

SALES AUTOMOBILE USING SALESFORCE CRM

The project aims to leverage Salesforce CRM to streamline and enhance the sales processes of an automobile company. By implementing a centralized, cloud-based system, the company can efficiently manage customer relationships, track leads, automate workflows, and analyze sales performance. The dealership will gain a centralized platform to manage customer interactions, optimize lead-to-sale conversion rates, and foster long-term customer relationships.

With advanced tools like AI-driven predictions, automated workflows, and real-time dashboards, the solution empowers sales teams to deliver personalized experiences, reduce response times, and improve overall sales efficiency. Furthermore, the mobile-friendly features of Salesforce enable sales representatives to operate effectively from anywhere, enhancing productivity and customer satisfaction.

The project will involve:

- **Requirements Gathering:**

- Collaborating with stakeholders to understand business needs, sales workflows, and pain points in the current system.

- **Salesforce CRM Implementation:**

- Configuring **Sales Cloud** for lead and opportunity management.
- Setting up **Service Cloud** for after-sales support and customer service.

- **Integration with Third-Party Tools:**

- Connecting Salesforce with inventory management systems to track vehicle availability.
- Enabling payment gateway and financing options for seamless transactions.

- **Testing and Deployment:**

- Rigorous testing of the configured Salesforce system to ensure all features work as intended.
- Deploying the solution and monitoring its performance in the live environment.

- **Post-Implementation Support:**

- Offering maintenance, support, and periodic updates to enhance the platform's efficiency.
- Gathering user feedback to implement improvements and additional features.

Task 1: Create Objects from Spreadsheet

Objective:

- The primary objective of this project is to implement Salesforce CRM to streamline and optimize the sales processes of an automobile business.
- Automate repetitive tasks and simplify workflows to allow the sales team to focus on building relationships and closing deals.

Objects Created:

1. Automobile Information, Invoice:

- These are key objects designed to identify Information like (Manufacturer, Model, Engine number, Bilt Date, Price)
- The Invoice information is created as Invoice ID, Total Price, Quantity, Unit Price, Purchase Date.

2. Automobile Object:

- The automobile information created like:
 - Record Name
 - Date Type
 - Display Format

Customization and Configuration:

- Maintain a comprehensive inventory of all vehicles, including details like make, model, year, color, price, and availability.
- Integrate with lead and opportunity management processes for test drives, quotes, and purchase orders.

Creating Tabs:

- Tabs serve as the primary interface for users to view, create, and manage records associated with custom or standard objects.
- Provides access to the inventory of vehicles, allowing users to quickly find details about models, pricing, and availability.
- A sales representative can access the Automobile Information tab to confirm vehicle availability while discussing options with a customer

Create a custom object from a spreadsheet

Define object and fields

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

CSV File Details

Encoding Format ¹ Values Separated By Field Label Source ☐ Enter manually ☒ Detect from row * Field Labels Row Import 0 rows of Data? ¹ ☐ No, skip import ☒ Yes, import data Record Name Field ¹

Fields 9 of 9 to import ☐ Hide mapped fields

IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS ¹	FIELD PREVIEW
✓ Manufacturer	✕	Manufacturer	Text	<input checked="" type="checkbox"/>	
✓ Model	✕	Model	Text	<input checked="" type="checkbox"/>	
✓ Engine Number	✕	Engine Number	Text	<input checked="" type="checkbox"/>	
✓ VIN	✕	VIN	Text	<input checked="" type="checkbox"/>	
✓ Total Cylinders	✕	Total Cylinders	Picklist	<input checked="" type="checkbox"/>	
✓ Colour	✕	Colour	Picklist	<input checked="" type="checkbox"/>	

Back ☒ ☐ ☐ ☐ ☐ Next

Create a custom object from a spreadsheet

Define object and fields

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

CSV File Details

Encoding Format ⓘ

Unicode (UTF8)

Values Separated By

Comma

Field Label Source

☐ Enter manually
☒ Detect from row

* Field Labels Row

1

Import 0 rows of Data? ⓘ

☒ No, skip import
☐ Yes, import data


Record Name Field ⓘ

Invoice ID

Fields 5 of 5 to import

☐ Hide mapped fields

IMPORT FILE FIELD NAME		SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS ⓘ	FIELD PREVIEW
✓ Invoice ID	×	Invoice ID	Text	<input checked="" type="checkbox"/>	
✓ Total Price	×	Total Price	Integer	<input checked="" type="checkbox"/>	
✓ Quantity	×	Quantity	Integer	<input checked="" type="checkbox"/>	
✓ Unit Price	×	Unit Price	Integer	<input checked="" type="checkbox"/>	
✓ Purchase Date	×	Purchase Date	Date	<input checked="" type="checkbox"/>	



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

Record Types

Related Lookup Filters

Search Layouts

Details

Description

API Name
Automobile_Information__c

Custom
✓

Singular Label
Automobile Information

Plural Label
Automobile Information

Enable Reports
✓

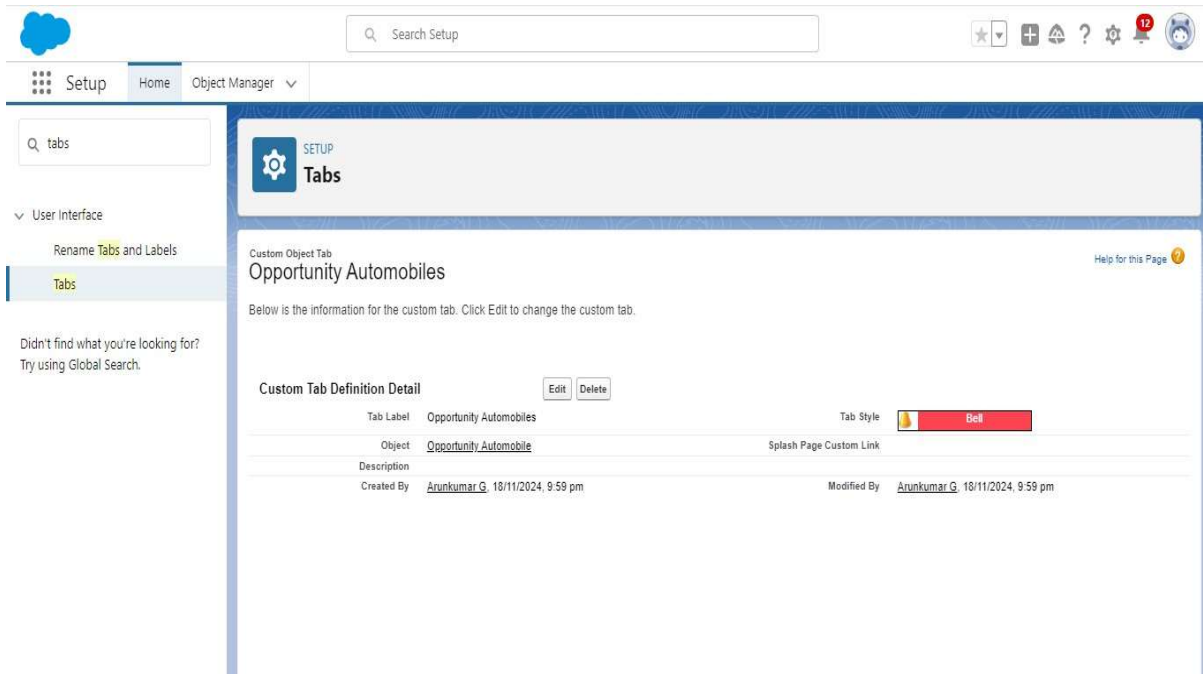
Track Activities
✓

Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

EditDelete



Task 2: Create a Lightning App

Creating a Lightning App in Salesforce provides users with a tailored, efficient, and visually appealing environment to perform their tasks. In an automobile sales context, such apps can streamline processes, enhance collaboration, and improve overall operational efficiency, ultimately leading to better customer service and higher sales performance is created.

- **Unified Experience:**
 - A **Sales App** for an automobile dealership can bring together tabs like **Leads**, **Opportunities**, **Automobile Inventory**, and **Invoices** in a single interface, reducing the need for users to switch between different views.
- **Role-Based Customization:**
 - **Sales Team App:** Focuses on leads, opportunities, and vehicle information.
 - **Finance App:** Includes invoices, payments, and financial reports.
 - **Service App:** Highlights post-sale service requests and customer history.

Lightning App Builder | App Settings | Pages | Sales Automobile Using Salesforce CRM | ? Help

App Settings

- App Details & Branding**
- App Options
- Utility Items (Desktop Only)
- Navigation Items
- User Profiles

App Details & Branding

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

App Details

* App Name ⓘ
Sales Automobile Using Salesforce CRM

* Developer Name ⓘ
Sales_Automobile_Using_Salesforce_CRM

Description ⓘ
Give a meaningful description

App Branding

Image ⓘ
Upload

Primary Color Hex Value ⓘ
#0070D2

Org Theme Options
☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview

SA Sales Automobile Using Sal...
Give a meaningful description

Task 3: Create Fields and Relationships:

Step 1: Create Opportunity Master Details

1. Navigate to **Setup** → **Object Manager** → **Opportunity Automobile**.
2. Click on Fields & Relationships
3. Enter the following details :
 - **Data Type:** Master Details
 - **Field Label:** Opportunity
 - **Field Name:** Opportunity

Step 2: Create Opportunity Automobile

1. Navigate to **Setup** → **Object Manager** → **Opportunity Automobile**
2. Click on Fields & Relationships
3. Enter the following details:
 - **Data Type:** Lookup Relationship
 - **Field Label:** Automob

Setup Home Object Manager

SETUP > OBJECT MANAGER

Opportunity Automobile

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

Automobile

Back to Opportunity Automobile

[Validation Rules](#)

Custom Field Definition Detail [Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

Field Information

Field Label	Automobile	Object Name	Opportunity Automobile
Field Name	Automobile	Data Type	Lookup
API Name	Automobile__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Anunkumar G. 18/11/2024, 10:32 pm	Modified By	Anunkumar G. 18/11/2024, 10:32 pm

Lookup Options

Related To	Automobile Information	Child Relationship Name	Opportunity_Automobiles
Related List Label	Opportunity Automobiles		
Required	<input type="checkbox"/>		

Step 3: Create Quantity Number Field

1. Navigate to **Setup** → **Object Manager** → **Opportunity Automobile**
2. Click on **Fields & Relationships**
3. Enter the following details:
 - **Data Type:** Numbers
 - **Field Label:** Automobile
 - **Field Name:** Automobile

SETUP > OBJECT MANAGER

Opportunity Automobile

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Opportunity Automobile Custom Field

Opportunity

Back to Opportunity Automobile

Validation Rules (0)

Custom Field Definition Detail

EditSet Field-Level SecurityView Field AccessibilityWhere is this used?

Field Information

Field Label	Opportunity	Object Name	Opportunity Automobile
Field Name	Opportunity	Data Type	Master-Detail
API Name	Opportunity__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Arunkumar G. 18/11/2024, 10:30 pm	Modified By	Arunkumar G. 18/11/2024, 10:30 pm

Master-Detail Options

Related To	Opportunity	Child Relationship Name	Opportunity_Automobiles
Related List Label	Opportunity Automobiles		
Sharing Setting	Read/Write: Allows users with at least Read/Write access to the Master record to create, edit, or delete related Detail records.		
Reparentable Master Detail	<input type="checkbox"/>		

Lookup Filter

No lookup filters defined.

SetupHomeObject Manager

SETUP > OBJECT MANAGER

Opportunity Automobile

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

Opportunity Automobile Custom Field

Quantity

Back to Opportunity Automobile

Validation Rules (0)

Custom Field Definition Detail

EditSet Field-Level SecurityView Field AccessibilityWhere is this used?

Field Information

Field Label	Quantity	Object Name	Opportunity Automobile
Field Name	Quantity	Data Type	Number
API Name	Quantity__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Arunkumar G. 18/11/2024, 10:34 pm	Modified By	Arunkumar G. 18/11/2024, 10:34 pm

General Options

Required	<input checked="" type="checkbox"/>
Unique	<input type="checkbox"/>

Step 4: Create Formula Field

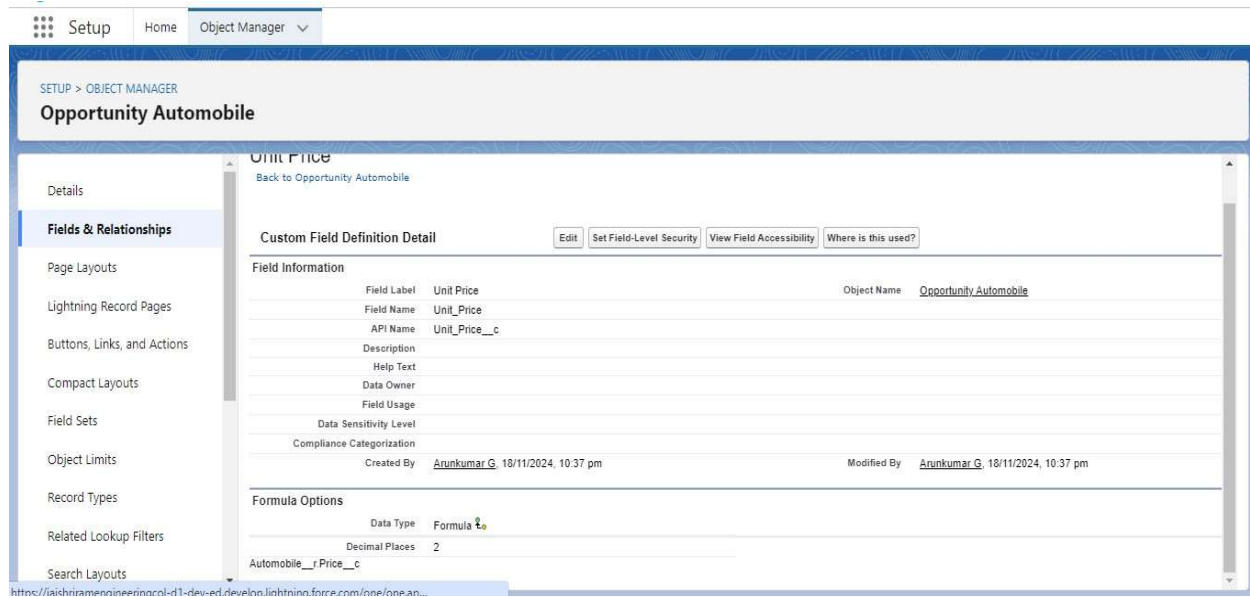
1. Navigate to **Setup → Object Manager→ Opportunity Automobile**
2. Click on Fields & Relationships
3. Enter the following details:
 - **Data Type:** Formula
 - **Field Label:** Unit Price
 - **Field Name:** Unit Price

Step 5: Update Fields in Invoice Object

1. Navigate to **Setup → Object Manager→ Opportunity Automobile**
2. Click on Fields & Relationships
3. Enter the following details:
 - **Data Type:** Auto Number
 - **Display Format:** I-{0000}
 - **Starting Nuber:** 1

Step 6: Create Another Field

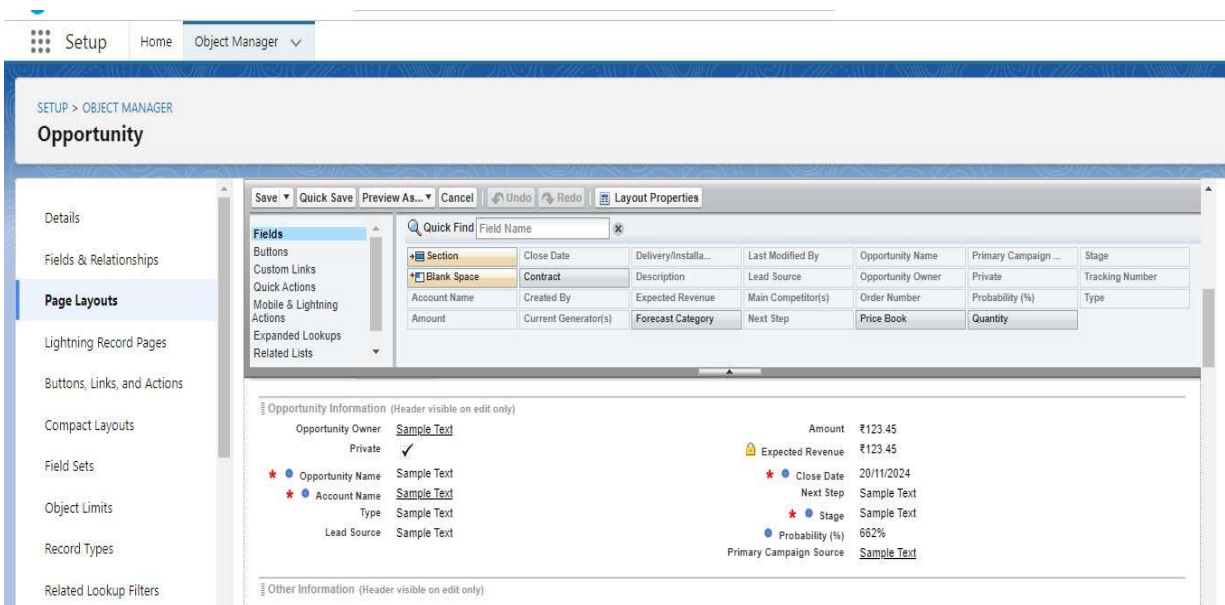
1. Navigate to **Setup → Object Manager→ Opportunity Automobile**
2. Click on Fields & Relationships
3. Enter the following details:
 - **Data Type:** Master Details Relationship
 - **Field Name:** Opportunity
 - **Object:** Opportunity



Task 4: Page Layouts:

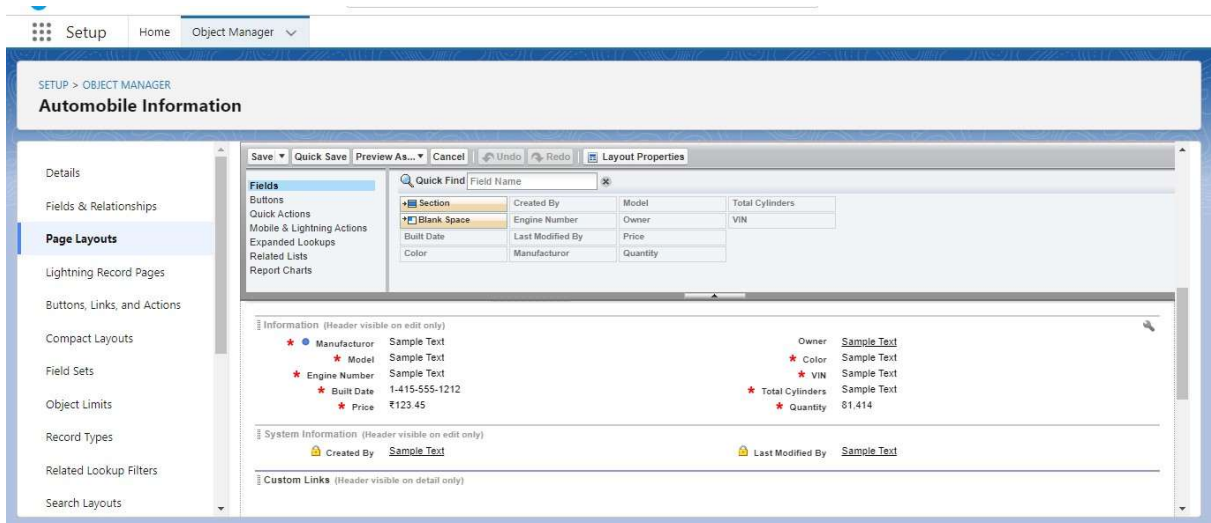
Step 1: Edit Opportunity Object Page Layout

1. Go to Setup → Object Manager → Search → Opportunity Layout.
2. Next Page Layouts → Opportunity Layouts.



Step 2: Edit Automobile Information Page Layout

1. Go to **Setup** → **Object Manager** → **Search** → **Automobile Information**.
2. Next **Page Layouts** → **Automobile Information**.



Task 5: Creating Apex Triggers

Step 1: Trigger for Automobile Quantity

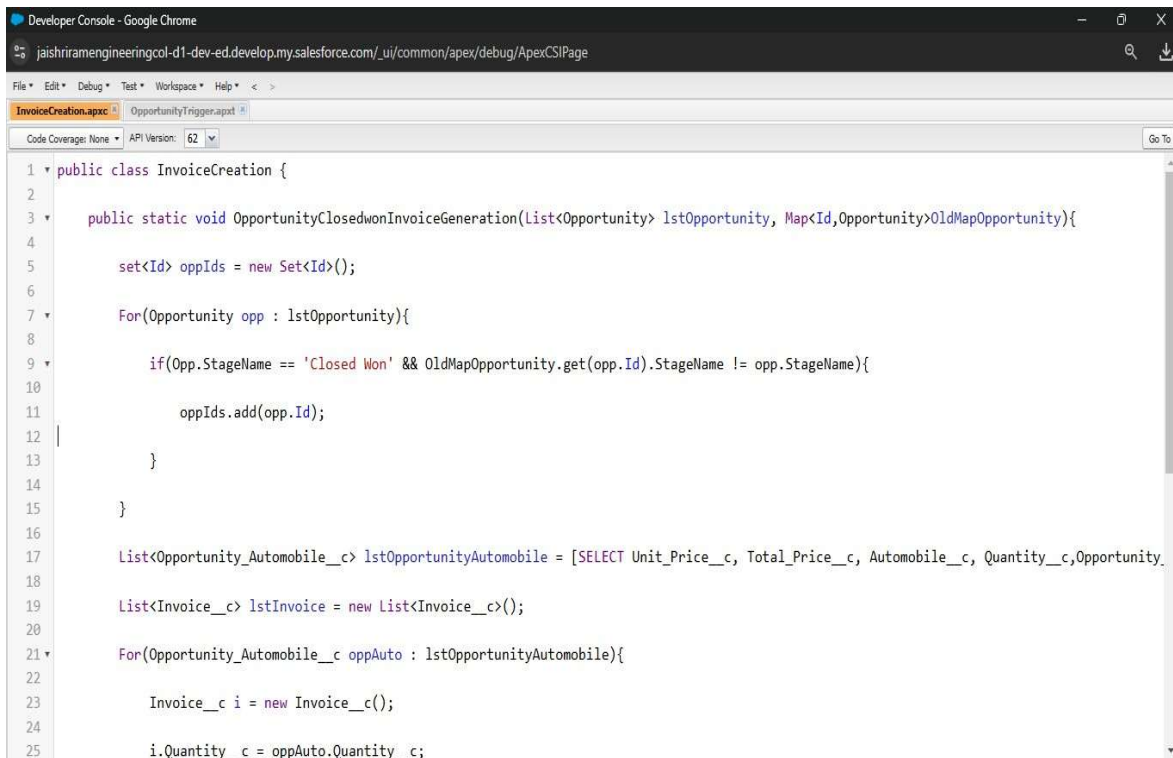
- Navigate to the gear icon on top right corner, click on Developer Console.
- Find FILE in tool bar and click on it to create new Apex Class with name “OpportunityHandler Class”.
 - Add the Java program to the Apex class & create a Trigger Handler class.
 - Create an Apex Trigger & type the code.

Step 2: Trigger for Opportunity-Automobile Error

- Same as the previous step create an Apex class with class name “OpportunityAutomobileHandler”.
- Also create a new Trigger Handler & new Apex Trigger.

Step 3: Invoice Creation Trigger

- Follow the last step and create another new Apex class with “InvoiceCreation” as the class name.
- It also has its own Trigger Handler with unique Trigger as code.

The screenshot shows the Salesforce Developer Console in Google Chrome. The browser address bar displays the URL: jaishriramengineeringcol-d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The console window has tabs for 'InvoiceCreation.apex' and 'OpportunityTrigger.apex'. The 'InvoiceCreation.apex' tab is active, showing the following Apex code:

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

Task 6: Creation of LCW Component

Step 1: Apex class to get Invoice

- New Apex class with class name “OppertunityInvoiceswithLWC” must be created

Step 2: Install Salesforce CLI

- 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.

```

C:\> Command Prompt
C:\sf\bin>sfdx
The Salesforce CLI

VERSION
@salesforce/cli/2.66.7 win32-x64 node-v22.10.0

USAGE
$ sf [COMMAND]

TOPICS
alias          Use the alias commands to manage your aliases.
analytics      Work with analytics assets.
apex           Use the apex commands to create Apex classes, execute anonymous blocks, view your logs, run Apex
               tests, and view Apex test results.
api            Commands to interact with API calls.
cmdt           Generate a field for a custom metadata type based on the provided field type.
community      Create an Experience Cloud site using a template.
config         Commands to configure Salesforce CLI.
data           Manage records in your org.
deploy         Deploy a Salesforce Function to an org from your local project.
dev            Audit messages in a plugin's messages directory to locate unused messages and missing messages that
               have references in source code.
env            Add a Heroku user as a collaborator on this Functions account, allowing them to attach Heroku add-ons
force          Legacy commands for backward compatibility.
generate       Create a Salesforce Function with basic scaffolding specific to a given language.
info           Access Salesforce CLI information from the command line.
lightning      Work with Lightning Web and Aura components.
login          Log in to Salesforce Functions.
logout         Log out of your Salesforce Functions account.
org            Commands to create and manage orgs and scratch org users.
package        Commands to develop and install unlocked packages and managed 2GP packages.
package1       Commands to develop first-generation managed and unmanaged packages.
plugins        list installed plugins
project        Work with projects, such as deploy and retrieve metadata.
run            Send a cloudevent to a function.
scanner        Add custom rules to Salesforce Code Analyzer's registry to run them along with the built-in rules.
schema         Generate metadata files.
subject        Commands to interact with Salesforce objects.
static-resource Work with static resources.
update         update the sf CLI
version
visualforce    Work with Visualforce components.

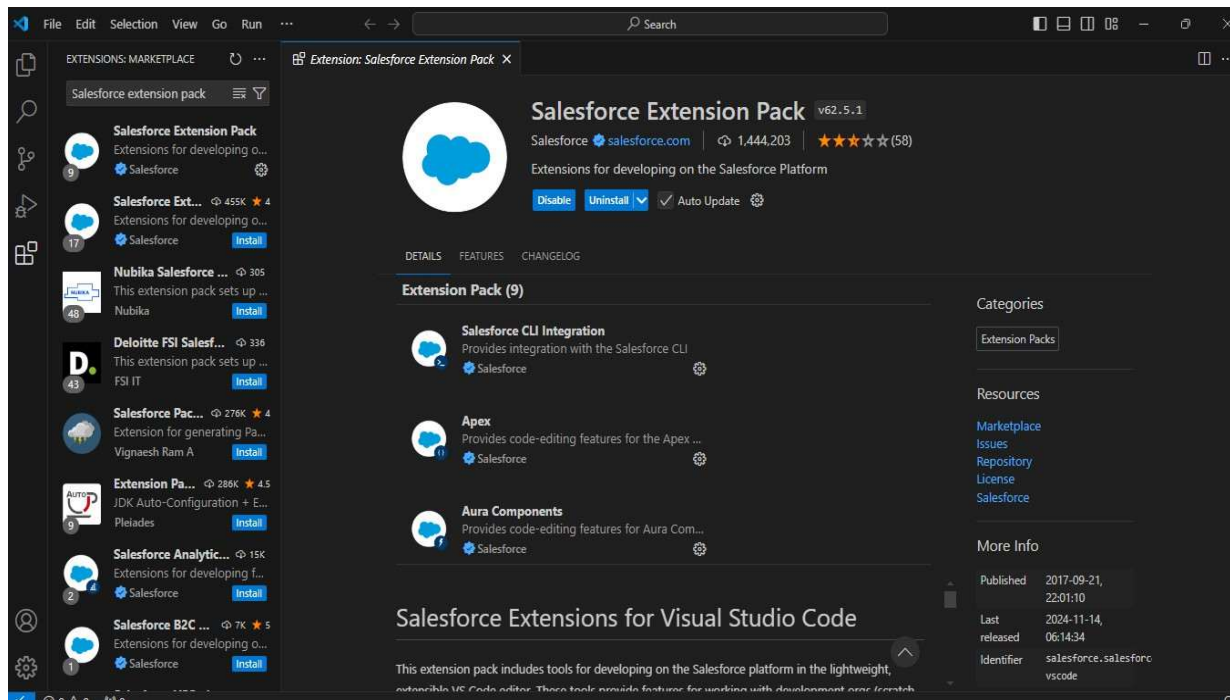
```

Step 3: 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.

Step 4: Install Salesforce Extension Packs

- Go to extensions in VS Code & install the “Salesforce Extension”.
- The Salesforce extensions help us analyze or deploy a code faster.
- There are many packs available so be careful to install the proper salesforce packs.



Step 5: Create a Project in VS Code

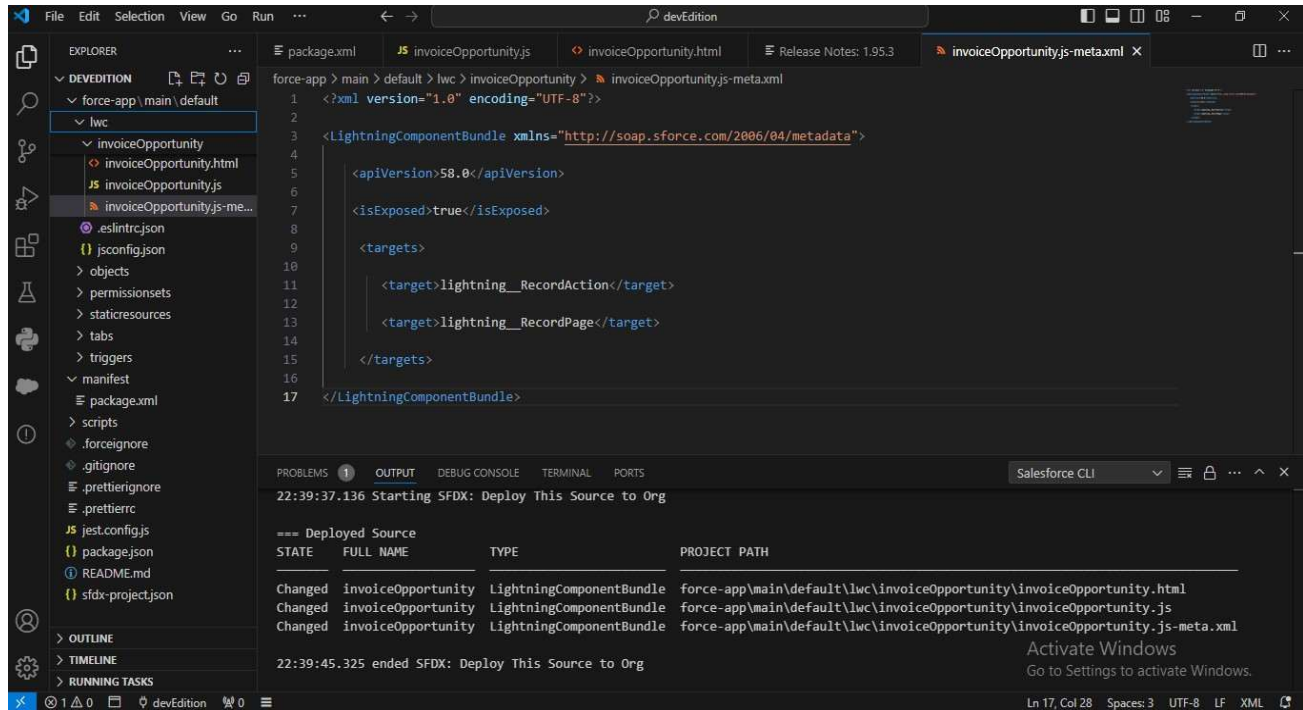
- Press CTRL + SHIFT + P, type sfdx: create.
- Select SFDX: Create Project with Manifest.
- Select the Standard project template
- Type a project name and Click Enter.
- Type a project name and Click Enter.

Step 6: 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.

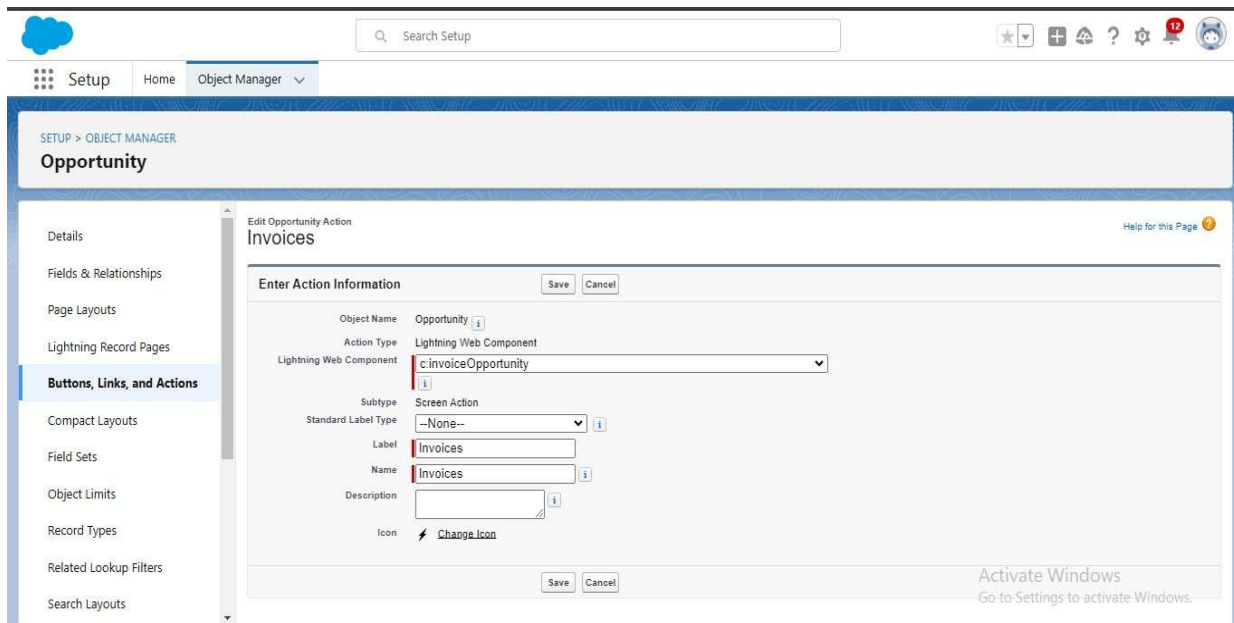
Step 7: Creating Lightning Web Component

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



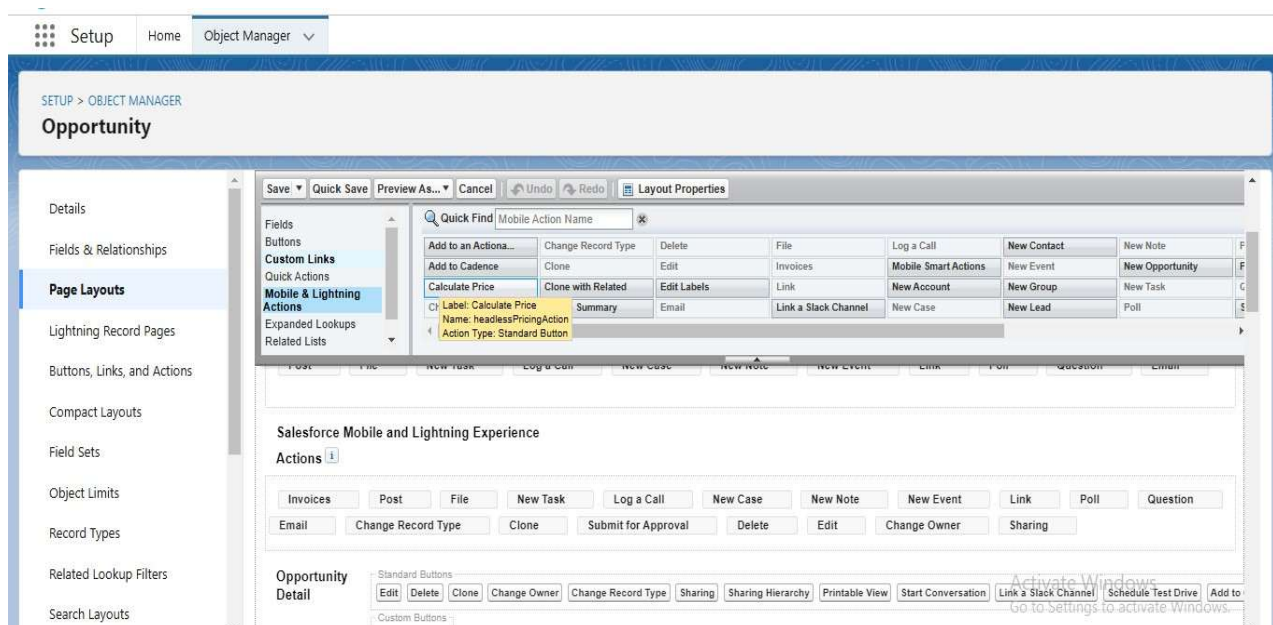
Step 8: Create Button to Add on Opportunity

1. Click on Object Manager
2. Search Opportunity and Click on it .
3. click on the New Action.
4. InvoiceOpportunity component
 - a. Label :- Invoices, Name :- Invoices



Step 9: 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. Drag and Drop the Invoice into Salesforce Mobile and Lightning Experience Actions.



Task 7: Apex Schedulers

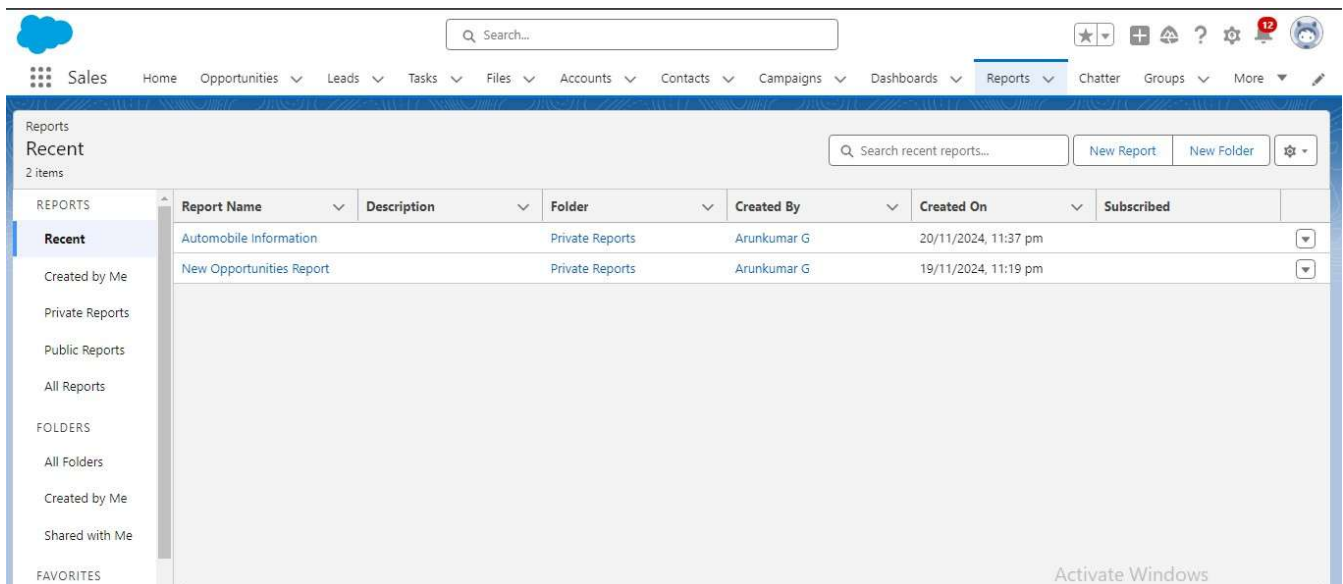
Step 1: Delete opportunity Schedule Class

- Click on the Developer console. Now you will see a new console window.
- In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
- Name the class as “DeleteClosedLostOpportunities ”
- Click on Schedule Apex and enter the Job name.
- Job Name : DeleteOpportunitySchedule

The screenshot shows the Salesforce 'Apex Classes' setup page. On the left is a navigation sidebar with a search bar containing 'apex'. The sidebar lists categories: Email (Apex Exception Email), Custom Code (Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers), Environments, and Jobs (Apex Flex Queue, Apex Jobs). The 'Apex Classes' item is selected. The main content area is titled 'Apex Classes' and contains a 'Schedule Apex Execution' form. The form has a 'Save' button and a 'Cancel' button. It includes fields for 'Job Name' (DeleteOpportunitySchedule) and 'Apex Class' (DeleteClosedLostOpportunit). Below these is a 'Schedule Using' section with 'Schedule Builder' selected and 'Cron Expression' unselected. The 'Schedule Apex Execution' section shows 'Frequency' set to 'Weekly' (with 'Monthly' as an option). A dropdown menu for 'Recurs every week on' is open, showing days of the week with checkboxes: Sunday (unchecked), Monday (checked), Tuesday (unchecked), Wednesday (checked), Thursday (unchecked), Friday (unchecked), and Saturday (unchecked). At the bottom, there are date pickers for 'Start' (20/11/2024) and 'End' (20/12/2024), and a 'Preferred Start Time' dropdown set to '10:00 am'. A note at the bottom states 'Exact start time will depend on job queue activity.' An 'Activate Windows' watermark is visible in the bottom right corner.

Step 2: Report on Automobile Information

- Create a report with a report type: “Automobile Information”.
- Create a Report by using “Opportunities with Opportunity Automobiles and Automobile” Report Type.
- Types of Reports:
 - Tabular
 - Summary
 - Matrix
 - Joined Reports



The screenshot displays the Salesforce Reports interface. At the top, there is a navigation bar with the Salesforce logo, a search bar, and various utility icons. Below this is a main navigation menu with options like Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports (selected), Chatter, Groups, and More. The 'Reports' section is active, showing a 'Recent' view with a search bar and buttons for 'New Report' and 'New Folder'. A table lists the recent reports:

REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Automobile Information		Private Reports	Arunkumar G	20/11/2024, 11:37 pm	<input type="checkbox"/>
Created by Me	New Opportunities Report		Private Reports	Arunkumar G	19/11/2024, 11:19 pm	<input type="checkbox"/>

On the left side of the table, there is a sidebar with a list of report folders: Recent, Created by Me, Private Reports, Public Reports, All Reports, FOLDERS, All Folders, Created by Me, Shared with Me, and FAVORITES. At the bottom right of the interface, there is a watermark that says 'Activate Windows'.

The screenshot shows the Salesforce Reports interface. The top navigation bar includes the Salesforce logo, a search bar, and various utility icons. The main navigation menu is visible, with 'Reports' selected. The 'Recent' reports section is active, displaying a list of reports. The left sidebar shows the 'REPORTS' section with 'Recent' selected. The main content area shows a table of recent reports.

Report Name	Description	Folder	Created By	Created On	Subscribed
Automobile Information		Private Reports	Arunkumar G	20/11/2024, 11:37 pm	
New Opportunities Report		Private Reports	Arunkumar G	19/11/2024, 11:19 pm	

Step 3: Create Sales Dashboard

- Go to the app ? click on the Dashboards tabs.
- Give a Name and click on Create.
- Name : Automobile Sales
- Select add component & Report



Dashboard

Dashboard 1

⚠ Last refreshed 1 day ago. Refresh this dashboard to see the latest data.

As of 19-Nov-2024, 11:34 pm-Viewing as Arunkumar G

New Opportunities Report

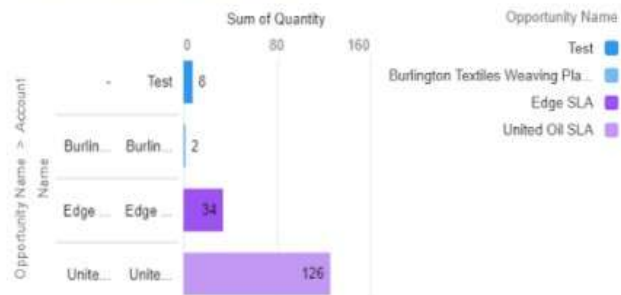
Account Na...	O...	Op...	Opportunity Name	Stage
Edge Communicatio ns	-	Arunkumar G	Edge Emergency Generator	Closed V
Edge Communicatio ns	-	Arunkumar G	Edge Emergency Generator	Id. Decis
GenePoint	-	Arunkumar	GenePoint Lab Generators	Id. Decis
GenePoint	-	Arunkumar	GenePoint SLA	Closed V

[View Report \(New Opportunities Report\)](#)



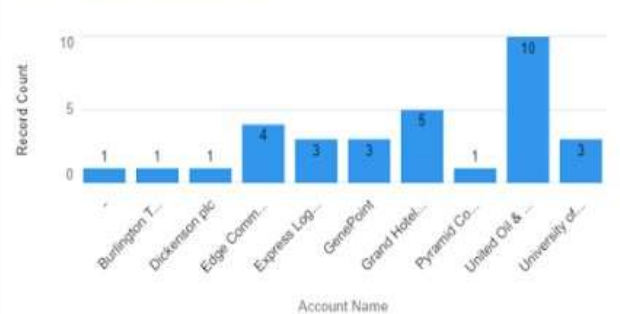
Sales Dashboard

Opportunity With Automobile Data



[View Report \(Opportunity With Automobile Data\)](#)

Opportunity Closed Won Report



[View Report \(Opportunity Closed Won Report\)](#)