# WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence

**Business Name:** WhatNext Vision Motors

**Industry**: Automotive & Mobility Technology (Electric Vehicles / Future Mobility Solutions)

## Project Overview

WhatsNext Vision Motors is implementing a Salesforce CRM solution to revolutionize its customer ordering, sales, and service management processes. This project focuses on enhancing the customer experience through features like intelligent dealer suggestions, real-time stock validation, and automated order status updates. By centralizing vehicle, dealer, and customer data and automating key workflows, the system will streamline operations, reduce manual intervention, and provide clear, real-time communication to customers. Additionally, automated batch processes will keep inventory and order statuses up to date, improving order accuracy and operational efficiency.

## Project Objectives

The Main Objective of the project is to implement a Salesforce CRM solution that centralizes vehicle, dealer, and customer management while automating sales, order processing, and service workflows to enhance operational efficiency and customer experience.

**1.Enhance Customer Ordering Experience**

- Suggest the nearest dealer to customers based on their location.
- Prevent orders for vehicles that are out of stock.

**2.Streamline Operations**

- Automate bulk order status updates based on stock availability**.**
- Reduce manual effort with scheduled batch jobs for inventory and order processing**.**

**3.Improve Communication & Transparency**

- Provide real-time order status updates to customers.
- Send automated reminders for test drives and notifications for stock updates.

**4.Leverage Salesforce Capabilities**

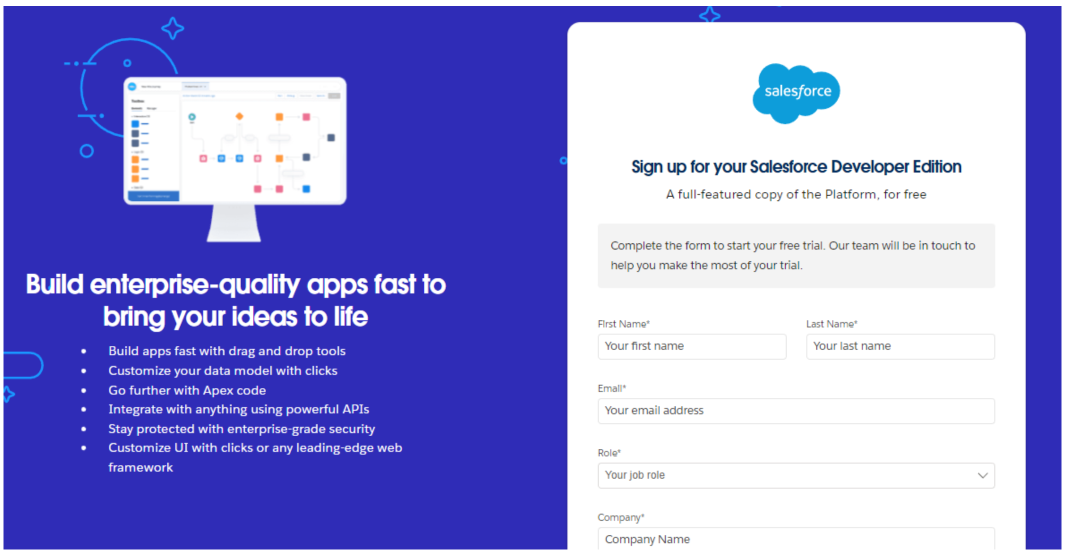
- Implement custom objects, fields, and relationships for vehicles, customers, dealers, and orders.
- Use Apex triggers, record-triggered flows, and batch Apex for enforcing business rules and process automation.

## Phase 1: Salesforce Credentials Creation

### Activity 1: Creating Developer Account

Creating a developer org in salesforce.

1. Go to https://developer.salesforce.com/signup
2. On the sign up form, enter the following details:


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

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

Sign up for your Salesforce Developer Edition

A full-featured copy of the Platform, 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* Your first name
Last Name* Your last name
Email* Your email address
Role* Your job role
Company* Company Name

1) First name & Last name
2) Email
3) Role: Developer
4) Company: College Name
5) Country/Region: India
6) Postal Code: pin code
7) Username: should be a combination of your name and company

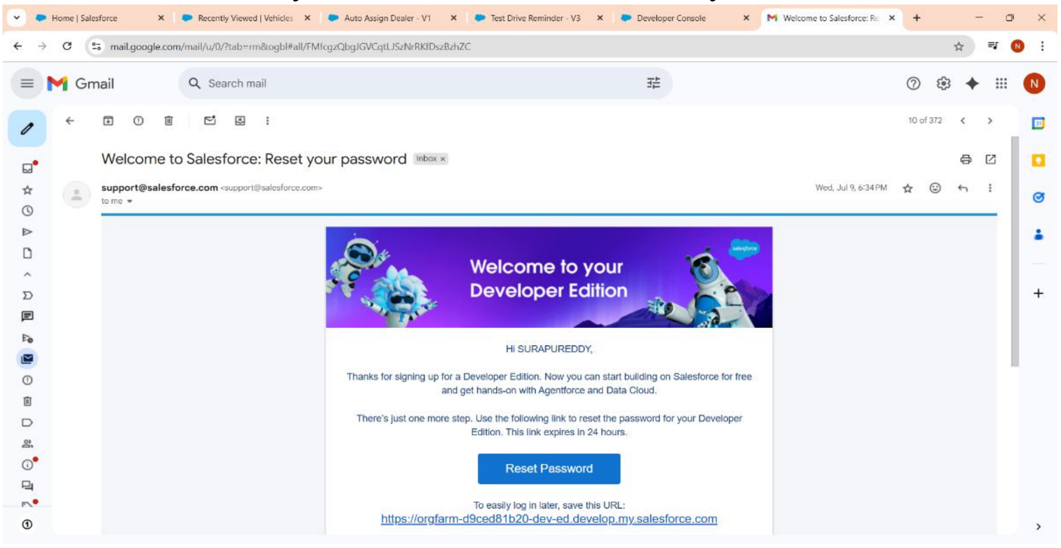
This need not be an actual email id, you can give anything in the format: username@organization.com

Click on sign me up after filling these.

### Activity 2: Verify Account

Account Activation

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



Welcome to Salesforce: Reset your password

support@salesforce.com <support@salesforce.com>

to me

Wed, Jul 9, 6:34 PM

Welcome to your Developer Edition

Hi SURAPUREDDY,

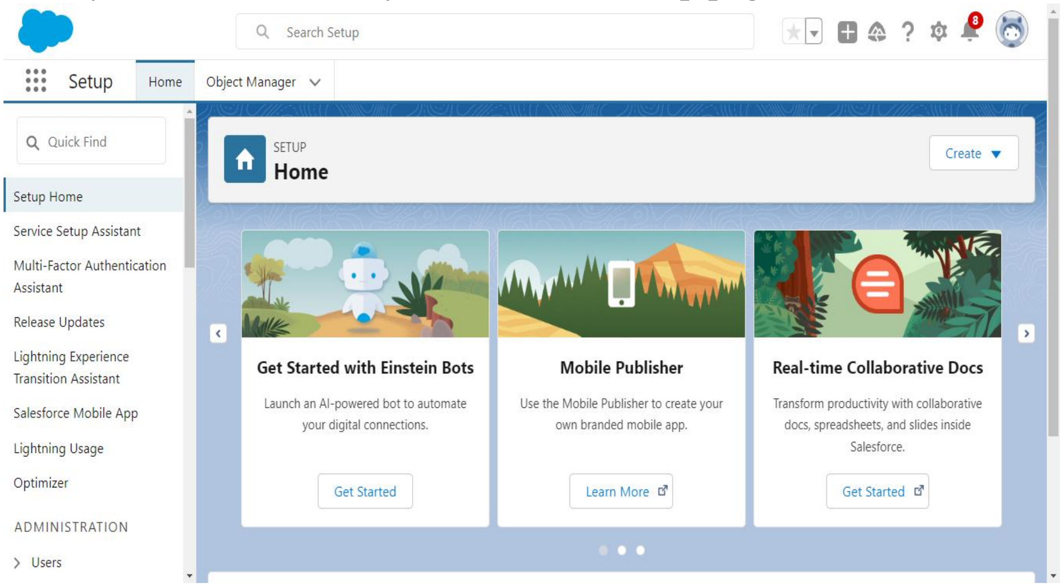
Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

There's just one more step. Use the following link to reset the password for your Developer Edition. This link expires in 24 hours.

Reset Password

To easily log in later, save this URL:
https://orgfarm-d9ced81b20-dev-ed.develop.my.salesforce.com

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



Get Started with Einstein Bots

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

Get Started

Mobile Publisher

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

Learn More

Real-time Collaborative Docs

Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Get Started

# Phase 2: Data Modeling

## Activity 1: Data Management-Objects

### 1.1 Vehicle__c

- Stores vehicle details
- Fields: Name, Model, Stock Quantity, Price, Dealer, Status

### 1.2 Vehicle_Dealer__c

- Stores authorized dealer info

- Fields: Dealer Name, Dealer Location, Dealer Code, Phone, Email

**1.3 Vehicle_Customer__c**

- Stores customer details
- Fields: Customer, Vehicle, Order_Date, Status

**1.4 Vehicle_Order__c**

- Tracks vehicle purchases
- Fields: Customer Name, Phone, Email, Address, Preferred Vehicle Type

**1.5 Vehicle_Test_Drive__c**

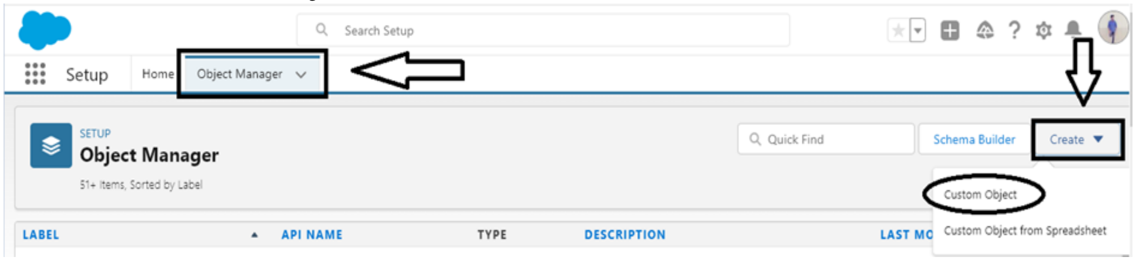
- Tracks test drive bookings
- Fields: Customer, Vehicle, Test Drive Date, Status

**1.6 Vehicle_Service_Request__c**

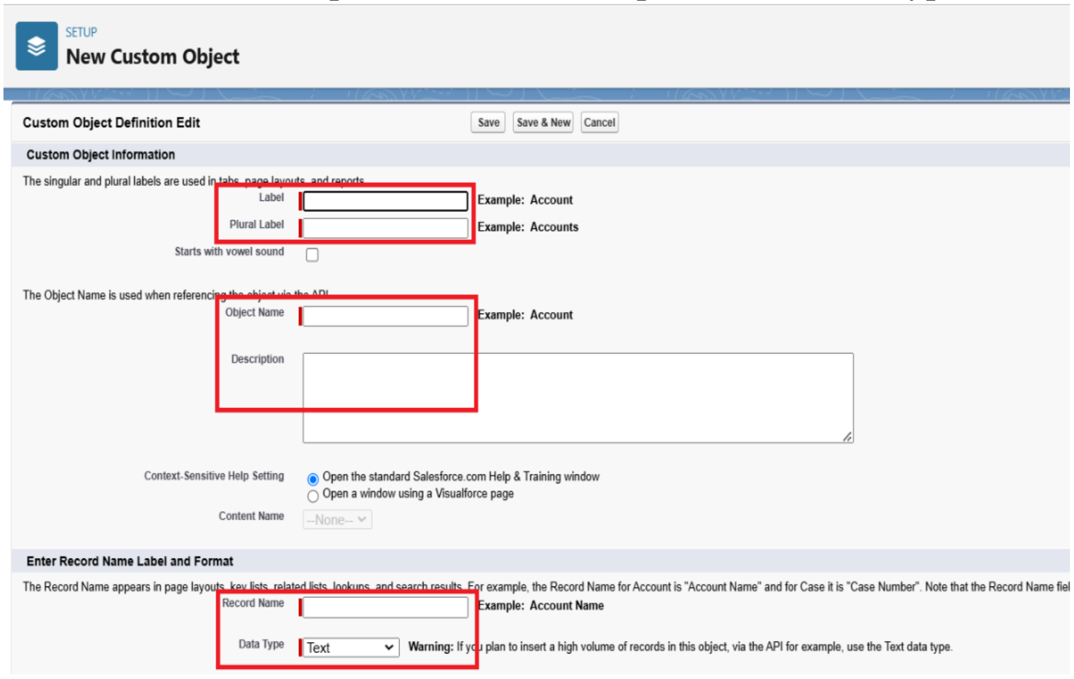
- Tracks vehicle servicing requests
- Fields: Customer, Vehicle, Service Date, Issue Description, Status

**Steps followed:**

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



- Enter the label name, plural label name, report name & datatype.



SETUP

New Custom Object

Custom Object Definition Edit

Save | Save & New | Cancel

Custom Object Information

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

| | | |
|---|---|---|
| Label | | Example: Account |
| Plural Label | | Example: Accounts |
| Starts with vowel sound | ☐ | |

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

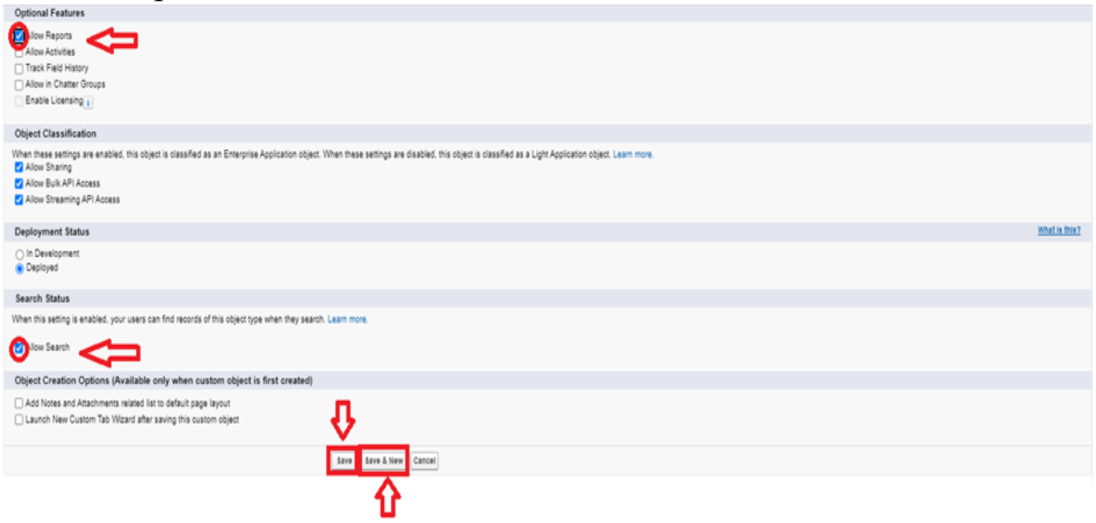
| | | |
|---|---|---|
| Object Name | | Example: Account |
| Description | | |
| Context-Sensitive Help Setting | ● Open the standard Salesforce.com Help & Training window<br>○ Open a window using a Visualforce page | |
| Content Name | --None-- | |

Enter Record Name Label and Format

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

| | | |
|---|---|---|
| Record Name | | Example: Account Name |
| Data Type | Text | Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type. |

- Allow Reports→Allow Search→Save.



Optional Features

- ✓ Allow Reports
- ☐ Allow Activities
- ☐ Track Field History
- ☐ Allow in Chatter Groups
- ☐ Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.

- ✓ Allow Sharing
- ✓ Allow Bulk API Access
- ✓ Allow Streaming API Access

Deployment Status

What is this?

- ○ In Development
- ● Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. Learn more.

- ✓ Allow Search

Object Creation Options (Available only when custom object is first created)

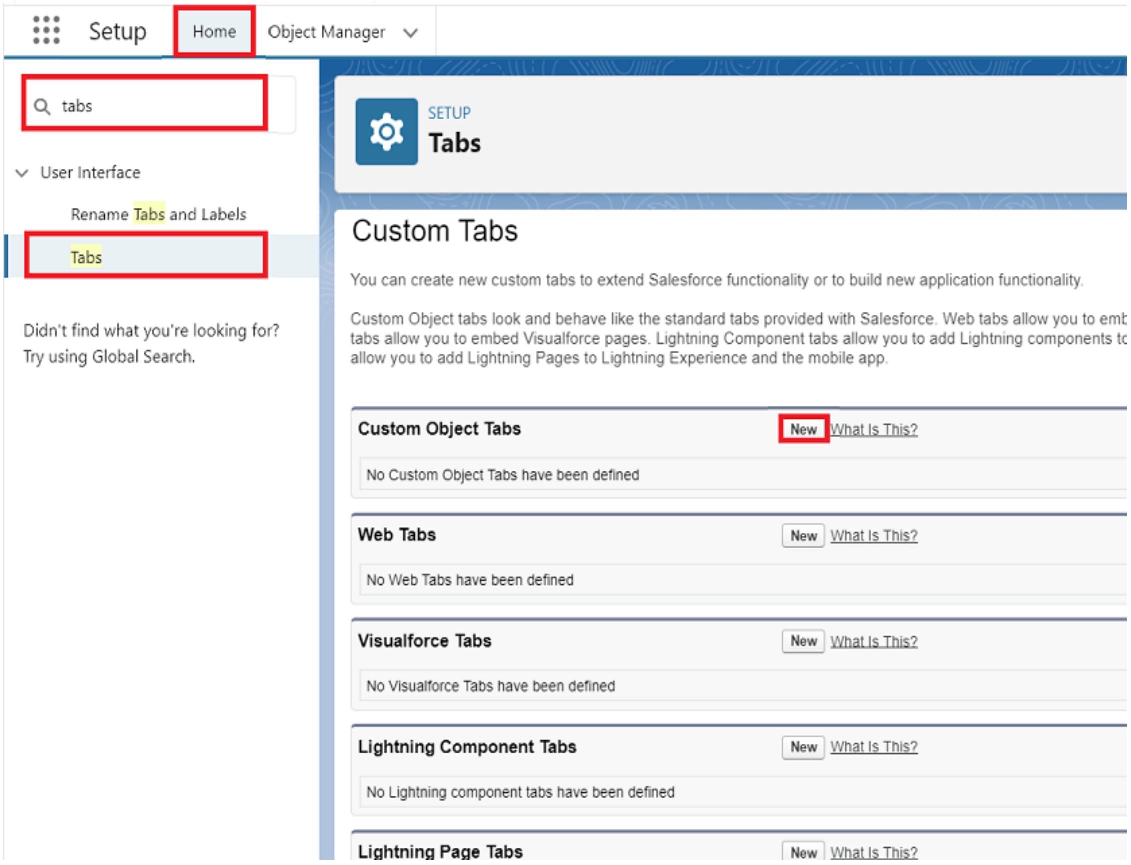
- ☐ Add Notes and Attachments related list to default page layout
- ☐ Launch New Custom Tab Wizard after saving this custom object

Save | Save & New | Cancel

## Activity 2: Data Management-Tabs

Creating a Custom Tab for Vehicles, Vehicle Customers, Vehicle Dealers, Vehicle Orders, Vehicle Test Drives, Vehicle Service Requests.

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



Custom Tabs

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

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to emt
tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components tc
allow you to add Lightning Pages to Lightning Experience and the mobile app.

**Custom Object Tabs** New What Is This?
No Custom Object Tabs have been defined

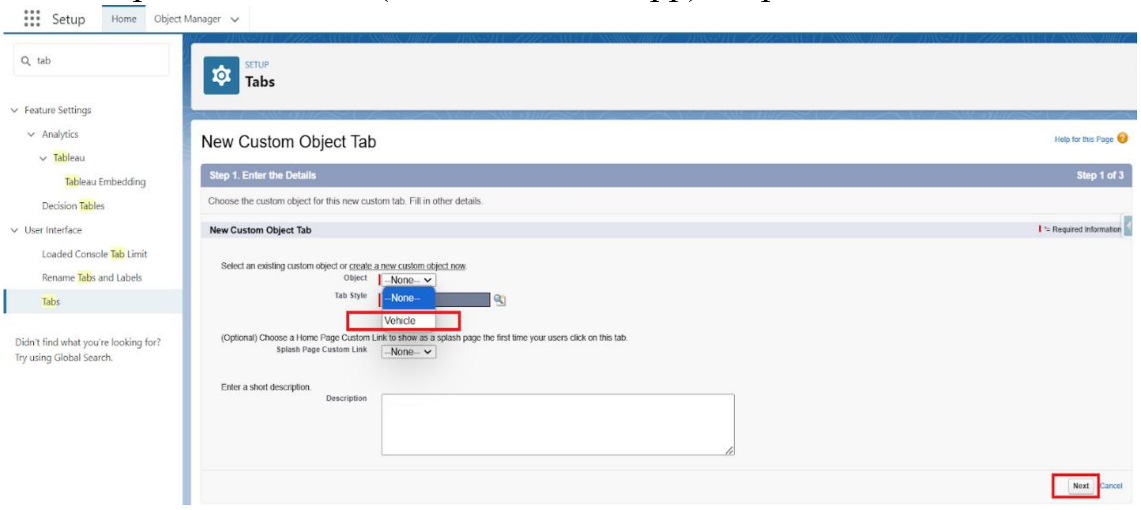
**Web Tabs** New What Is This?
No Web Tabs have been defined

**Visualforce Tabs** New What Is This?
No Visualforce Tabs have been defined

**Lightning Component Tabs** New What Is This?
No Lightning component tabs have been defined

**Lightning Page Tabs** New What Is This?

- Select Object (Vehicle) → Select any tab style → Assign the tab to relevant profiles→ Next (Add to Custom App) keep it as default → Save.



New Custom Object Tab

Step 1. Enter the Details Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

New Custom Object Tab | = Required Information

Select an existing custom object or create a new custom object now.
Object --None--
Tab Style --None--
Vehicle

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.
Splash Page Custom Link --None--

Enter a short description.
Description

## Activity 3: Data Management-Fields

### Fields & Relationships

### 1. Vehicle__c (Custom Object)

- Vehicle_Name__c (Text)

- Vehicle_Model__c (Picklist: Sedan, SUV, EV, etc.)
- Stock_Quantity__c (Number)
- Price__c (Currency)
- Dealer__c (Lookup to Dealer__c)
- Status__c (Picklist: Available, Out of Stock, Discontinued)

**2.Vehicle_ Dealer__c (Custom Object)**

- Dealer_Name__c (Text)
- Dealer_Location__c (Text)
- Dealer_Code__c (Auto Number)
- Phone__c (Phone)
- Email__c (Email)

**3. Vehicle_Order__c (Custom Object)**

- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Order_Date__c (Date)
- Status__c (Picklist: Pending, Confirmed, Delivered, Canceled)

**4. Vehicle_Customer__c (Custom Object)**

- Customer_Name__c (Text)
- Email__c (Email)
- Phone__c (Phone)
- Address__c (Text)
- Preferred_Vehicle_Type__c (Picklist: Sedan, SUV, EV, etc.)

**5. Vehicle_Test_Drive__c (Custom Object)**

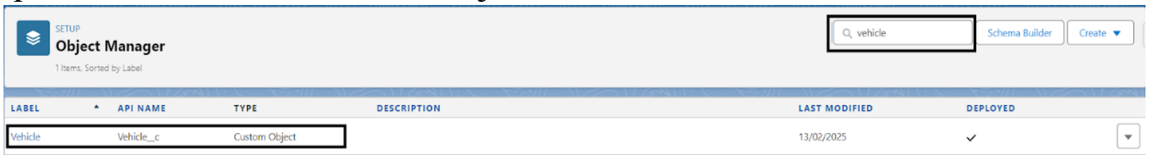
- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Test_Drive_Date__c (Date)
- Status__c (Picklist: Scheduled, Completed, Canceled)

**6. Vehicle_Service_Request__c (Custom Object)**

- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Service_Date__c (Date)
- Issue_Description__c (Text)
- Status__c (Picklist: Requested, In Progress, Completed)

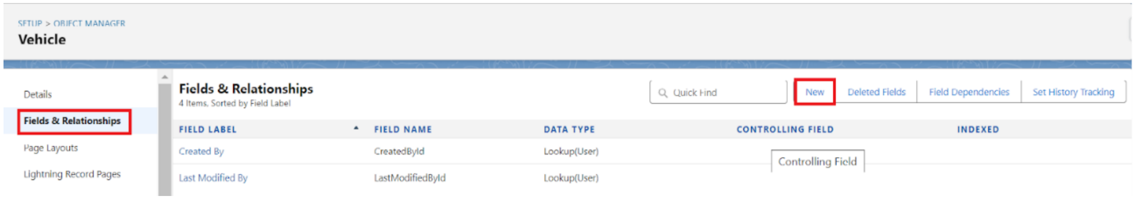
**Steps followed:**

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



| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED |
|---|---|---|---|---|---|
| Vehicle | Vehicle__c | Custom Object | | 13/02/2025 | ✓ |

- Now click on "Fields & Relationships" → New



| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|---|---|---|---|---|
| Created By | CreatedById | Lookup(User) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |

- Select Data type
- Field Label,Field Name
- Click on Next → Next → Save and new.

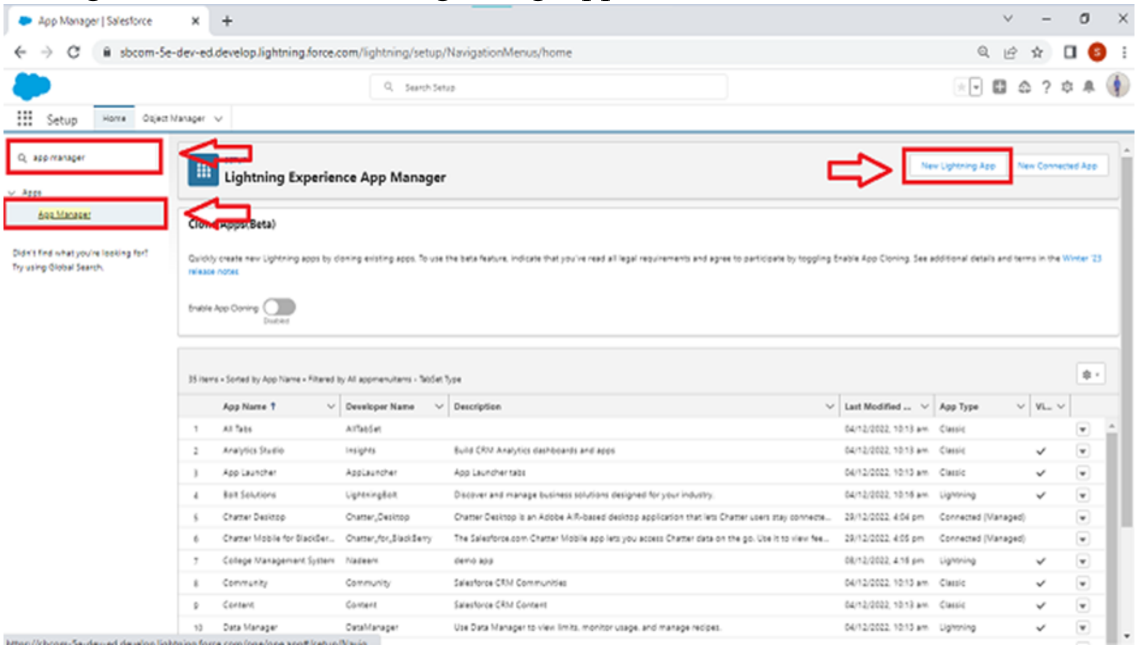
## Phase 3: App Building

### Activity 1: Data Management-App Manager

### Create a Lightning App

To create a lightning app page:

1. Go to setup page → search "app manager" in quick find → select "app manager" → click on New lightning App.


App Manager | Salesforce
sbcom-5e-dev-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home
Search Setup
Setup
Home
Object Manager
app manager
Apps
App Manager
Didn't find what you're looking for? Try using Global Search.
Lightning Experience App Manager
New Lightning App
New Connected App
Clone Apps (Beta)
Enable App Cloning
Disabled

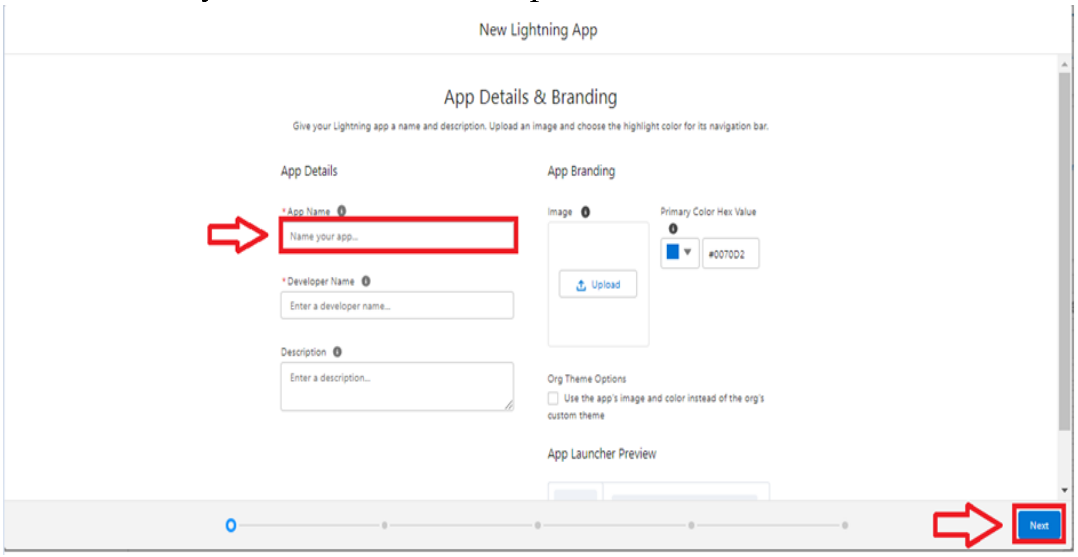
Quickly create new Lightning apps by cloning existing apps. To use the beta feature, indicate that you've read all legal requirements and agree to participate by toggling Enable App Cloning. See additional details and terms in the Winter '23 release notes

35 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type

| | App Name | Developer Name | Description | Last Modified ... | App Type | Vi... |
|---|---|---|---|---|---|---|
| 1 | All Tabs | AllTabSet | | 04/12/2022, 10:13 am | Classic | |
| 2 | Analytics Studio | Insights | Build CRM Analytics dashboards and apps | 04/12/2022, 10:13 am | Classic | ✓ |
| 3 | App Launcher | AppLauncher | App Launcher tabs | 04/12/2022, 10:13 am | Classic | ✓ |
| 4 | Bolt Solutions | LightningBolt | Discover and manage business solutions designed for your industry. | 04/12/2022, 10:16 am | Lightning | ✓ |
| 5 | Chatter Desktop | Chatter_Desktop | Chatter Desktop is an Adobe AIR-based desktop application that lets Chatter users stay connecte... | 29/12/2022, 4:04 pm | Connected (Managed) | |
| 6 | Chatter Mobile for BlackBer... | Chatter_for_BlackBerry | The Salesforce.com Chatter Mobile app lets you access Chatter data on the go. Use it to view fee... | 29/12/2022, 4:05 pm | Connected (Managed) | |
| 7 | College Management System | Nadeem | demo app | 08/12/2022, 4:16 pm | Lightning | ✓ |
| 8 | Community | Community | Salesforce CRM Communities | 04/12/2022, 10:13 am | Classic | ✓ |
| 9 | Content | Content | Salesforce CRM Content | 04/12/2022, 10:13 am | Classic | ✓ |
| 10 | Data Manager | DataManager | Use Data Manager to view limits, monitor usage, and manage recipes. | 04/12/2022, 10:13 am | Lightning | ✓ |

2. Fill the app name in app details and branding as follow
   - App Name: WhatNext Vision Motors

- Developer Name: this will auto populated
- Description: Give a meaningful description
- Image: optional (if you want to give any image you can otherwise not mandatory)
- Primary color hex value: keep this default


New Lightning App

App Details & Branding

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

App Details

* App Name: Name your app...

* Developer Name: Enter a developer name...

Description: Enter a description...

App Branding

Image

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

3. Then click Next → (App option page) keep it as default → Next → (Utility Items) keep it as default → Next.
4. To Add Navigation Items:
    - Search the items in the search bar (Vehicle, Dealer, Customer, Order, Test Drive, Service Request, Reports, Dashboard) from the search bar and move it using the arrow button → Next.

Note: select the custom object which we have created in the previous activity.

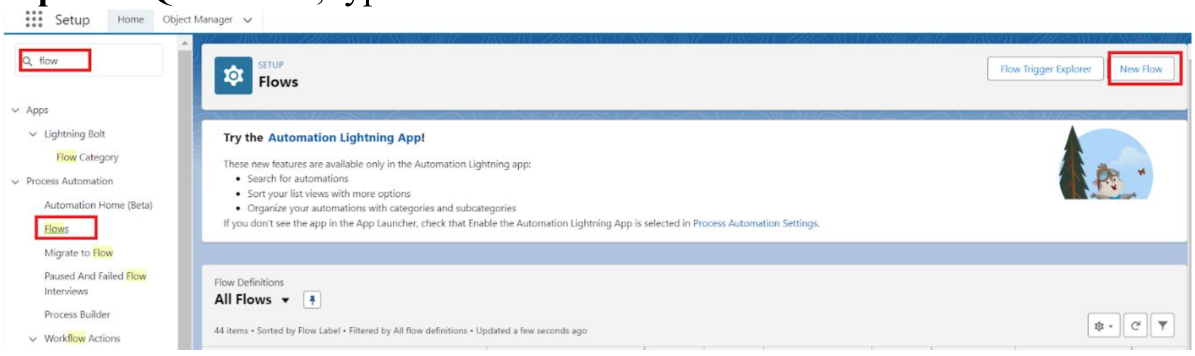
5. To Add User Profiles:
    - Search profiles (System administrator) in the search bar → click on the arrow button → save & finish.

## Phase 4: Automation

### Activity 1: Flow Creation

Creating a record triggered flow to assign nearest dealer to the customer's location

**Step 1**: In Quick Find, type Flows and click on Flows. Click New Flow.



Try the Automation Lightning App!

These new features are available only in the Automation Lightning app:

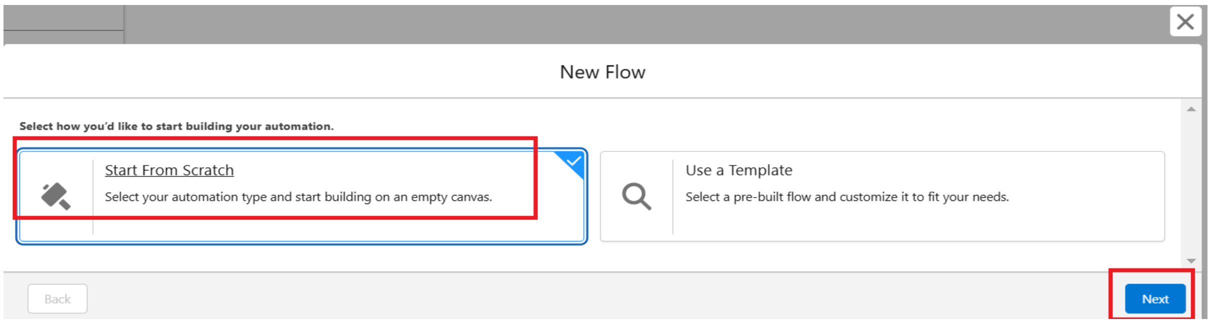
- Search for automations
- Sort your list views with more options
- Organize your automations with categories and subcategories

If you don't see the app in the App Launcher, check that Enable the Automation Lightning App is selected in Process Automation Settings.

Flow Definitions
All Flows

44 items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

**Step 2**: Select Start From Scratch and click Next.

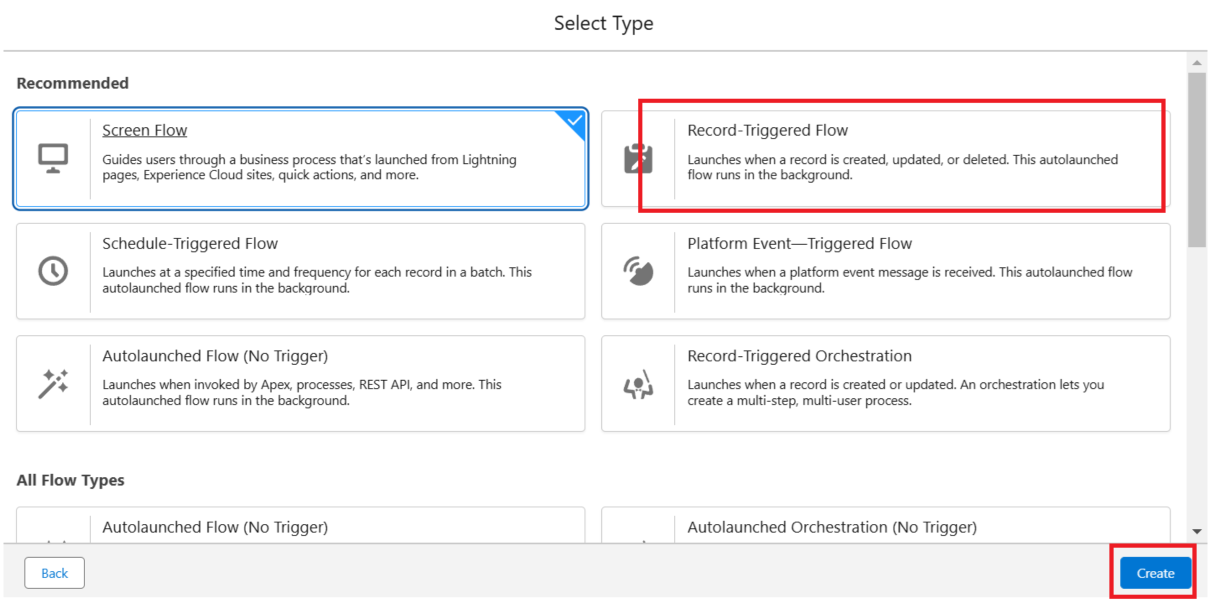


Select how you'd like to start building your automation.

Start From Scratch
Select your automation type and start building on an empty canvas.

Use a Template
Select a pre-built flow and customize it to fit your needs.

**Step 3**: Select Record-Triggered Flow and click Create



Recommended

Screen Flow
Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.

Record-Triggered Flow
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.

Schedule-Triggered Flow
Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.

Platform Event—Triggered Flow
Launches when a platform event message is received. This autolaunched flow runs in the background.

Autolaunched Flow (No Trigger)
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.

Record-Triggered Orchestration
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

All Flow Types

Autolaunched Flow (No Trigger)

Autolaunched Orchestration (No Trigger)

**Step 4**: Select Vehicle Order Object

Trigger the Flow When: Select Record is Created.

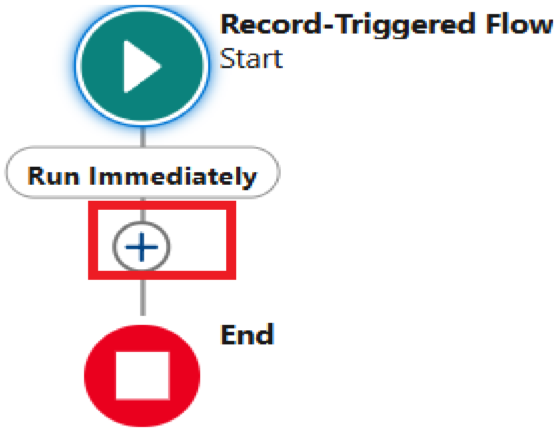
**Set Entry Condition:**

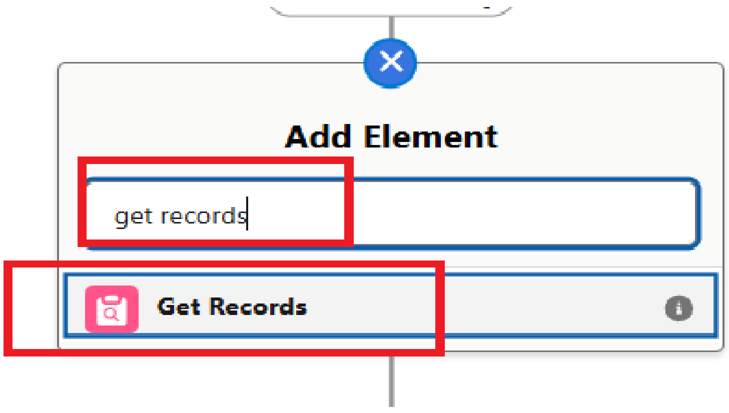
All Conditions Are Met (AND)

- Filed: Status__c

- Operator: Equals
- Value: Pending

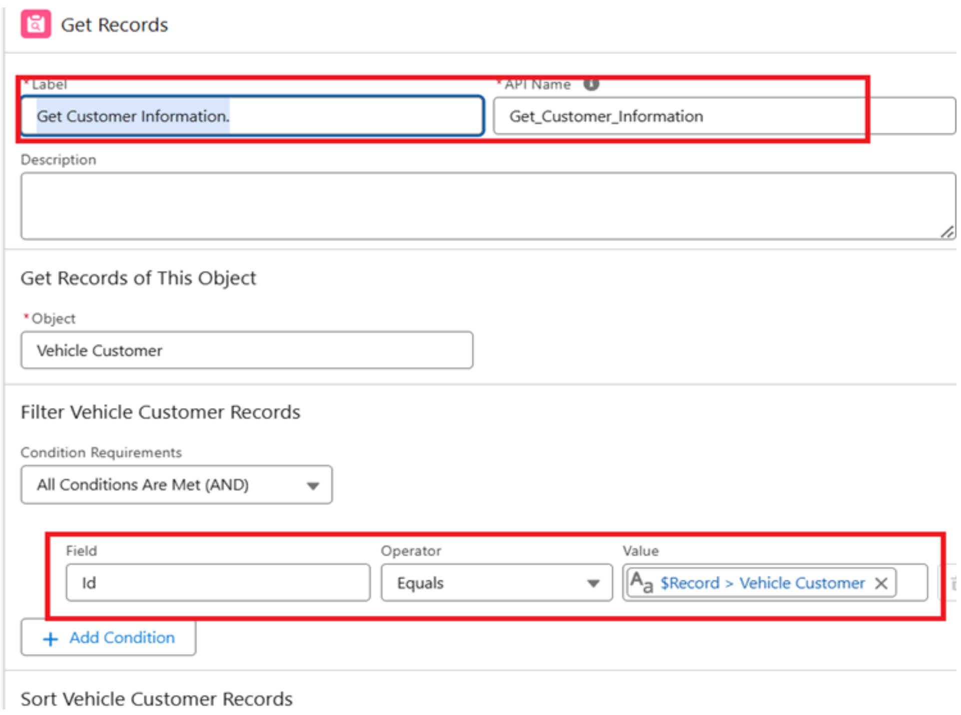
**Step 5**: Click + → Select Get Records





**Step 6**:

- Label: Get Customer Information
- Object: Vehicle Customer
- Condition: Field: Id,Operator : Equals
  Value: {!$Record.Vehicle_Customer__c}


Get Records

*Label: Get Customer Information.

*API Name: Get_Customer_Information

Description

Get Records of This Object

*Object: Vehicle Customer

Filter Vehicle Customer Records

Condition Requirements: All Conditions Are Met (AND)

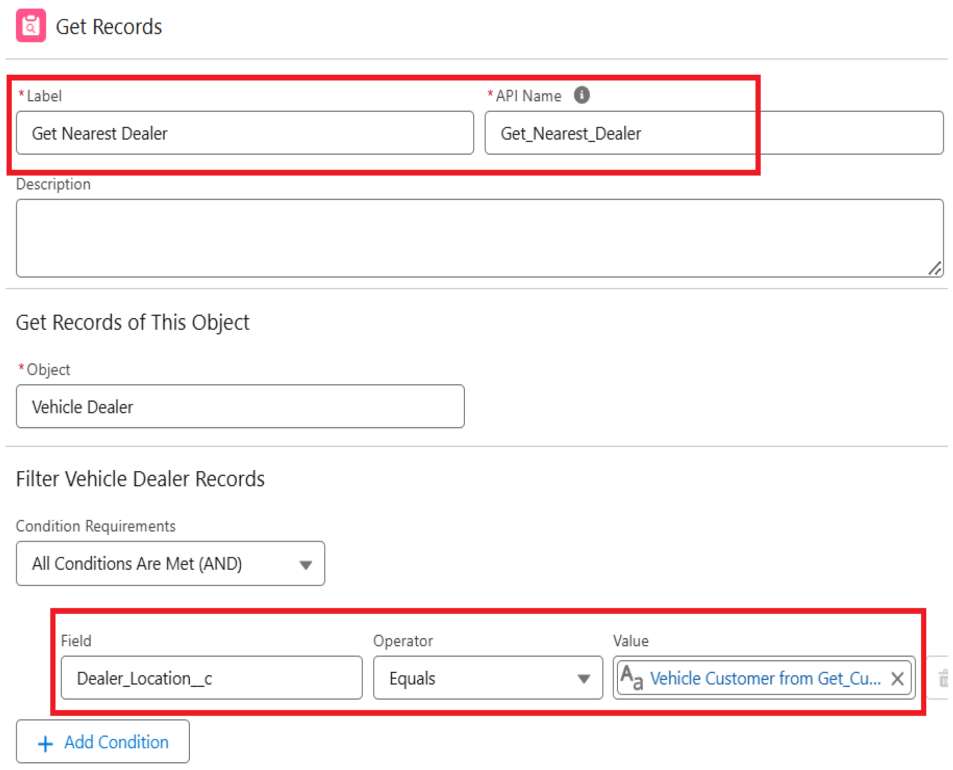
| Field | Operator | Value |
|---|---|---|
| Id | Equals | $Record > Vehicle Customer |

+ Add Condition

Sort Vehicle Customer Records

**Step 7**: Click + → Select Get Records

- Label : Get Nearest Dealer
- Object : Vehicle Dealer
- Condition : Field : Dealer_Location__c Operator : Equals Value : {!Get_Customer_Information.Address__c}


Get Records

*Label: Get Nearest Dealer

*API Name: Get_Nearest_Dealer

Description

Get Records of This Object

*Object: Vehicle Dealer

Filter Vehicle Dealer Records

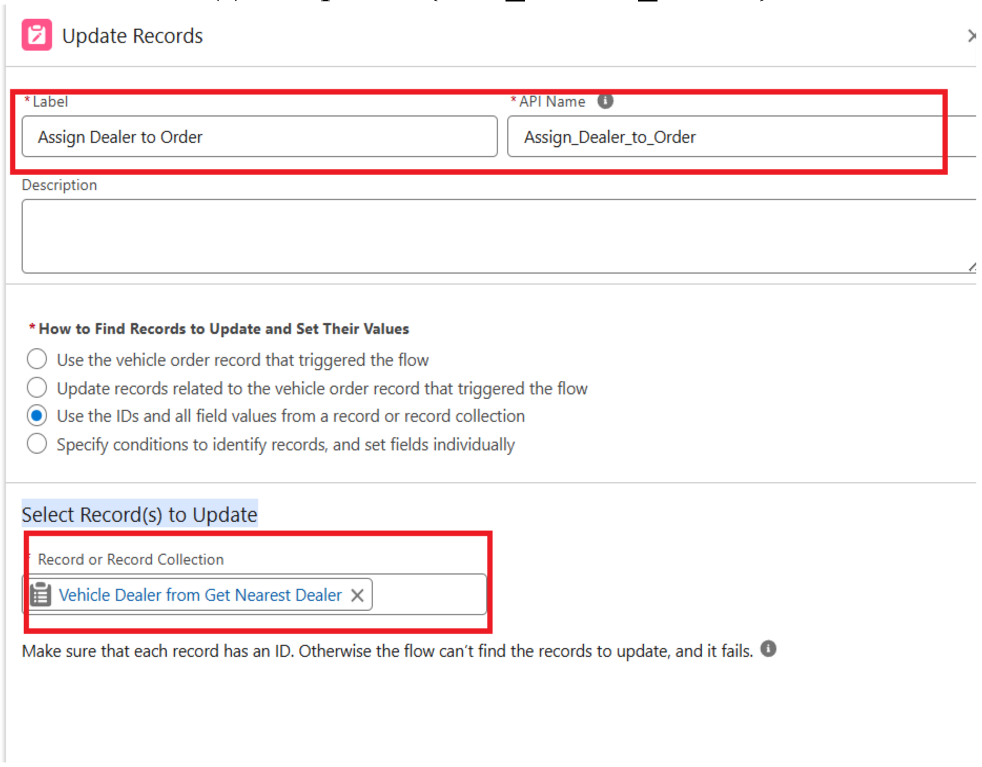
Condition Requirements: All Conditions Are Met (AND)

| Field | Operator | Value |
|---|---|---|
| Dealer_Location__c | Equals | Vehicle Customer from Get_Cu... |

+ Add Condition

**Step 8**: Click + → Select Update Records

- Label: Assign Dealer to Order
- How to Find Records to Update and Set Their Values: Use the IDs and all field values from a record or record collection
- Select Record(s) to Update: {!Get_Nearest_Dealer}



Update Records

*Label: Assign Dealer to Order

*API Name: Assign_Dealer_to_Order

Description

* How to Find Records to Update and Set Their Values

- ○ Use the vehicle order record that triggered the flow
- ○ Update records related to the vehicle order record that triggered the flow
- ● Use the IDs and all field values from a record or record collection
- ○ Specify conditions to identify records, and set fields individually

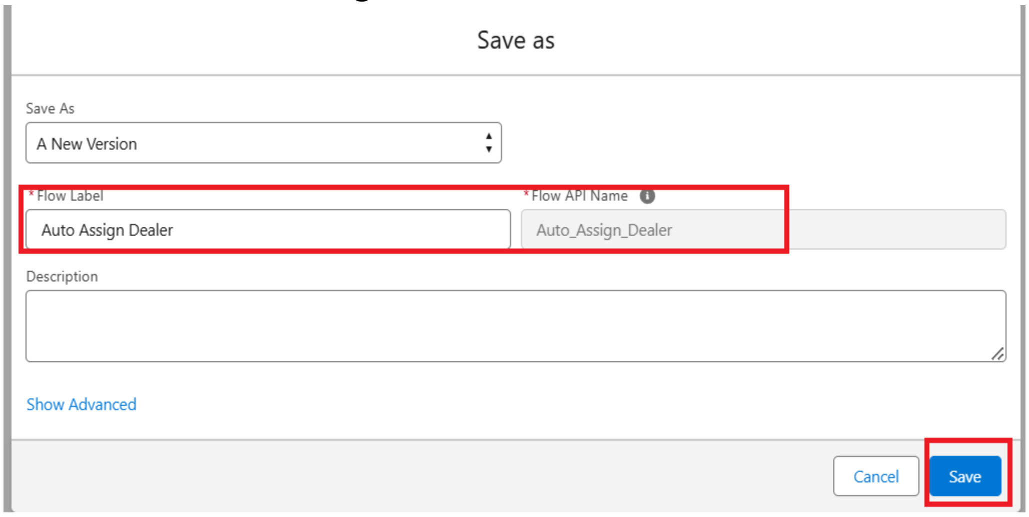
Select Record(s) to Update

Record or Record Collection: Vehicle Dealer from Get Nearest Dealer

Make sure that each record has an ID. Otherwise the flow can't find the records to update, and it fails.

**Step 9**: Click Save and Give label Name and Activate Flow.

- Label Name: Auto Assign Dealer



Save as

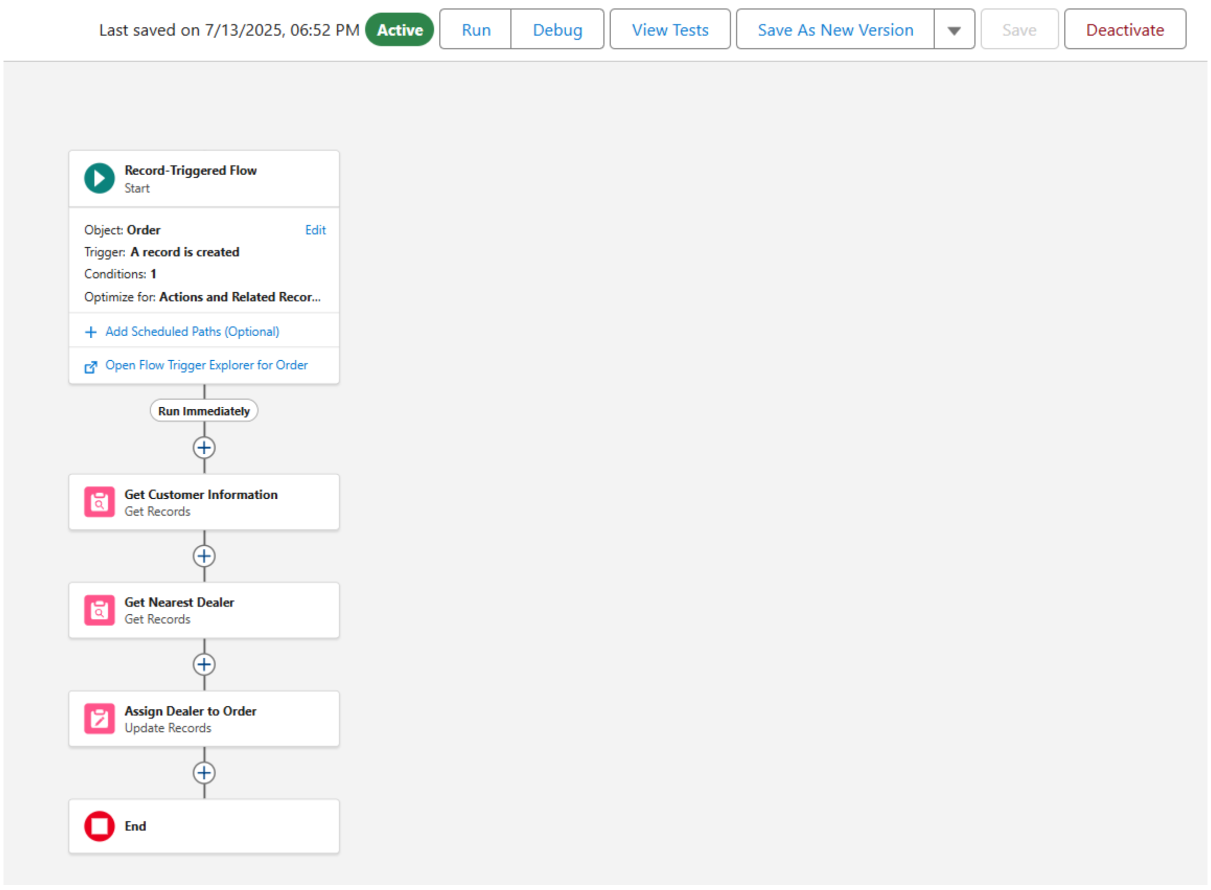
Save As: A New Version

*Flow Label: Auto Assign Dealer

*Flow API Name: Auto_Assign_Dealer

Description

## **Step 10**: Activate Flow


Last saved on 7/13/2025, 06:52 PM Active

Record-Triggered Flow
Start

Object: Order
Trigger: A record is created
Conditions: 1
Optimize for: Actions and Related Recor...

Edit
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Order
Run Immediately

Get Customer Information
Get Records

Get Nearest Dealer
Get Records

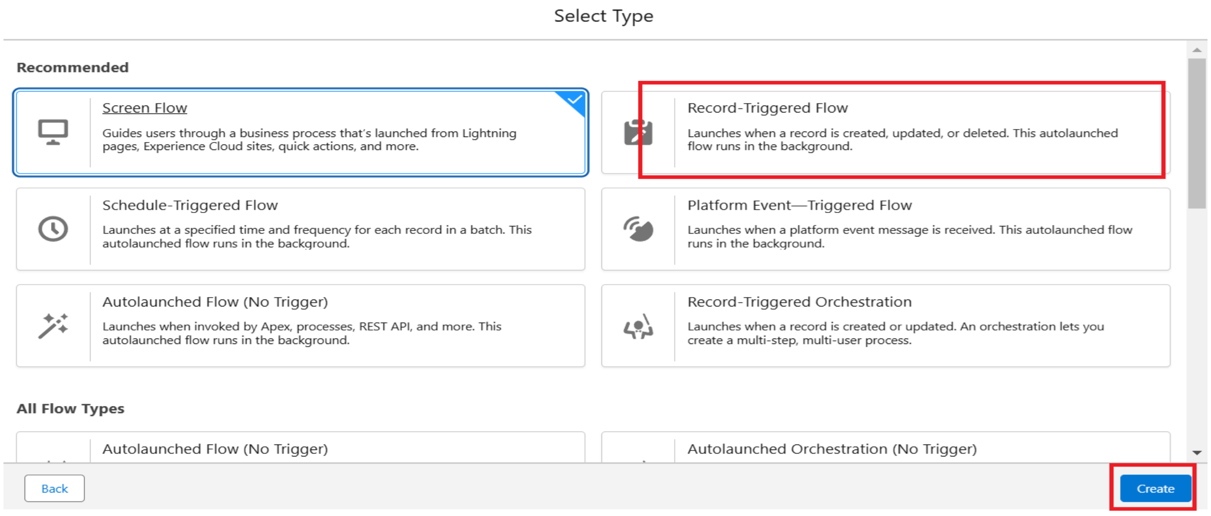
Assign Dealer to Order
Update Records

End

## **Activity 2: Record-Trigger Flow Creation**

Creating record triggered flow to send an email to the customer reminding about the test drive.

## **Step 1**: Select Record-Triggered Flow and click Create


Select Type

Recommended

Screen Flow
Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.

Record-Triggered Flow
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.

Schedule-Triggered Flow
Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.

Platform Event—Triggered Flow
Launches when a platform event message is received. This autolaunched flow runs in the background.

Autolaunched Flow (No Trigger)
Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.

Record-Triggered Orchestration
Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

All Flow Types

Autolaunched Flow (No Trigger)

Autolaunched Orchestration (No Trigger)

Step 2: Select Vehicle Test Drive Object

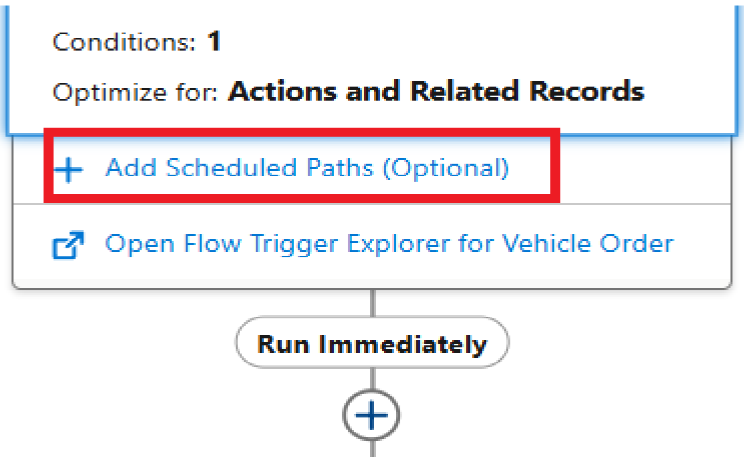
Trigger the Flow When: A record is created or updated

**Set Entry Condition**:

- All Conditions Are Met (AND)
- Filed: Status__c
- Operator: Equals
- Value: Scheduled

**Step 3**: Click + Add Scheduled Paths (below the trigger).

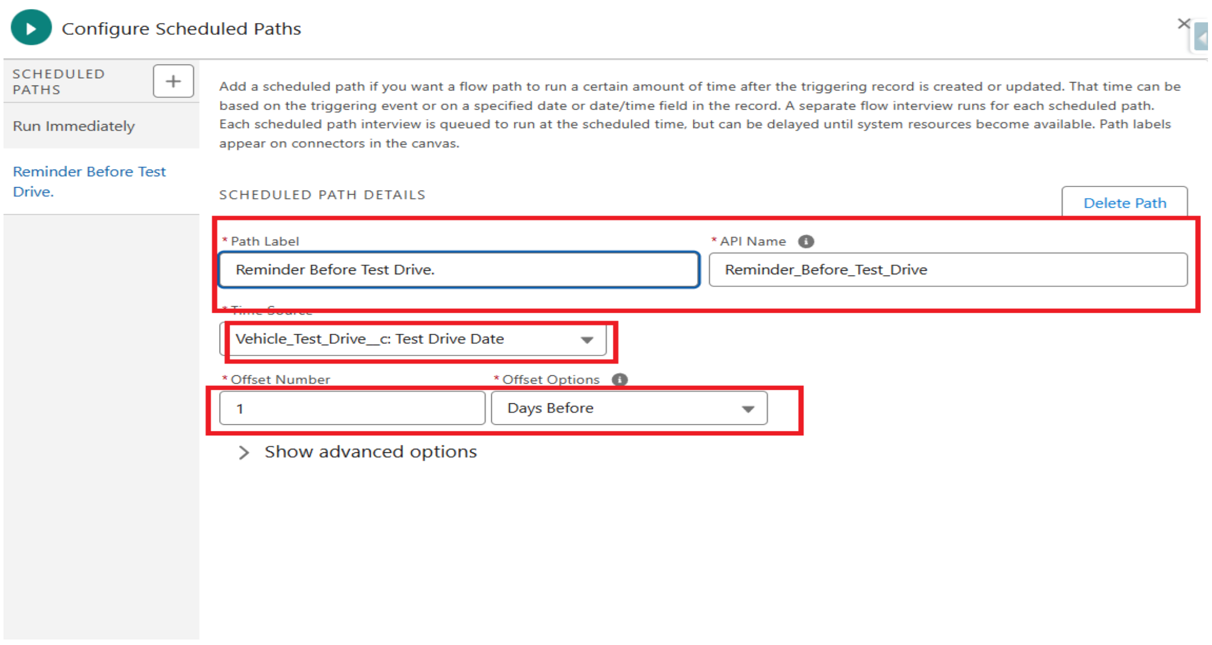
- Label: Reminder Before Test Drive.Time Source: Test_Drive_Date__c
- Offset Number: 1.
- Offset Options: Days Before.
- Click Done.



Conditions: 1

Optimize for: Actions and Related Records

+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Vehicle Order
Run Immediately


Configure Scheduled Paths

SCHEDULED PATHS

Run Immediately

Reminder Before Test Drive.

Add a scheduled path if you want a flow path to run a certain amount of time after the triggering record is created or updated. That time can be based on the triggering event or on a specified date or date/time field in the record. A separate flow interview runs for each scheduled path. Each scheduled path interview is queued to run at the scheduled time, but can be delayed until system resources become available. Path labels appear on connectors in the canvas.

SCHEDULED PATH DETAILS

Delete Path

* Path Label: Reminder Before Test Drive.

* API Name: Reminder_Before_Test_Drive

* Time Source: Vehicle_Test_Drive__c: Test Drive Date

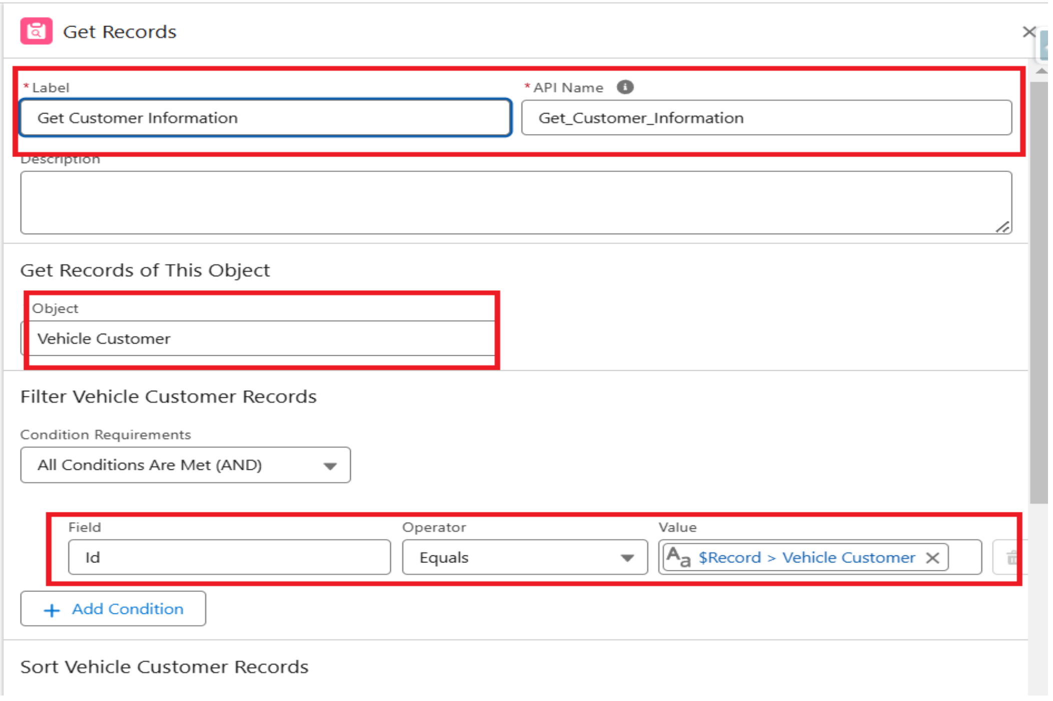
* Offset Number: 1

* Offset Options: Days Before

Show advanced options

**Step 4**: Click + Add Element → Get Records

- Label: Get Customer Information.
- Object: Vehicle_Customer__c.
- Filter Conditions: Id = {!$Record.Customer__c}.
- How Many Records to Store: Select Only the first record.
- How to Store Record Data: Choose Automatically store all fields.


Get Records

* Label: Get Customer Information

* API Name: Get_Customer_Information

Description

Get Records of This Object

Object: Vehicle Customer

Filter Vehicle Customer Records

Condition Requirements: All Conditions Are Met (AND)

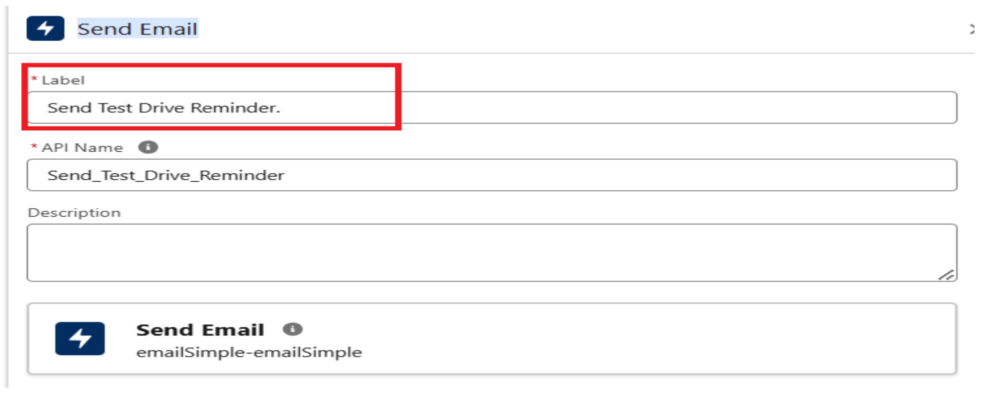
| Field | Operator | Value |
|---|---|---|
| Id | Equals | $Record > Vehicle Customer |

Add Condition

Sort Vehicle Customer Records

**Step 5**: Send Reminder Email

- Click + Add Element → Action.
- Action Type: Send Email
- Label: Send Test Drive Reminder.
- Subject: "Reminder: Your Test Drive is Tomorrow!".
- Recipient Address List: {!Get_Customer_Information.Email__c}
- Rich-Text-Formatted Body: True


Send Email

* Label
Send Test Drive Reminder.

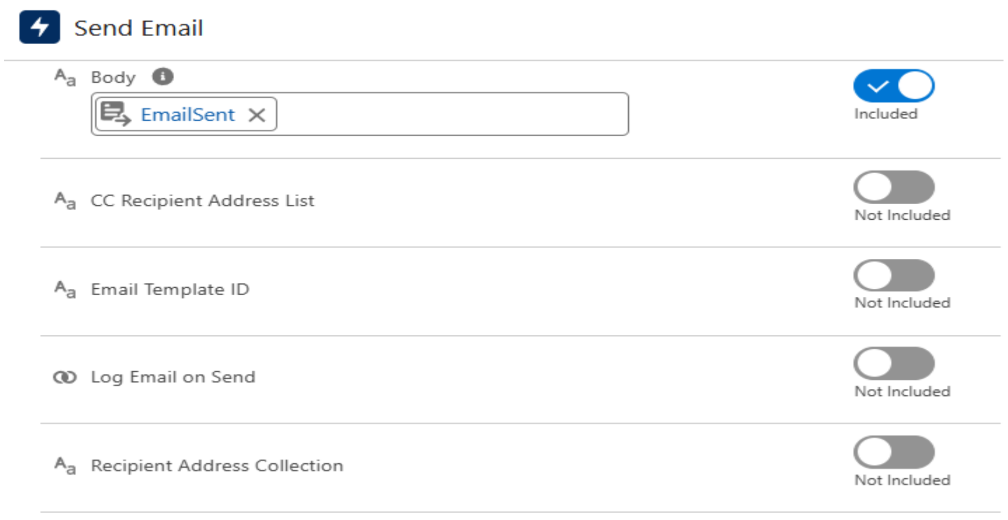
* API Name
Send_Test_Drive_Reminder

Description

Send Email
emailSimple-emailSimple

- Body: Create Variable

- Api Name: EmailSent


Send Email

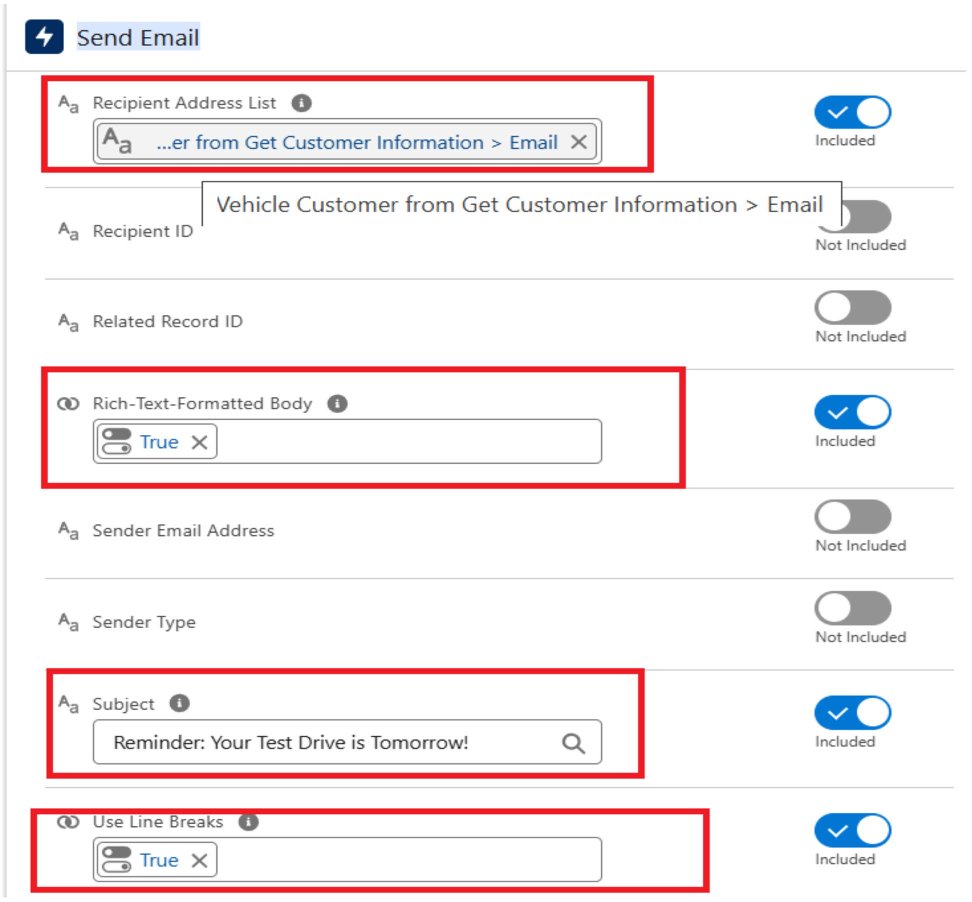
Body: EmailSent — Included

CC Recipient Address List — Not Included

Email Template ID — Not Included

Log Email on Send — Not Included

Recipient Address Collection — Not Included


Send Email

Recipient Address List: ...er from Get Customer Information > Email — Included

Vehicle Customer from Get Customer Information > Email

Recipient ID — Not Included

Related Record ID — Not Included

Rich-Text-Formatted Body: True — Included

Sender Email Address — Not Included

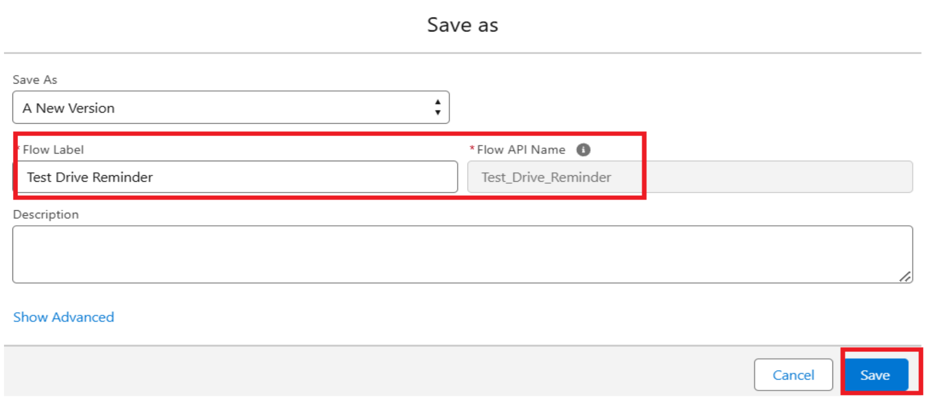
Sender Type — Not Included

Subject: Reminder: Your Test Drive is Tomorrow! — Included

Use Line Breaks: True — Included

**Step 6**: Click save

- Label Name: Test Drive Reminder


Save as

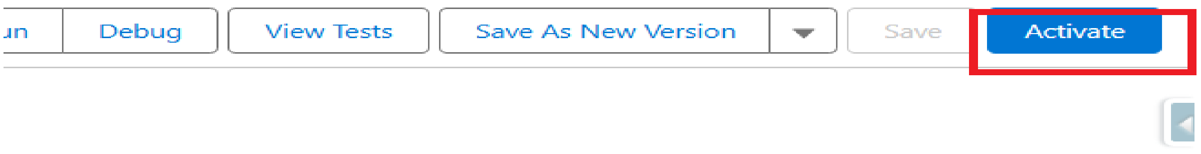
Save As: A New Version

Flow Label: Test Drive Reminder

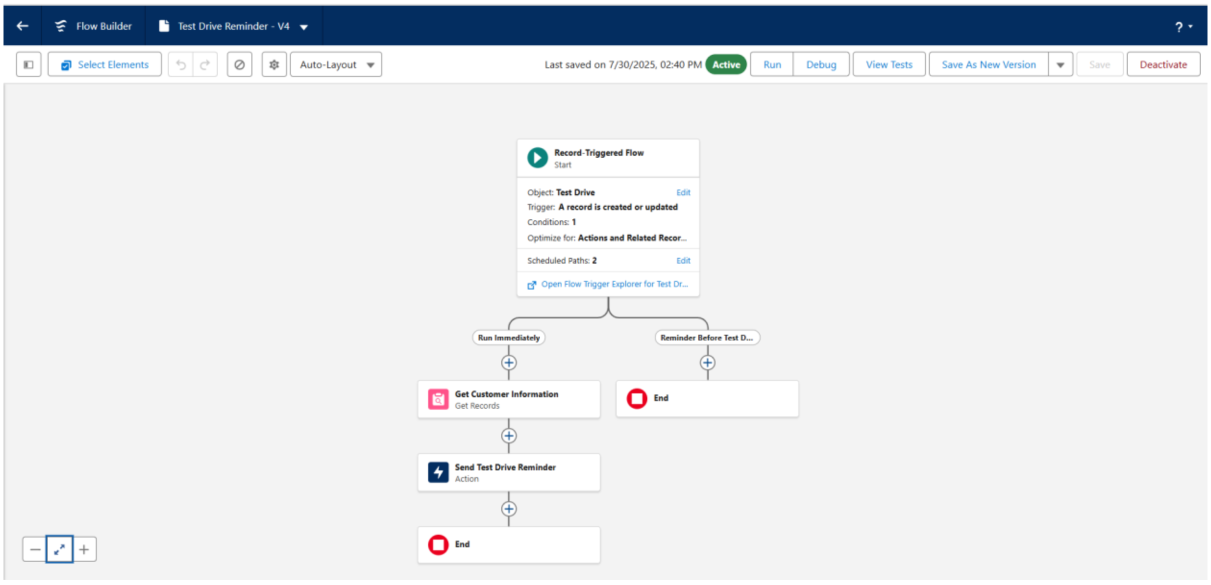
* Flow API Name: Test_Drive_Reminder

Description:

## Step 7: Activate Flow



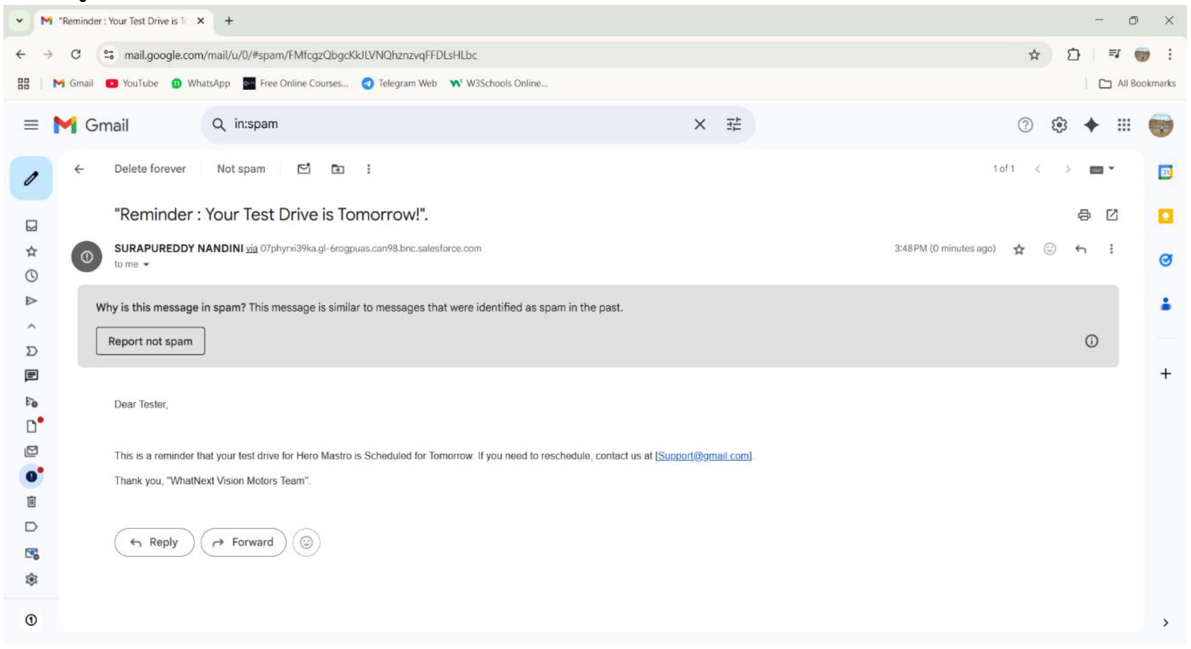
## Flow Page After Creation:



Last saved on 7/30/2025, 02:40 PM

Object: Test Drive
Trigger: A record is created or updated
Conditions: 1
Optimize for: Actions and Related Recor...
Scheduled Paths: 2

## Test your Flow:



"Reminder : Your Test Drive is Tomorrow!".

SURAPUREDDY NANDINI via 07phyrxi39ka.gl-6rogpuas.can98.bnc.salesforce.com
to me

3:48 PM (0 minutes ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Tester,

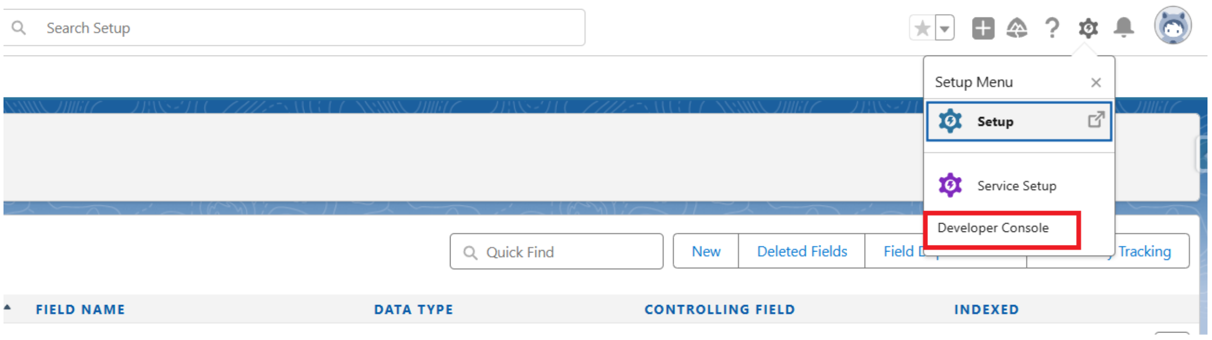
This is a reminder that your test drive for Hero Mastro is Scheduled for Tomorrow. If you need to reschedule, contact us at [Support@gmail.com]

Thank you, "WhatNext Vision Motors Team".

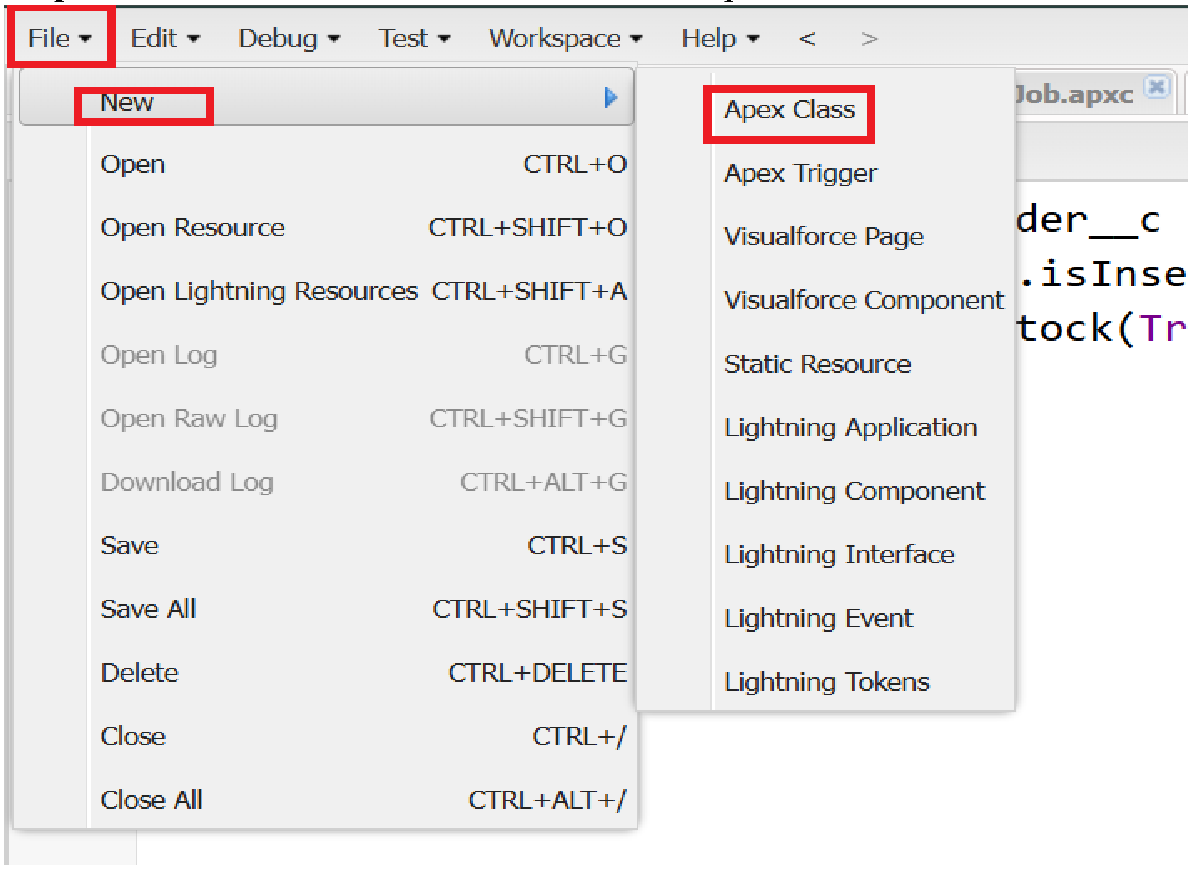
# Phase 5: Development

## Create Apex and Trigger Batch Jobs

### Step 1: Click Developer Console from Gear icon

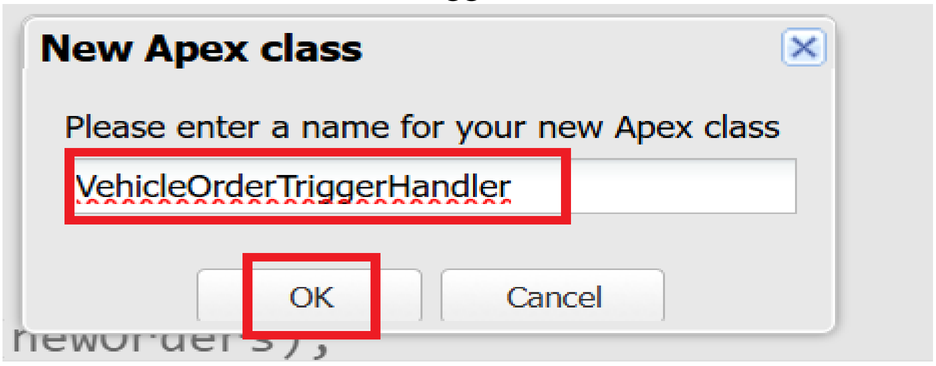


**Step 2**: click File then Select New then Select Apex.

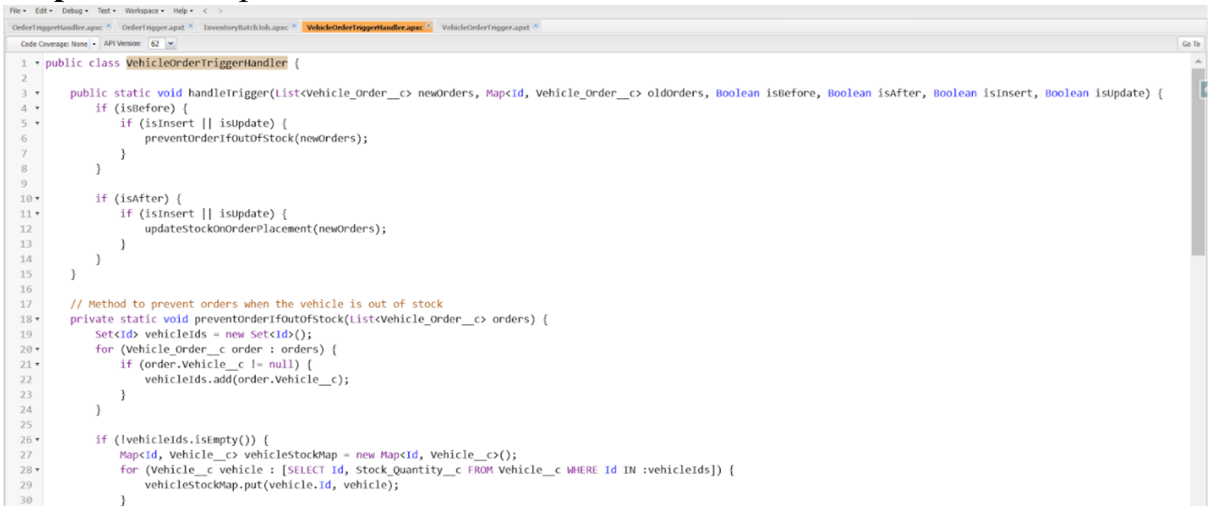


**Step 3**: Give Name for Apex Class

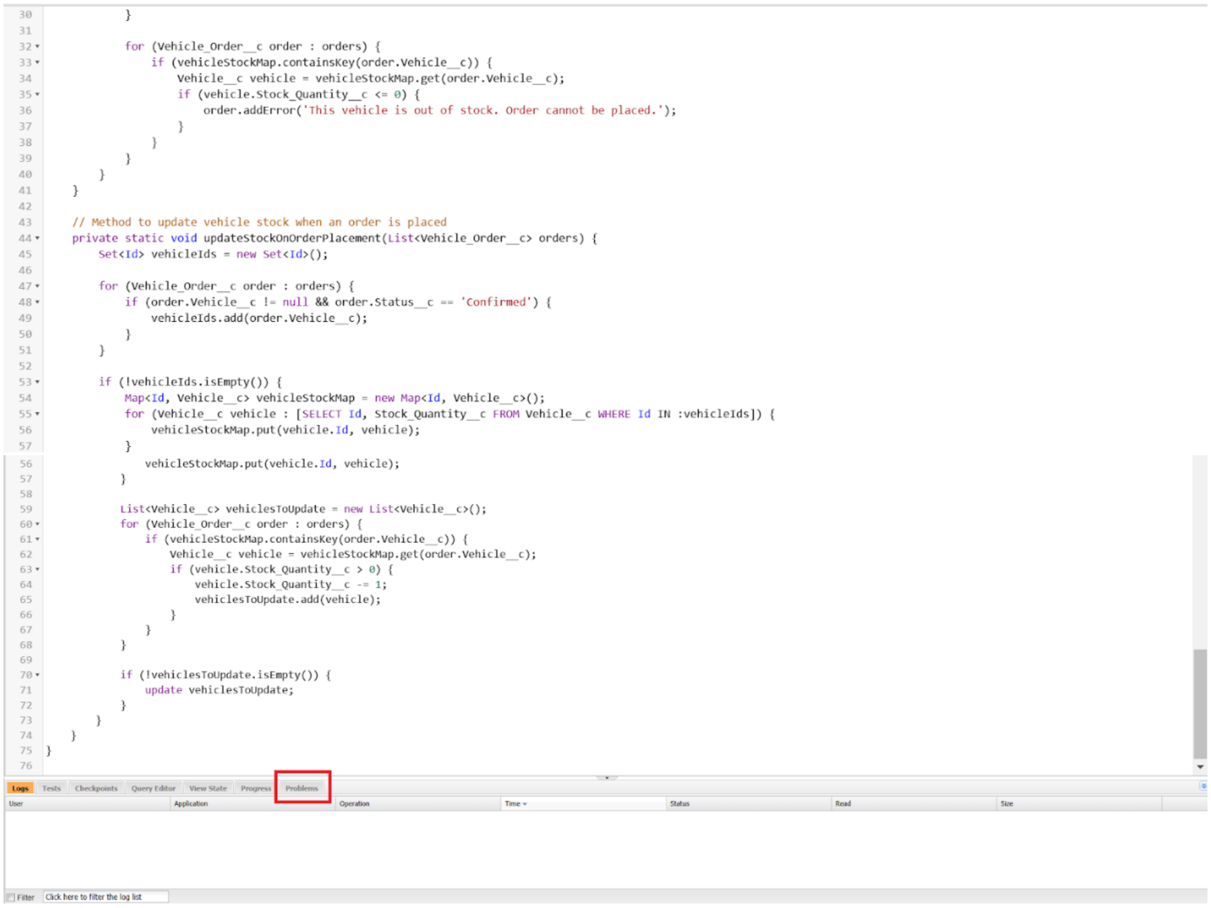
- Class Name: VehicleOrderTriggerHandler


New Apex class
Please enter a name for your new Apex class
VehicleOrderTriggerHandler
OK
Cancel

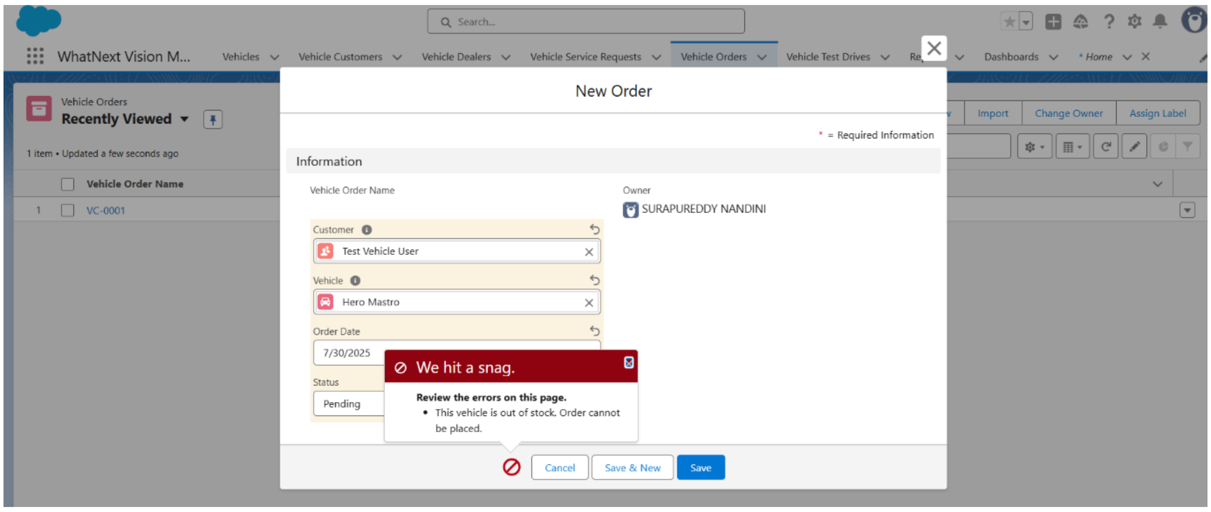
## **Step 4**: Write Apex Code



```
public class VehicleOrderTriggerHandler {

    public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isInsert, Boolean isUpdate) {
        if (isBefore) {
            if (isInsert || isUpdate) {
                preventOrderIfOutOfStock(newOrders);
            }
        }

        if (isAfter) {
            if (isInsert || isUpdate) {
                updateStockOnOrderPlacement(newOrders);
            }
        }
    }

    // Method to prevent orders when the vehicle is out of stock
    private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }

        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }
```



```
            }

            for (Vehicle_Order__c order : orders) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c <= 0) {
                        order.addError('This vehicle is out of stock. Order cannot be placed.');
                    }
                }
            }
        }
    }

    // Method to update vehicle stock when an order is placed
    private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();

        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null && order.Status__c == 'Confirmed') {
                vehicleIds.add(order.Vehicle__c);
            }
        }

        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }
                vehicleStockMap.put(vehicle.Id, vehicle);
            }

            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
            for (Vehicle_Order__c order : orders) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c > 0) {
                        vehicle.Stock_Quantity__c -= 1;
                        vehiclesToUpdate.add(vehicle);
                    }
                }
            }

            if (!vehiclesToUpdate.isEmpty()) {
                update vehiclesToUpdate;
            }
        }
    }
}
```



1 item • Updated a few seconds ago

1 VC-0001

New Order

* = Required Information

Information

Vehicle Order Name

Owner: SURAPUREDDY NANDINI

Customer: Test Vehicle User

Vehicle: Hero Mastro

Order Date: 7/30/2025

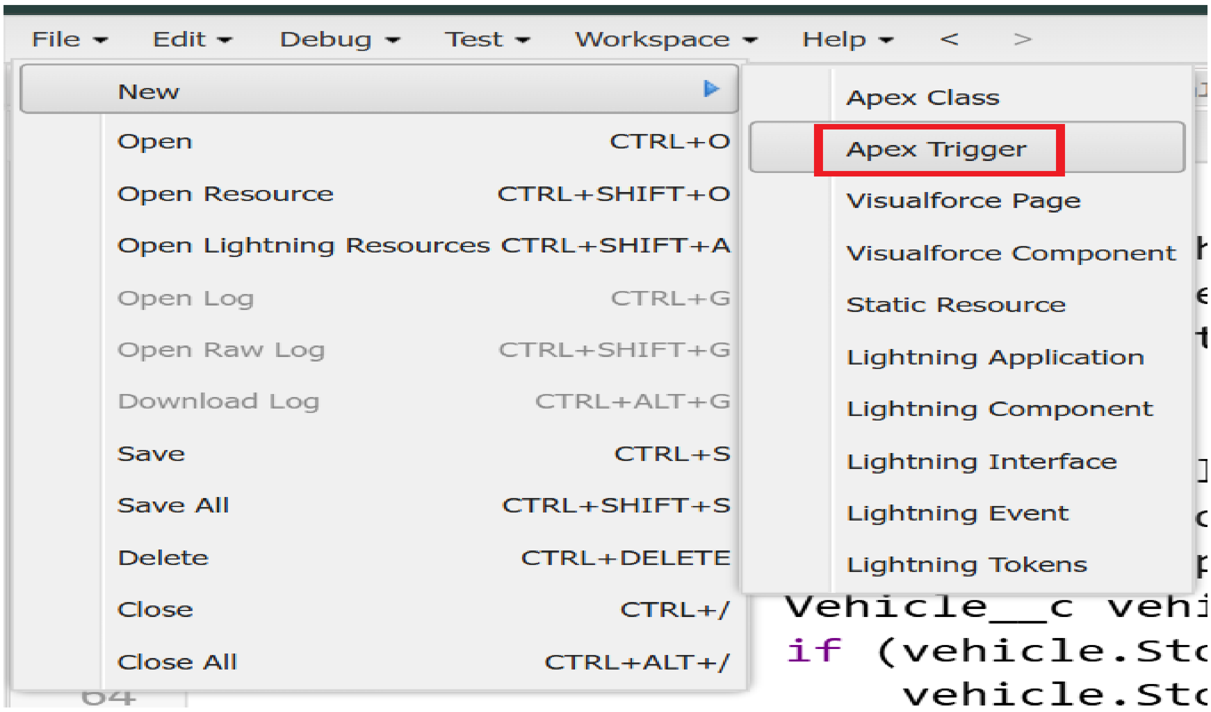
Status: Pending

We hit a snag.

Review the errors on this page.

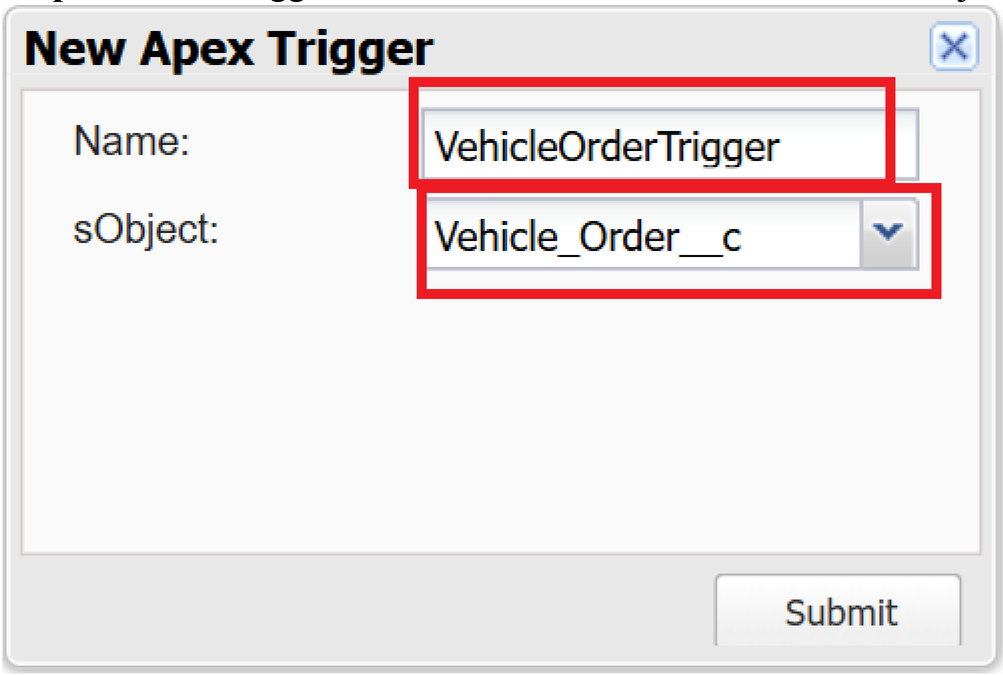
- This vehicle is out of stock. Order cannot be placed.

**Step 5**: Write Trigger Handler



```
Vehicle__c vehi
if (vehicle.Stc
    vehicle.Stc
```

**Step 6**: Writer Trigger Class Name and Select Vehicle Order Object



Name: VehicleOrderTrigger

sObject: Vehicle_Order__c

**Step 7**: Call Apex Class in Trigger Class

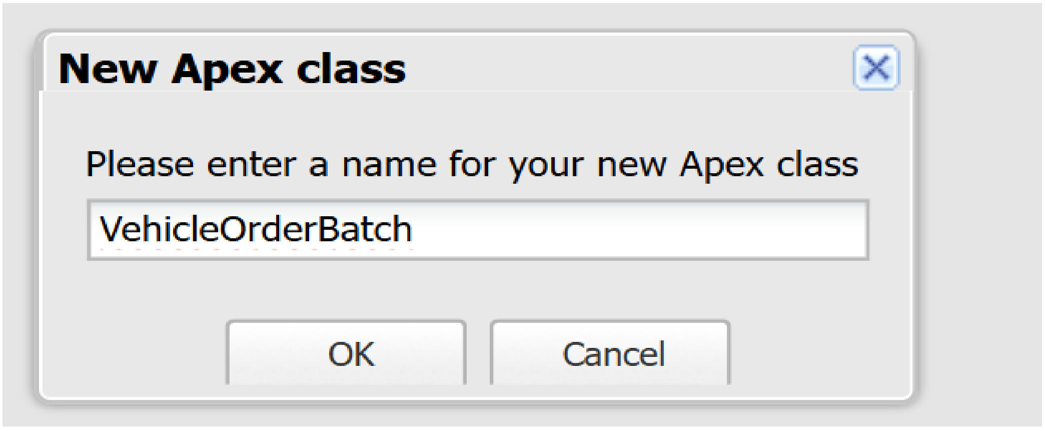
**Source Code:**

```
trigger VehicleOrderTrigger on Vehicle_Order__c (before insert,before update, after insert, after update) {
    VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore,Trigger.isAfter, Trigger.isInsert, Trigger.isUpdate);
}
```

**Step 8**: Create Batch Job

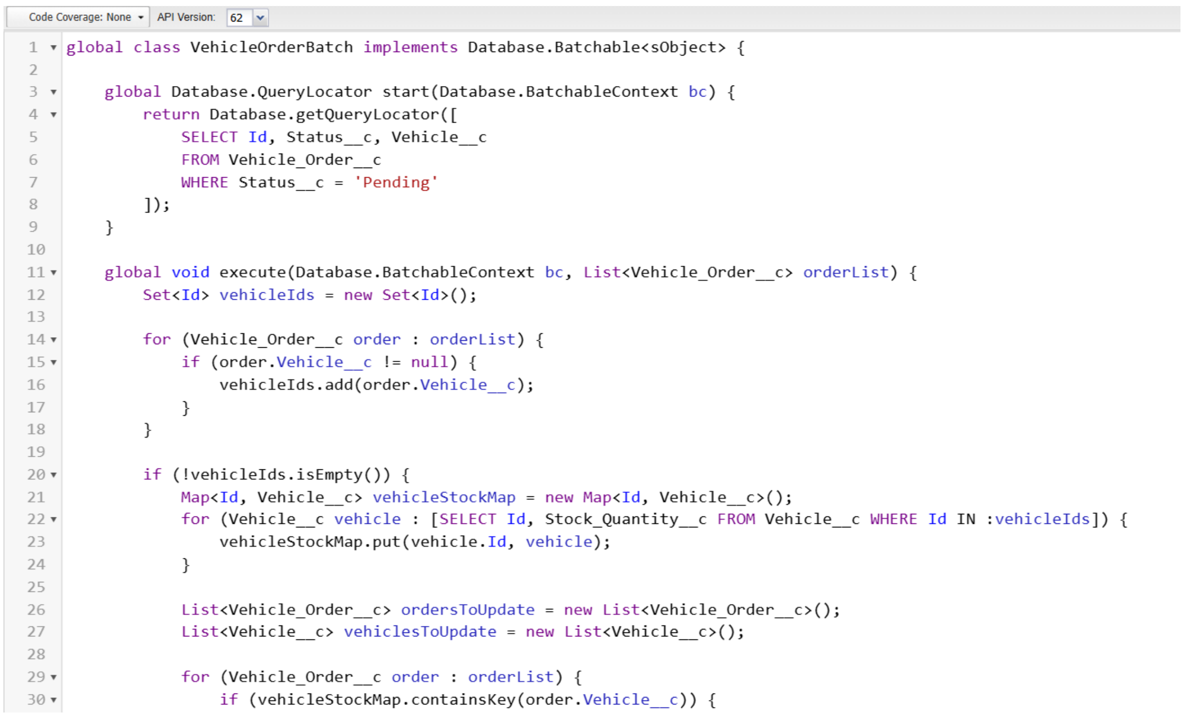
A customer places an order, but the vehicle is out of stock.

- The order remains pending.
- After new stock is added, the batch job updates the order to confirmed.

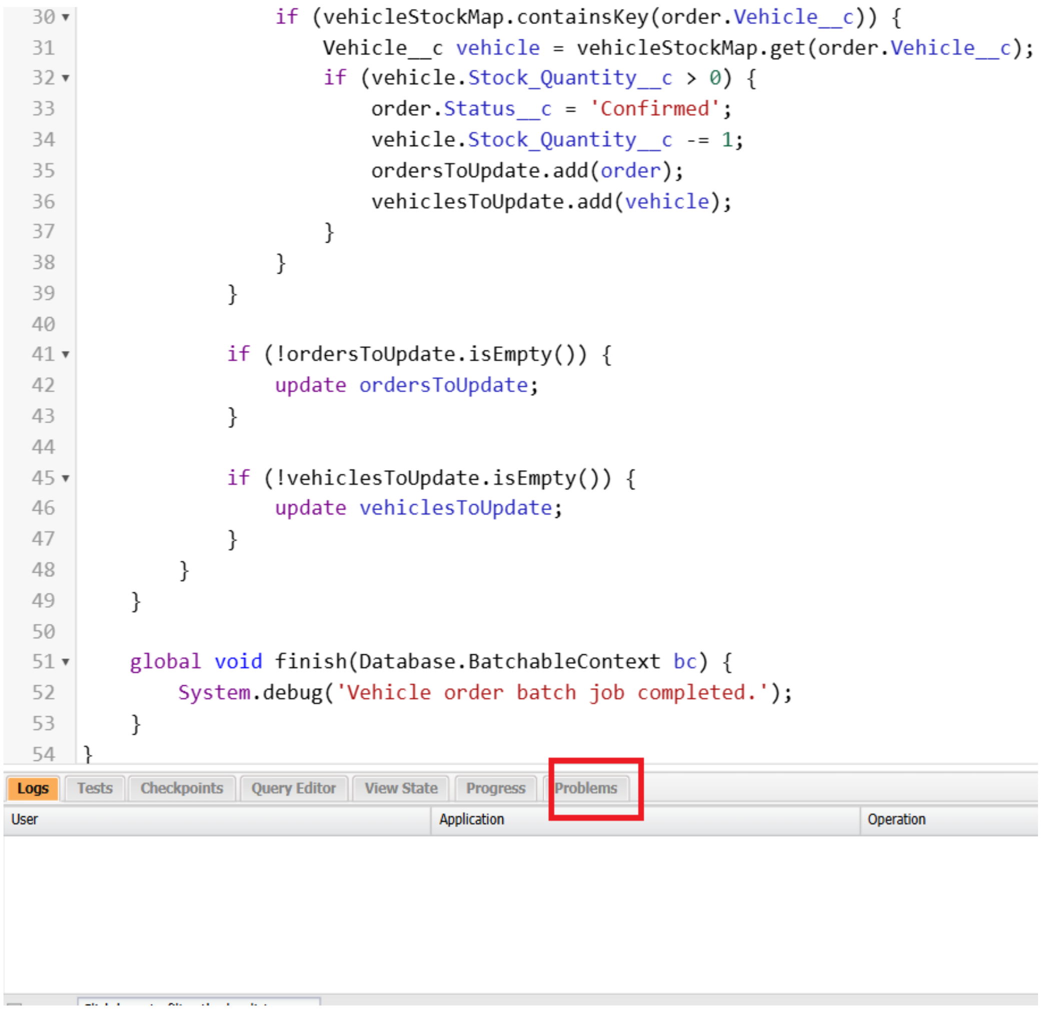


Please enter a name for your new Apex class

VehicleOrderBatch

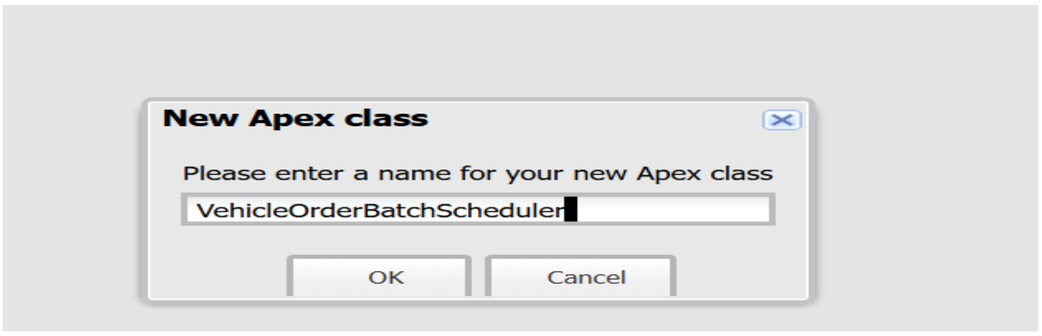


```
global class VehicleOrderBatch implements Database.Batchable<sObject> {

    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator([
            SELECT Id, Status__c, Vehicle__c
            FROM Vehicle_Order__c
            WHERE Status__c = 'Pending'
        ]);
    }

    global void execute(Database.BatchableContext bc, List<Vehicle_Order__c> orderList) {
        Set<Id> vehicleIds = new Set<Id>();

        for (Vehicle_Order__c order : orderList) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }

        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }

            List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();

            for (Vehicle_Order__c order : orderList) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
```



```
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c > 0) {
                        order.Status__c = 'Confirmed';
                        vehicle.Stock_Quantity__c -= 1;
                        ordersToUpdate.add(order);
                        vehiclesToUpdate.add(vehicle);
                    }
                }
            }

            if (!ordersToUpdate.isEmpty()) {
                update ordersToUpdate;
            }

            if (!vehiclesToUpdate.isEmpty()) {
                update vehiclesToUpdate;
            }
        }
    }

    global void finish(Database.BatchableContext bc) {
        System.debug('Vehicle order batch job completed.');
    }
}
```

**Step 9:** Create Schedule Class then revoked Batch Class in Schedule Class


New Apex class

Please enter a name for your new Apex class

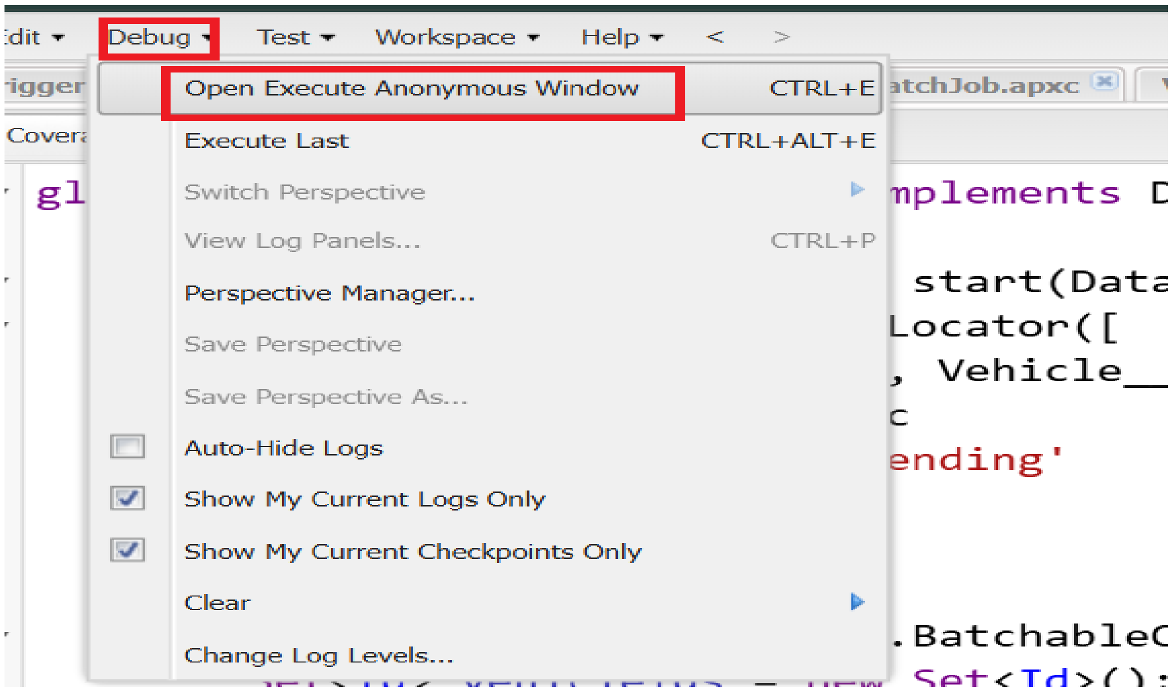
VehicleOrderBatchScheduler

OK Cancel

**Source Code**:

```
global class VehicleOrderBatchScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        VehicleOrderBatch batchJob = new VehicleOrderBatch();
        Database.executeBatch(batchJob, 50); // 50 is the batch size
    }
}
```

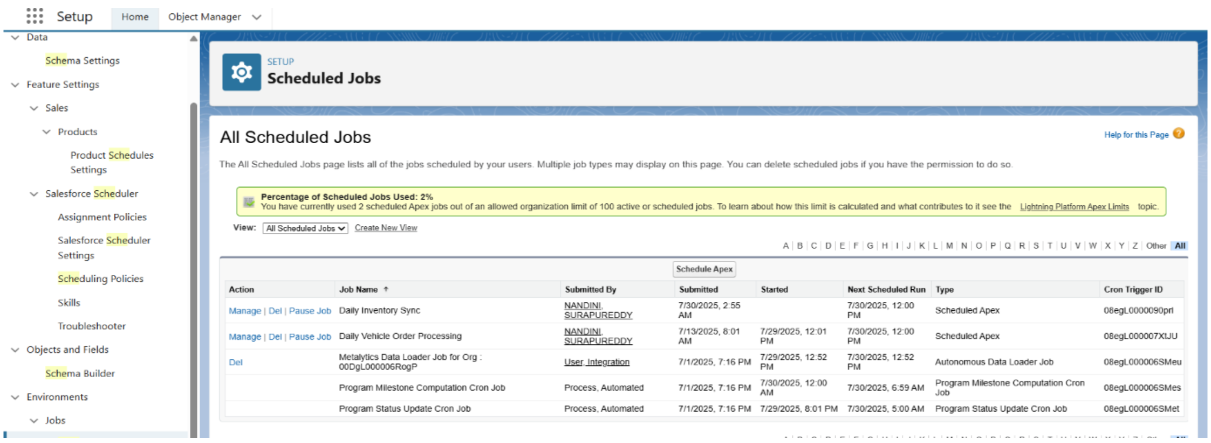
**Step 10**: Schedule the Batch Job To run the batch job every night at midnight:



**Source Code**:

```
String cronExp = '0 0 12 * * ?'; // Runs daily at 12:00 PM
System.schedule('Daily Vehicle Order Processing', cronExp, new
VehicleOrderBatchScheduler());
```

**Step 11**: You Can Check where this Schedule Job is Running



All Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

**Percentage of Scheduled Jobs Used: 2%**
You have currently used 2 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the Lightning Platform Apex Limits topic.

| Action | Job Name ↑ | Submitted By | Submitted | Started | Next Scheduled Run | Type | Cron Trigger ID |
|---|---|---|---|---|---|---|---|
| Manage \| Del \| Pause Job | Daily Inventory Sync | NANDINI, SURAPUREDDY | 7/30/2025, 2:55 AM | | 7/30/2025, 12:00 PM | Scheduled Apex | 08egL0000090prl |
| Manage \| Del \| Pause Job | Daily Vehicle Order Processing | NANDINI, SURAPUREDDY | 7/13/2025, 8:01 AM | 7/29/2025, 12:01 PM | 7/30/2025, 12:00 PM | Scheduled Apex | 08egL000007XtJU |
| Del | Metalytics Data Loader Job for Org : 00DgL000006RogP | User, Integration | 7/1/2025, 7:16 PM | 7/29/2025, 12:52 PM | 7/30/2025, 12:52 PM | Autonomous Data Loader Job | 08egL000006SMeu |
| | Program Milestone Computation Cron Job | Process, Automated | 7/1/2025, 7:16 PM | 7/30/2025, 12:00 AM | 7/30/2025, 6:59 AM | Program Milestone Computation Cron Job | 08egL000006SMes |
| | Program Status Update Cron Job | Process, Automated | 7/1/2025, 7:16 PM | 7/29/2025, 8:01 PM | 7/30/2025, 5:00 AM | Program Status Update Cron Job | 08egL000006SMet |

## Conclusion:

This Salesforce implementation at WhatsNext Vision Motors is a strategic move towards digital transformation in the automotive industry. By focusing on critical pain points in the ordering and fulfillment process, the project delivers tangible improvements in customer service and internal operations. The end result is a robust, intelligent, and user-centric ordering system that aligns with the company's mission to lead the future of mobility.

## Future Enhancements:

- Customer Self-Service Portal for order tracking and service requests.
- AI-Powered Vehicle Recommendations using Salesforce Einstein.
- Advanced Dealer & Sales Analytics with detailed dashboards.
- Mobile App Integration for on-the-go dealer and sales management.
- Predictive Stock Management using demand forecasting.