

WhatsNext Vision Motors

Shaping the Future of Mobility with Innovation and Excellence

WhatsNext Vision Motors, a pioneering force in the automotive industry, is dedicated to transforming the mobility sector through innovative technology and customer-focused solutions. This Salesforce project aims to enhance customer experience, streamline the ordering process, and strengthen operational efficiency across the organization.

Project Overview

This Salesforce implementation improves the customer ordering process by automatically suggesting the nearest dealer based on the customer's address. This feature enhances convenience and reduces the effort required for customers when placing vehicle orders.

The system also prevents order placement for vehicles that are out of stock. This ensures that customers can only place orders for available units, reducing confusion and boosting order accuracy.

A scheduled process is included to automatically update bulk order records. If a vehicle is out of stock, the order status becomes Pending. If the vehicle is available, the order status becomes Confirmed. This automation promotes transparency and accurate communication with customers.

Overall, this project enhances efficiency, reduces manual workload, minimizes errors, and improves customer satisfaction.

Requirements

1. Salesforce CRM Implementation

- Store and manage vehicle details, stock availability, and dealer information.
- Track customer orders, test drives, and service requests.
- Automatically assign orders to the nearest dealer based on customer location.

2. Process Automation

- Prevent order placement for out-of-stock vehicles.
- Auto-assign orders to the nearest dealer.
- Send automated email reminders for scheduled test drives.

3. Apex and Triggers

- Implement Apex triggers for stock validation and dealer assignment.
- Use a trigger handler framework for modular and maintainable code.

4. Batch Jobs

- Create a Batch Apex job to periodically check and update stock availability.
- Send scheduled email notifications for stock replenishment and order processing.

What You'll Learn

- Data Modelling
- Fields and Relationships
- Lightning App Builder
- Record-Triggered Flows
- Apex and Apex Triggers
- Batch Apex
- Scheduled Apex

Project Progress

Salesforce Credentials Creation

Setup

Home

Object Manager

Q Search Setup

Star

Share

Help

Settings

Notifications

Profile

Q Quick Find

Setup Home

Salesforce Go

Service Setup Assistant

Commerce Setup Assistant

Hyperforce Assistant

Release Updates

Salesforce Mobile App

Lightning Usage

Optimizer

Sales Cloud Everywhere

ADMINISTRATION

> Users

> Data

> Email

PLATFORM TOOLS

> Subscription Management

> Apps

> Feature Settings

> Slack

> Data Cloud

> Heroku

> MuleSoft

> Einstein

> Objects and Fields

Welcome, Jermaine Micah

Manage and customize Salesforce from Setup. Browse suggestions, explore features, and more.

Achieve Popular Business Goals

Cross Cloud

Connect with Sales Prospects and Customers

Help sales reps prioritize and engage with customers.

Includes

- Sales Cloud Everywhere
- Agentforce (Default)
- Einstein Sales Emails

3 Completed1 In Progress

Keep Going

Cross Cloud

Track & Manage Customer Data

Collect, organize, and analyze customer data to drive insights, personalize experiences, and boost engagement.

Includes

- Case Management
- Feedback Management
- Einstein Activity Capture

2 Completed1 In Progress

Keep Going

Sales

Capture & Auto-Qualify Leads

Qualify and pass leads to reps and convert qualified leads quickly.

Includes

- Prospecting Center
- Sales Engagement
- Cadences

1 Completed

Keep Going

View All

Recent Items

NAME	TYPE	OBJECT
SFDC_DevConsole	Debug Level	
VehicleOrderBatchScheduler	Apex Class	
VehicleOrderBatch	Apex Class	
VehicleOrderTrigger	Apex Trigger	Vehicle Order

Data Management-Objects

Setup

Home

Object Manager

Q Search Setup

Star

Share

Help

Settings

Notifications

Profile

SETUP

Object Manager

6 Items, Sorted by Label

Q Vehicle

Schema Builder

Create

Data Management-Tabs

The screenshot shows the Salesforce Setup interface for the 'Custom Tabs' section. The left sidebar contains navigation links for 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Custom Tabs' and includes a 'Help for this Page' link. Below the title, there is a brief explanation of custom tabs and their types: Custom Object Tabs, Web Tabs, and Visualforce Tabs. The 'Custom Object Tabs' section contains a table with columns for 'Action', 'Label', 'Tab Style', and 'Description'. The table lists six tabs: 'Vehicle Customers' (People style), 'Vehicle Dealers' (Building style), 'Vehicle Orders' (Box style), 'Vehicles' (Car style), 'Vehicle Service Requests' (Form style), and 'Vehicle Test Drives' (Cards style). The 'Web Tabs' and 'Visualforce Tabs' sections both indicate that no tabs have been defined.

Action	Label	Tab Style	Description
Edit Del	Vehicle Customers	People	
Edit Del	Vehicle Dealers	Building	
Edit Del	Vehicle Orders	Box	
Edit Del	Vehicles	Car	
Edit Del	Vehicle Service Requests	Form	
Edit Del	Vehicle Test Drives	Cards	

Data Management-App Manager

The screenshot shows the Salesforce App Manager interface for the 'Vehicle Customers' app. The top navigation bar includes links for 'WhatNext Vision Motors', 'Vehicle Customers', 'Vehicle Dealers', 'Vehicle Service Requests', 'Vehicle Orders', 'Vehicle Test Drives', 'Vehicles', 'Reports', and 'Dashboards'. The main content area is titled 'Vehicle Customers' and includes a 'Recently Viewed' section. Below this, there is a table with columns for 'Vehicle Customer Name' and 'John'. The table contains one row with the name 'John'.

Vehicle Customer Name	John
John	

Data Management-Fields

SETUP > OBJECT MANAGER

Vehicle Customer

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

Object Access

Triggers

Flow Triggers

Validation Rules

Conditional Field Formatting

Fields & Relationships

8 Items, Sorted by Field Label

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(60)		<input type="checkbox"/>
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		<input type="checkbox"/>
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>
Phone	Phone__c	Phone		<input type="checkbox"/>
Preferred Vehicle Type	Preferred_Vehicle_Type__c	Picklist		<input type="checkbox"/>
Vehicle Customer Name	Name	Text(80)		<input checked="" type="checkbox"/>

SETUP > OBJECT MANAGER

Vehicle Dealer

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

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

8 Items, Sorted by Field Label

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Dealer Code	Dealer_Code__c	Auto Number		<input type="checkbox"/>
Dealer Location	Dealer_Location__c	Text(60)		<input type="checkbox"/>
Dealer Name	Name	Text(80)		<input checked="" type="checkbox"/>
Email	Email__c	Email		<input type="checkbox"/>
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		<input checked="" type="checkbox"/>
Phone	Phone__c	Phone		<input type="checkbox"/>

SETUP > OBJECT MANAGER

Vehicle

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

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

9 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Price	Price__c	Currency(18, 0)		
Status	Status__c	Picklist		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		
Vehicle Dealer	Vehicle_Dealer__c	Lookup(Vehicle Dealer)		✓
Vehicle Model	Vehicle_Model__c	Picklist		
Vehicle Name	Name	Text(80)		✓

SETUP > OBJECT MANAGER

Vehicle Order

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

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

9 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assigned Dealer	Assigned_Dealer__c	Lookup(Vehicle Dealer)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User,Group)		✓
Status	Status__c	Picklist		
Vehicle	Vehicle__c	Lookup(Vehicle)		✓
Vehicle Customer	Vehicle_Customer__c	Lookup(Vehicle Customer)		✓
Vehicle Order Number	Name	Auto Number		✓

Setup

Home

Object Manager

Search Setup

Star

Grid

Cloud

Help

Settings

Notifications

Profile

Setup > OBJECT MANAGER

Vehicle Service Request

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

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

9 Items, Sorted by Field Label

Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(User)			
Customer	Customer__c	Lookup(Customer)		✓	
Issue Description	Issue_Description__c	Text(60)			
Last Modified By	LastModifiedById	Lookup(User)			
Owner	OwnerId	Lookup(User/Group)		✓	
Service Date	Service_Date__c	Date			
Status	Status__c	Picklist			
Vehicle	Vehicle__c	Lookup(Vehicle)		✓	
Vehicle Service Request Name	Name	Text(80)		✓	

Star

Grid

Cloud

Help

Settings

Notifications

Profile

Setup

Home

Object Manager

Search Setup

Star

Grid

Cloud

Help

Settings

Notifications

Profile

Setup > OBJECT MANAGER

Vehicle Test Drive

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

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

8 Items, Sorted by Field Label

Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(User)			
Last Modified By	LastModifiedById	Lookup(User)			
Owner	OwnerId	Lookup(User/Group)		✓	
Status	Status__c	Picklist			
Test Drive Date	Test_Drive_Date__c	Date			
Vehicle	Vehicle__c	Lookup(Vehicle)		✓	
Vehicle Customer	Vehicle_Customer__c	Lookup(Vehicle Customer)		✓	
Vehicle Test Drive Name	Name	Text(80)		✓	

Star

Grid

Cloud

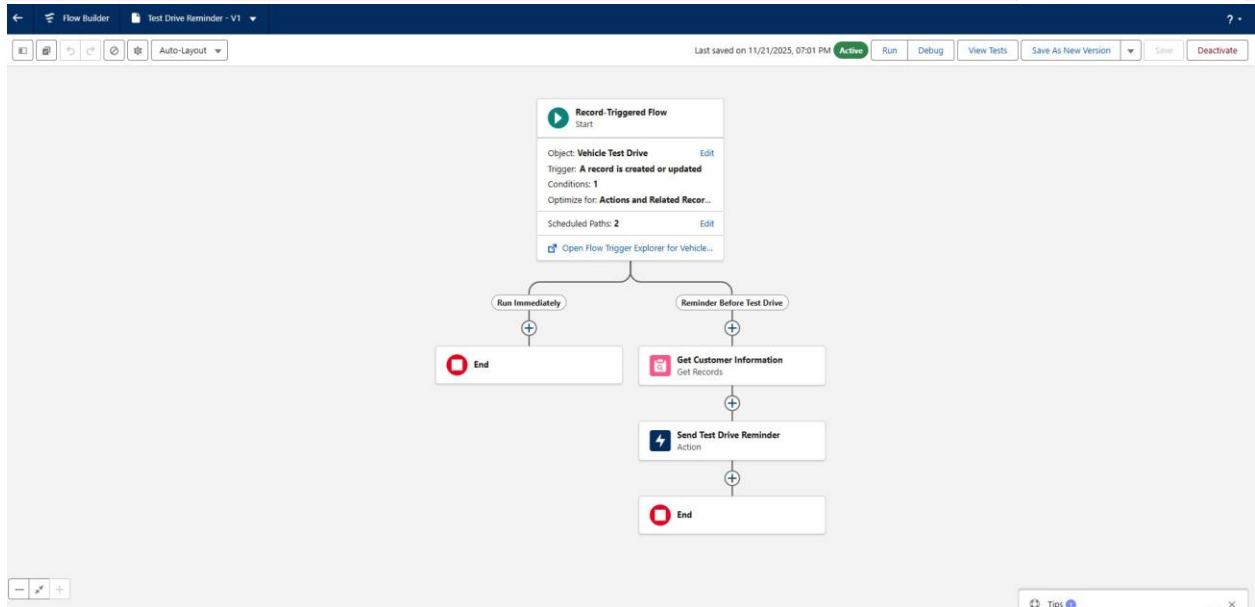
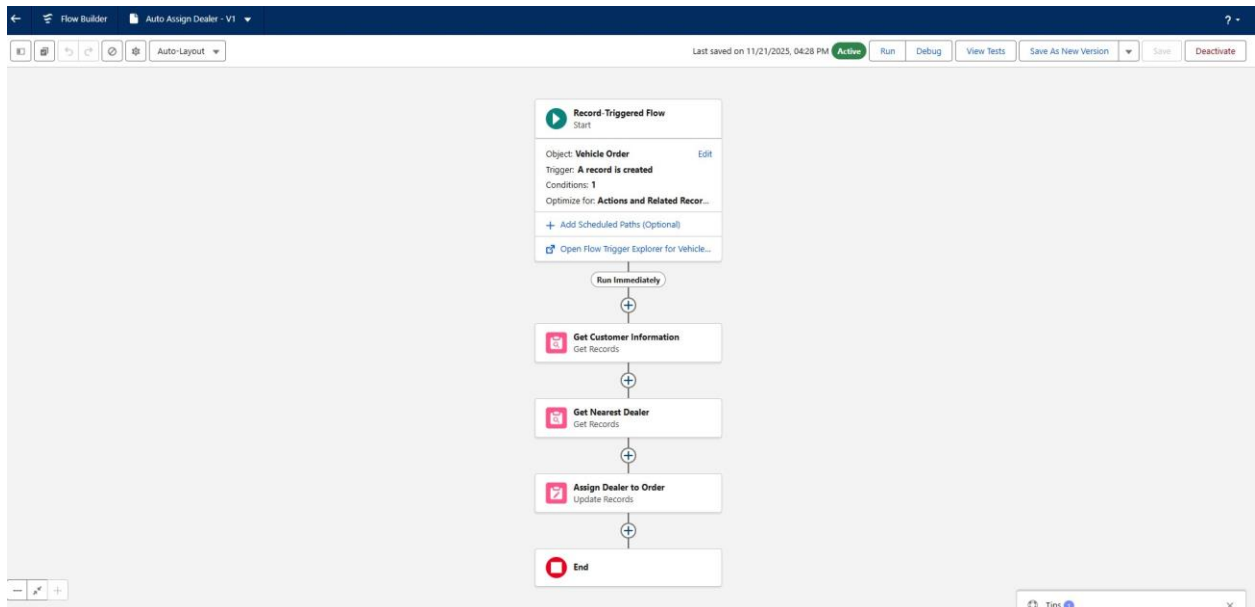
Help

Settings

Notifications

Profile

Automation



WhatNext Vision Motors
Vehicle Customers
Vehicle Dealers
Vehicle Service Requests
Vehicle Orders
Vehicle Test Drives
Vehicles
Reports
Dashboards

Vehicle Order
O-0001

New Contact
Edit
New Opportunity

Related
Details

Vehicle Order Number	O-0001	Owner	Jermaine Micah Rianzares
Vehicle	Honda		
Order Date	11/21/2025		
Status	Pending		
Assigned Dealer			
Vehicle Customer	John		
Created By	Jermaine Micah Rianzares, 11/21/2025, 12:42 AM	Last Modified By	Jermaine Micah Rianzares, 11/21/2025, 12:42 AM

Apex and Batch Class

```

File • Edit • Debug • Test • Workspace • Help • < >
VehicleOrderTriggerHandler.apex • VehicleOrderTrigger.apex • VehicleOrderBatch.apex • VehicleOrderBatchScheduler.apex
Code Coverage: None • API Version: 65
1 public class VehicleOrderTriggerHandler {
2
3     public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isInsert, Boolean isUpdate) {
4         if (isBefore && (isInsert || isUpdate)) {
5             preventOrderIfoutofStock(newOrders);
6         }
7
8         if (isAfter && (isInsert || isUpdate)) {
9             updateStockOnOrderPlacement(newOrders);
10        }
11    }
12
13    private static void preventOrderIfoutofStock(List<Vehicle_Order__c> orders) {
14        Set<Id> vehicleIds = new Set<Id>();
15        for (Vehicle_Order__c order : orders) {
16            if (order.Vehicle__c != null) {
17                vehicleIds.add(order.Vehicle__c);
18            }
19        }
20
21        if (!vehicleIds.isEmpty()) {
22            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>{
23                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
24            };
25
26            for (Vehicle_Order__c order : orders) {
27                Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
28                if (vehicle != null && vehicle.Stock_Quantity__c <= 0) {
29                    order.addError('This vehicle is out of stock. Order cannot be placed.');
```

```
File • Edit • Debug • Test • Workspace • Help • < >
VehicleOrderTriggerHandler.apxc | VehicleOrderTrigger.apxt | VehicleOrderBatch.apxc | VehicleOrderBatchScheduler.apxc
Code Coverage: None | API Version: 65 | Go To

30     }
31   }
32 }
33
34
35 private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> orders) {
36   Set<Id> vehicleIds = new Set<Id>();
37   for (Vehicle_Order__c order : orders) {
38     if (order.Vehicle__c != null && order.Status__c == 'Confirmed') {
39       vehicleIds.add(order.Vehicle__c);
40     }
41   }
42
43   if (!vehicleIds.isEmpty()) {
44     Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>{
45       [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
46     };
47
48     List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
49     for (Vehicle_Order__c order : orders) {
50       Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
51       if (vehicle != null && vehicle.Stock_Quantity__c > 0) {
52         vehicle.Stock_Quantity__c -= 1;
53         vehiclesToUpdate.add(vehicle);
54       }
55     }
56
57     if (!vehiclesToUpdate.isEmpty()) {
58       update vehiclesToUpdate;
59     }
60   }
61 }
62 }
```

```
File • Edit • Debug • Test • Workspace • Help • < >
VehicleOrderTriggerHandler.apxc | VehicleOrderTrigger.apxt | VehicleOrderBatch.apxc | VehicleOrderBatchScheduler.apxc
Code Coverage: None | API Version: 65 | Go To

1 trigger VehicleOrderTrigger on Vehicle_Order__c (before insert, before update, after insert, after update) {
2   VehicleOrderTriggerHandler.handleTrigger(trigger.new, trigger.oldMap, trigger.isBefore, trigger.isAfter, trigger.isInsert, trigger.isUpdate);
3 }
4 }
```

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt VehicleOrderBatch.apxc VehicleOrderBatchScheduler.apxc
Code Coverage: None API Version: 65 ▾

1 global class VehicleOrderBatch implements Database.Batchable<sObject> {
2
3     global Database.QueryLocator start(Database.BatchableContext bc) {
4         return Database.getQueryLocator([
5             SELECT Id, Status__c, Vehicle__c FROM Vehicle_Order__c WHERE Status__c = 'Pending'
6         ]);
7     }
8
9     global void execute(Database.BatchableContext bc, List<Vehicle_Order__c> orderList) {
10         Set<Id> vehicleIds = new Set<Id>();
11         for (Vehicle_Order__c order : orderList) {
12             if (order.Vehicle__c != null) {
13                 vehicleIds.add(order.Vehicle__c);
14             }
15         }
16
17         if (!vehicleIds.isEmpty()) {
18             Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>([
19                 SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds
20             ]);
21
22             List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
23             List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
24
25             for (Vehicle_Order__c order : orderList) {
26                 Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
27                 if (vehicle != null && vehicle.Stock_Quantity__c > 0) {
28                     order.Status__c = 'Confirmed';
29                     vehicle.Stock_Quantity__c -= 1;
30                     ordersToUpdate.add(order);
31                     vehiclesToUpdate.add(vehicle);
32                 }
33             }
34         }
35     }
36 }
```

```
File Edit Debug Test Workspace Help < >
VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt VehicleOrderBatch.apxc VehicleOrderBatchScheduler.apxc
Code Coverage: None API Version: 65
11 for (Vehicle_Order__c order : orderList) {
12     if (order.Vehicle__c != null) {
13         vehicleIds.add(order.Vehicle__c);
14     }
15 }
16
17 if (!vehicleIds.isEmpty()) {
18     Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
19         [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
20     );
21
22     List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
23     List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
24
25     for (Vehicle_Order__c order : orderList) {
26         Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
27         if (vehicle != null && vehicle.Stock_Quantity__c > 0) {
28             order.Status__c = 'Confirmed';
29             vehicle.Stock_Quantity__c -= 1;
30             ordersToUpdate.add(order);
31             vehiclesToUpdate.add(vehicle);
32         }
33     }
34
35     if (!ordersToUpdate.isEmpty()) update ordersToUpdate;
36     if (!vehiclesToUpdate.isEmpty()) update vehiclesToUpdate;
37 }
38
39
40 global void finish(Database.BatchableContext bc) {
41     System.debug('Vehicle order batch job completed.');
```

```
File Edit Debug Test Workspace Help < >
VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt VehicleOrderBatch.apxc VehicleOrderBatchScheduler.apxc
Code Coverage: None API Version: 65
1 global class VehicleOrderBatchScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         VehicleOrderBatch batchJob = new VehicleOrderBatch();
4         Database.executeBatch(batchJob, 50); // 50 = batch size
5     }
6 }
```