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

ABSTRACT

WhatsNext Vision Motors, a pioneering force in the automotive industry, is dedicated to transforming the mobility sector with innovative technology and solutions that prioritize customer needs. The company has embarked on an ambitious Salesforce project with the core objective of enhancing the customer experience and streamlining its operational processes.

The Project consist of 6 Major phases

1. Data Management-Objects
2. Data Management-Tabs
3. Data Management-App Manager
4. Data Management-Fields
5. Automation
6. Apex and Batch Classes

It aims to create a more efficient ordering system that reduces the potential for errors and improves the overall service provided to customers.

The main objective of the project is to make it easy for the customer to order vehicles and to help the company to manage the stock and orders effectively using Salesforce CRM.

Benefits:

1. Customers:

- Get faster and clearer Service.
- Don’t face issues due to out-of-stock vehicles.

2. Company:

- Staff save time (less manual checking).
- Order system becomes smoother and more accurate.
- Company looks more professional and reliable.

PHASE 1: REQUIREMENT ANALYSIS & PLANNING

Scope:

The project streamlines the vehicle ordering process by auto-assigning orders to the nearest dealer based on customer location and preventing orders for out-of-stock vehicles. Automated workflows update order statuses dynamically and send scheduled email reminders for test drives.

Objective:

1. The system will automatically suggest the nearest car dealer to the customer based on where they live thus saving customer time and effort while ordering.

2. Customers can order only those vehicles which are in stock, if the vehicle is not available the system won't let them to order thus avoiding confusion and disappointments.

3. There is scheduled system process that regularly checks the order status: If the vehicle is available, the order becomes confirmed or else pending thus keeps order status up-to-date and clear for both company and customers through emails.

Data Model:

1. Data Management-Objects : creating different objects like dealer, vehicle, customer etc

2. Data Management-Tabs : Tabs were created for easy navigation and access to key objects in the Salesforce UI

3. Data Management-App Manager : Combine related tabs, objects, and processes into a single app interface

4. Data Management-Fields : Custom Fields were added to support the business logic

5. Automation : Assign nearest dealer automatically based on customer address using geolocation logic and giving them email for test drive remainder

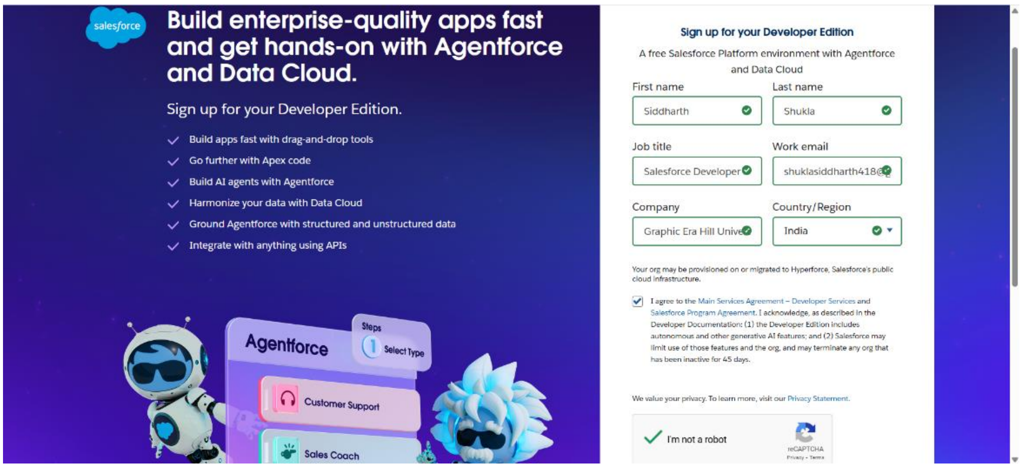
6. Apex and Batch Classes : Used for more complex logic where Flows weren't enough

Security Model:

- Vehicle: Private (to protect stock details)
- Order: Controlled by Parent
- Dealer: Public Read Only

PHASE 2:SALESFORCE DEVELOPMENT - BACKEND & CONFIGURATIONS

1. Developer Account Creation and verification of credentials



Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- Build apps fast with drag-and-drop tools
- Go further with Apex code
- Build AI agents with Agentforce
- Harmonize your data with Data Cloud
- Ground Agentforce with structured and unstructured data
- Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Siddharth ✓	Last name: Shukla ✓
Job title: Salesforce Developer ✓	Work email: shuklasiddharth418@ ✓
Company: Graphic Era Hill Unive ✓	Country/Region: India ✓

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

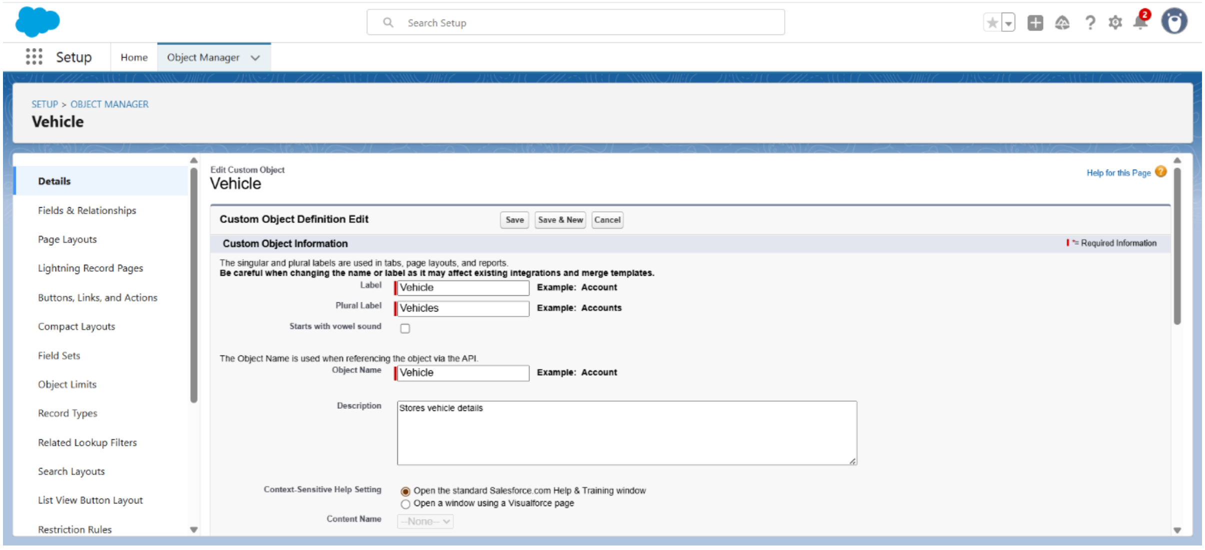
☑ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

✓ I'm not a robot

2. Creation of objects from object manager and creating relationships

Object Name	Purpose	Relationships
Vehicle__c	Stores vehicle details	Related to Dealer & Orders
Vehicle_Dealer__c	Stores authorized dealer info	Related to Orders
Vehicle_Customer__c	Stores customer details	Related to Orders & Test Drives
Vehicle_Order__c	Tracks vehicle purchases	Related to Customer & Vehicle
Vehicle_Test_Drive__c	Tracks test drive bookings	Related to Customer & Vehicle
Vehicle_Service_Request__c	Tracks vehicle servicing requests	Related to Customer & Vehicle



Edit Custom Object

Vehicle

Custom Object Definition Edit

Custom Object Information | = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.
Be careful when changing the name or label as it may affect existing integrations and merge templates.

Label: Vehicle — **Example: Account**

Plural Label: Vehicles — **Example: Accounts**

Starts with vowel sound: □

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

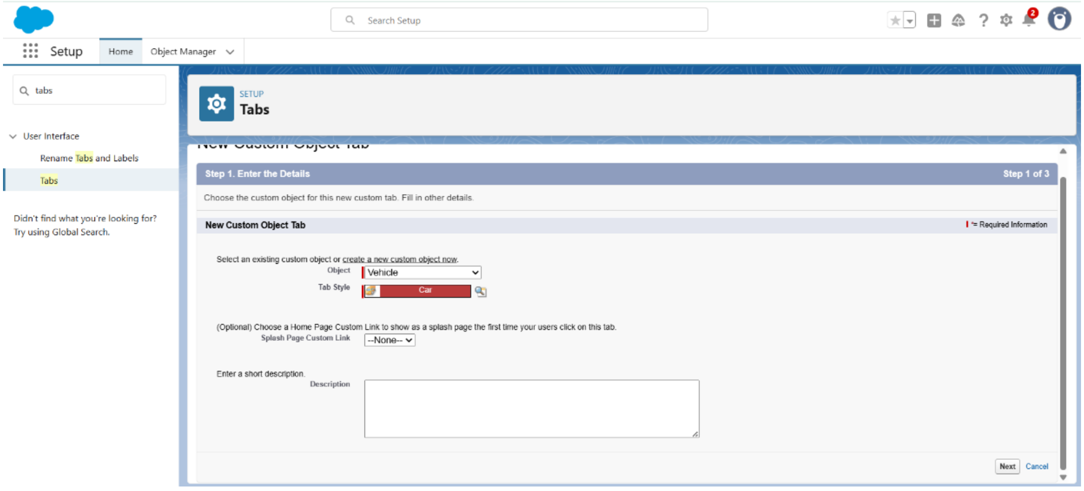
Object Name: Vehicle — **Example: Account**

Description: Stores vehicle details

Context-Sensitive Help Setting: ● Open the standard Salesforce.com Help & Training window
○ Open a window using a Visualforce page

Content Name: --None--

3. Creating custom tabs for lightning app for easing the use of customers



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: Vehicle

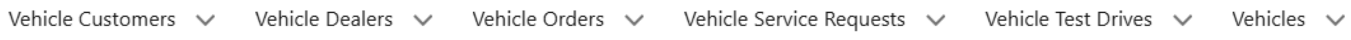
Tab Style: Car

(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:



4. Creation of fields and relationships

Vehicle__c: Stores details of each vehicle, including model type, stock availability, price, and associated dealer.

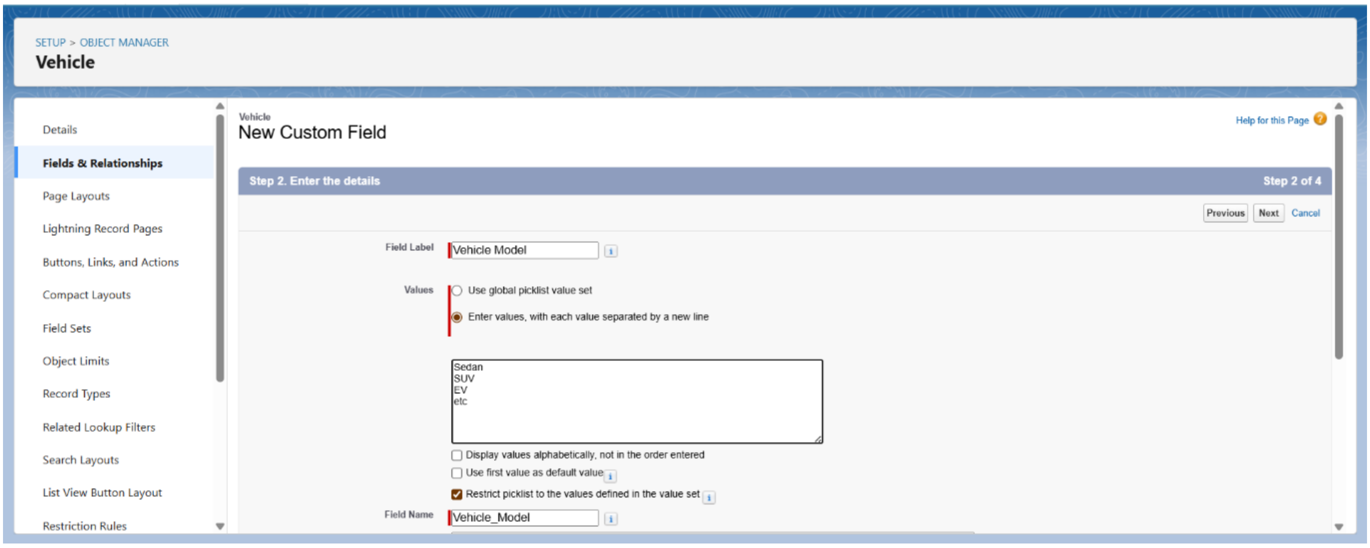
Vehicle_Dealer__c: Represents dealership information such as name, location, contact details, and unique dealer code.

Vehicle_Order__c: Captures customer vehicle orders, linking to the selected vehicle and tracking order status and date.

Vehicle_Customer__c: Stores customer profile data, including contact info, address, and preferred vehicle type.

Vehicle_Test_Drive__c: Logs customer test drive bookings with selected vehicle, date, and status.

Vehicle_Service_Request__c: Tracks vehicle service requests made by customers, including issue details and current service status.



Step 2. Enter the details — Step 2 of 4

Field Label: Vehicle Model

Values:
○ Use global picklist value set
● Enter values, with each value separated by a new line

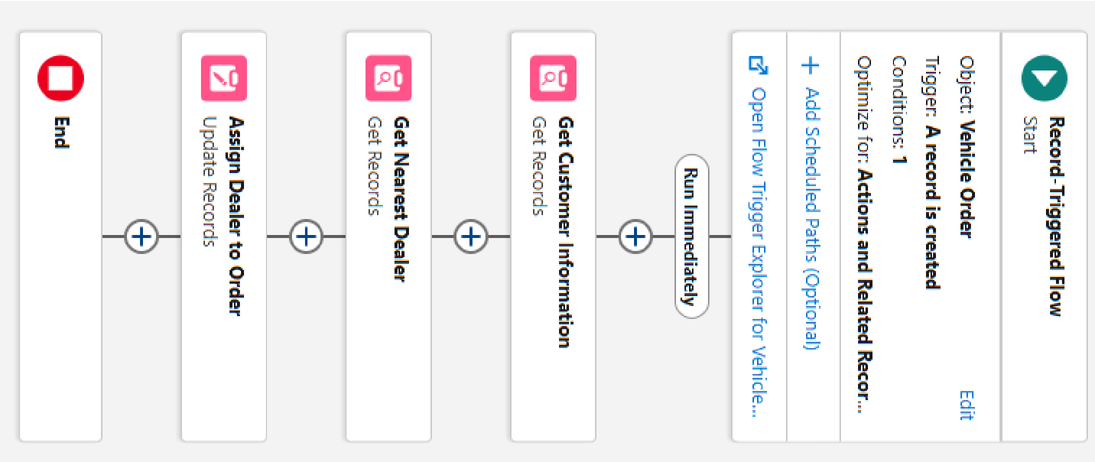
Sedan
SUV
EV
etc

□ Display values alphabetically, not in the order entered
□ Use first value as default value
☑ Restrict picklist to the values defined in the value set

Field Name: Vehicle_Model

5. Automation using flows and creation of flows using flow builder

- record triggered flow to assign nearest dealer to the customer’s location



Record-Triggered Flow
Start

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

Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Vehicle...

Run Immediately

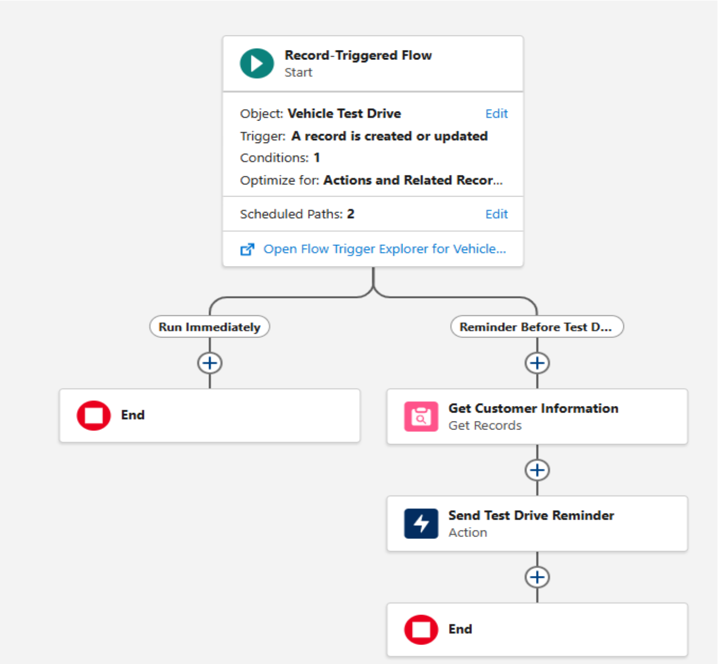
Get Customer Information
Get Records

Get Nearest Dealer
Get Records

Assign Dealer to Order
Update Records

End

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



Record-Triggered Flow
Start

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

Scheduled Paths: 2
Edit

Open Flow Trigger Explorer for Vehicle...

Run Immediately

End

Reminder Before Test D...

Get Customer Information
Get Records

Send Test Drive Reminder
Action

End

6. Apex Classes and Triggers.

- This Apex trigger handler class ensures that vehicle orders cannot be placed if the stock is zero, and automatically reduces the stock quantity by 1 when an order is confirmed.

```
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 && (isInsert || isUpdate)) {
            preventOrderIfOutOfStock(newOrders);
        }
        if (isAfter && (isInsert || isUpdate)) {
            updateStockOnOrderPlacement(newOrders);
        }
    }
   //Prevent placing an order if stock is zero
   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>(
            [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
         );
         for (Vehicle_Order__c order : orders) {
            Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
            if (vehicle != null && vehicle.Stock_Quantity__c <= 0) {
               order.addError('This vehicle is out of stock. Order cannot be placed.');
            }
         }
      }
   }
   //Decrease stock when an order is confirmed
   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>(
            [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
         );
         List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
         for (Vehicle_Order__c order : orders) {
            Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
            if (vehicle != null && vehicle.Stock_Quantity__c > 0) {
               vehicle.Stock_Quantity__c -= 1;
               vehiclesToUpdate.add(vehicle);
            }
         }
         if (!vehiclesToUpdate.isEmpty()) {
            update vehiclesToUpdate;
         }
      }
   }
}
```

- This **Batch Apex class** processes all Pending vehicle orders in bulk, confirms them if stock is available, and updates the vehicle stock accordingly — ensuring automated and scheduled order fulfillment.

```
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>(
                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
            );

            List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
            for (Vehicle_Order__c order : orderList) {
                Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                if (vehicle != null && 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.');
    }
}
```

- This **Scheduler class** runs the VehicleOrderBatch job at specified intervals, automatically processing pending vehicle orders in batches of 50 to update their status and manage stock.

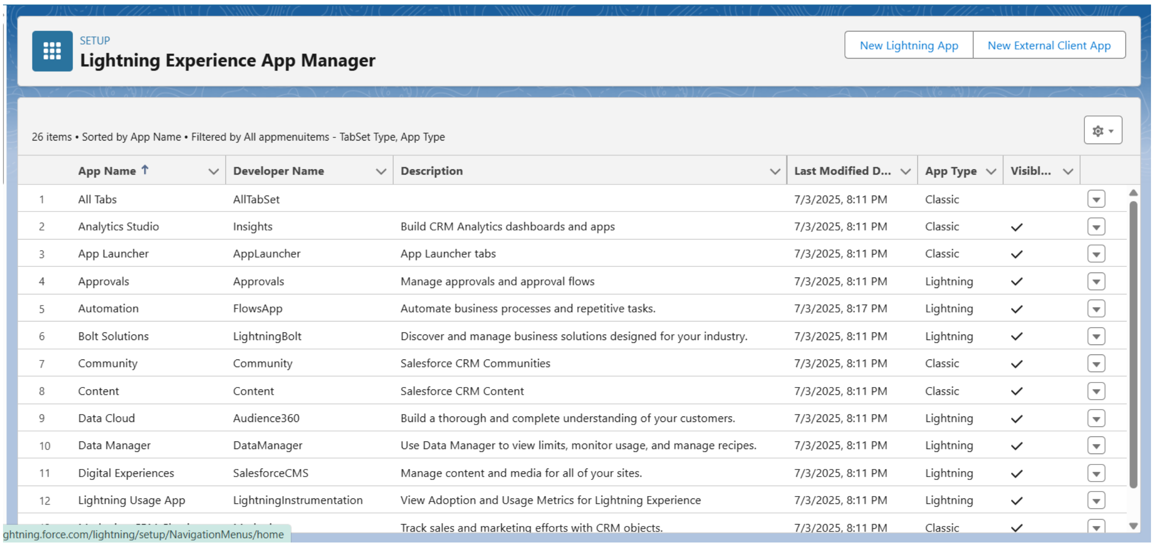
```
global class VehicleOrderBatchScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        VehicleOrderBatch batchJob = new VehicleOrderBatch();
        Database.executeBatch(batchJob, 50); // 50 = batch size
    }
}
```

- This trigger calls the VehicleOrderTriggerHandler to manage logic before and after inserting or updating vehicle orders, ensuring order validation and stock updates are handled centrally and efficiently.

```
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);
}
```

PHASE 3:UI/UX DEVELOPMENT & CUSTOMIZATION

1. Creating of lightning app using lightning app builder by go to app manager.



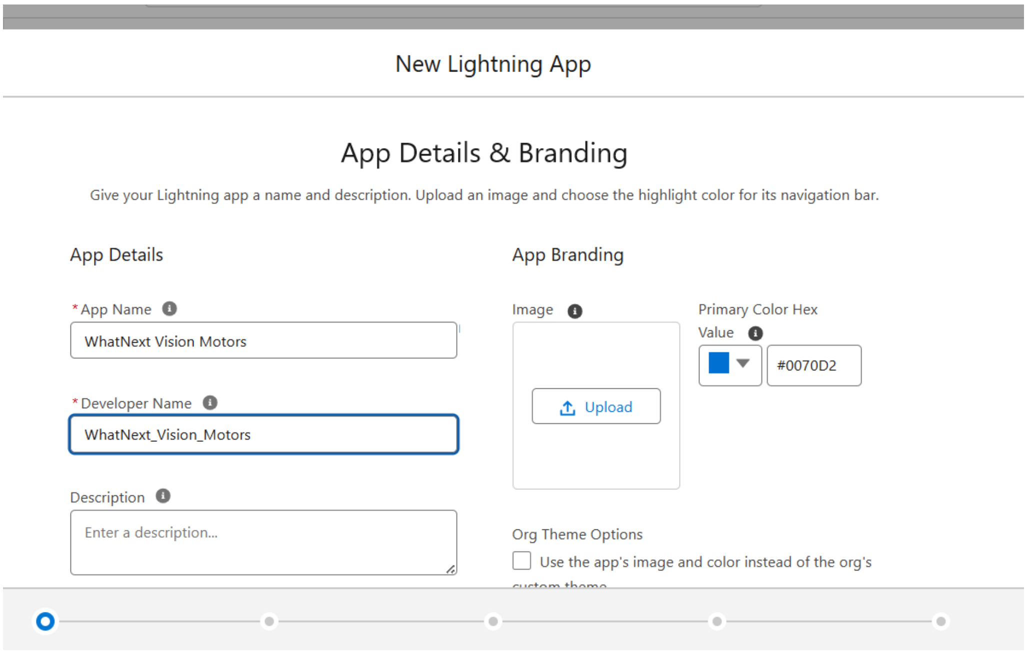
SETUP

Lightning Experience App Manager

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

	App Name	Developer Name	Description	Last Modified D...	App Type	Visibl...
1	All Tabs	AllTabSet		7/3/2025, 8:11 PM	Classic	
2	Analytics Studio	Insights	Build CRM Analytics dashboards and apps	7/3/2025, 8:11 PM	Classic	✓
3	App Launcher	AppLauncher	App Launcher tabs	7/3/2025, 8:11 PM	Classic	✓
4	Approvals	Approvals	Manage approvals and approval flows	7/3/2025, 8:11 PM	Lightning	✓
5	Automation	FlowsApp	Automate business processes and repetitive tasks.	7/3/2025, 8:17 PM	Lightning	✓
6	Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	7/3/2025, 8:11 PM	Lightning	✓
7	Community	Community	Salesforce CRM Communities	7/3/2025, 8:11 PM	Classic	✓
8	Content	Content	Salesforce CRM Content	7/3/2025, 8:11 PM	Classic	✓
9	Data Cloud	Audience360	Build a thorough and complete understanding of your customers.	7/3/2025, 8:11 PM	Lightning	✓
10	Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	7/3/2025, 8:11 PM	Lightning	✓
11	Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	7/3/2025, 8:11 PM	Lightning	✓
12	Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	7/3/2025, 8:11 PM	Lightning	✓
			Track sales and marketing efforts with CRM objects.	7/3/2025, 8:11 PM	Classic	✓

ghtning.force.com/lightning/setup/NavigationMenus/home



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: WhatNext Vision Motors

*Developer Name: WhatNext_Vision_Motors

Description: Enter a description...

App Branding

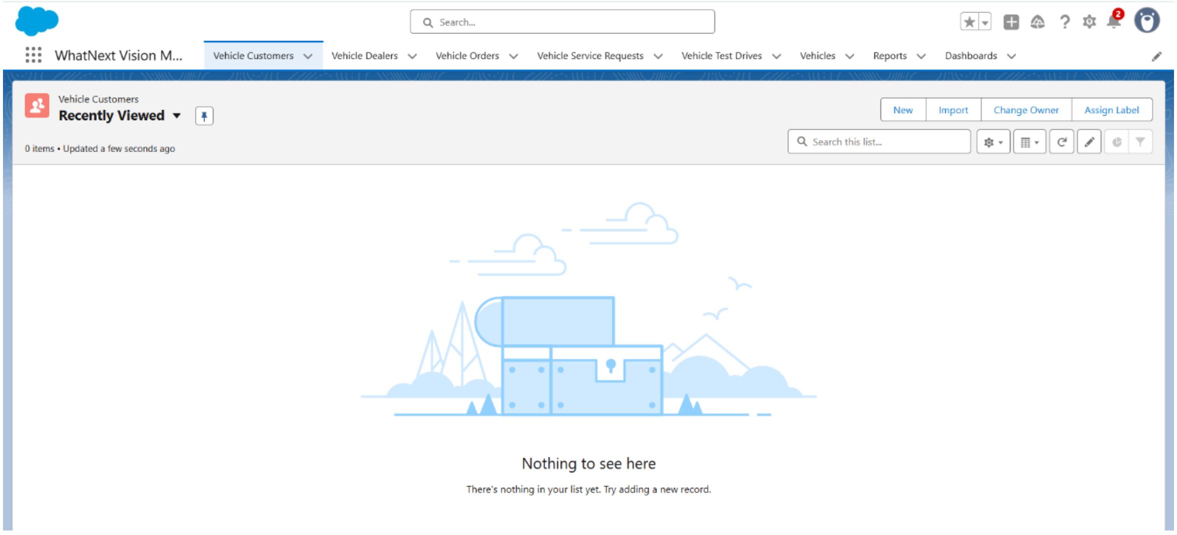
Image

Primary Color Hex Value: #0070D2

Org Theme Options

□ Use the app's image and color instead of the org's custom theme

2. After that add all the objects to lightning app and it will be shown like this



Vehicle Customers

Recently Viewed

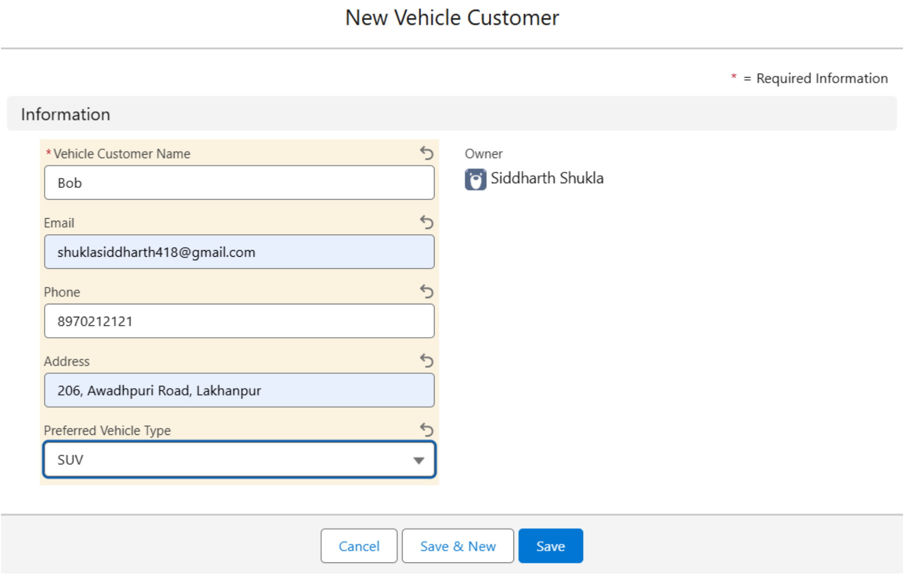
0 items • Updated a few seconds ago

Nothing to see here

There's nothing in your list yet. Try adding a new record.

PHASE 4:DATA MIGRATION, TESTING & SECURITY

1. Creation of new customers



New Vehicle Customer

* = Required Information

Information

*Vehicle Customer Name: Bob

Email: shuklasiddharth418@gmail.com

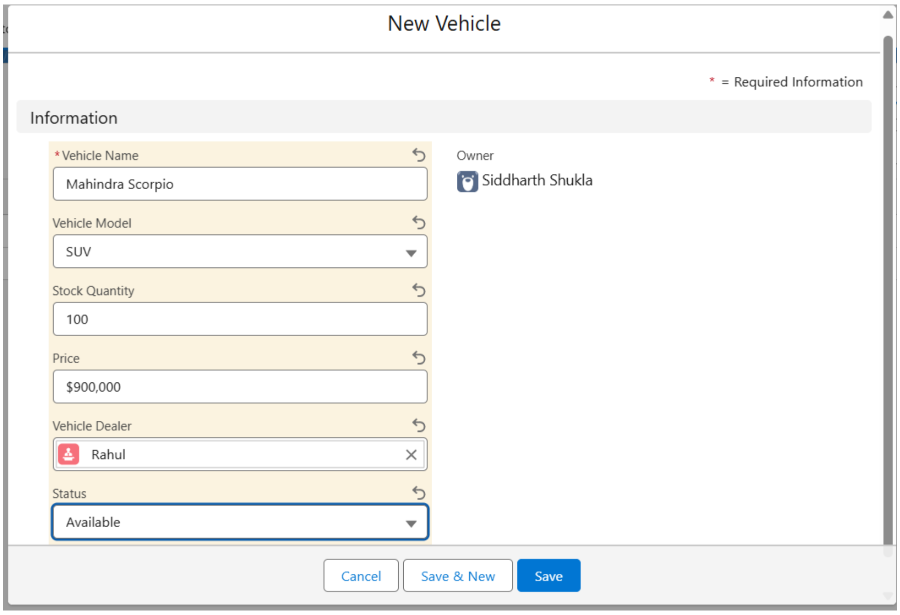
Phone: 8970212121

Address: 206, Awadhpuri Road, Lakhanpur

Preferred Vehicle Type: SUV

Owner: Siddharth Shukla

2. Creation of new vehicle



New Vehicle

* = Required Information

Information

*Vehicle Name: Mahindra Scorpio

Vehicle Model: SUV

Stock Quantity: 100

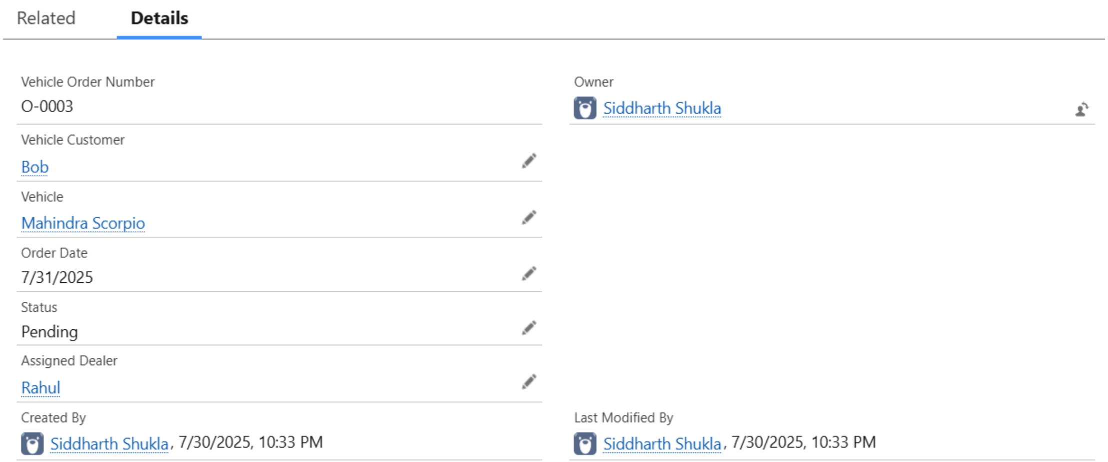
Price: $900,000

Vehicle Dealer: Rahul

Status: Available

Owner: Siddharth Shukla

3. Automatic assigning of nearest dealer on creating vehicle order



Vehicle Order Number: O-0003

Owner: Siddharth Shukla

Vehicle Customer: Bob

Vehicle: Mahindra Scorpio

Order Date: 7/31/2025

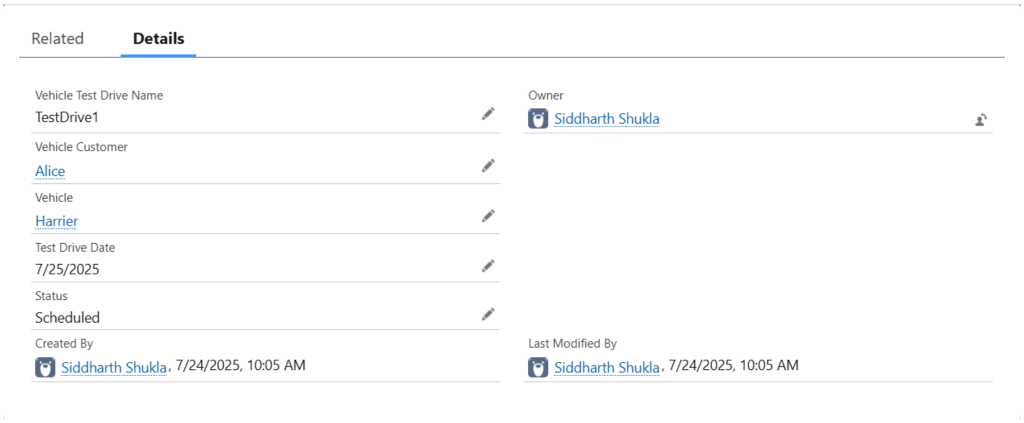
Status: Pending

Assigned Dealer: Rahul

Created By: Siddharth Shukla, 7/30/2025, 10:33 PM

Last Modified By: Siddharth Shukla, 7/30/2025, 10:33 PM

4. Email for remainder of test drive one day before



Vehicle Test Drive Name
TestDrive1

Owner
Siddharth Shukla

Vehicle Customer
Alice

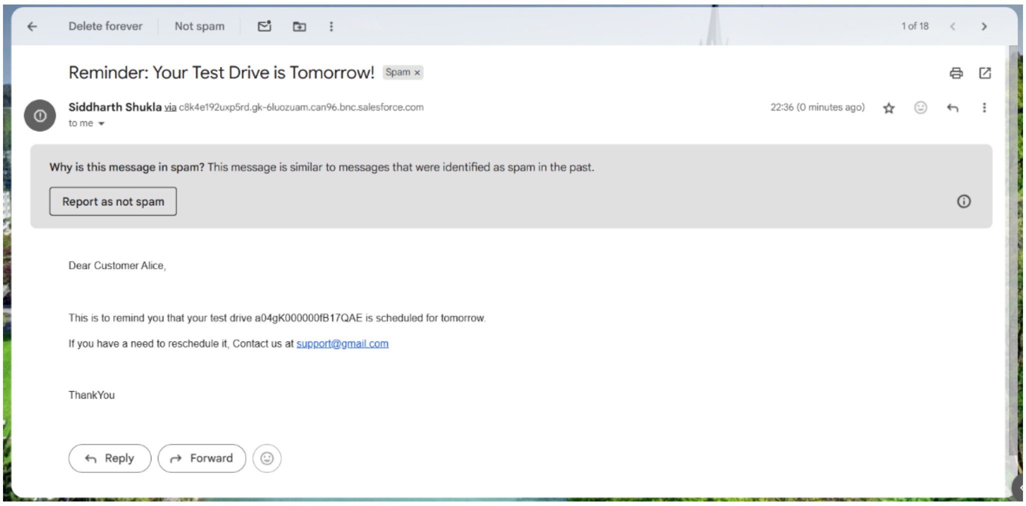
Vehicle
Harrier

Test Drive Date
7/25/2025

Status
Scheduled

Created By
Siddharth Shukla, 7/24/2025, 10:05 AM

Last Modified By
Siddharth Shukla, 7/24/2025, 10:05 AM



Reminder: Your Test Drive is Tomorrow!

Siddharth Shukla via c8k4e192uxp5rd.gk-6luozuam.can96.bnc.salesforce.com
to me
22:36 (0 minutes ago)

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

Dear Customer Alice,

This is to remind you that your test drive a04gK000000fB17QAE is scheduled for tomorrow.

If you have a need to reschedule it, Contact us at support@gmail.com

ThankYou

Test Case 1: Vehicle Order Creation

- **Feature:** Prevent order if vehicle is out of stock
- **Input:** Try placing an order for a vehicle with 0 stock
- **Expected Output:** Error message “Vehicle is out of stock”

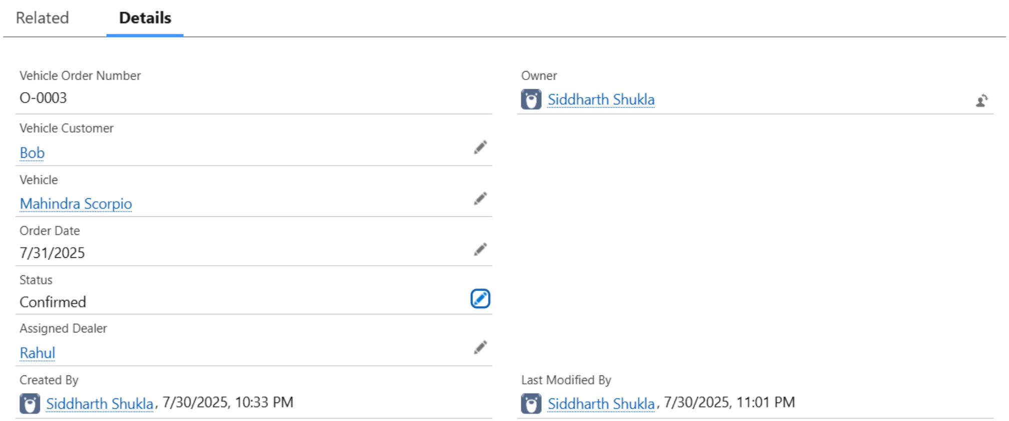
Test Case 2: Flow - Nearest Dealer Suggestion

- **Input:** Enter a customer address in the order form
- **Expected Output:** Nearest dealer is suggested automatically

Test Case 3: Email for remainder of test drive one day before

- **Input:** Enter the date in the form one after the current date
- **Expected Output:** You will get the email for test drive remainder

5. Changing the status to confirm and to sure that stock quantity decreases



Vehicle Order Number
O-0003

Owner
Siddharth Shukla

Vehicle Customer
Bob

Vehicle
Mahindra Scorpio

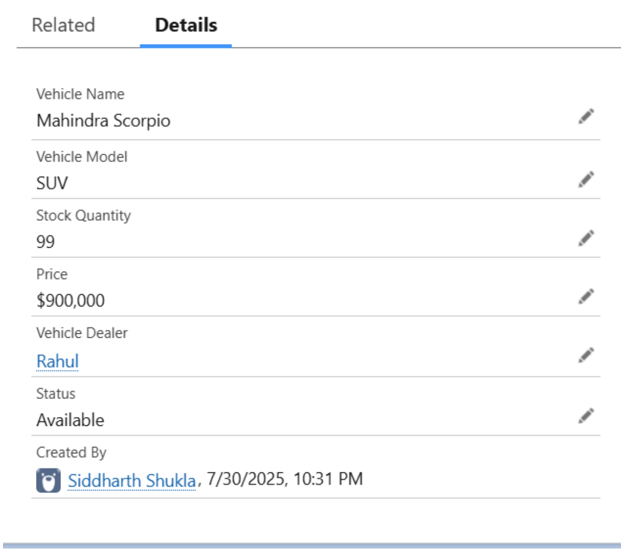
Order Date
7/31/2025

Status
Confirmed

Assigned Dealer
Rahul

Created By
Siddharth Shukla, 7/30/2025, 10:33 PM

Last Modified By
Siddharth Shukla, 7/30/2025, 11:01 PM



Related | **Details**

Vehicle Name
Mahindra Scorpio

Vehicle Model
SUV

Stock Quantity
99

Price
$900,000

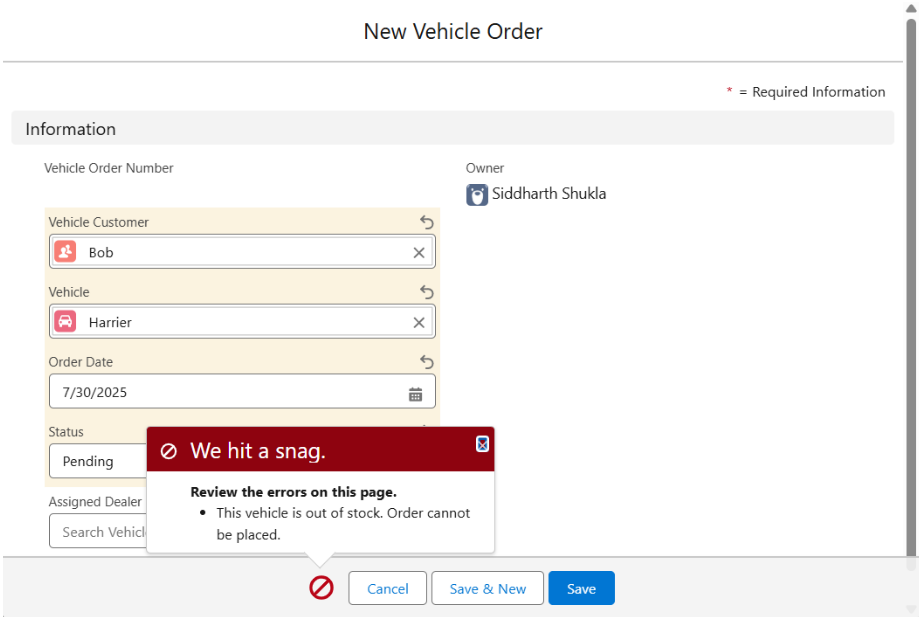
Vehicle Dealer
Rahul

Status
Available

Created By
Siddharth Shukla, 7/30/2025, 10:31 PM

(decrease to 99)

6. For checking vehicle out of stock if we set the quantity to zero and if we try to save the new order



New Vehicle Order

* = Required Information

Information

Vehicle Order Number

Owner
Siddharth Shukla

Vehicle Customer
Bob

Vehicle
Harrier

Order Date
7/30/2025

Status
Pending

Assigned Dealer
Search Vehicl

We hit a snag.

Review the errors on this page.

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

PHASE 5:DEPLOYMENT, DOCUMENTATION & MAINTENANCE

1. The deployment from the **Sandbox** to the **Production environment** was carried out using **Change Sets**, Salesforce's native point-and-click deployment method. The following steps were followed:
All metadata components such as:

  - Custom Objects and Fields (Vehicle__c, Vehicle_Order__c, etc.)
  - Validation Rules and Flows
  - Apex Classes (Trigger Handler, Batch, Scheduler)
  - Triggers

- Were added to an **Outbound Change Set** in the Developer Sandbox.

- The change set was uploaded to the Production org and validated for deployment errors.

- Post-validation, the deployment was completed, and smoke testing was done to confirm successful setup.

2. To ensure the solution continues to work as expected in production, the following monitoring and maintenance practices will be used:

Monitoring Scheduled Jobs

- The VehicleOrderBatchScheduler job is scheduled to update order statuses daily.
- Apex Job logs will be monitored through **Setup → Apex Jobs** for success/failure.

Debug Logs and Error Tracking

- Debug logs will be enabled for key users and developers to trace trigger or flow failures.
- Errors such as "Vehicle out of stock" are caught via validations and addError() in Apex.

User Feedback & Support

- Periodic reviews of user feedback will help identify gaps and bugs.
- Admins will monitor for unexpected behavior in order processing, dealer assignment, or stock updates.

Data Backup

- Salesforce's native weekly export or external backup tools will be considered to protect critical records like orders and customers.

3. The following table outlines common issues, how to identify them, and recommended resolutions:

Issue	**Diagnosis**	**Resolution**
Order fails to save	Review error message — might show "Vehicle is out of stock"	Ensure Stock_Quantity__c on Vehicle__c is greater than 0
Flow not triggering	Check if the flow is activated and debug logs for failures	Activate latest version of flow and test
Batch job not running	Go to Setup → Apex Jobs to see job status	Re-schedule job via Apex Scheduler or run manually
Dealer not suggested	Lookup field not auto-filling	Revisit flow or formula to fetch nearest dealer

CONCLUSION

The Salesforce implementation at WhatsNext Vision Motors successfully streamlines key business processes, particularly enhancing the customer vehicle ordering experience, dealer allocation, and real-time stock management. Through the use of standard Salesforce features—alongside custom Apex triggers, batch jobs, flows, and validation rules—the system ensures orders are processed accurately, stock levels are reliably updated, and customers are presented with only available options.

Ongoing **maintenance and monitoring mechanisms** like scheduled jobs, debug logs, and user feedback loops have been established to ensure long-term reliability and performance.

With detailed troubleshooting documentation and user support plans in place, the system is well-prepared for scalability and future enhancements. Overall, this solution not only improves operational efficiency but also elevates the overall customer experience for WhatsNext Vision Motors.

FUTURE ENHANCEMENTS

- Implement an AI/ML model or use Einstein Recommendation Builder to suggest the most optimal dealer to the customer, considering not just location but also stock availability, delivery time, and service rating.
- Create a **Customer Community (Experience Cloud)** portal where users can Get personalized offers based on vehicle preferences
- Advanced Inventory Management to Enable **Automatic reorder alerts** for low stock and **Multi-dealer stock visibility**
- Use **Approval Processes** for high-value or customized vehicle orders that require managerial approval before confirmation.