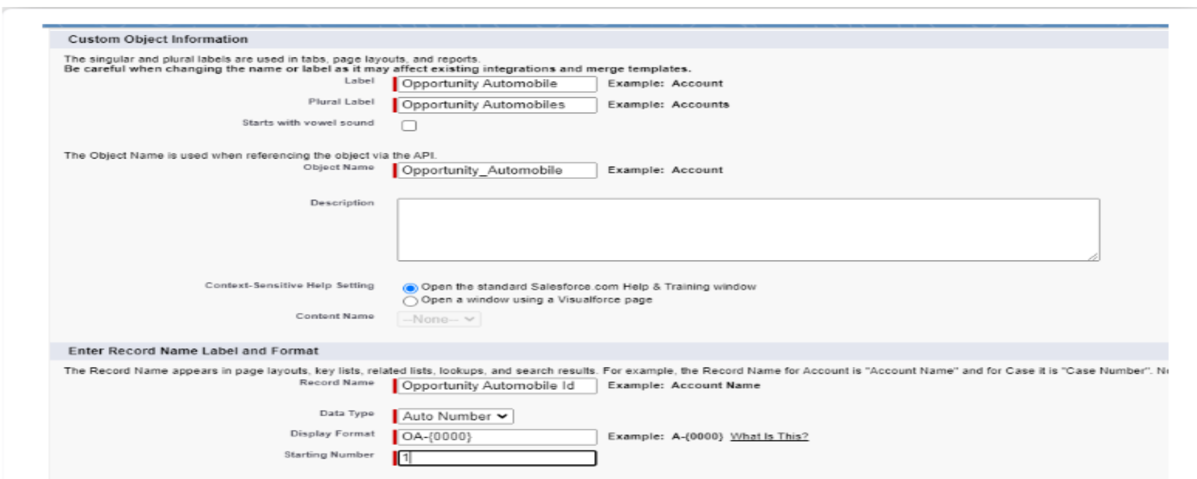


Activity 2: Create Invoice Object



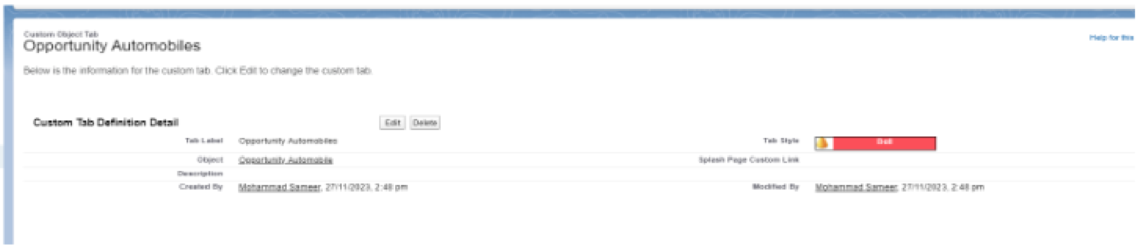
Activity 3-Create Automobile Object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name>> Opportunity Automobile
3. Plural label name>>Opportunity Automobiles
4. Enter Record Name Label and Format
   - Record Name >> Opportunity Automobile Id
   - Data Type >> Auto Number
   - Display Format >> OA-{0000}
   - Starting Number >> 1
5. Click on Allow reports.
6. Allow search
7. Save.



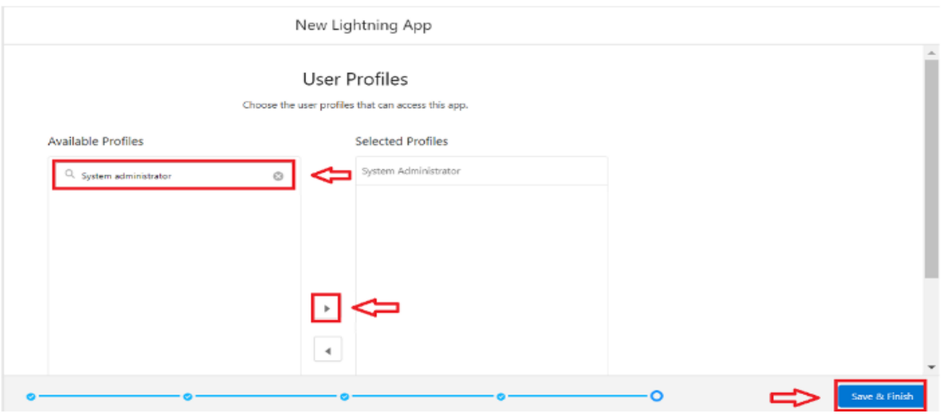
Milestone 3- Creating customer tab:

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



Milestone 4- Create a Lightning App

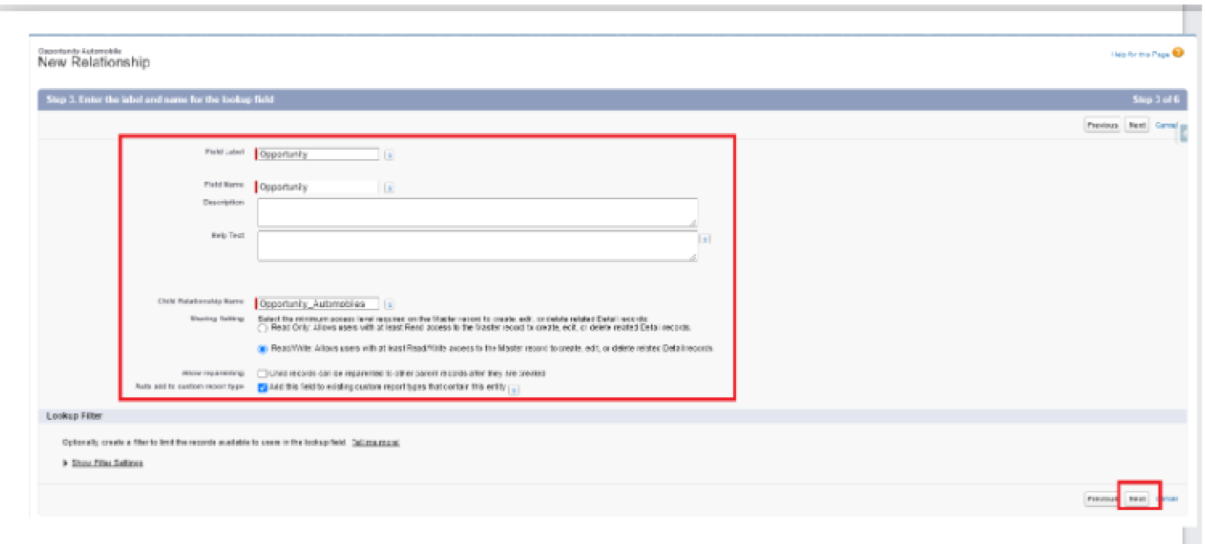
1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
2. Fill the app name in app details and branding as follow
   - App Name :Sales Automobile Using Salesforce CRM
   - Developer Name : this will auto populated
   - Description : Give a meaningful description
   - Image : optional (if you want to give any image you can otherwise not mandatory)
   - Primary color hex value : keep this default
3. Then click Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.
4. Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Milestone 5- Fields and Relationships:

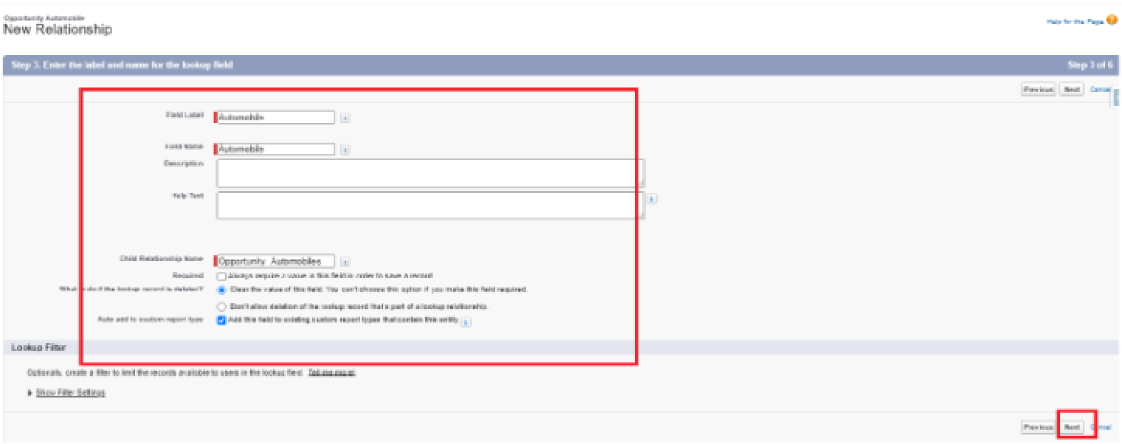
Activity 1: Creating Opportunity Master Detail Relationship Field in Opportunity AutoMobile Object

1. Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar>> click on the object
2. Now click on “Fields & Relationships” >> New
3. Click on Next
4. Fill the above as following:
   - Field Label: gets auto Generated(Opportunity)
   - Field Name : gets auto generated(Opportunity)
   - Click on Next >> Next >> Save and new.



Activity 2 : Creating the AutoMobile Information Lookup Field in Opportunity Automobile Object

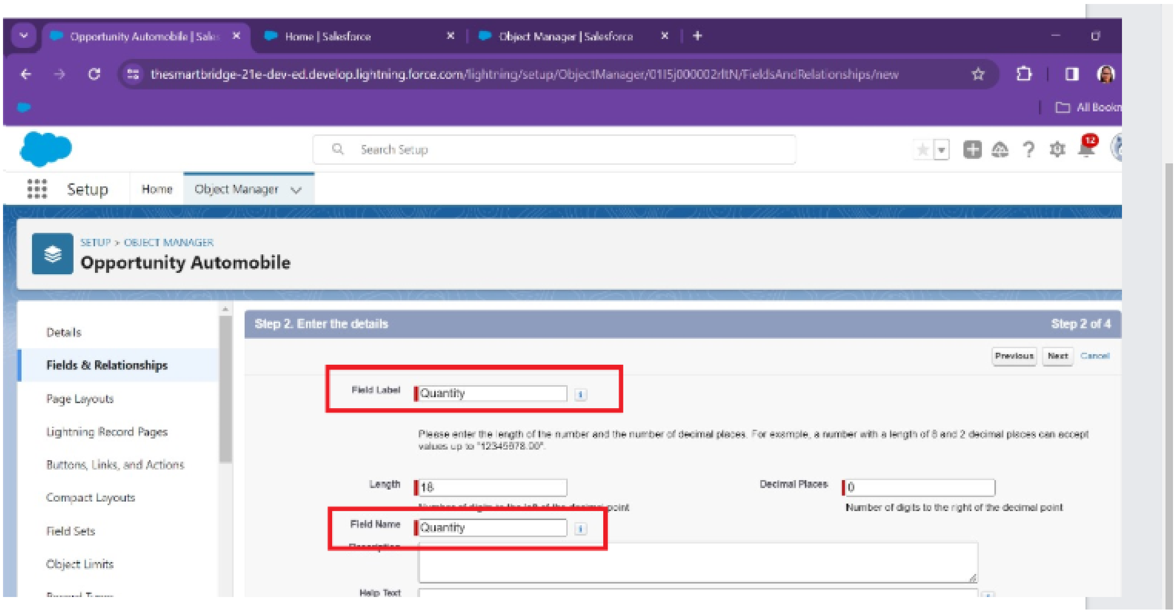
1. Fill the above as following:
   - Field Label: Automobile
   - Field Name : Automobile
2. Click on Next >> Next>> Save and new.



Activity 5- Creating Quantity Number Field in Opportunity Automobile Object:

1. Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar >> click on the object.
2. Now click on "Fields & Relationships" >> New.
3. Select Data type as "Number" and click Next.

- Field Label >> Quantity
- Field Name >> Quantity



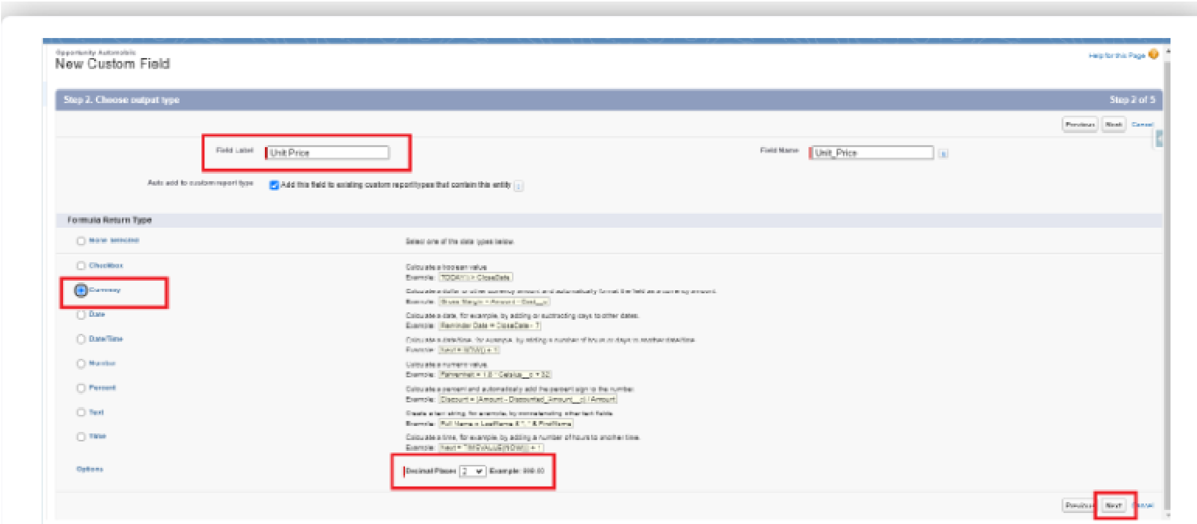
Activity 6- Creating Formula Field in Opportunity Automobile Object

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.

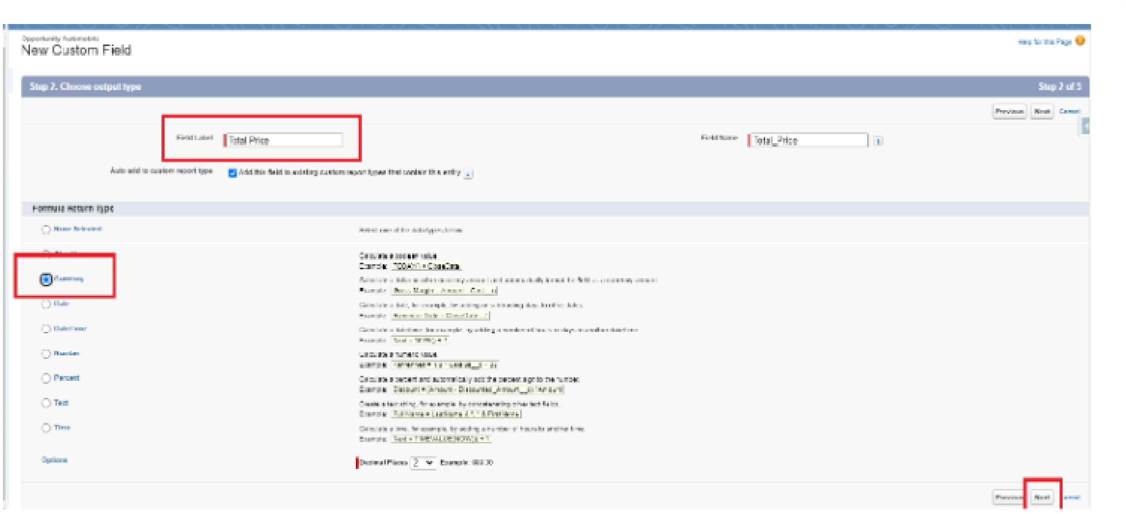


Activity 7- Creating the Formula field in Opportunity Automobile Object

To create fields in an object:

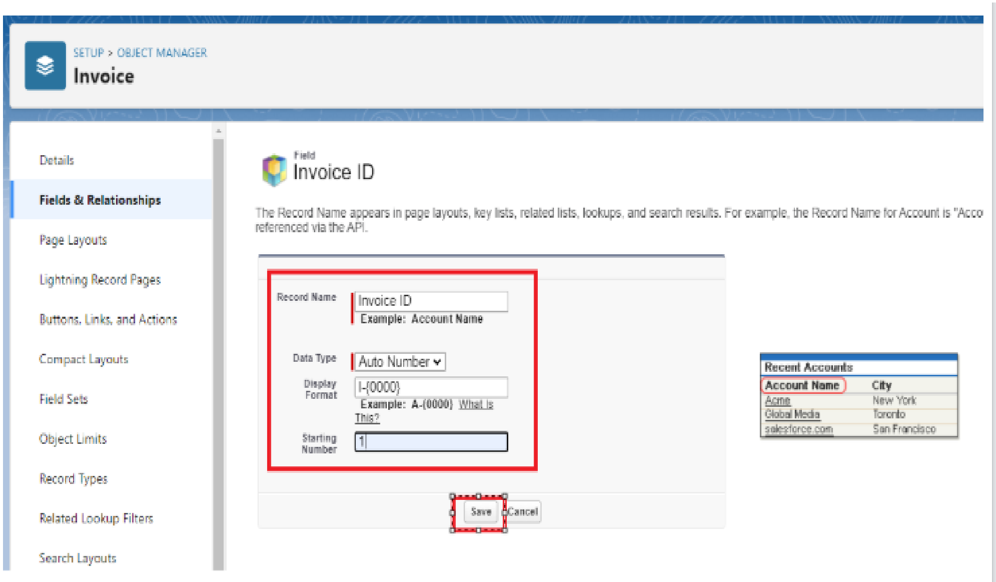
1. Go to setup >> click on Object Manager >> type object name(Opportunity Automobile) in quick find bar >> click on the object.

- 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.



Activity 8- Updating field in Invoice Object

1. Go to setup ? click on Object Manager ? type object name(Invoice) in quick find bar? click on the object.
2. Now click on “Fields & Relationships” , Click on the edit of Invoice Id field.
3. Select Data type as “Auto Number ” and click Next.



Milestone 6-Page Layout

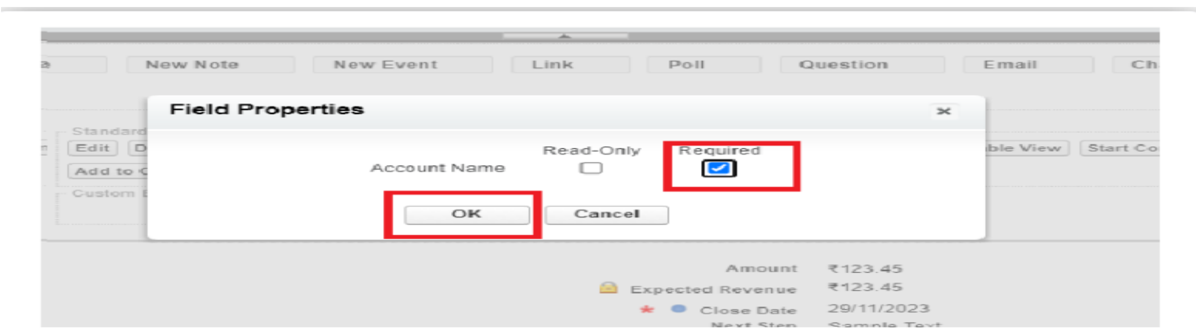
Activity 1- Edit the Page layout for Opportunity Object

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’.

Step 3: check the Required box for Account name and click on Ok.

Step 4: Click on Save.



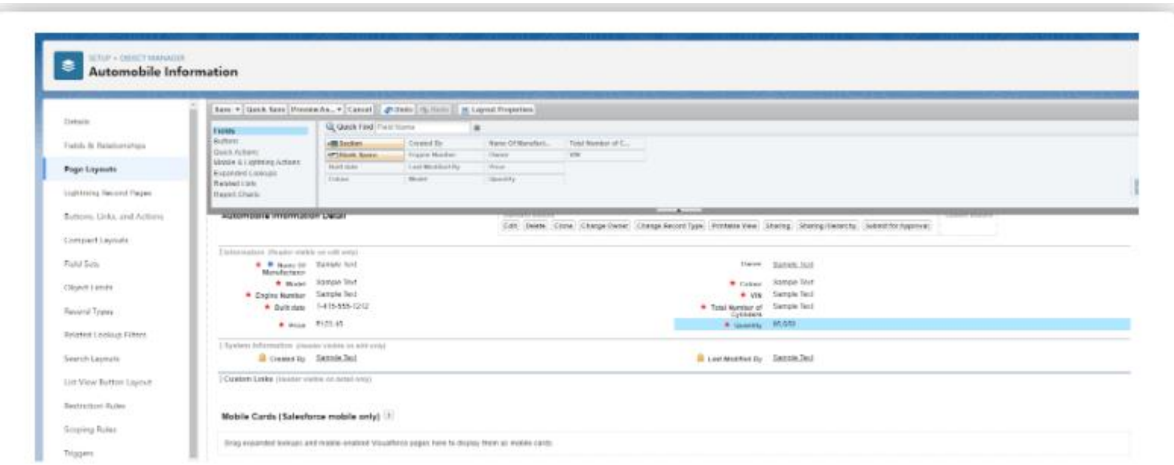
Activity 2- Edit the Page layout for Automobiles Information

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’.

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

Step 5 : Click on Save.



Milestone 7-Apex Trigger

Activity 1-Opportunity Automobile quantity:

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 for Opportunity Object.

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

Opportunity-Automobile Error

```
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){
            OppAutomobile.addError('the Number of Automobile u want are not Available !!
the Automobile are Available Count is ' +
lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c );
         }
      }
```

```
    }

}
```

Trigger handler for the Opportunity_Automobile__c Object

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

Invoice Creation Trigger

```
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:

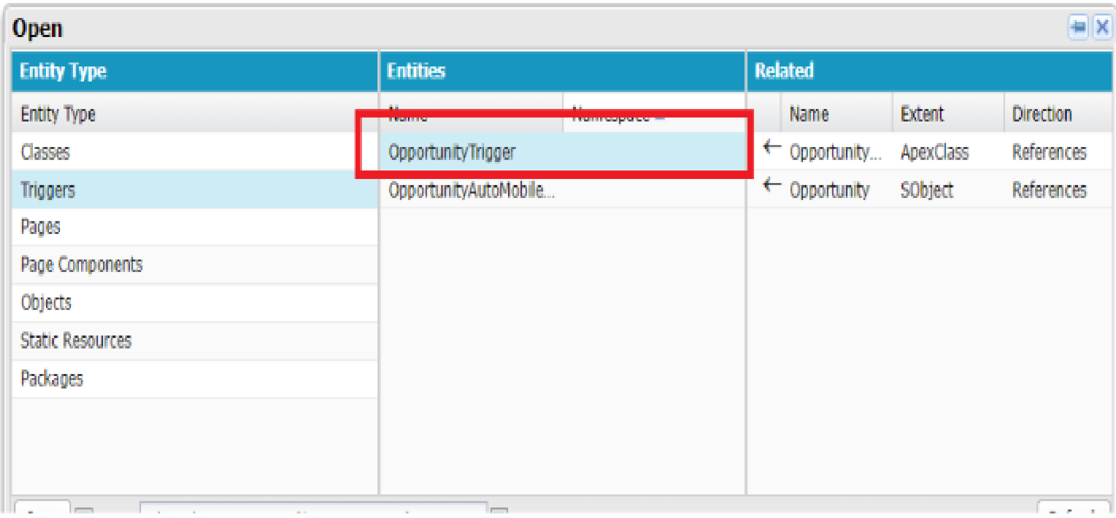
```
trigger OpportunityTrigger on Opportunity (before update, After Update) {
    if(trigger.isbefore && trigger.isUpdate){
          OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new,
trigger.oldMap);
    }
    IF(trigger.isafter && trigger.isupdate){
        InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new,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

check contact role

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.');
}
}
}
}
}
```



Milestone 8- LWC Components

Activity 1-Create Apex Class to Get Invoices

```
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];

   }


}
```

Activity 2- 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

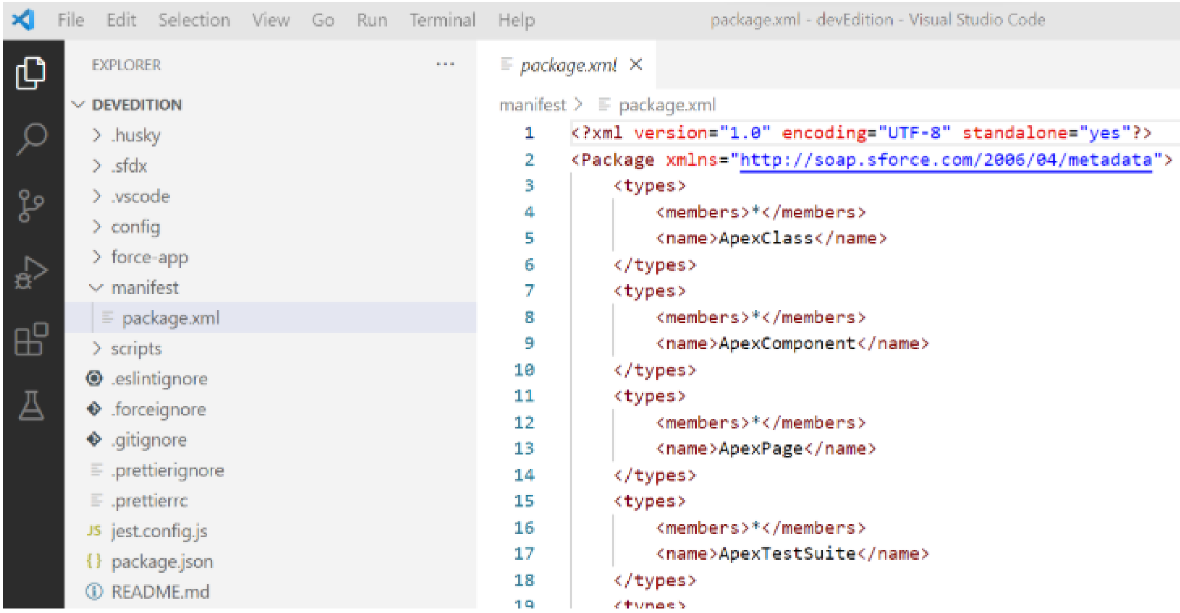
To confirm that the Salesforce CLI is installed and working correctly, you can open a command prompt and type sfdx. This will display the version number of the Salesforce CLI that is currently installed on your system.

```
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 Salesforce CLI
  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
```

Activity 3- Create a project in VS Code

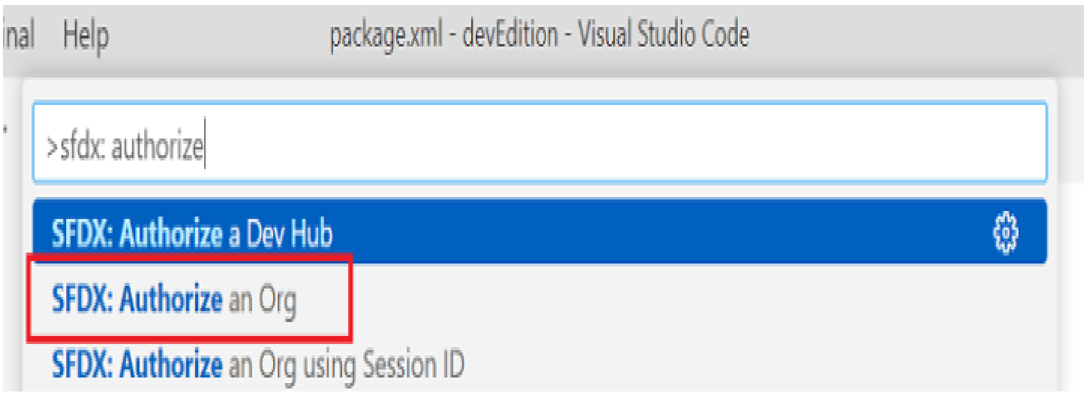
1. Press CTRL + SHIFT + P, type sfdx: create

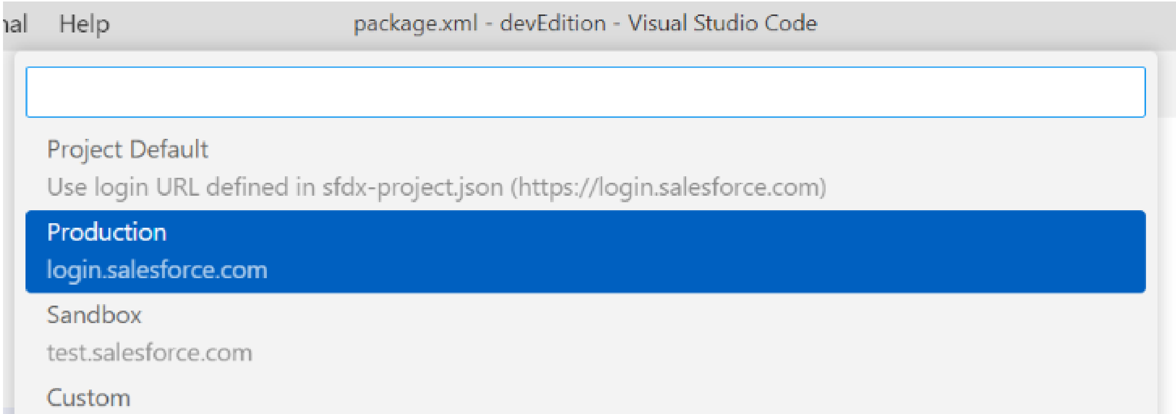
2. select SFDX: Create Project with Manifest
3. Select the Standard project template
4. The new project is created with package.xml



Activity 5- Authorize an org

1. Press CTRL + SHIFT + P, type sfdx: authorize.
2. select SFDX: Authorize an Org from the list
3. Choose your Salesforce instance.
4. The Salesforce login page opens in the browser.
5. Enter the credentials and click Log In



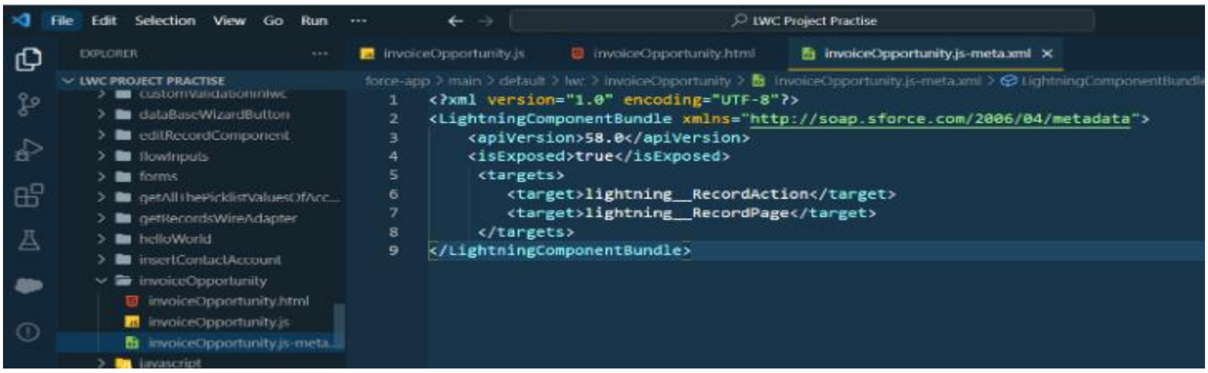


Activity 6- Create Lightning Web Component

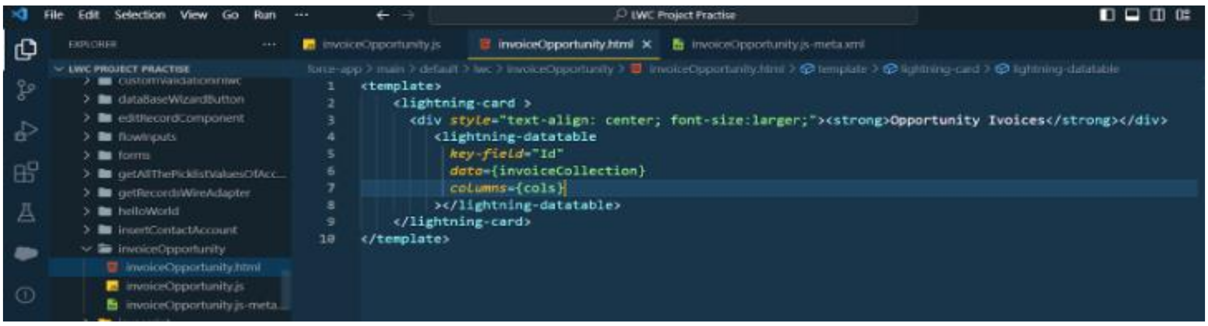
XML File :

1. In the VS Code, press CTRL + SHIFT + P, type sfdx: create lightning in the search bar, and select SFDX: Create Lightning Web Component
2. Give the name “InvoiceOpportunity” and press Enter.
3. Choose the directory.
4. LWC is created successfully.

JS File :



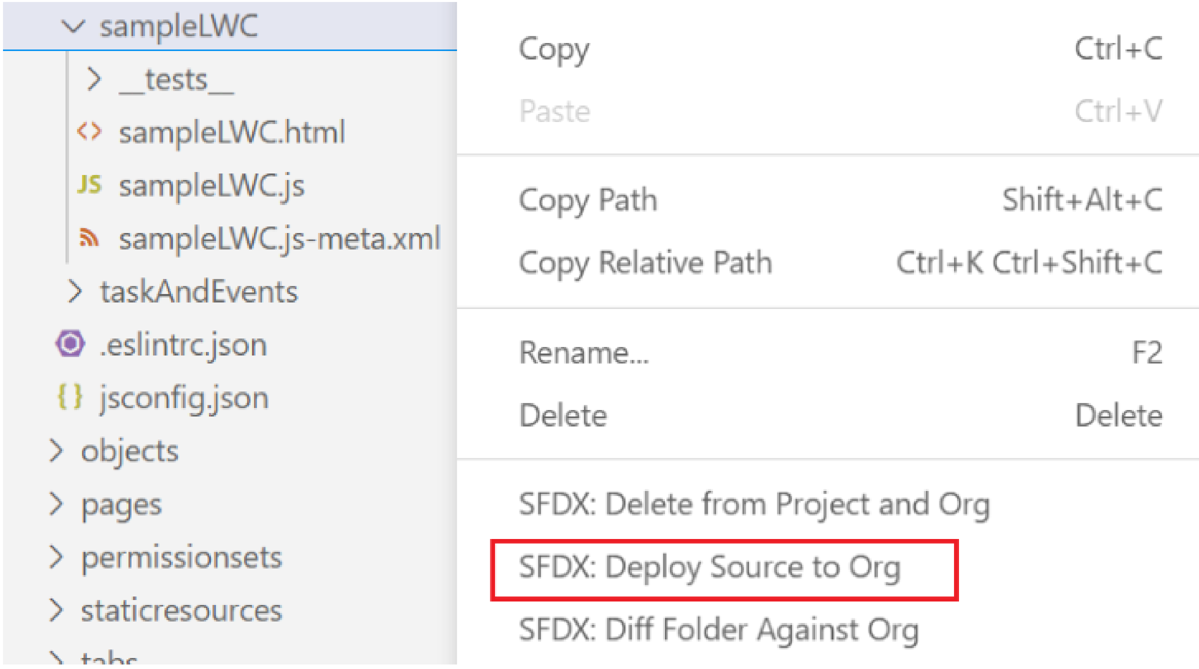
HTML File :

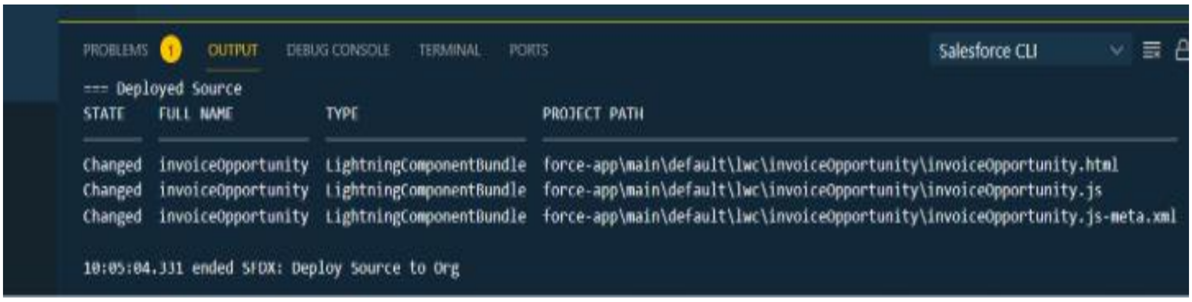


Deploy Component:

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

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





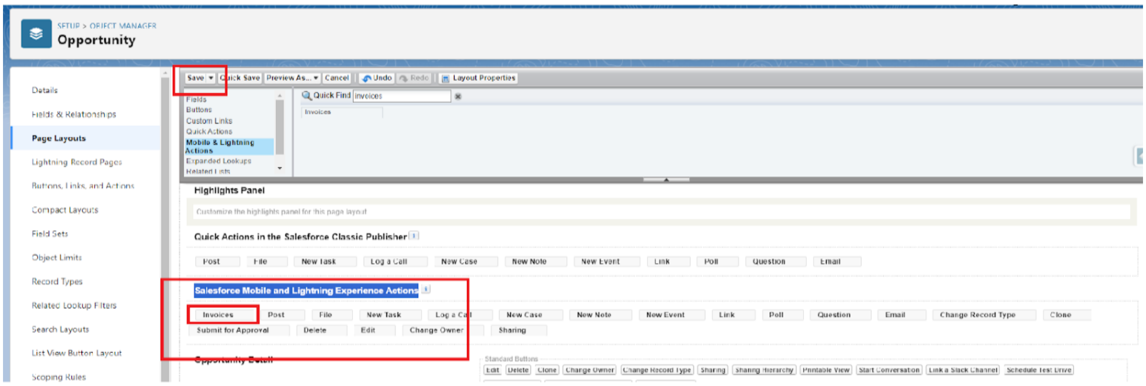
Activity 7- Create Button to Add on Opportunity

1. To add the newly created component to the view, Go to Salesforce Setup
2. Click on Object Manager
3. Search Opportunity and Click on it .
4. click on Button Links and Action.
5. click on the New Action.
6. Select the InvoiceOpportunity component
   - Label :- Invoices
   - Name :- Invoices
7. Click on Save and your action Button is Ready.

Activity 8- Add InvoiceOpportunity into Opportunity Record Page

1. On Opportunity Object Manager Click on Page layout.
2. Click on OpportunityLayout.
3. Search for invoice on Quick Find.

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



Milestone 9-Appex Schedulers

Activity 1- Delete opportunity Schedule Class

Objective :

Through this schedulable class, we can see all the Closed Lost Opportunities.

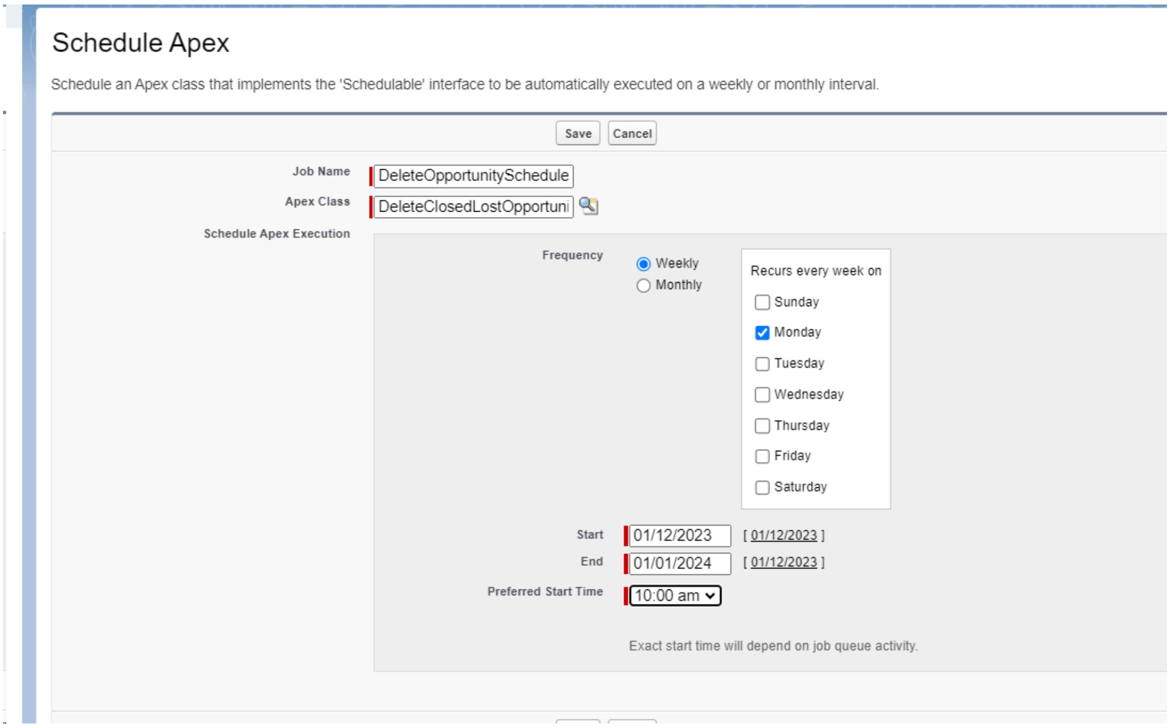
We can delete all the Closed lost Opportunities by this Scheduled method on every monday as weekly.

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

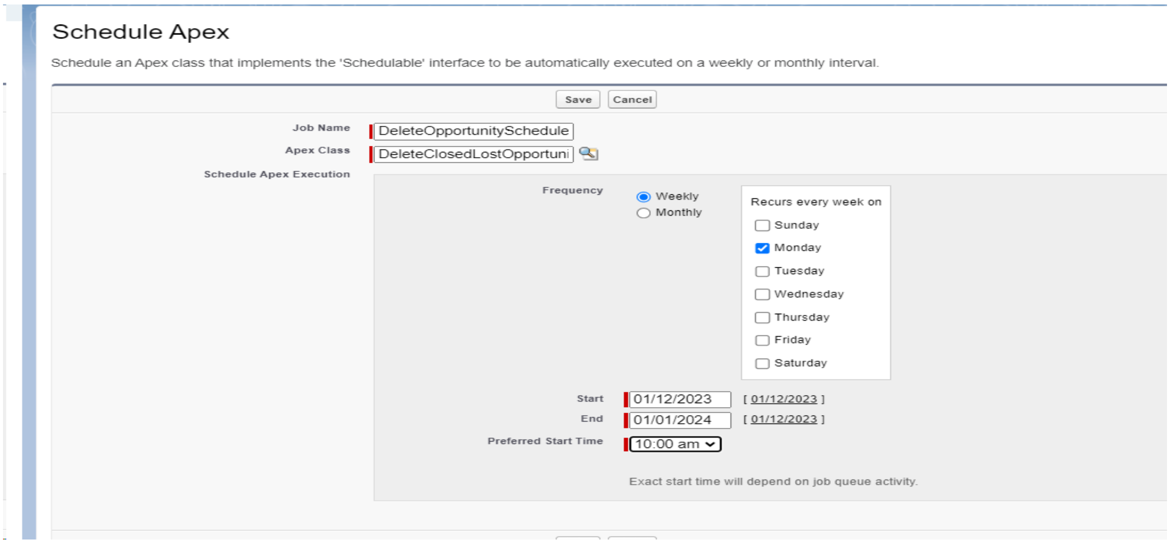
Code

```
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;

      }

   }

}
```

1. Click on Schedule Apex and enter the Job name.
2. Job Name : DeleteOpportunitySchedule



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



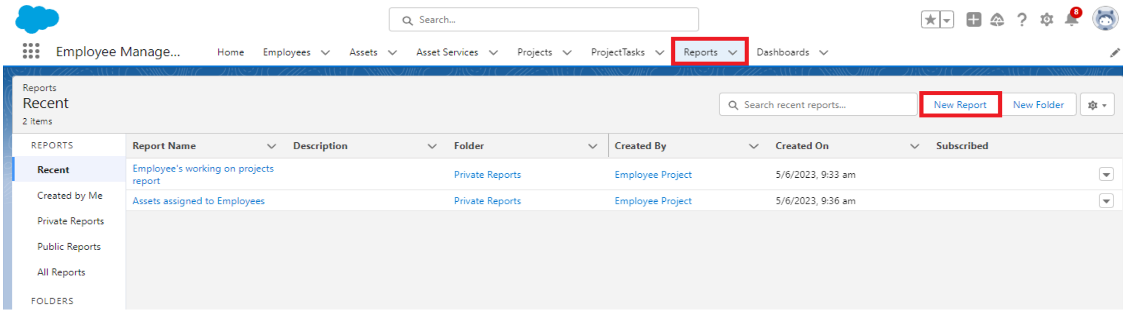
Click on Save. Now enter Apex in the search box and select Apex jobs.

Milestone 10-Reports

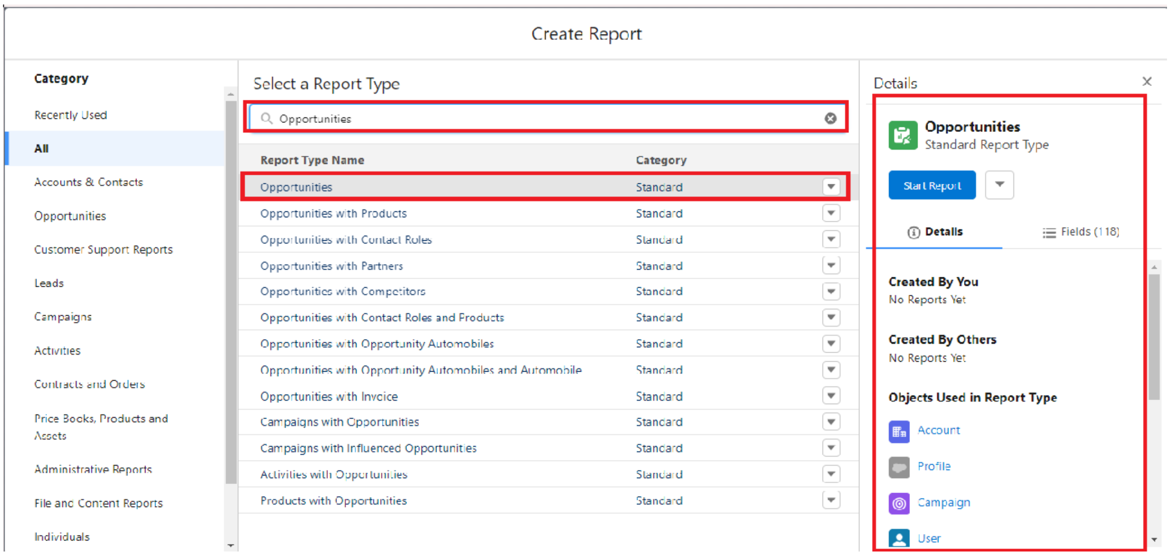
Activity 1-Create Report on Opportunity

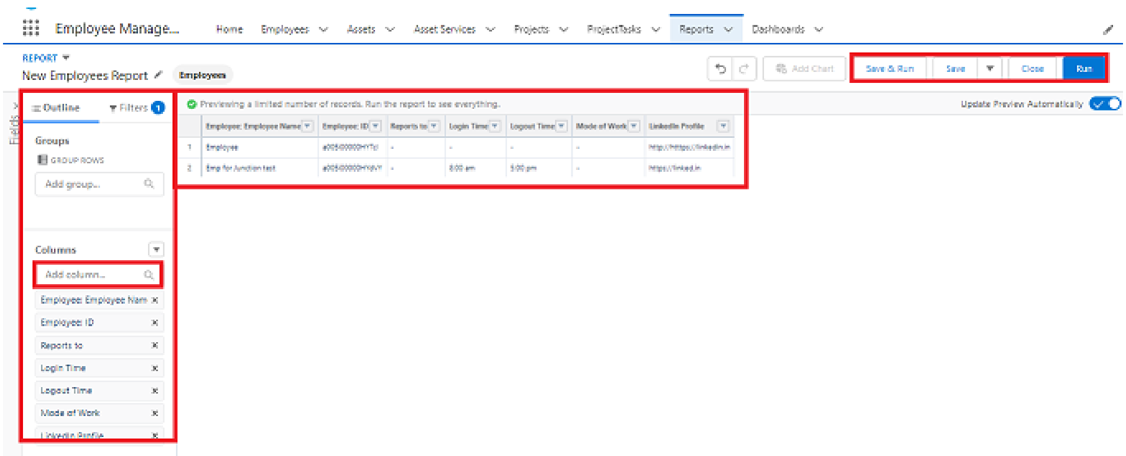
Go to the app >> click on the reports tab

Click New Report.



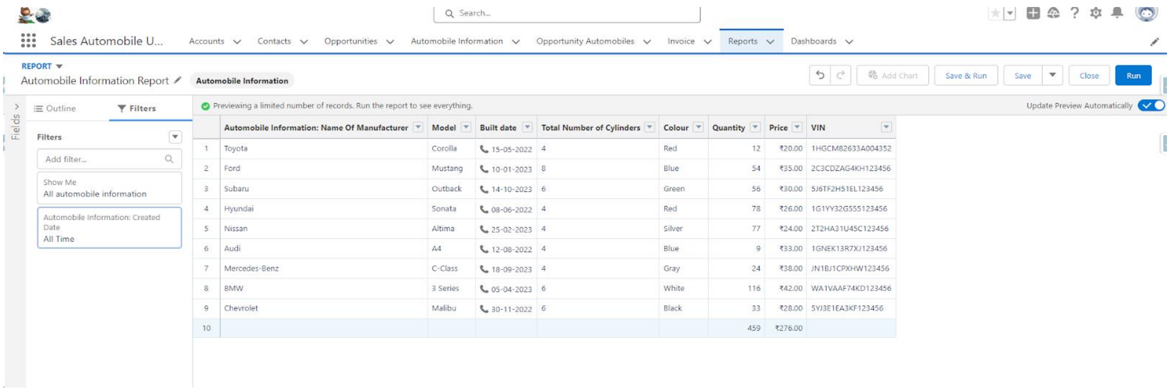
Select report type from category or from report type panel or from search panel >> click on start report.





Activity 2- Create Report on Automobile Information

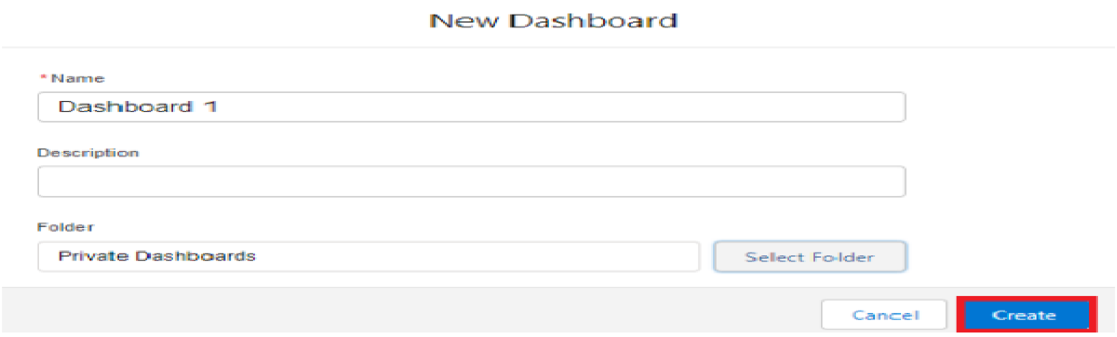
Create a report with a report type : “Automobile Information”.



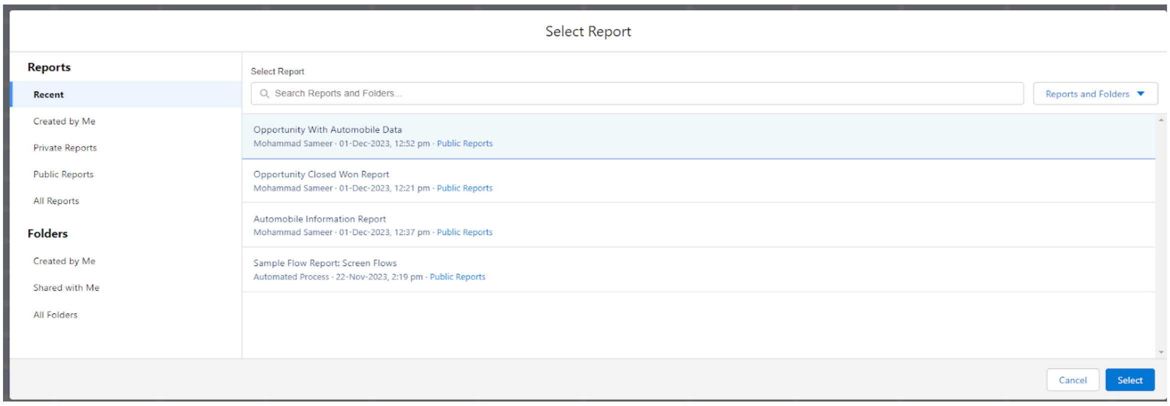
Milestone 11-Dashboard

Sales Dashboard

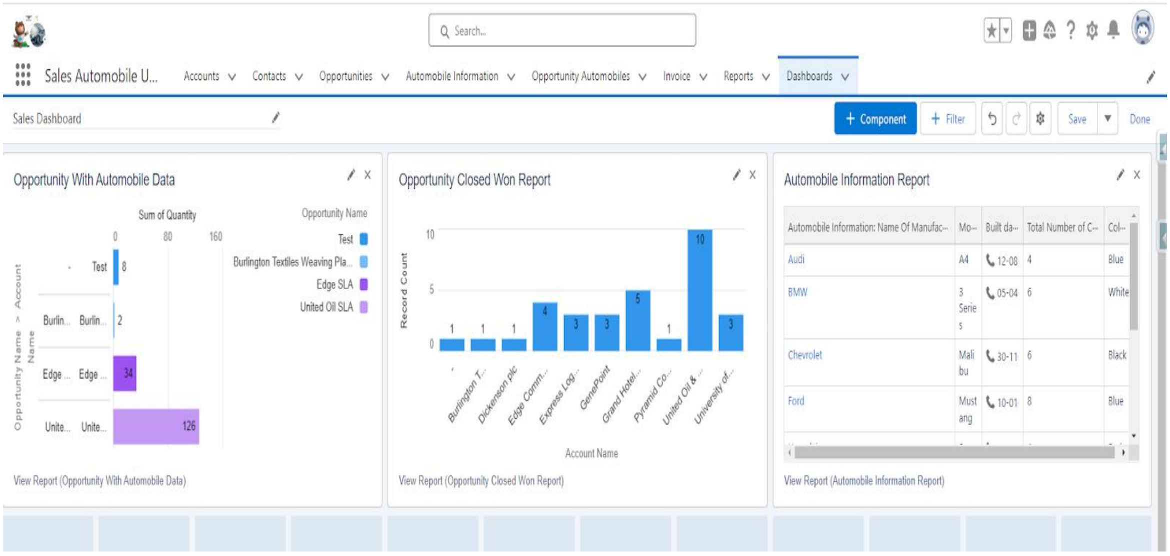
1. Create Dashboard
2. Go to the app ? click on the Dashboards tabs.
3. Give a Name and click on Create.



4. Select a Report and click on select.



The Created Dashboard will look like this.



Conclusion:

Implementing Salesforce CRM in the automobile industry revolutionizes sales operations by providing a robust platform for efficient lead management, personalized customer engagement, and seamless integration with inventory and marketing tools. Its insightful analytics empower data-driven decision-making, enabling businesses to forecast demand accurately, identify sales trends, and optimize strategies. By streamlining processes and fostering stronger customer relationships, Salesforce CRM not only boosts operational efficiency but also drives significant revenue growth, positioning the automobile industry for sustained success in a competitive market.