

# PROJECT 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](https://developer.salesforce.com)
- Click on sign up.
- On the signup form, enter the following details
- First name & Last name- TABASSUM ARA SHAIK
- Email- sonushaik1625@gmail.com
- Role- SALESFORCE DEVELOPER
- Company- GAYATRI DEGREE COLLEGE- TIRUPATI
- County- INDIA
- Postal Code- 517501
- Username- sonushaik1625@ltgdcproject.com



## 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, for free

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

First Name\* TABASSUM ARA Last Name\* SHAIK

Email\* sonushaik1625@gmail.com

Role\* Developer

Company\* GAYATRI DEGREE COLLEGE - TIRUPATI

Country/Region\* India

State/Province\* Andhra Pradesh

Postal Code\* 517501

Username\* sonushaik1625@itgdcproject.com

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

☒ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement.

Yes, I would like to receive marketing communications regarding Salesforce products, services, and events. I can unsubscribe at any time.

By registering, you confirm that you agree to the processing of your personal data by Salesforce as described in the [Privacy Statement](#).

Sign me Up

Already have a Salesforce Developer Environment?

Log In

## Thanks for signing up with Salesforce!



Click below to verify your account.

Verify Account

To easily log in later, save this URL:

<https://gayatridegreecollege453-dev-ed.develop.my.salesforce.com>

Username:

[sonushaik1625@itgdcproject.com](mailto:sonushaik1625@itgdcproject.com)


Again, welcome to Salesforce!

## 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.

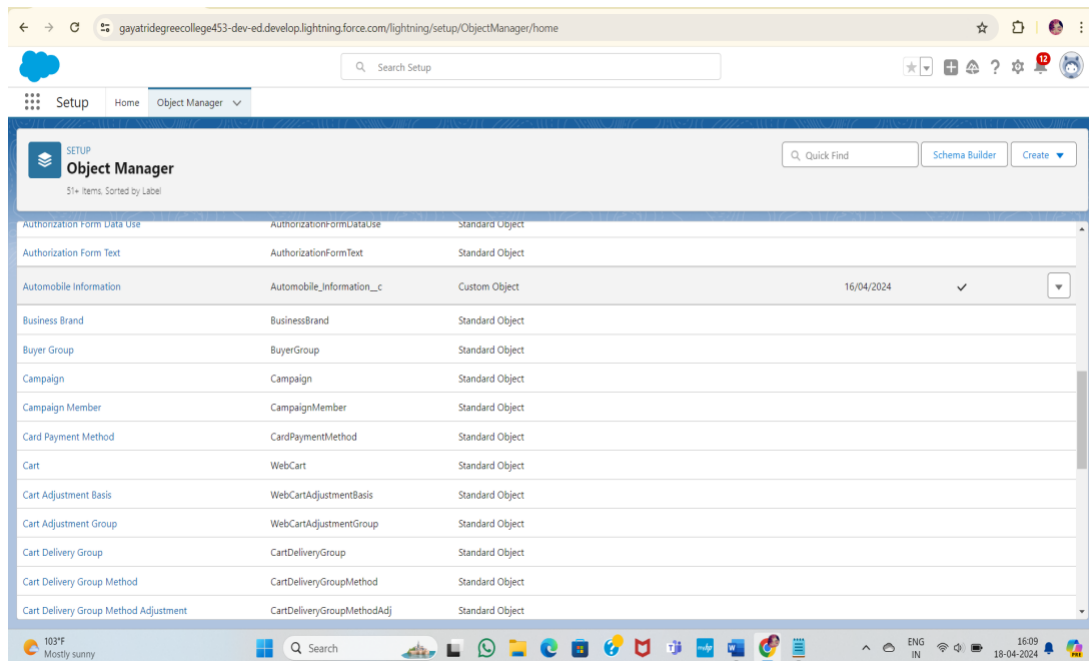
### ➤ Create Automobile Object:

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.

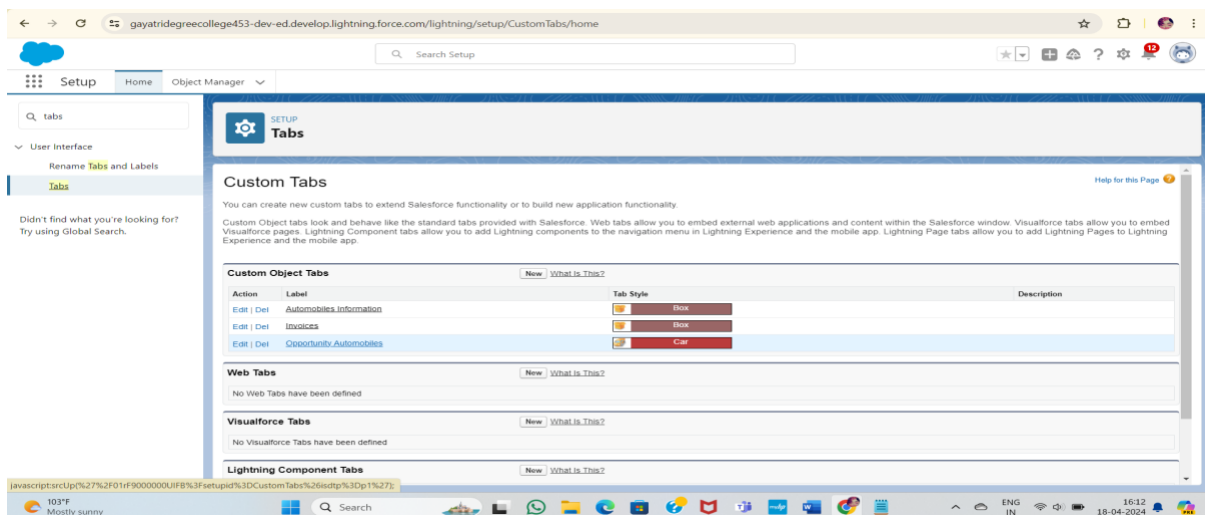
3. Allow search
4. Save



## 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

- 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 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.

App Name	Developer Name	Description	Last Modified Date	App Ty...	Visibl...
10 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	16/04/2024, 7:08 pm	Lightning	✓
11 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	16/04/2024, 7:08 pm	Classic	✓
12 Platform	Platform	The fundamental Lightning Platform	16/04/2024, 7:08 pm	Classic	✓
13 Queue Management	QueueManagement	Create and manage queues for your business.	16/04/2024, 7:08 pm	Lightning	✓
14 Sales	Sales	The world's most popular sales force automation (SFA) solution	16/04/2024, 7:08 pm	Classic	✓
15 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	16/04/2024, 7:08 pm	Lightning	✓
16 Sales Automobile Using ...	Sales_Automobile_Using_S...	Sales Automobile Using Salesforce CRM	16/04/2024, 9:31 pm	Lightning	✓
17 Sales Console	LightningSalesConsole	(Sales Automobile Using Salesforce CRM) with multiple records on one sc...	16/04/2024, 7:08 pm	Lightning	✓
18 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	16/04/2024, 7:08 pm	Classic	✓
19 Salesforce Scheduler Setup	LightningScheduler	Set up personalized appointment scheduling.	16/04/2024, 7:10 pm	Lightning	✓
20 Service	Service	Manage customer service with accounts, contacts, cases, and more	16/04/2024, 7:08 pm	Classic	✓
21 Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records acro...	16/04/2024, 7:08 pm	Lightning	✓
22 Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com a...	16/04/2024, 7:08 pm	Classic	✓
23 Subscription Management	RevenueCloudConsole	Get started automating your revenue processes	16/04/2024, 7:08 pm	Lightning	✓

## 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.

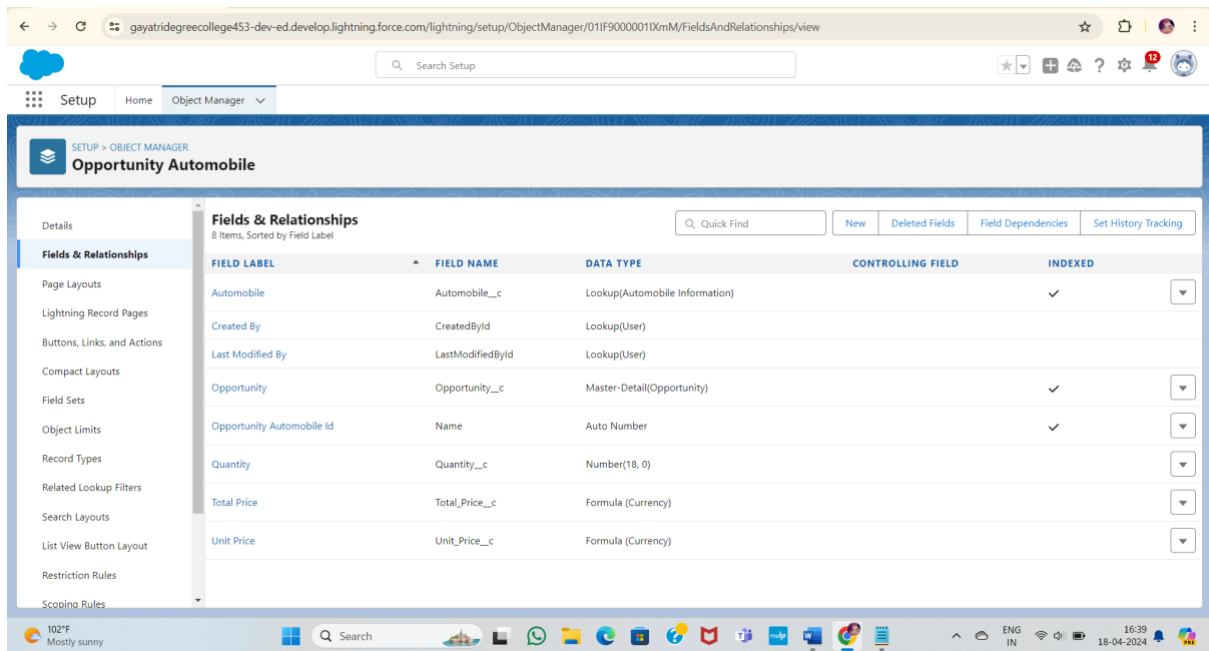
**Updating Field in Invoice Object:**

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.

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



## 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.



← → ↺

gayatridgecollege453-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/Opportunity/PageLayouts/view

☆ 📄 ? ⚙️ 12 🗨️

Setup

Home

Object Manager

SETUP > OBJECT MANAGER

Opportunity

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

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	TABASSUM ARA SHAIK, 16/04/2024, 7:08 pm	TABASSUM ARA SHAIK, 16/04/2024, 10:19 pm	▼
Opportunity (Sales) Layout	TABASSUM ARA SHAIK, 16/04/2024, 7:08 pm	TABASSUM ARA SHAIK, 16/04/2024, 10:19 pm	▼
Opportunity (Support) Layout	TABASSUM ARA SHAIK, 16/04/2024, 7:08 pm	TABASSUM ARA SHAIK, 16/04/2024, 10:19 pm	▼
Opportunity Layout	TABASSUM ARA SHAIK, 16/04/2024, 7:08 pm	TABASSUM ARA SHAIK, 16/04/2024, 11:16 pm	▼

← → ↺

gayatridgecollege453-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/Opportunity/PageLayouts/00hf900000AyAueIAF/view

☆ 📄 ? ⚙️ 12 🗨️

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

List View Button Layout

Scoping Rules

Save

Quick Save

Preview As...

Cancel

Undo

Redo

Layout Properties

Quick Find

Mobile Action Name

Add to an Action...	Change Record Type	Delete	Invoices	Mobile Smart Actions	New Event	New Opportunity	Printable View	Send Survey
Add to Cadence	Clone	Edit	Link	New Account	New Group	New Task	Questions	Sharing
Calculate Price	Clone with Related	Email	Link a Slack Channel	New Case	New Lead	Poll	Schedule Appointment	Sharing Hier
Change Owner	Create Summary	File	Log a Call	New Contact	New Note	Post	Schedule Test Drive	Start Cover

Customize the highlights panel for this page layout...

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

Edit Delete Clone Change Owner Change Record Type Sharing Sharing Hierarchy Printable View Start Conversation Link a Slack Channel Schedule Test Drive Add to Cadence Add to an Actionable Lis

Custom Buttons

← → ↺

gayatridgecollege453-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01f9000001XmC/PageLayouts/view

☆ 📄 ? ⚙️ 12 🗨️

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

Page Layouts

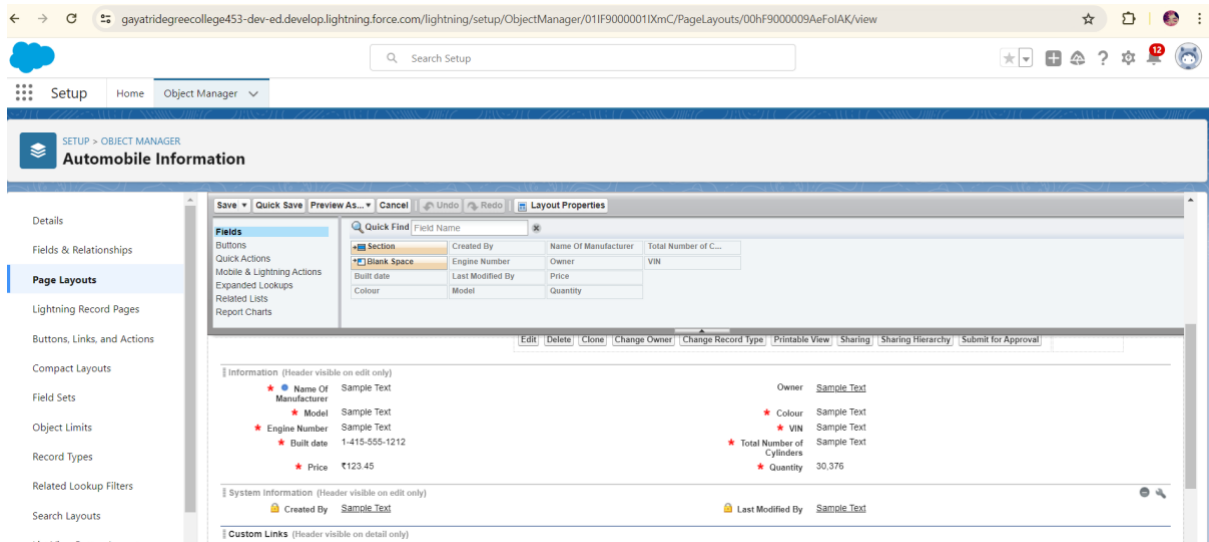
1 Items, Sorted by Page Layout Name

Quick Find

New

Page Layout Assignment

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY	
Automobile Information Layout	TABASSUM ARA SHAIK, 16/04/2024, 8:55 pm	TABASSUM ARA SHAIK, 16/04/2024, 10:34 pm	▼



## Milestone 07- APEX TRIGGER

### ➤ Opportunity Automobile Quantity:

**Use case:** 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".

```

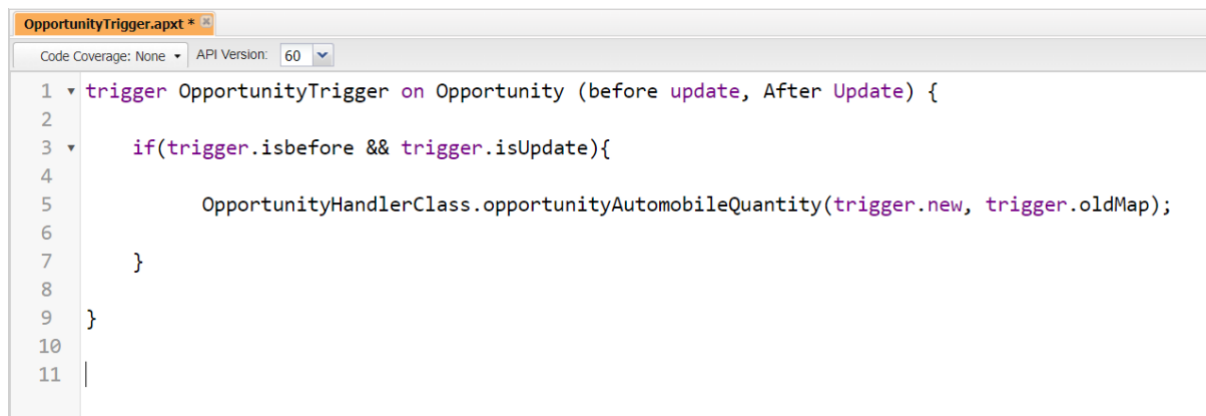
1 public class OpportunityHandlerClass {
2
3
4
5     public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity, Map<Id,Opportunity> OldMapOpportunity){
6
7         set<Id> opportunityIds = new set<Id>();
8
9         for(Opportunity opp : LstOpportunity){
10
11             if(opp.StageName == 'Closed Won' ){
12
13                 opportunityIds.add(opp.Id);
14
15             }
16
17         }
18     }

```

## 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



```
OpportunityTrigger.apxt
Code Coverage: None API Version: 60
1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2
3     if(trigger.isbefore && trigger.isUpdate){
4
5         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
6
7     }
8
9 }
10
11
```

### ➤ 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”.

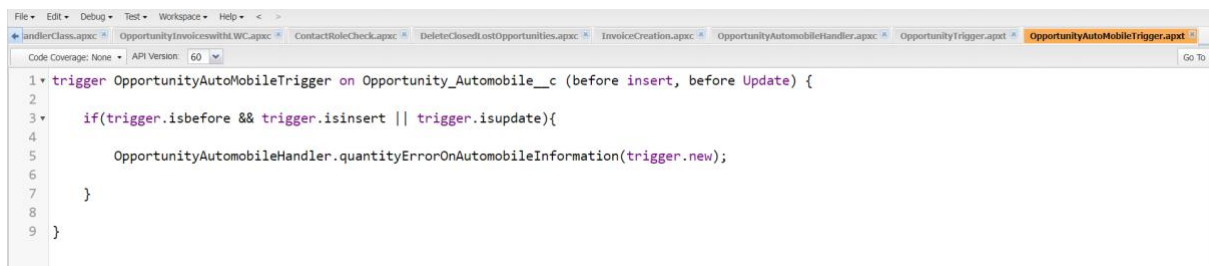


```
File Edit Debug Test Workspace Help
OpportunityHandlerClass.apxc OpportunityInvoiceswithLWC.apxc ContactRoleCheck.apxc DeleteClosedLostOpportunities.apxc InvoiceCreation.apxc OpportunityAutomobileHandler.apxc
Code Coverage: None API Version: 60 Go To
1 public class OpportunityAutomobileHandler {
2
3     public static void quantityErrorOnAutomobileInformation(List<Opportunity_Automobile__c> lstOpportunityAutomobile){
4
5         Set<Id> AutomobileIds = new Set<Id>();
6
7         For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
8
9             if(oppAutomobile.Automobile__c != null){
10
11                 AutomobileIds.add(oppAutomobile.Automobile__c);
12
13             }
14
15         }
16
17         Map<Id, Automobile_Information__c> lstAutomobileInformation = new map<Id, Automobile_Information__c>([SELECT Id, CreatedById, Quantity__c, Pr
18
```

## Trigger Handler:

### How to create a new trigger:

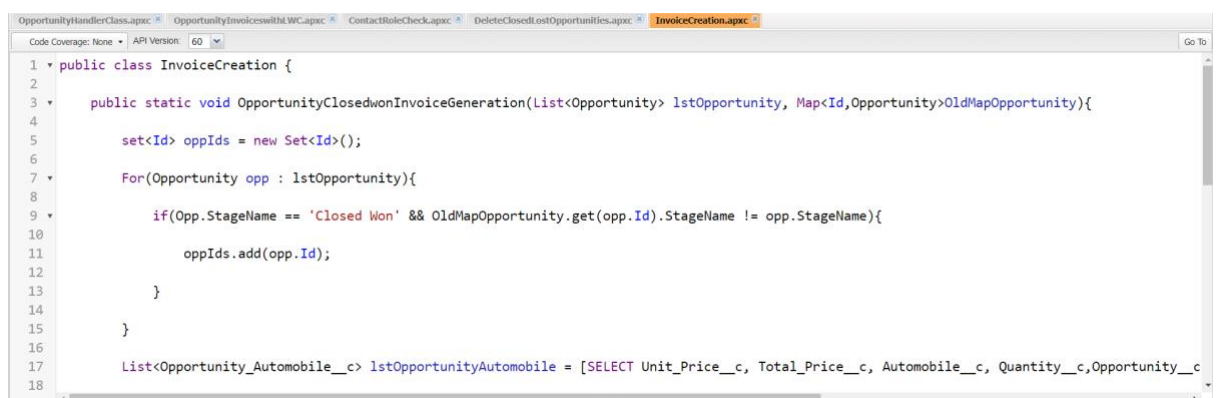
1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on the File menu in the toolbar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5.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



### ➤ Invoice Creation Trigger:

**Use case:** 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”.

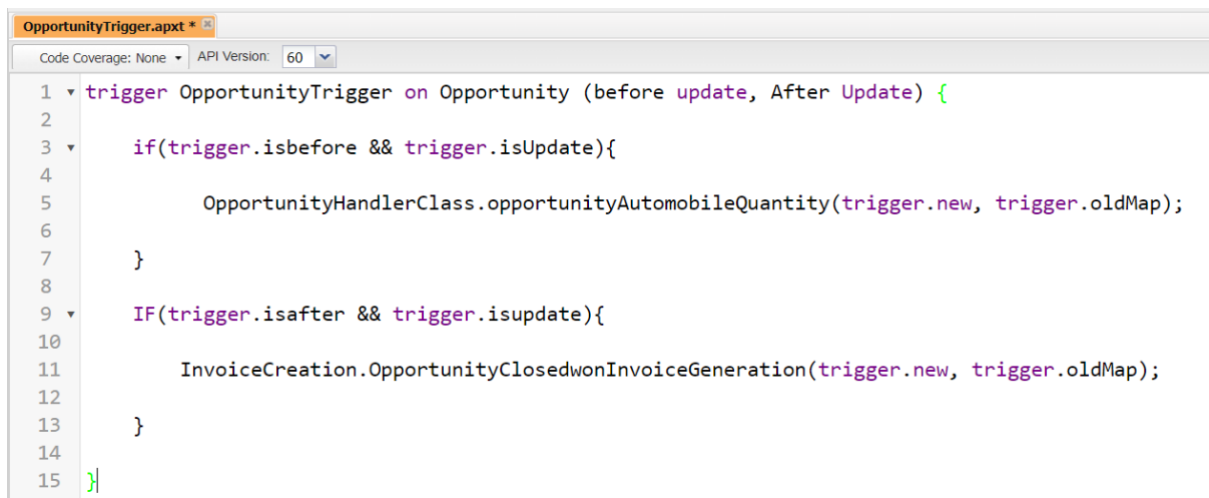


## Trigger Handler:

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.



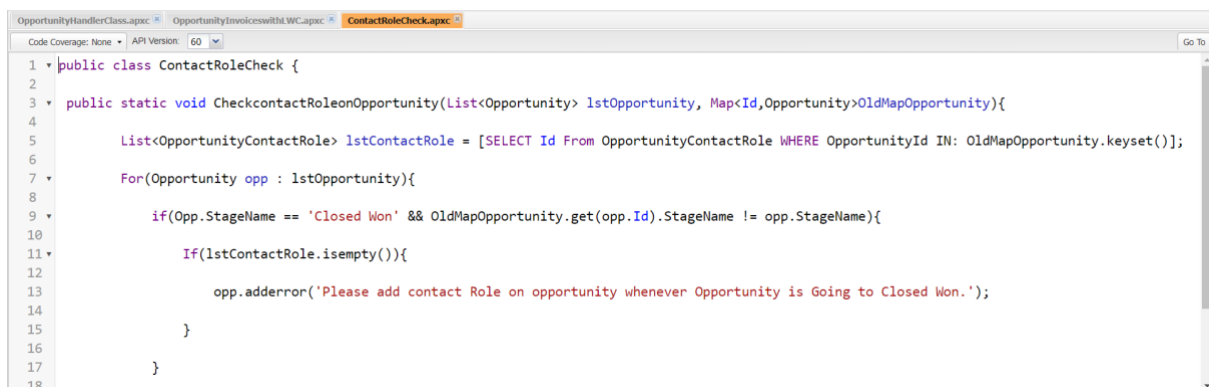
```
1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2
3     if(trigger.isbefore && trigger.isUpdate){
4
5         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
6
7     }
8
9     IF(trigger.isafter && trigger.isupdate){
10
11         InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);
12
13     }
14
15 }
```

#### ➤ 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”.

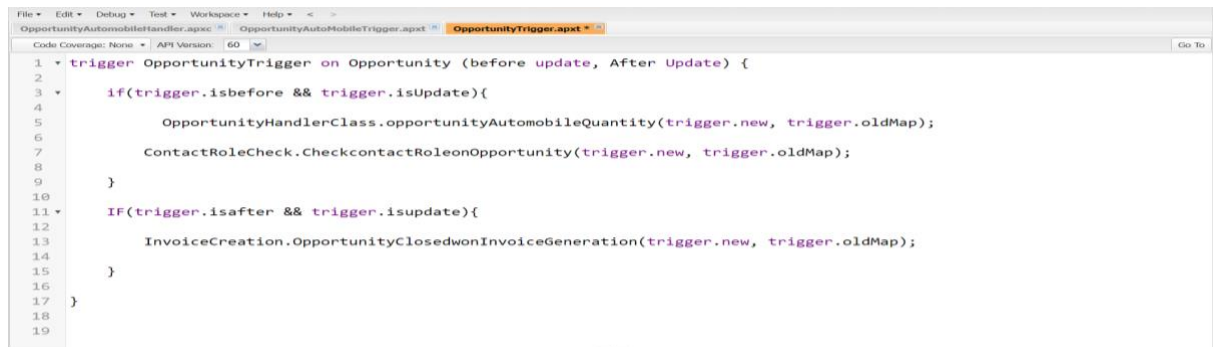


```
1 public class ContactRoleCheck {
2
3     public static void CheckcontactRoleonOpportunity(List<Opportunity> lstOpportunity, Map<Id,Opportunity>OldMapOpportunity){
4
5         List<OpportunityContactRole> lstContactRole = [SELECT Id From OpportunityContactRole WHERE OpportunityId IN: OldMapOpportunity.keySet()];
6
7         For(Opportunity opp : lstOpportunity){
8
9             if(opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
10
11                 If(lstContactRole.isEmpty()){
12
13                     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.

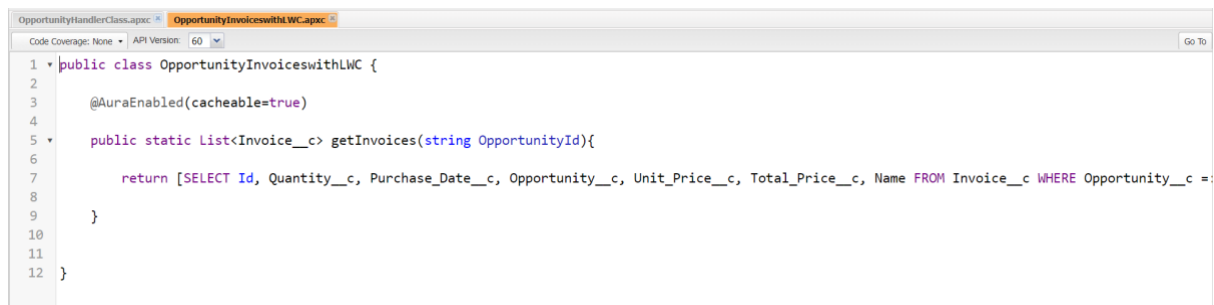


```
1 trigger OpportunityTrigger on Opportunity (before update, After Update) {
2
3     if(trigger.isbefore && trigger.isUpdate){
4
5         OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
6
7         ContactRoleCheck.CheckcontactRoleonOpportunity(trigger.new, trigger.oldMap);
8
9     }
10
11     IF(trigger.isafter && trigger.isupdate){
12
13         InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);
14
15     }
16
17 }
18
19
```

## 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”.



```
1 public class OpportunityInvoiceswithLWC {
2
3     @AuraEnabled(cacheable=true)
4
5     public static List<Invoice__c> getInvoices(string OpportunityId){
6
7         return [SELECT Id, Quantity__c, Purchase_Date__c, Opportunity__c, Unit_Price__c, Total_Price__c, Name FROM Invoice__c WHERE Opportunity__c = :OpportunityId];
8
9     }
10
11
12 }
```

## 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.

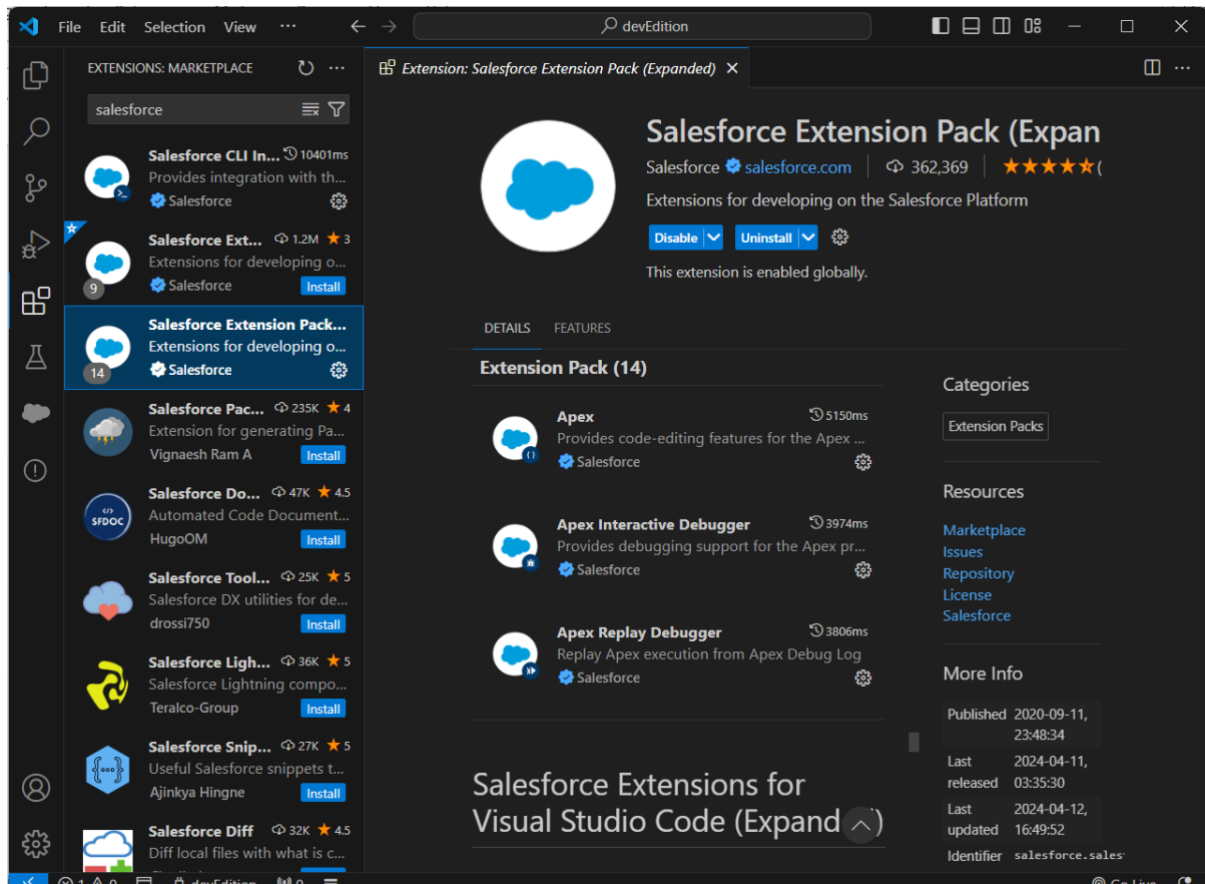
Download the version of the software that is compatible with your operating system and install it.

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

➤ **Install The Salesforce Extension Pack:**

In the VS Code,

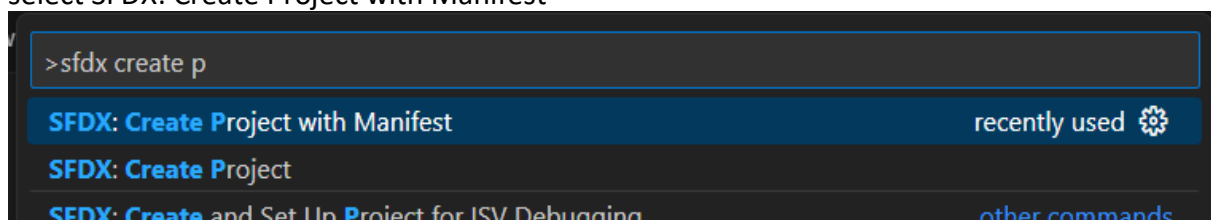
1. go to extensions (1) as shown in the image below.
2. Search with the Salesforce extension pack (2) as shown in the image below.
3. select Salesforce Extension Pack from the list (3) as shown in the image below.
4. Click the Install button (4) as shown in the image below.



The extension pack is installed successfully

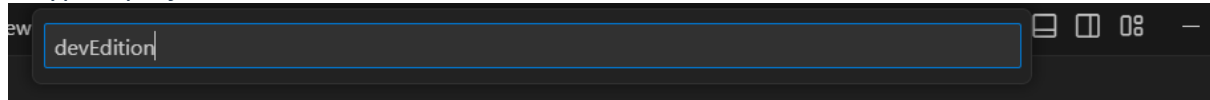
➤ **Create A Project in VS Code:**

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

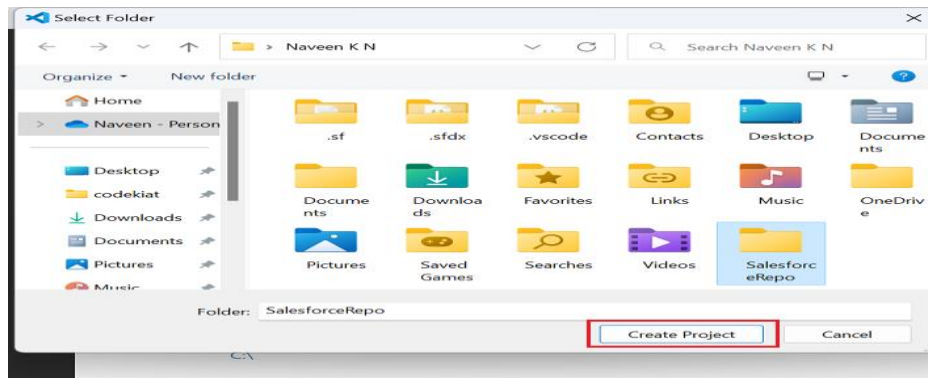


3. Select the standard project template.

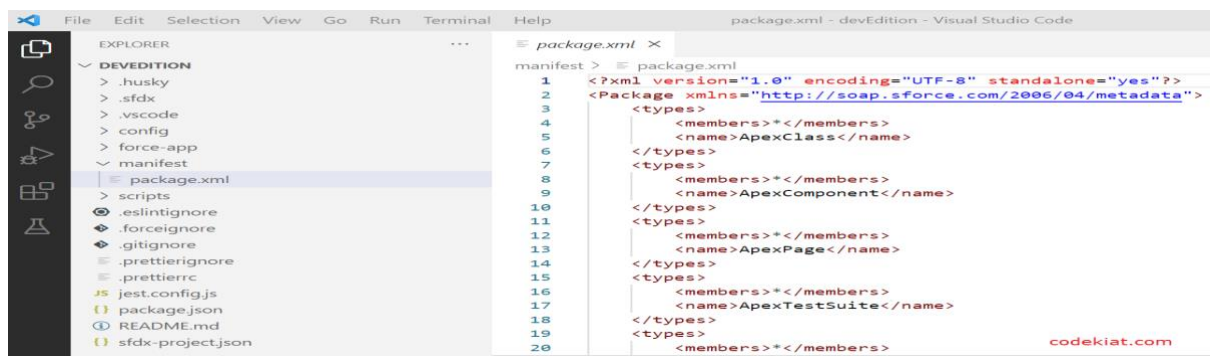
4.Type a project name and Click Enter.



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



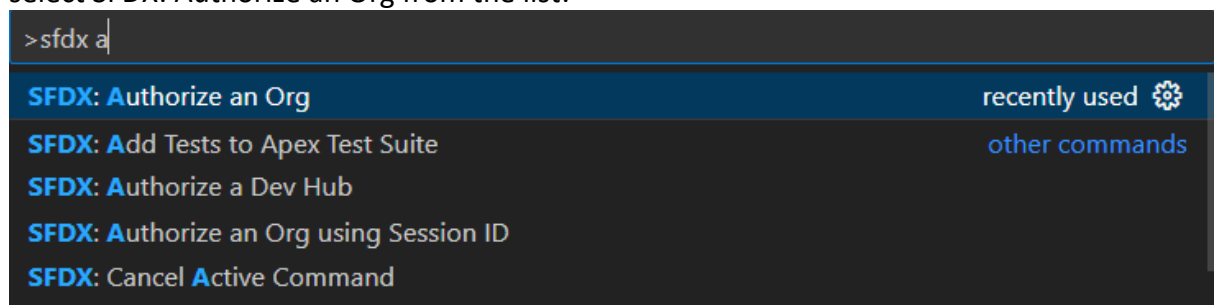
6.The new project is created with package.xml



### ➤ Authorize An Org:

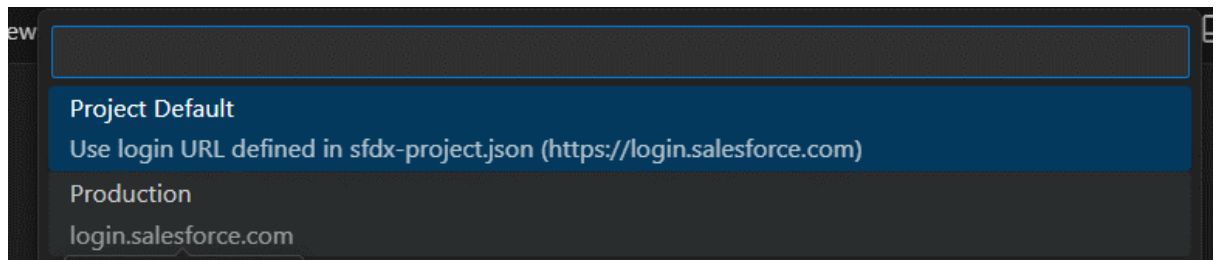
Establish a connection between the local project and the Salesforce instance to retrieve and deploy the components.

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

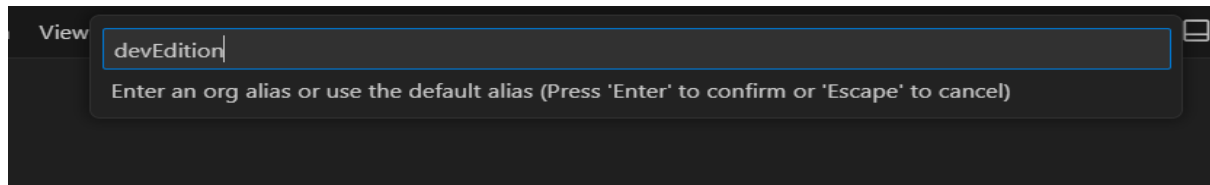


3. Choose your Salesforce instance.  
For developer edition and production instances select Production.

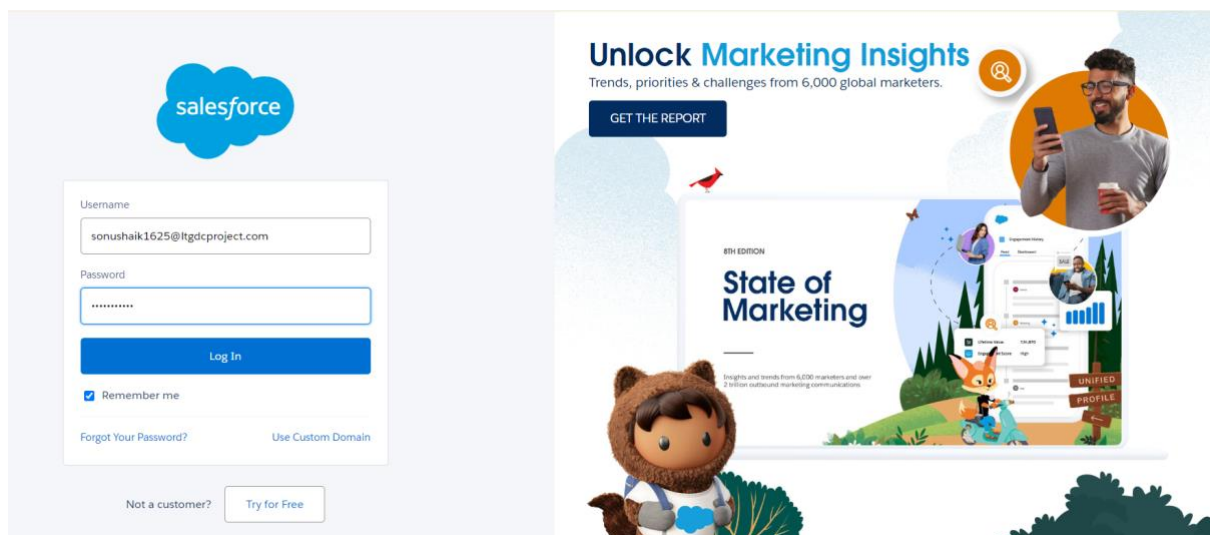




4. For this demonstration, I used the developer edition, hence it is Production.
5. Give a project name and press Enter



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

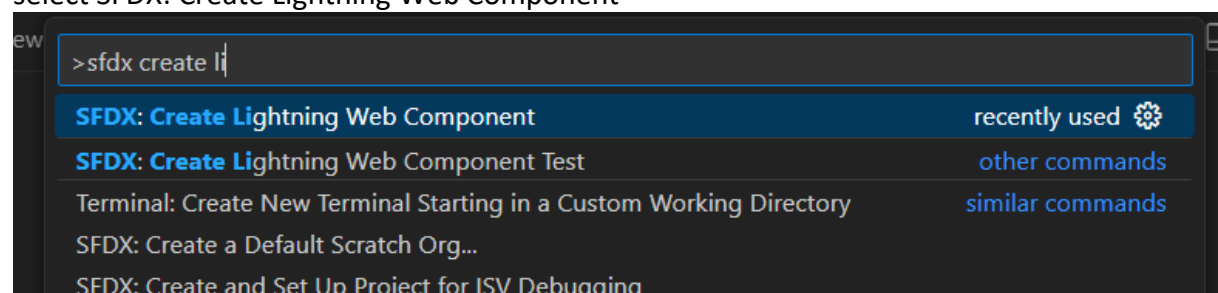


8. It will be successfully authorized.

### ➤ 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



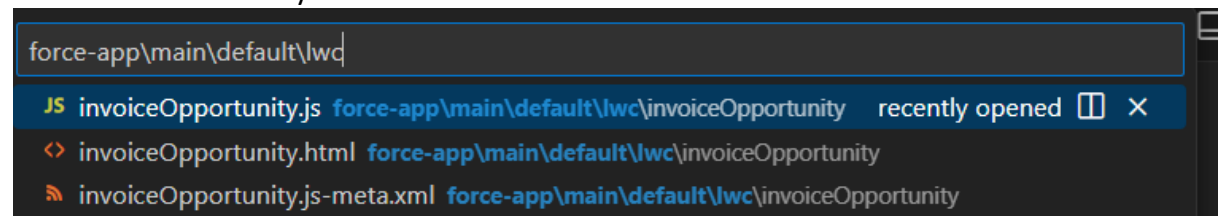
2. Give the name "InvoiceOpportunity" and press Enter.



The screenshot shows the VS Code interface with a terminal window open. The terminal prompt is 'force-app > main > default > lwc > invoiceOpportunity >'. The user has entered 'InvoiceOpportunity' and pressed Enter. The terminal shows the command 'invoiceOpportunity.js-meta.xml' and the resulting XML content for the Lightning Component Bundle.

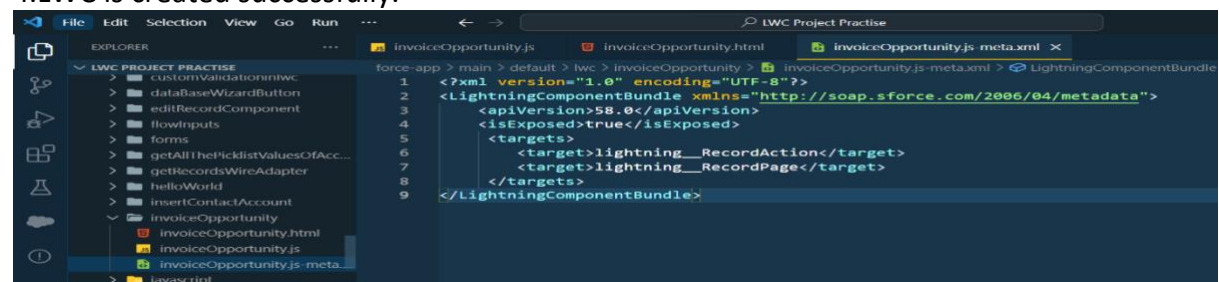
```
InvoiceOpportunity
force-app > main > default > lwc > invoiceOpportunity > invoiceOpportunity.js-meta.xml > LightningComponentBundle
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3   <apiVersion>58.0</apiVersion>
4   <isExposed>true</isExposed>
5   <targets>
6     <target>lightning__RecordAction</target>
7     <target>lightning__RecordPage</target>
8   </targets>
9 </LightningComponentBundle>
```

3. Choose the directory.



The screenshot shows the VS Code Explorer with the directory 'force-app/main/default/lwc' selected. The file 'invoiceOpportunity.js' is highlighted, and the path 'force-app/main/default/lwc/invoiceOpportunity' is shown in the breadcrumb. The file is also listed in the 'recently opened' section.

4. LWC is created successfully.



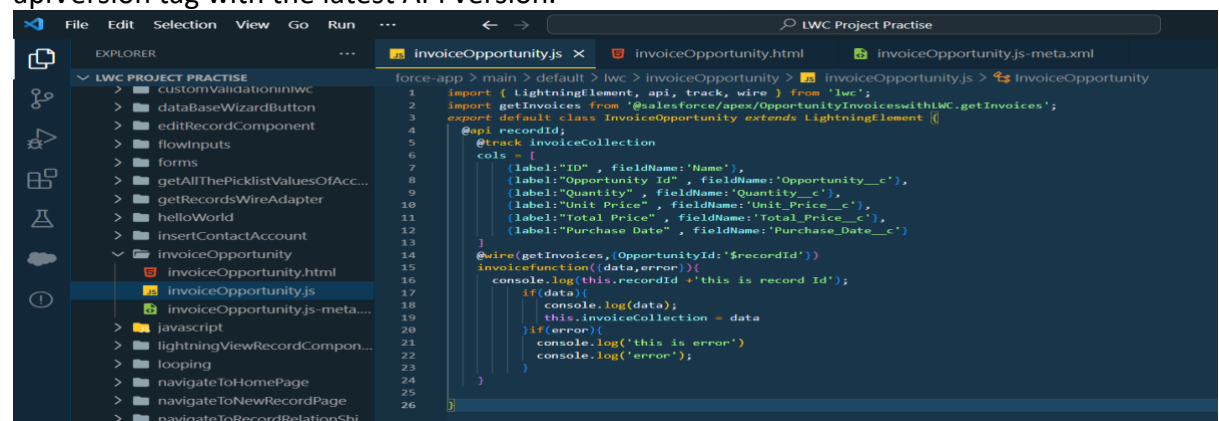
The screenshot shows the VS Code Explorer with the directory 'force-app/main/default/lwc/invoiceOpportunity' expanded. The files 'invoiceOpportunity.html', 'invoiceOpportunity.js', and 'invoiceOpportunity.js-meta.xml' are visible. The 'invoiceOpportunity.js-meta.xml' file is selected, and its content is shown in the editor.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
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>
```

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:**

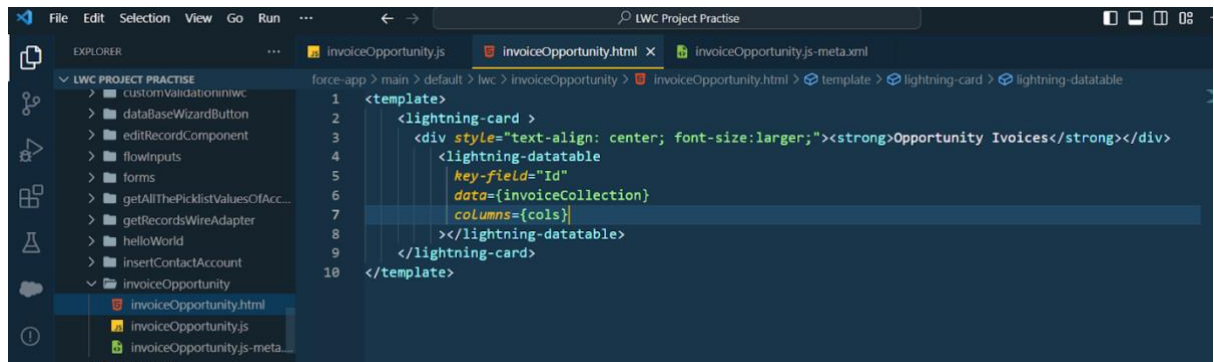
Copy and paste the below-mentioned code in the InvoiceOpportunity.js and update the apiVersion tag with the latest API version.



The screenshot shows the VS Code editor with the 'invoiceOpportunity.js' file open. The code defines a class 'InvoiceOpportunity' that extends 'LightningElement'. It includes a '@track' decorator for 'invoiceCollection' and a '@wire' decorator for 'getInvoices'. The 'getInvoices' method is a wire adapter that fetches invoice data and logs it to the console.

```
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}
```

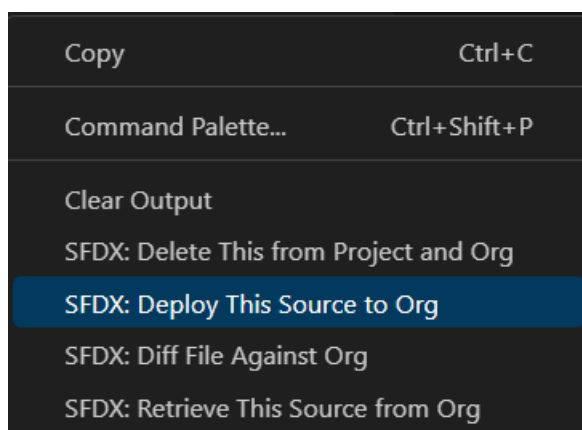
**HTML File:** Copy and paste the below-mentioned code in the InvoiceOpportunity.html and update the apiVersion tag with the latest API version.



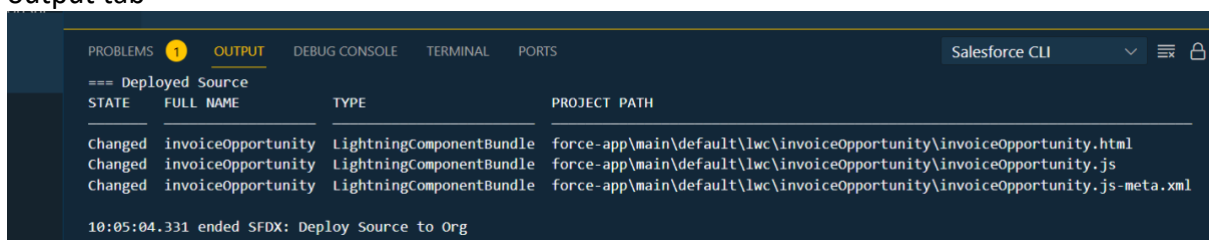
```
1 <template>
2   <lightning-card>
3     <div style="text-align: center; font-size: larger;"><strong>Opportunity Ivoices</strong></div>
4     <lightning-datatable
5       key-field="Id"
6       data={invoiceCollection}
7       columns={cols}>
8     </lightning-datatable>
9   </lightning-card>
10 </template>
```

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

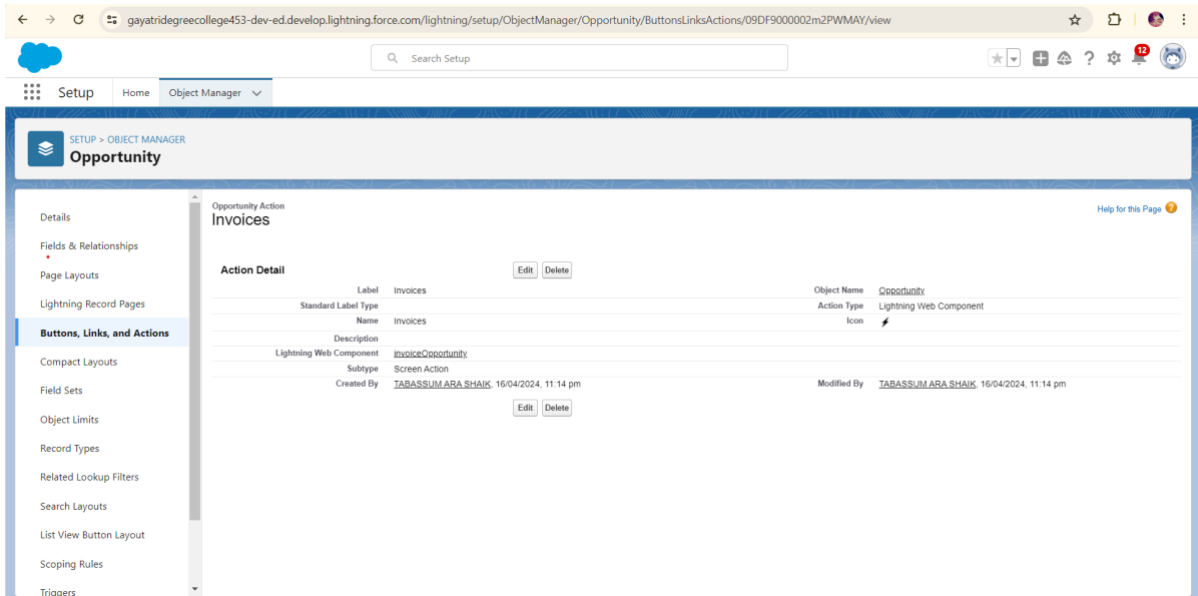


```
==== Deployed Source
STATE    FULL NAME          TYPE                      PROJECT PATH
-----
Changed  invoiceOpportunity  LightningComponentBundle force-app\main\default\lwc\invoiceOpportunity\invoiceOpportunity.html
Changed  invoiceOpportunity  LightningComponentBundle force-app\main\default\lwc\invoiceOpportunity\invoiceOpportunity.js
Changed  invoiceOpportunity  LightningComponentBundle force-app\main\default\lwc\invoiceOpportunity\invoiceOpportunity.js-meta.xml

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

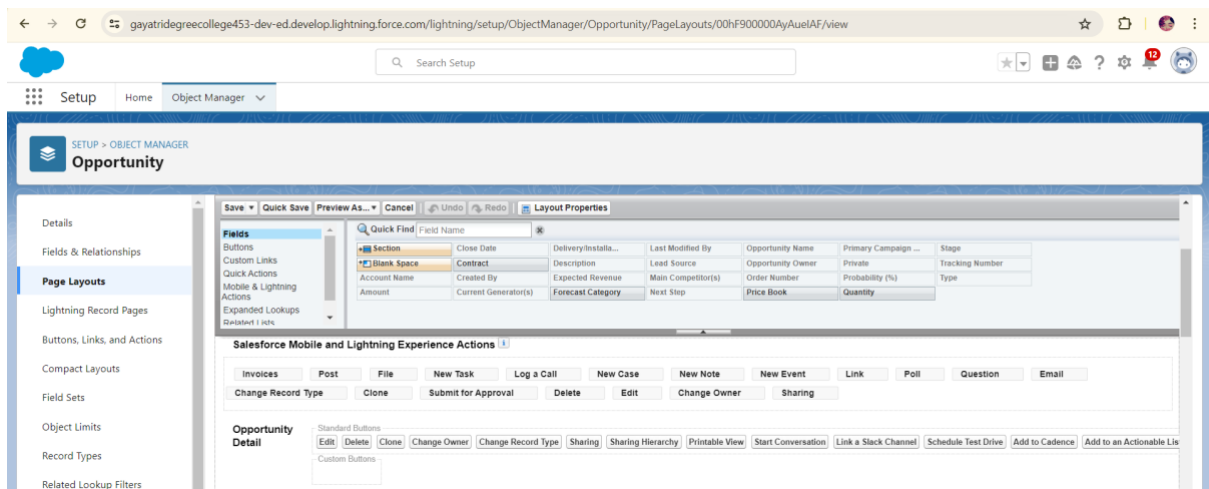
### ➤ Create Button to Add on Opportunity:

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.

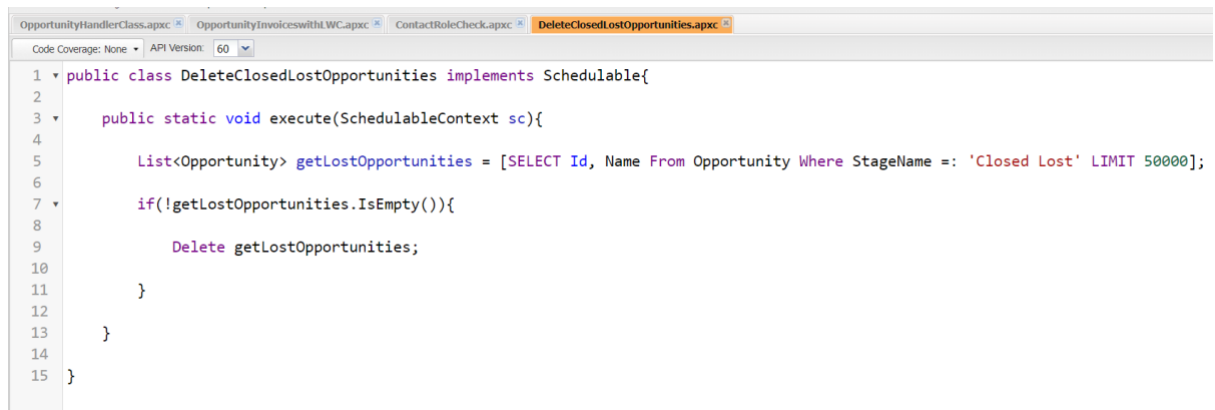
## Milestone 09- APEX SCHEDULERS:

### ➤ 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. 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”



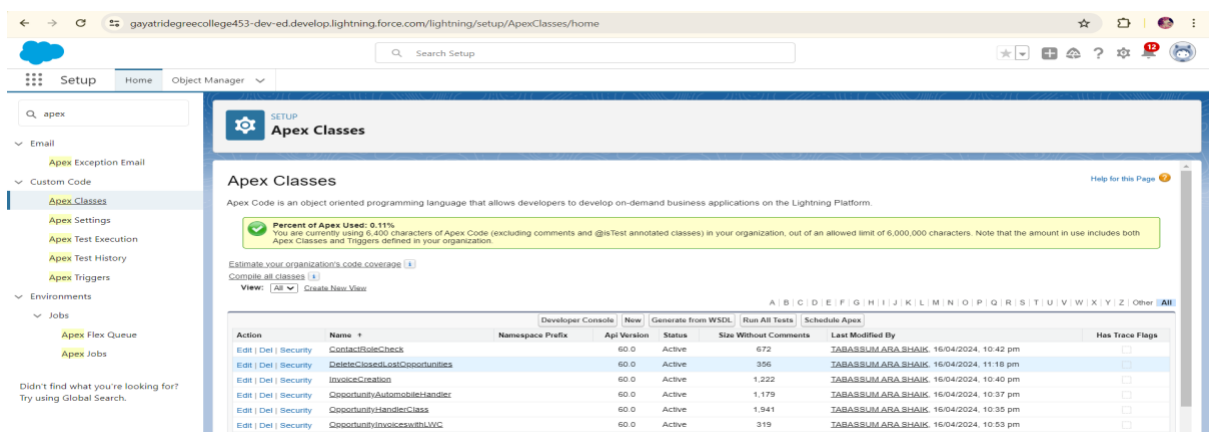
```

1 public class DeleteClosedLostOpportunities implements Schedulable{
2
3     public static void execute(SchedulableContext sc){
4
5         List<Opportunity> getLostOpportunities = [SELECT Id, Name From Opportunity Where StageName =: 'Closed Lost' LIMIT 50000];
6
7         if(!getLostOpportunities.isEmpty()){
8
9             Delete getLostOpportunities;
10
11         }
12     }
13 }
14
15

```

### 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.



Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	ContactRoleCheck		60.0	Active	672	TABASSUM.ARA.SHAIK, 16/04/2024, 10:42 pm	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	DeleteClosedLostOpportunities		60.0	Active	356	TABASSUM.ARA.SHAIK, 16/04/2024, 11:18 pm	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	InvoiceCreation		60.0	Active	1,222	TABASSUM.ARA.SHAIK, 16/04/2024, 10:40 pm	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	OpportunityAutomobileHandler		60.0	Active	1,179	TABASSUM.ARA.SHAIK, 16/04/2024, 10:37 pm	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	OpportunityHandlerClass		60.0	Active	1,941	TABASSUM.ARA.SHAIK, 16/04/2024, 10:35 pm	<input type="checkbox"/>
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	OpportunityInvoicesWithWC		60.0	Active	319	TABASSUM.ARA.SHAIK, 16/04/2024, 10:53 pm	<input type="checkbox"/>

### 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.

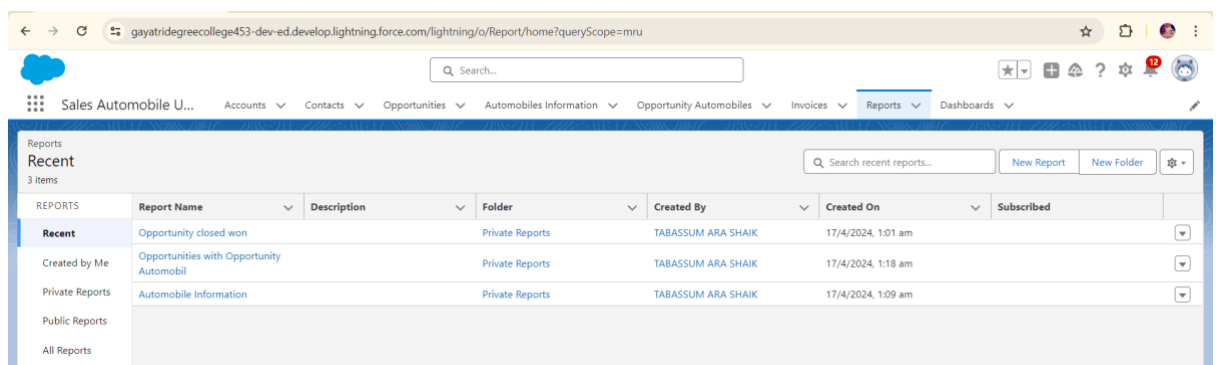
➤ **Create Report on Opportunity:**

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

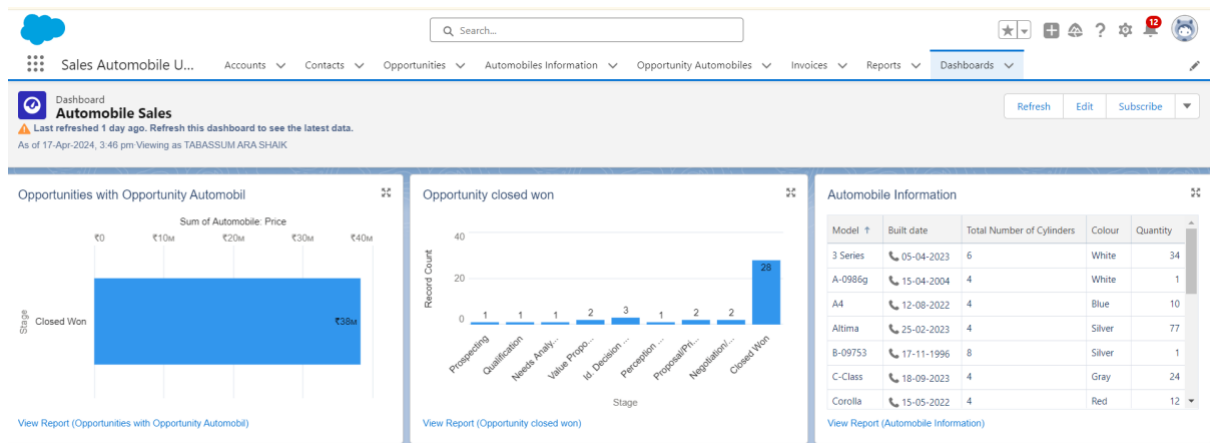
➤ **Sales Dashboard:**

**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

