PR0JECT **REPORT ON**

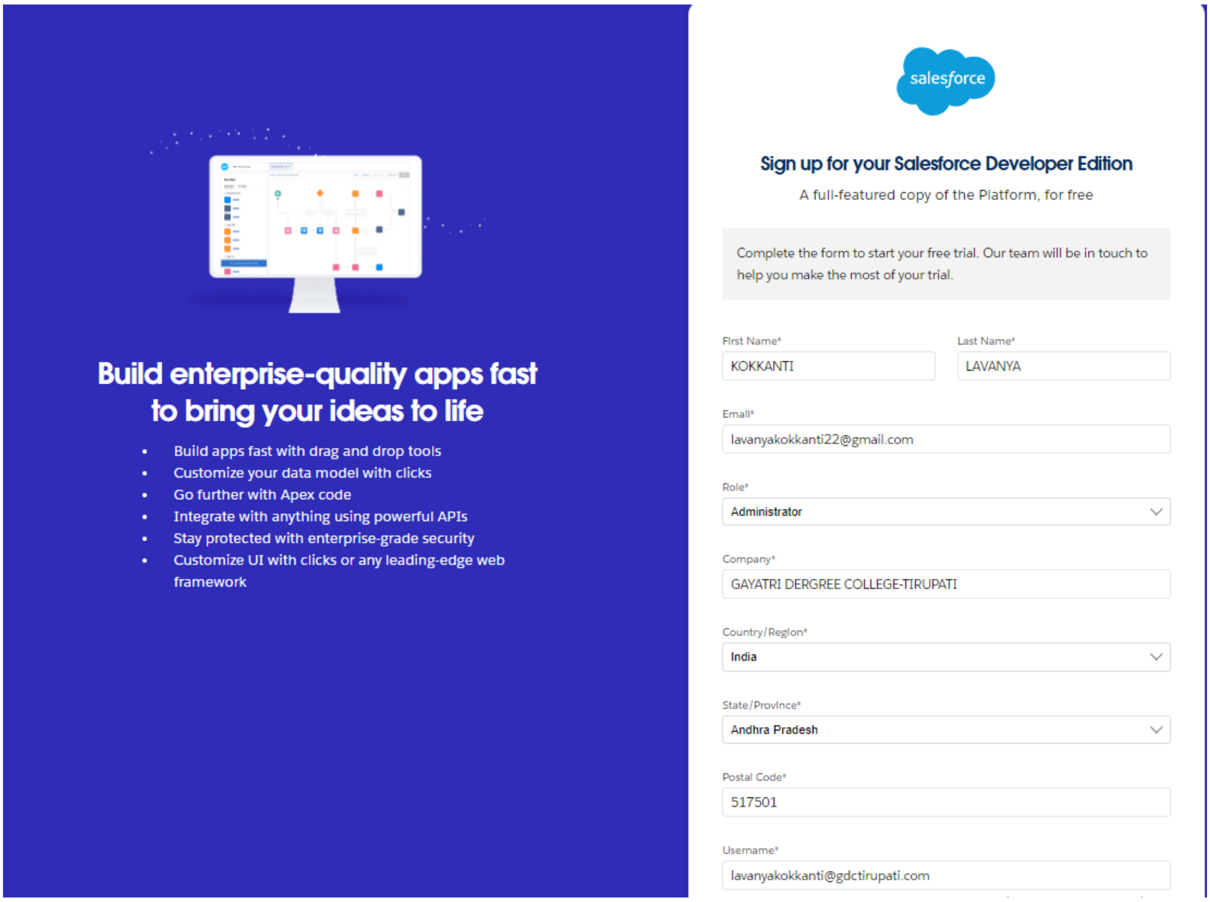
SALES AUTOMOBILE USING SALESFORCE CRM

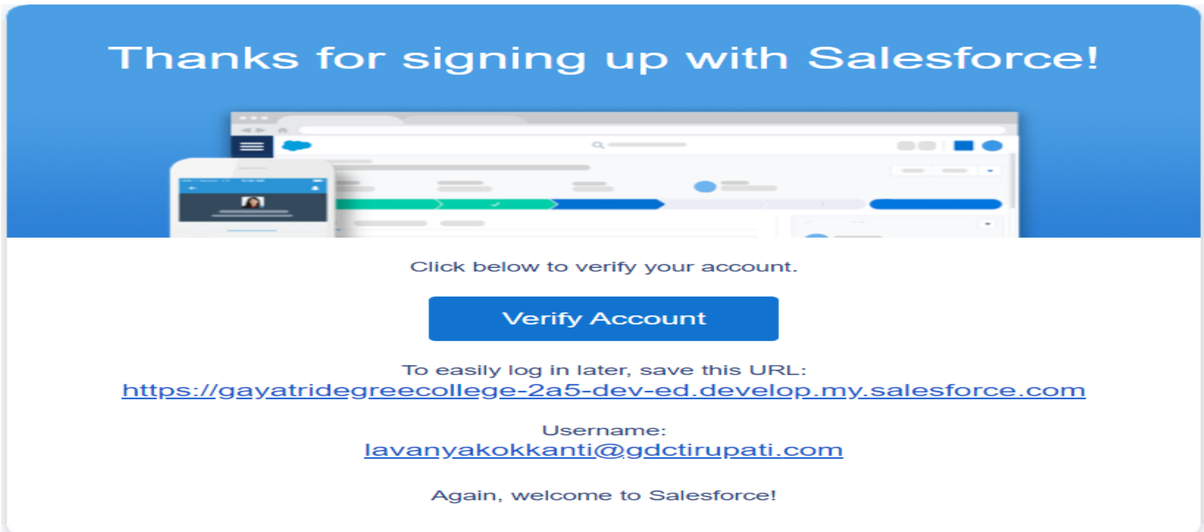
Introduction:

The Salesforce CRM implementation for automobile sales streamlines the entire sales process, enhancing efficiency and customer satisfaction. Through this system, sales teams can manage leads, track customer interactions, and automate follow-ups. It enables comprehensive customer profiling, allowing for personalized marketing strategies and targeted campaigns. The platform facilitates inventory management, ensuring real-time updates on available vehicles and their specifications. Integration with marketing tools enables seamless communication and lead nurturing. Additionally, the system provides insightful analytics, empowering decision-making by identifying sales trends and forecasting demand. Overall, the Salesforce CRM for automobile sales optimizes operations, fosters customer relationships, and drives revenue growth within the automotive industry.

Milestone 01 - CREATING DEVELOPER ORG

- Go to developer.salesforce.com
- Click on sign up.
- On the signup form, enter the following details
- First name & Last name- LAVANYA KOKKANTI
- Email- lavanyakokkanti22@gmail.com
- Role- SALESFORCE ADMINISTRATOR
- Company- GAYATRI DEGREE COLLEGE- TIRUPATI
- County- INDIA
- Postal Code- 517501
- Username- lavanyakokkanti@gdctirupati.com





Milestone 02 - CREATION OF OBJECT

➢ **Create Automobile Information Object:**

1. Download and open this spreadsheet, save it as AutomobileInformation.csv.
2. 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.

3. Click the Object Manager tab.
4. Click Create.
5. Select Custom Object from Spreadsheet.
6. Click Login WithSalesforce.
7. Enter your Salesforce account username and password. (which you have created in the Milestone 1, Activity 1)
8. Click Log In.
9. Click Allow.
10. Click Upload.
11. 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.
12. Click Next and enter the following settings.
13. Click Finish. The Automobile Information object is successfully created and data imported, all within minutes.

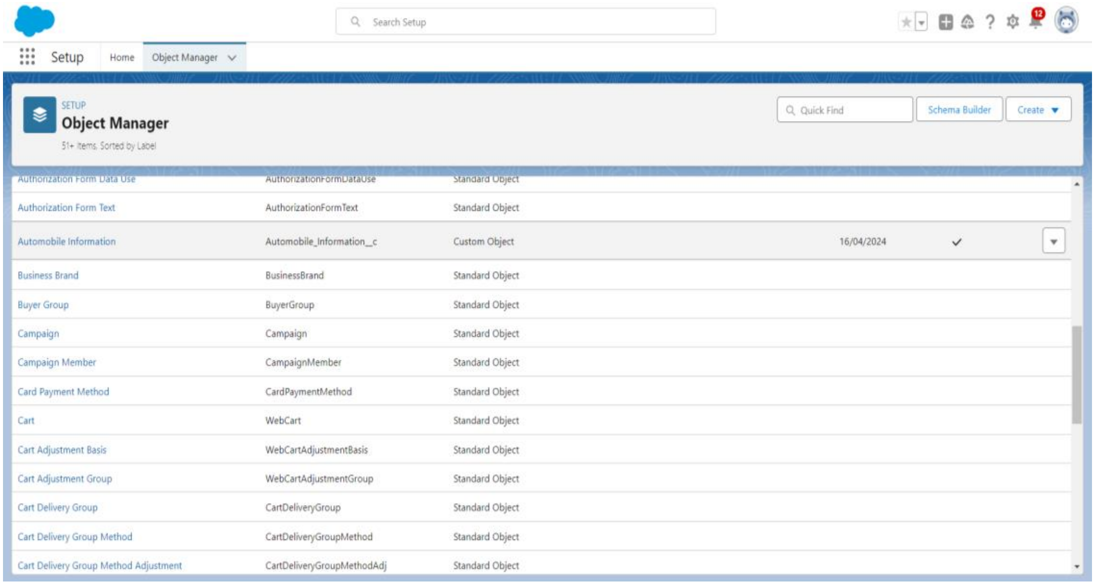
➢ **Create Invoice Object:**

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.

➢ **Create Automobile Object:**

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

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

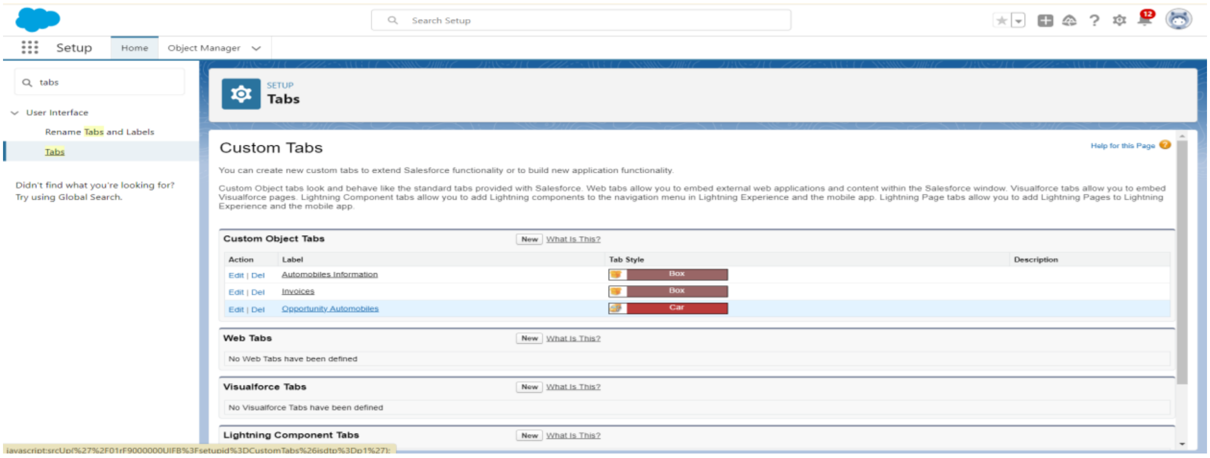


Milestone 03 – TABS

➢ Creating A Custom 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 04 - THE LIGHTNING APP

➢ Create A Lightning App:

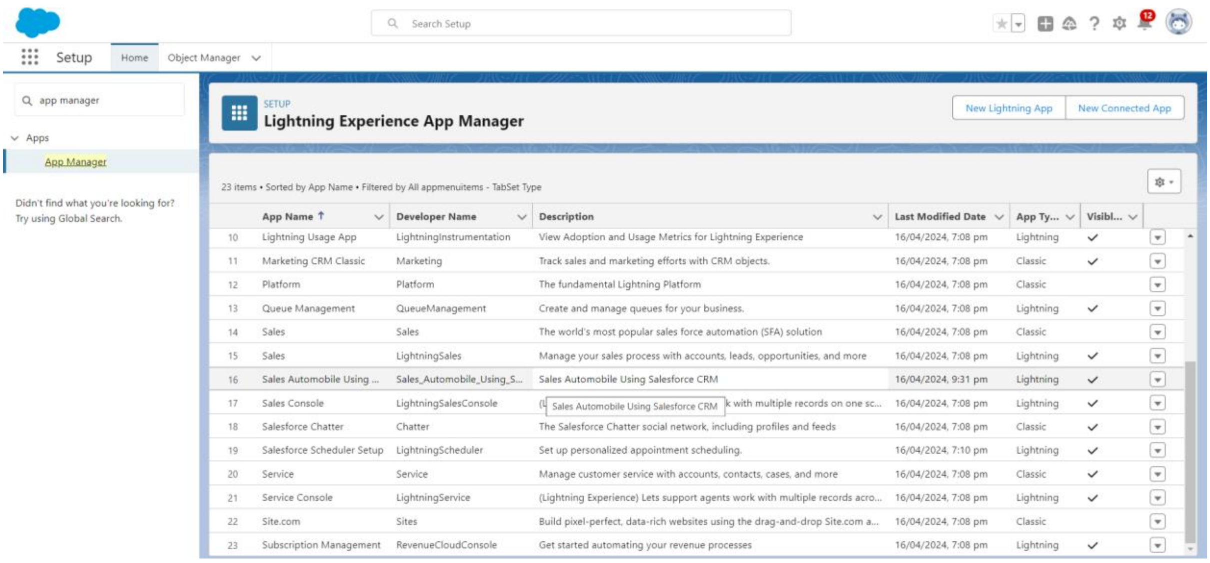
1.Go to setup page >> search “app manager” in quick find >> select “app manager” >>click on new lightning App.

2. Fill the app name in app details and branding as follow

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 colour 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. Add Navigation Items.
5. 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.6. Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.



Milestone 05- FIELDS & RELATIONSHIPS

- **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.Select Data type as “Master Details Relationship
4.Click on Next
5.Fill the above as following:
- Field Label: gets auto Generated (Opportunity)
- Field Name: gets auto generated (Opportunity)

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

- **Creating The AutoMobile Information Lookup 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
2. Now click on “Fields & Relationships” >> New
3. Select Data type as “Lookup RelationShip”.

4.Click on Next

5.Fill the above as following:
   a. Field Label: Automobile
   b. Field Name: Automobile

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

- **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.
4. Field Label >>Quantity
5. Field Name>> Quantity
6. Check that Required Check box.
7. Click Next >>Next>>Save & New.

- **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.

2. Now click on “Fields & Relationships” >> New.

3. Select Data type as “Formula” and click Next.

4. 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.

5.Under Advanced Formula write down the formula: (Automobile__r.Price__c)

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

- **Creating The Formula 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 “Formula” and click Next.

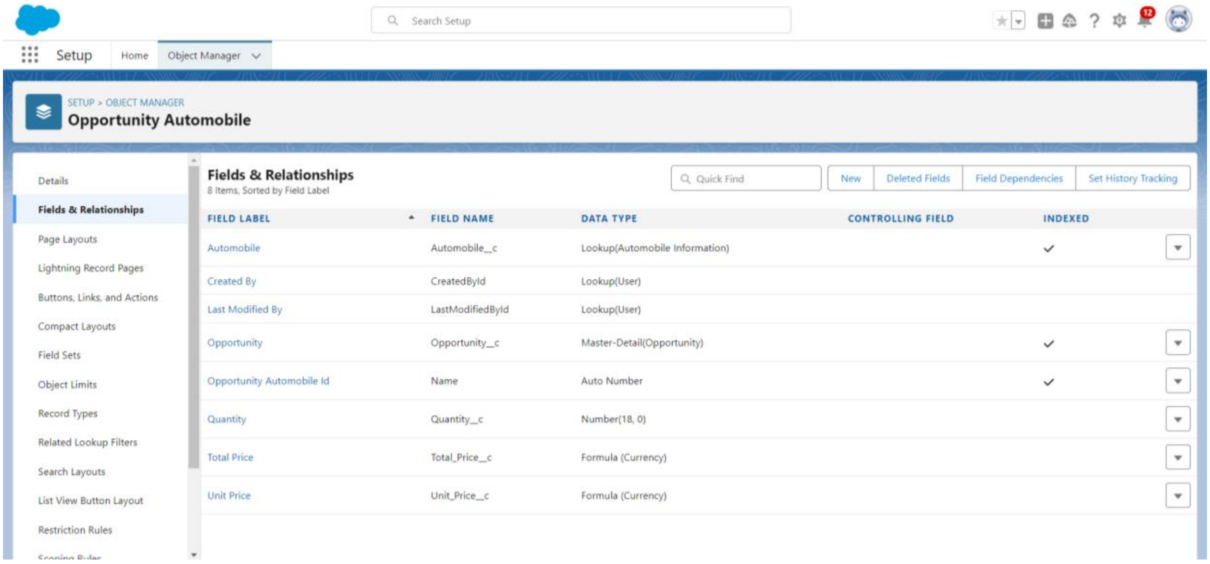
4. Give Field Label and Field Name as “Total Price” and select formula return type as “Currency” and change the decimal values to two and click next.
5. Under Advanced Formula write down the formula: (Unit_Price__c * Quantity__c)
6. click “Check Syntax” and Next >> Next >> Save.

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.
    a) Display Format: I-{0000}
    b) Starting Number: 1
4. Click Save.

- **Creating Remaining Fields in Objects:**
  Now create the remaining fields using the data types mentioned.

<table>
<tr><th>s.no</th><th>Object name</th><th>Fields</th></tr>
<tr><td>1</td><td>Invoice</td><td><table><tr><td>Field Name</td><td>Opportunity</td></tr><tr><td>Data type</td><td>Master Detail relationship
Object : Opportunity</td></tr></table></td></tr>
</table>



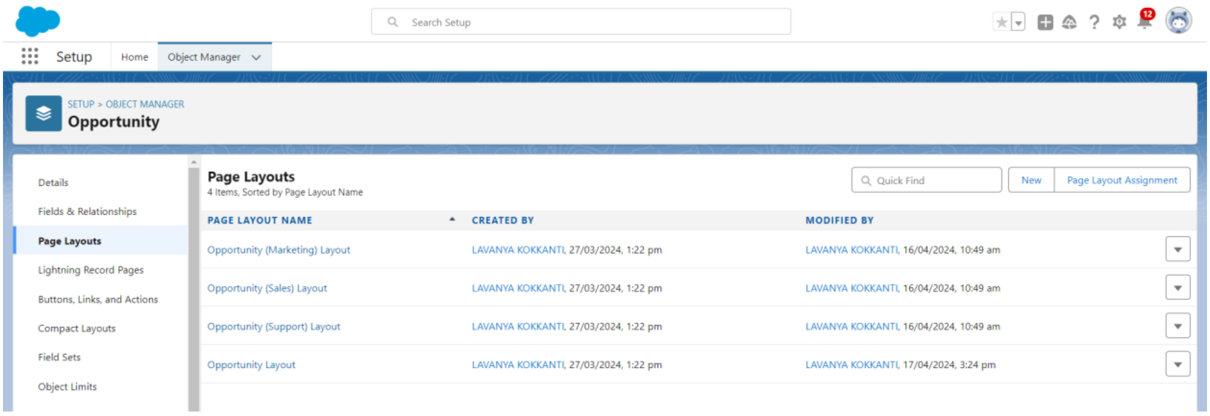
Milestone 06- PAGE LAYOUTS

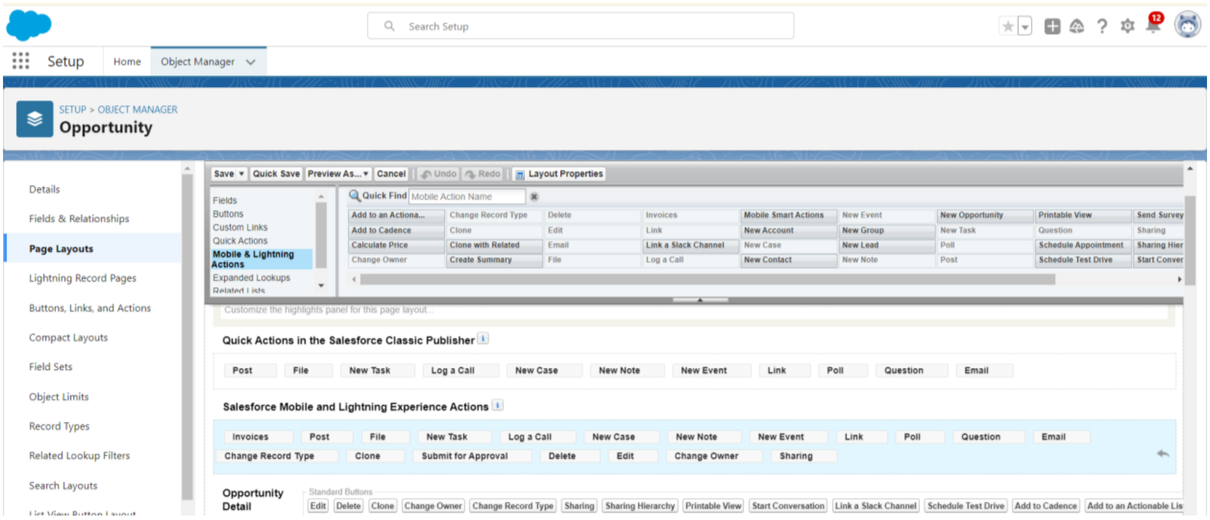
- **Edit The Page Layout for Opportunity Object:**

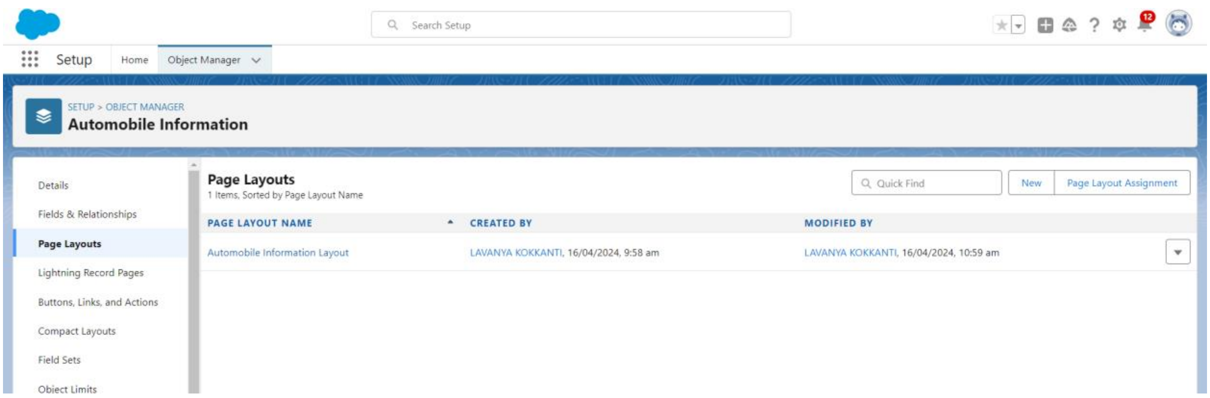
Step 1: Go to Setup >> Click on Object Manager >> On the search bar, select Opportunity Layout. You can notice Page Layouts on the left panel
Step 2: Click on Page Layouts, Click on 'Opportunity Layouts'.
Step 3: In the Opportunity Detail Section, you can see various fields. Go on Account and Click on that Properties icon of Account name Field.
Step 4: check the Required box for Account name and click on Ok.
Step 5: Click on Save.

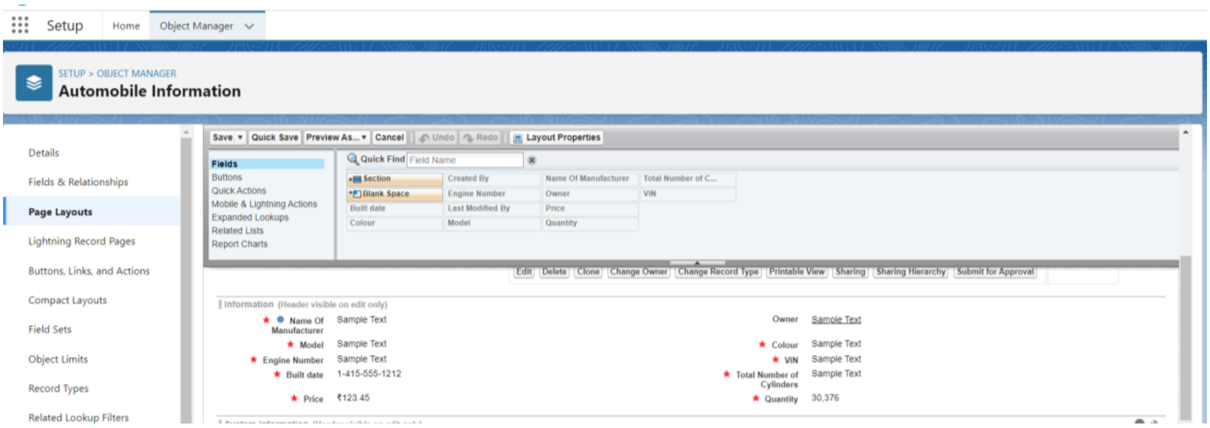
- **Edit The Page Layout for Automobiles Information:**

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'.
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.
Step 4: Adjust the Fields as given below for A good looking view.
Step 5: Click on Save.


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Opportunity
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Page Layouts
4 Items, Sorted by Page Layout Name
Quick Find
New
Page Layout Assignment
PAGE LAYOUT NAME
CREATED BY
MODIFIED BY
Opportunity (Marketing) Layout
LAVANYA KOKKANTI, 27/03/2024, 1:22 pm
LAVANYA KOKKANTI, 16/04/2024, 10:49 am
Opportunity (Sales) Layout
LAVANYA KOKKANTI, 27/03/2024, 1:22 pm
LAVANYA KOKKANTI, 16/04/2024, 10:49 am
Opportunity (Support) Layout
LAVANYA KOKKANTI, 27/03/2024, 1:22 pm
LAVANYA KOKKANTI, 16/04/2024, 10:49 am
Opportunity Layout
LAVANYA KOKKANTI, 27/03/2024, 1:22 pm
LAVANYA KOKKANTI, 17/04/2024, 3:24 pm


Search Setup
Setup
Home
Object Manager
SETUP > 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
Save
Quick Save
Preview As...
Cancel
Undo
Redo
Layout Properties
Fields
Buttons
Custom Links
Quick Actions
Mobile & Lightning Actions
Expanded Lookups
Quick Find
Mobile Action Name
Quick Actions in the Salesforce Classic Publisher
Post
File
New Task
Log a Call
New Case
New Note
New Event
Link
Poll
Question
Email
Salesforce Mobile and Lightning Experience Actions
Invoices
Post
File
New Task
Log a Call
New Case
New Note
New Event
Link
Poll
Question
Email
Change Record Type
Clone
Submit for Approval
Delete
Edit
Change Owner
Sharing
Opportunity Detail
Standard Buttons


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Automobile Information
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Page Layouts
1 Items, Sorted by Page Layout Name
Quick Find
New
Page Layout Assignment
PAGE LAYOUT NAME
CREATED BY
MODIFIED BY
Automobile Information Layout
LAVANYA KOKKANTI, 16/04/2024, 9:58 am
LAVANYA KOKKANTI, 16/04/2024, 10:59 am

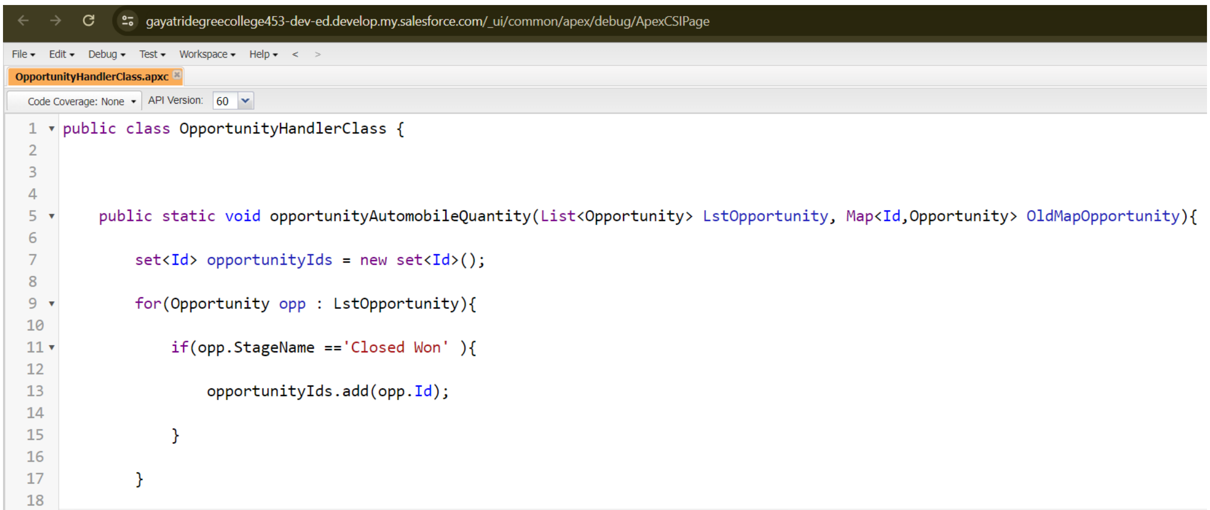


Milestone 07- APEX TRIGGER

- **Opportunity Automobile Quantity:**

Usecase: Whenever Opportunity Closed Won Than Neglect / Minus the Quantity from Automobile Information on the Bases of Opportunity Automobile quantity.

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



Trigger Handler:

How to create a new trigger:

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

```
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.

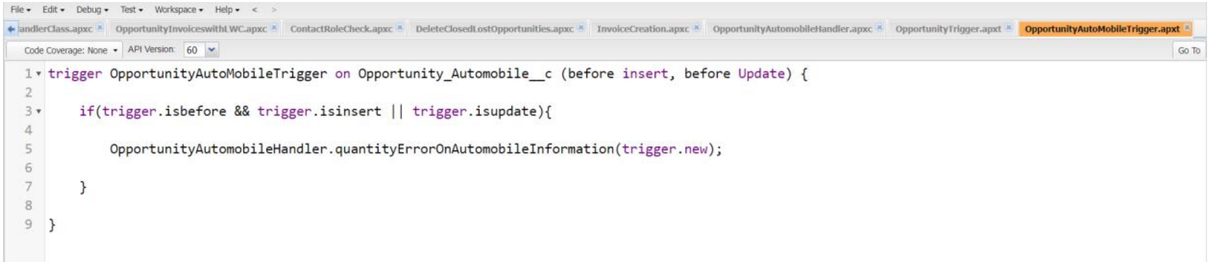
1. Click on the Developer console. Now you will see a new console window.
2. In the toolbar, you can see FILE. Click on it and navigate to new and create new apex class.
3. Name the class as “OpportunityAutomobileHandler”.

```
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, Pr
```

Trigger Handler:

How to create a new trigger:

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

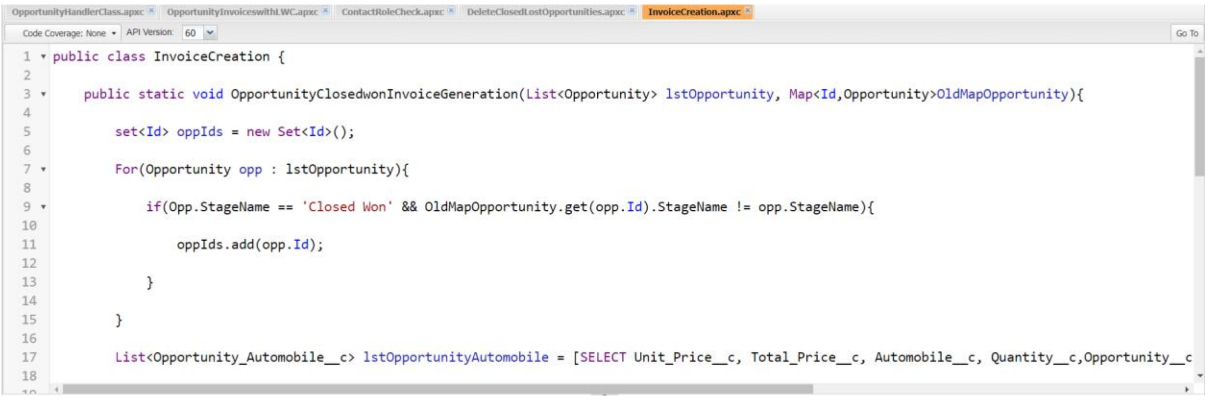


```
trigger OpportunityAutoMobileTrigger on Opportunity_Automobile__c (before insert, before Update) {

    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.

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



```
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
```

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

1. Go on files and click on open.
2. Click on triggers.
3. Double click on OpportunityTrigger.

OpportunityTrigger.apxt *

Code Coverage: None API Version: 60

```
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:**

Use Case: 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.

1. Click on the Developer console. Now you will see a new console window.
2. In the toolbar, you can see FILE. Click on it and navigate to new and create new apex class.
3. Name the class as “ContactRoleCheck”.

OpportunityHandlerClass.apxc OpportunityInvoiceswithLWC.apxc ContactRoleCheck.apxc

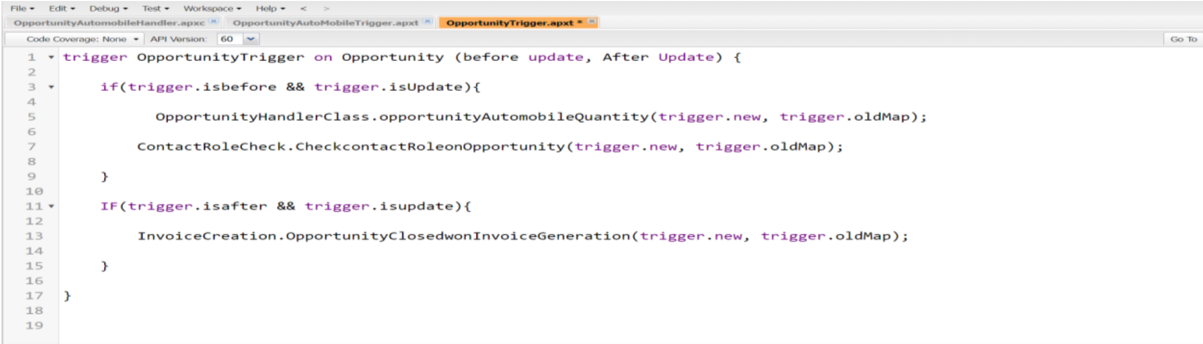
Code Coverage: None API Version: 60 Go To

```
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”.

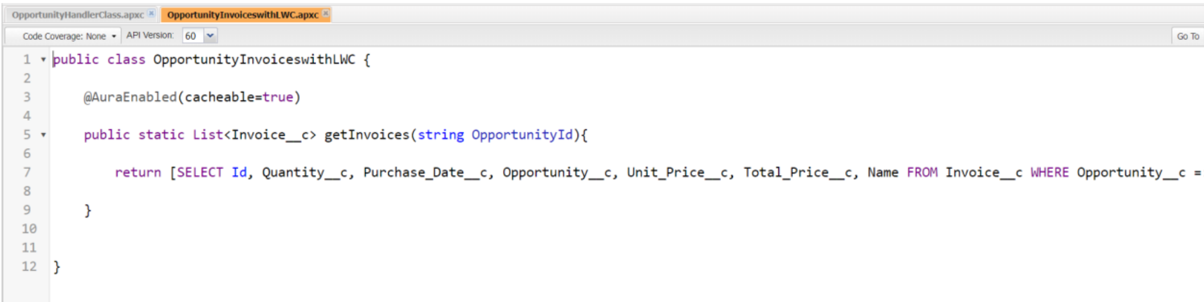
1. Go on files and click on open.
2. Click on triggers.
3. Double click on OpportunityTrigger.



Milestone 08- LWC COMPONENT

➢ **Create Apex Class to Get Invoices:**

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



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.

➢ **Install Microsoft VS Code:**

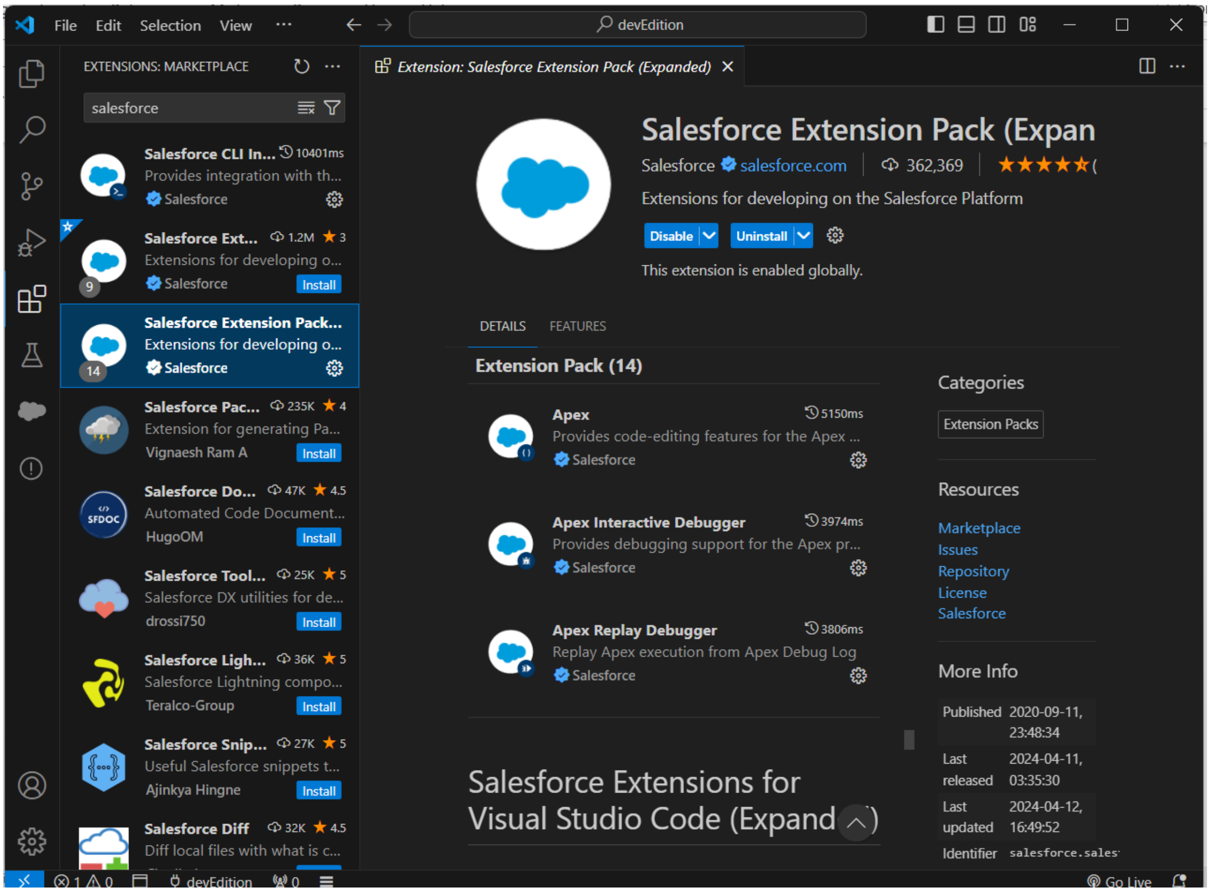
VS Code, or Visual Studio Code, is a free, open-source code editor developed by Microsoft. It is a lightweight, cross-platform code editor that provides features such as debugging, Git integration, and support for a wide range of programming languages.
Download the version of the software that is compatible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

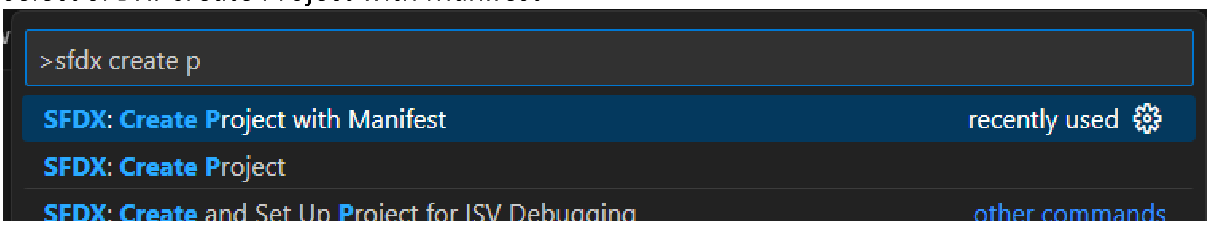
1. go to extensions (1) as shown in the image below.
2. Search with the Salesforce extension pack (2) as shown in the image below.
3. select Salesforce Extension Pack from the list (3) as shown in the image below.
4. Click the Install button (4) as shown in the image below.



The extension pack is installed successfully

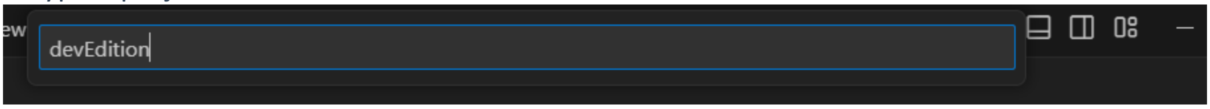
➢ **Create A Project in VS Code:**

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

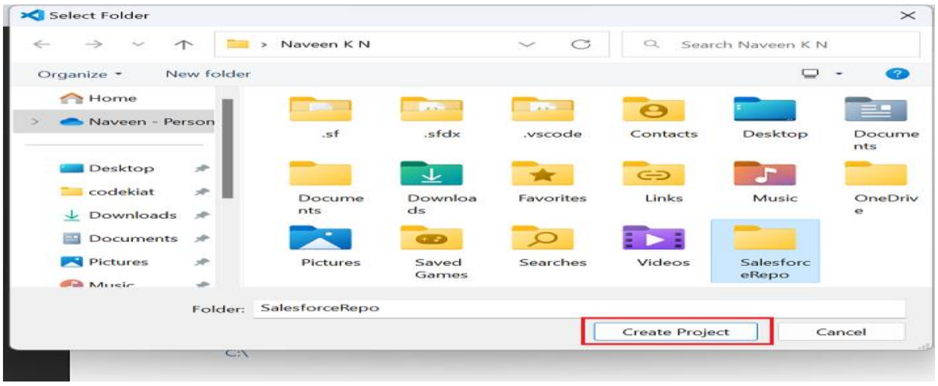


3. Select the standard project template.

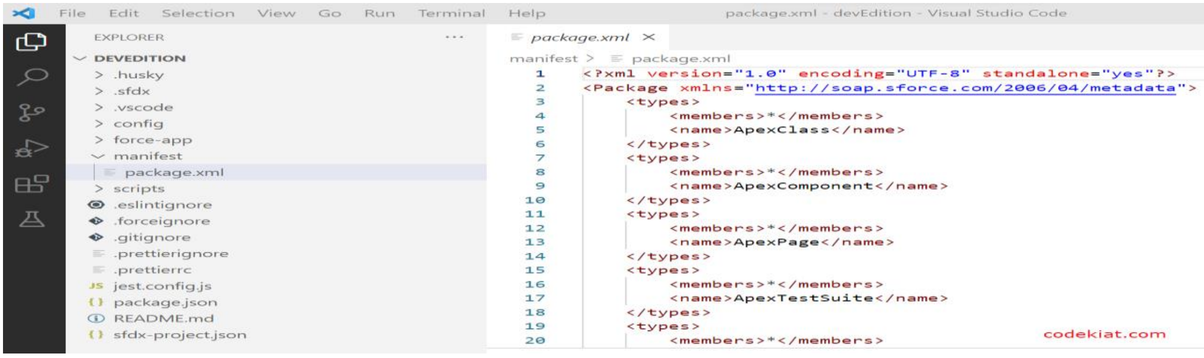
4.Type a project name and Click Enter.



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



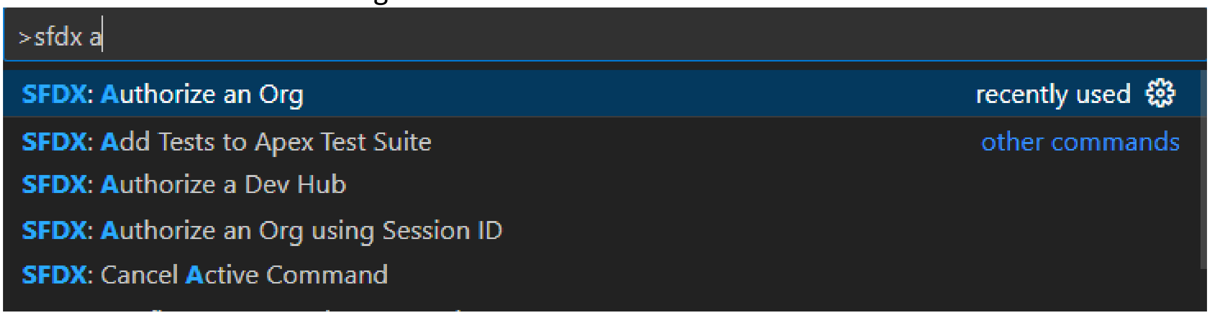
6.The new project is created with package.xml



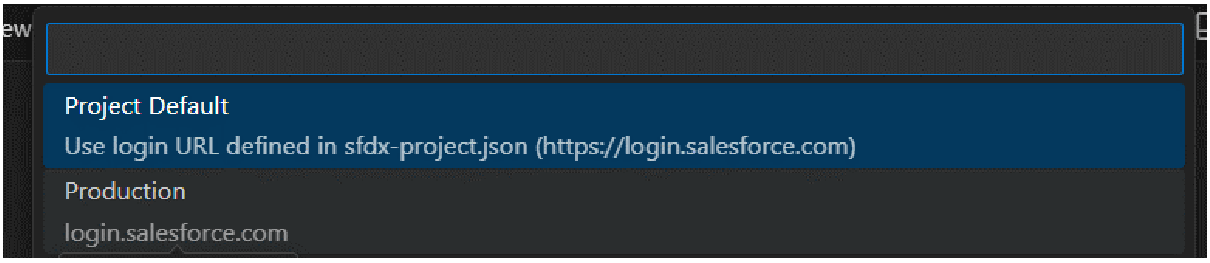
➢ **Authorize An Org:**

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

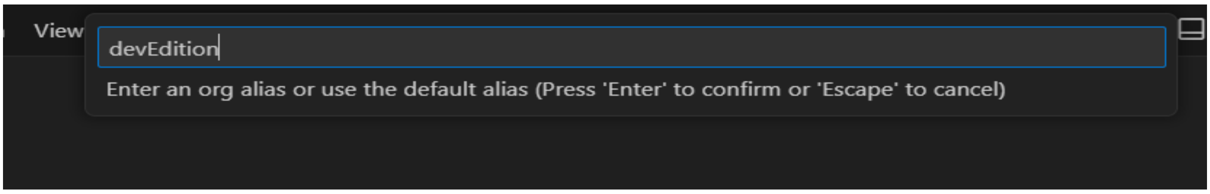
1. Press CTRL + SHIFT + P, type sfdx: authorize.
2. select SFDX: Authorize an Org from the list.



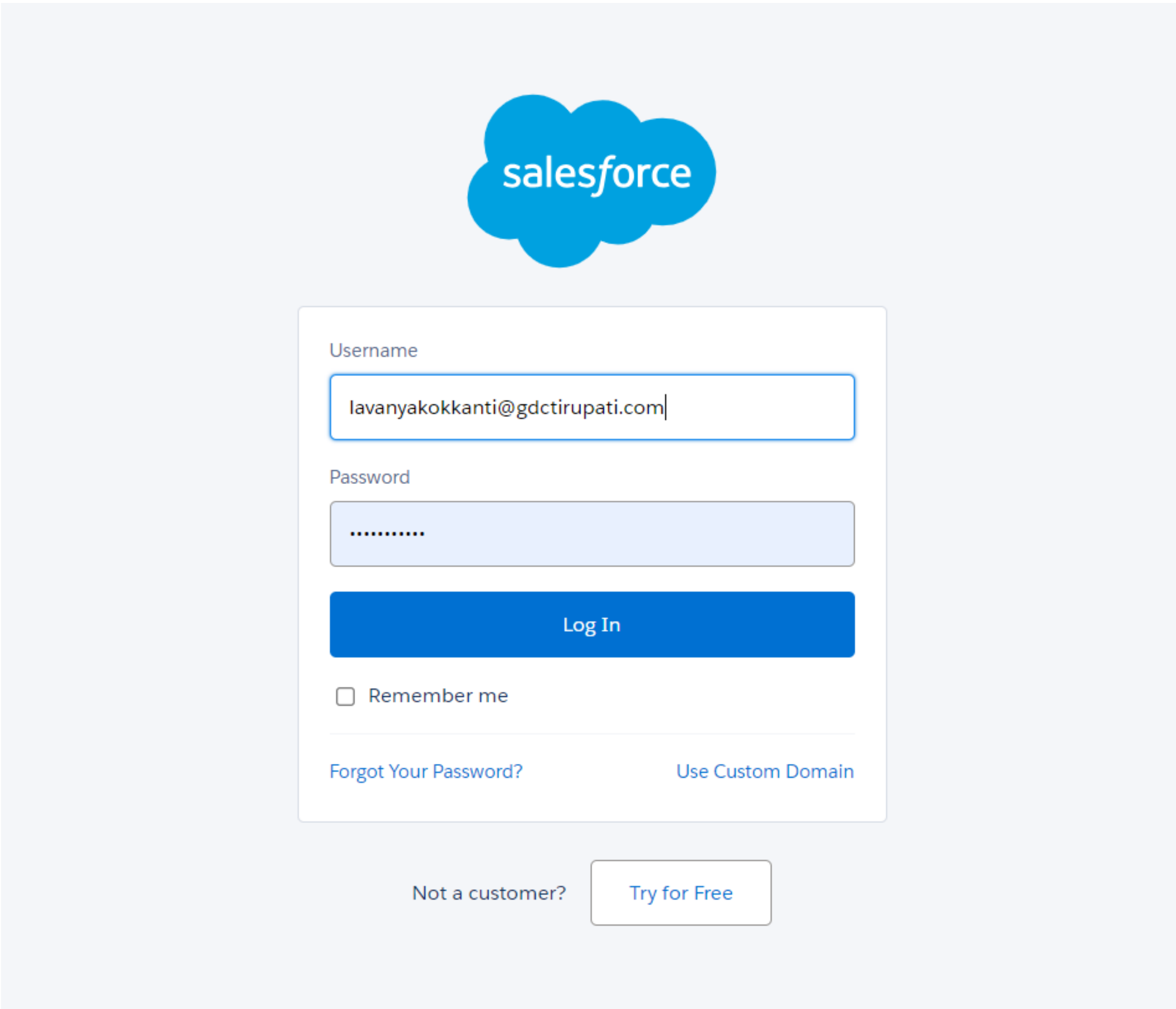
3. Choose your Salesforce instance.
   For developer edition and production instances select Production.



4. For this demonstration, I used the developer edition, hence it is Production.
5.Give a project name and press Enter



6.The Salesforce login page opens in the browser.
7.Enter the credentials and click Log In.

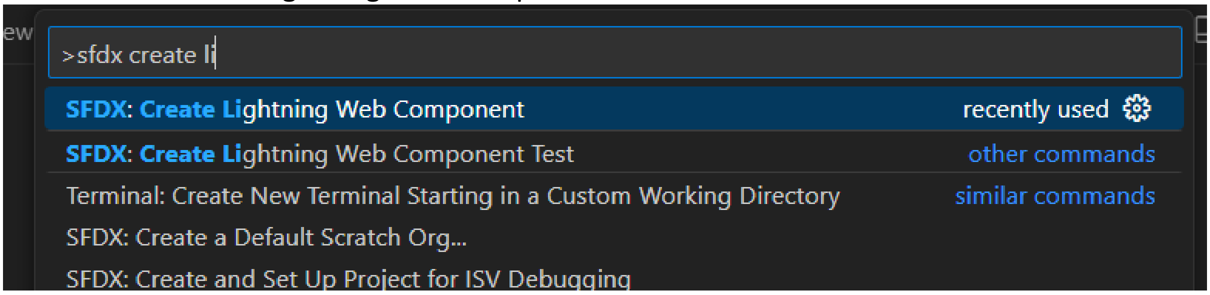


8.It will be successfully authorized.

- ➢ **Create Lightning Web Component:**

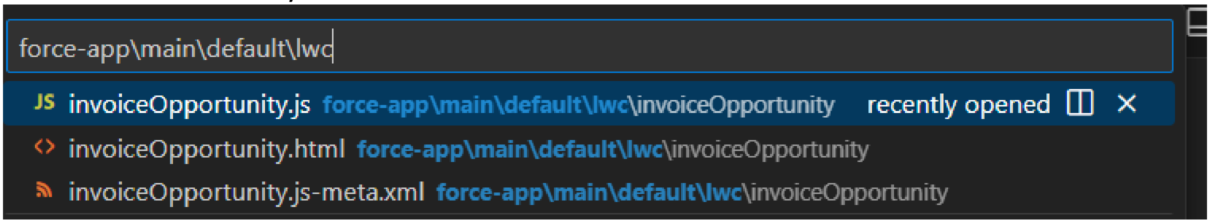
XML File:

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

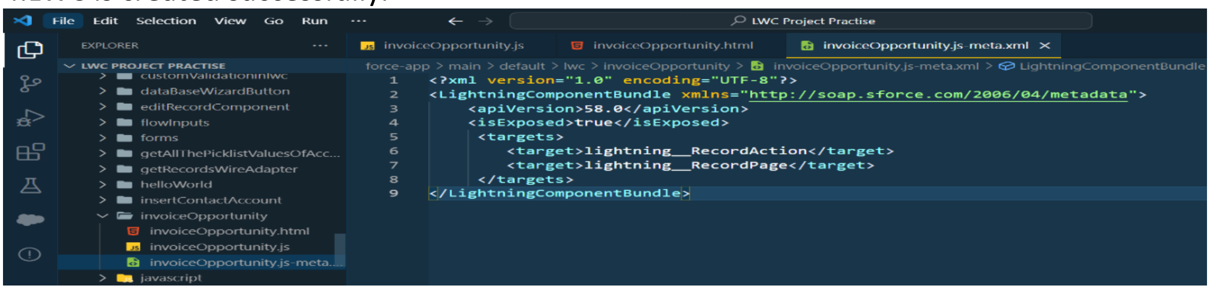


2. Give the name “InvoiceOpportunity” and press Enter.

3.Choose the directory.



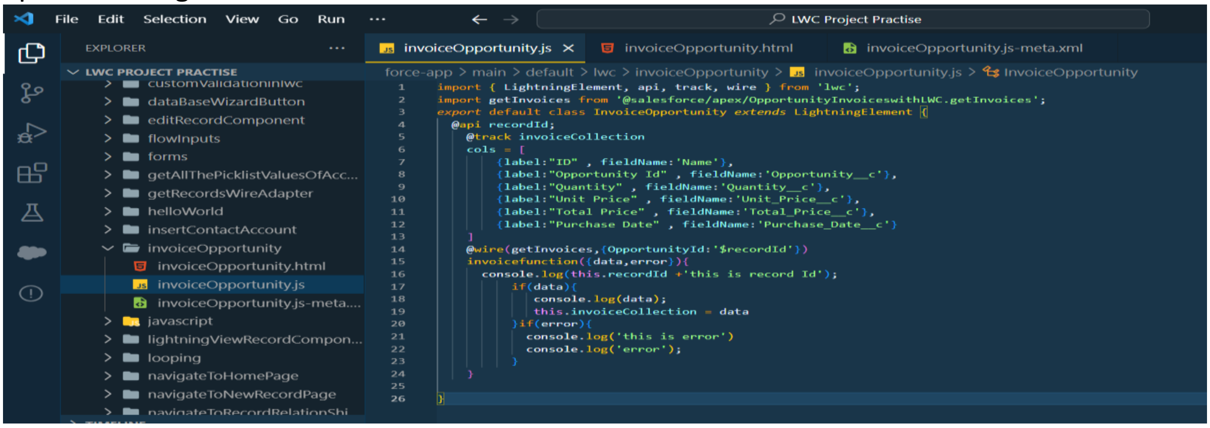
4.LWC is created successfully.



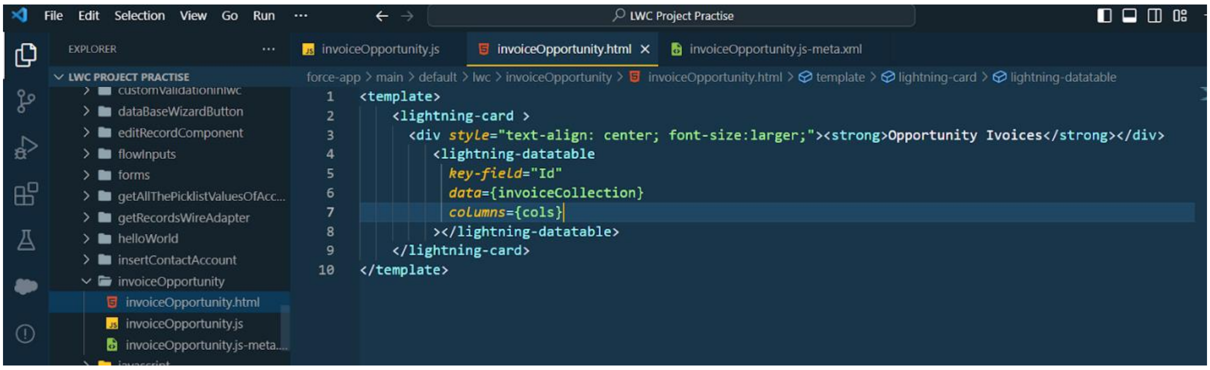
5.Copy and paste the below-mentioned code in the InvoiceOpportunity.js-meta.xml and update the apiVersion tag with the latest API version.

JS File:

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

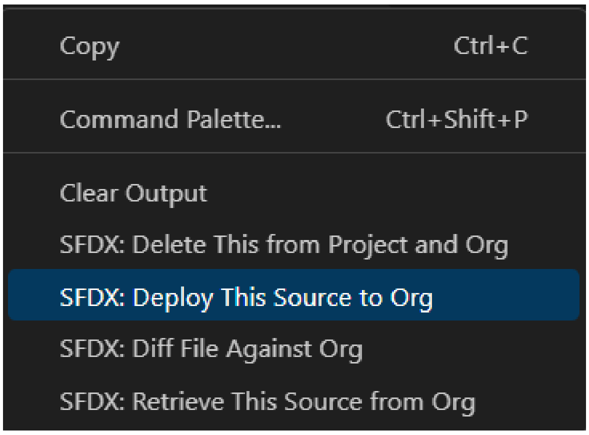


HTML File: Copy and paste the below-mentioned code in the InvoiceOpportunity.html and update the apiVersion tag with the latest API version.

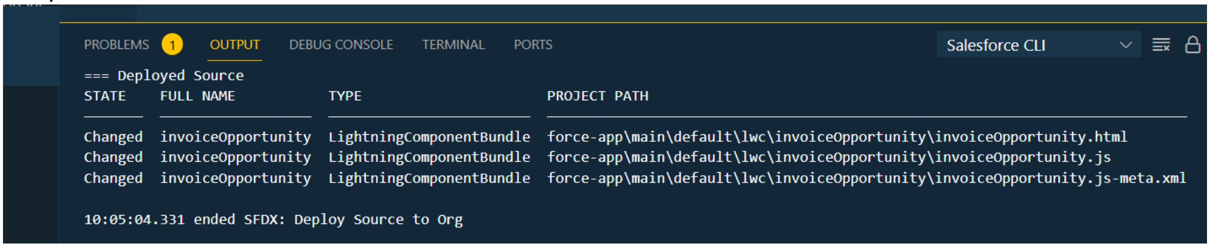


Deploy Component:

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

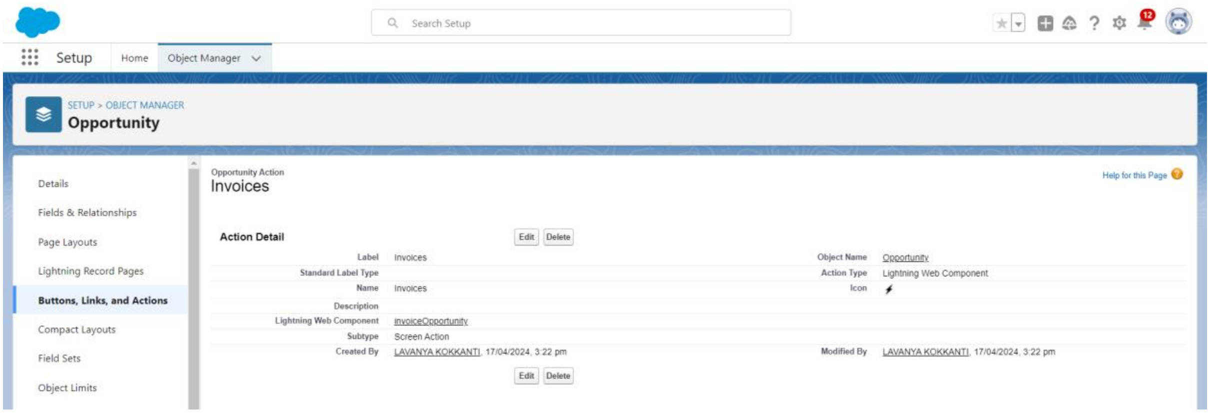


2. Once the deployment is complete, you will see the below-highlighted message in the output tab



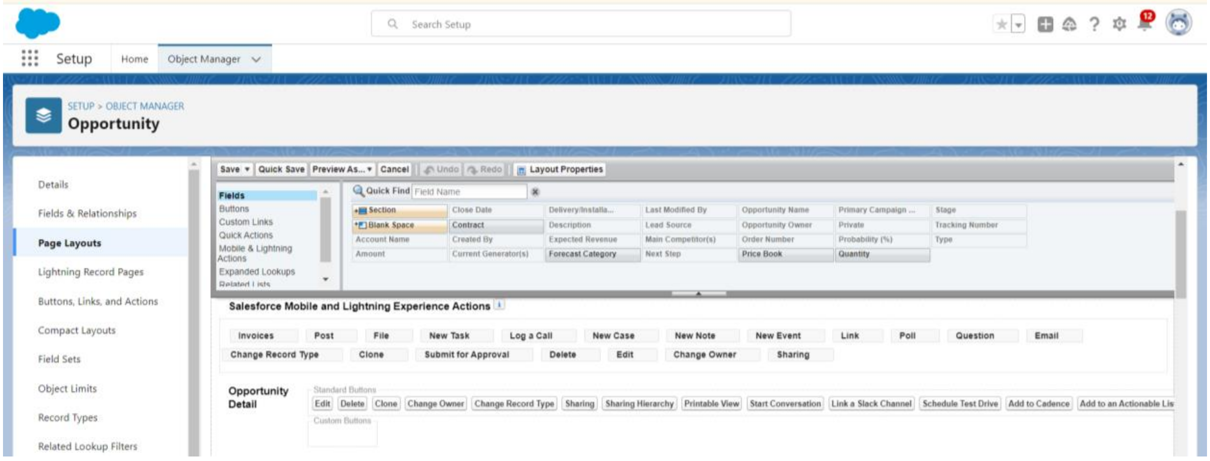
➢ **Create Button to Add on Opportunity:**

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.



➢ **Add InvoiceOpportunity Into Opportunity Record Page:**

1. On Opportunity Object Manager Click on Page layout.
2. Click on OpportunityLayout.
3. Click on Mobile and Lightning Action as show on below Image
4. Search for invoice on Quick Find.
5. Drag and Drop the Invoice into Salesforce Mobile and Lightning Experience Actions.



6.Click on Save.

Milestone 09- 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.

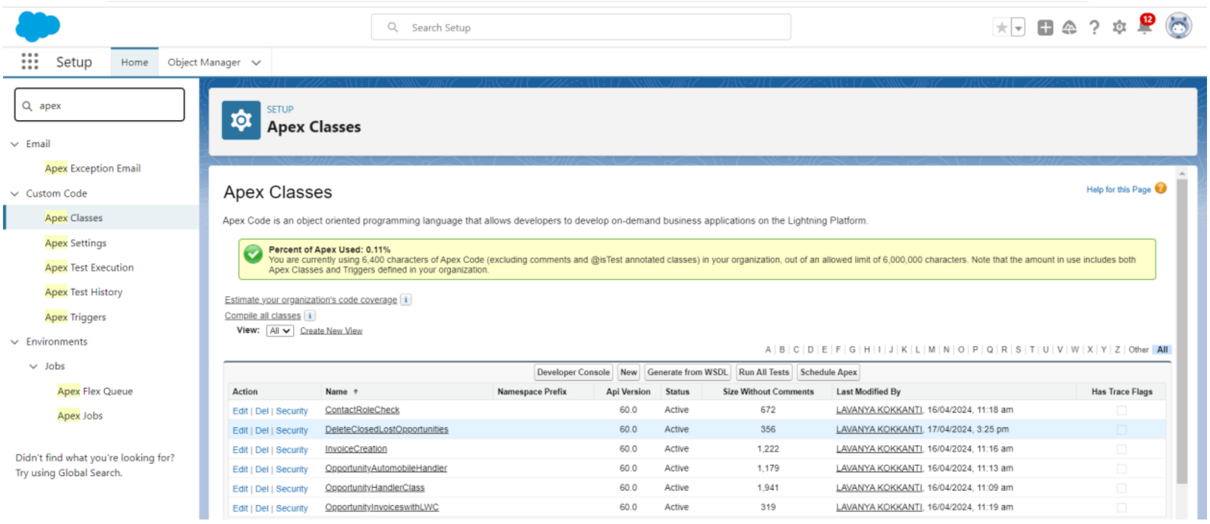
1. 1.Login to the respective account and navigate to the gear icon in the top right corner.
2. 2.Click on the Developer console. Now you will see a new console window.
3. 3.In the toolbar, you can see FILE. Click on it and navigate to new and create new apex class.
4. 4.Name the class as “DeleteClosedLostOpportunities”

```
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:

1. Go to the Home page in your salesforce account.
2. In the search bar, enter Apex and click on Apex Classes.
3. Click on Schedule Apex and enter the Job name.
    a. Job Name: DeleteOpportunitySchedule

4. Now click on the search icon present near the Apex class: Goto the Lookup icon beside? click on it? select DeleteClosedLostOpportunities.
5. In the Schedule Apex section, select weekly and select Monday mentioned and preferred time as 10:00 AM.
6. Click on Save. Now enter Apex in the search box and select Apex jobs.



Milestone 10- REPORTS

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics**.**

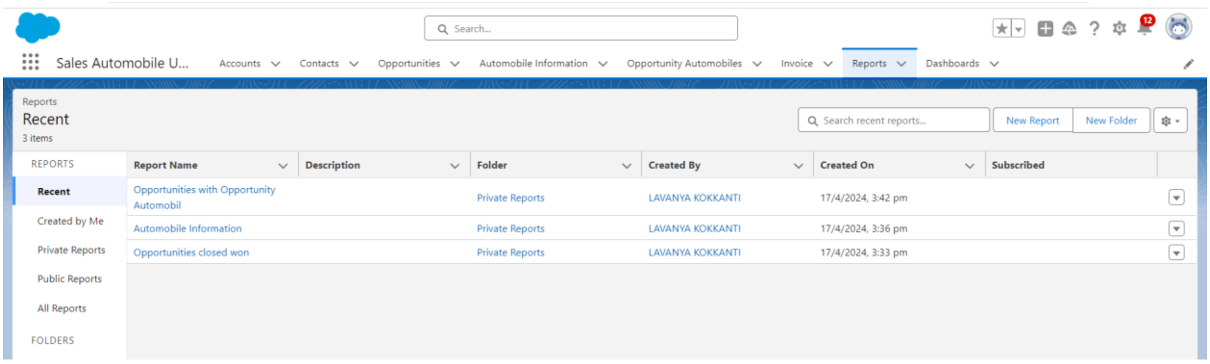
- **Create Report on Opportunity:**

1. Go to the app >> click on the reports tab
2. Click New Report.
3. 3.Select report type from category or from report type panel or from search panel >> click on start report.
4. 4.Customize your report
5. Add fields from left pane as shown below
6. Add the Above Filter as well.
7. 5.Save or run it.

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

- **Create Report on Automobile Information:**

1. Create a report with a report type: “Automobile Information”
2. Create a Report by using “Opportunities with Opportunity Automobiles and Automobile” Report Type.



Milestone 11- DASHBOARD

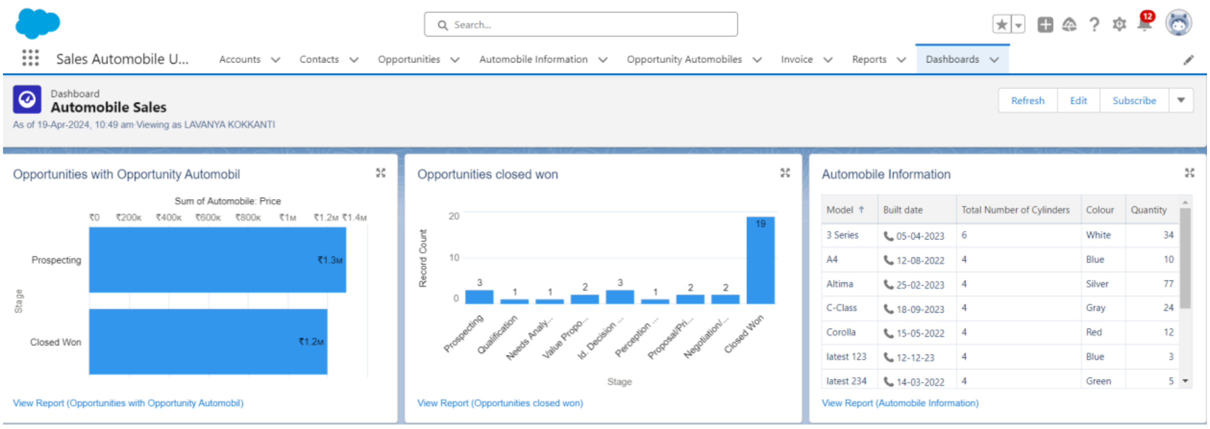
Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

- **Sales Dashboard:**

Create Dashboard

1. Go to the app? click on the Dashboards tabs.
2. Give a Name and click on Create.
   Name: Automobile Sales
3. Select add component.
4. Select a Report and click on select.
5. Click Add then click on Save and then click on Done.

The Created Dashboard will look like this.



THE END