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

Project Overview

WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence is a Salesforce-based system designed to make vehicle buying and servicing easier for both customers and dealers. It brings all important information such as customer details, available vehicles, orders, test drives, and service requests into one place. By automating tasks like finding the nearest dealer, checking vehicle availability, scheduling test drives, and tracking service updates, the system helps users save time and avoid confusion. It aims to create a smoother, faster, and more connected experience for everyone involved.

Objectives

The main goal of this project is to build a user-friendly Salesforce CRM solutions that improves the vehicle ordering process and enhances customer service. Specifically, it aims to:

- To simplify vehicle ordering and servicing by providing a single system where customers and dealers can manage everything easily.
- To improve customer experience through quick access to vehicle availability, dealer suggestions, and service updates.
- To automate common processes such as order management, test-drive scheduling, and service requests to reduce manual work.
- To keep all important information organized in one place to ensure accurate, real-time data for better decision-making.

- To help dealers and staff work more efficiently by providing tools that support faster responses and smoother workflows.

Phase 1: Requirement Analysis & Planning

• Understanding Business Needs

- Gather all important details about the users, such as their name, email, phone number, and address.
- Identify problems in the current process of ordering vehicles, managing dealers, and handling service requests.
- Make sure customer, vehicle, and order information is recorded correctly and easy to find.
- Set up secure, role-based access so only the right people like sales staff, dealer managers, and IT admins can view or edit certain data.

• Defining the Project Scope and Goals

- **Scope:** Build a Salesforce-based CRM system for WhatNext Vision Motors.
- **Main Goals:**
    - Make the vehicle ordering process faster and more organized.
    - Improve how stock or vehicle availability is tracked.
    - Give customers a smoother and more helpful experience.
    - Help the team work more efficiently by reducing manual work.
    - Build stronger customer satisfaction and long-term loyalty.

• Creating the Data and Security Structure

- Design how different objects will connect, such as Vehicle, Dealer, Customer, Order, Test Drive, and Service Request.
- Apply security features like profiles, roles, and permission sets to manage what each type of user can access.

• Identifying Stakeholders

- **Primary stakeholders:** Sales team, dealership managers, customers, and IT administrators.
- **Secondary stakeholders:** Senior leaders and support teams who assist in operations.

• Building the Execution Roadmap

- Prepare a clear step-by-step plan, starting from requirements analysis up to deployment.
- Set target dates and milestones for development, testing, and final implementation.

Phase 2: Salesforce Development – Backend & Configurations

• Setting Up the Development Environment & Workflow

- Prepared a Salesforce sandbox to safely build and test the system.
- Set up version control and deployment processes using tools like **VS Code** and **Git** to make development organized and track changes properly.

• Customizing Objects, Fields, Validation Rules, and Automations

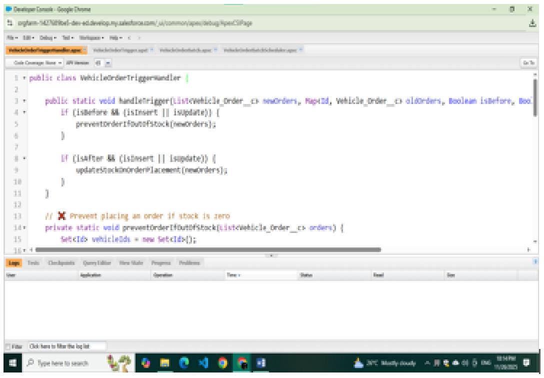
- Created custom objects for the system, including **Vehicle**, **Vehicle Dealer**, **Vehicle Customer**, **Vehicle Order**, **Vehicle Test Drive**, and **Vehicle Service Request**.
- Added customized fields to objects, for example in the **Vehicle object**:
  - Vehicle_Name__c — Text
  - Vehicle_Model__c — Picklist (Sedan, SUV, EV, etc.)
  - Stock_Quantity__c — Number
  - Price__c — Currency
  - Dealer__c — Lookup to Dealer
  - Status__c — Picklist (Available, Out of Stock, Discontinued)
- Set up validation rules, workflows, Process Builder actions, and Flow automations to make processes smoother and reduce manual work.

• Apex Classes, Triggers, and Asynchronous Apex

- Developed Apex classes and triggers to handle backend logic and automated tasks that cannot be done with simple configuration.
- Added asynchronous Apex where needed to process tasks without slowing down the system.

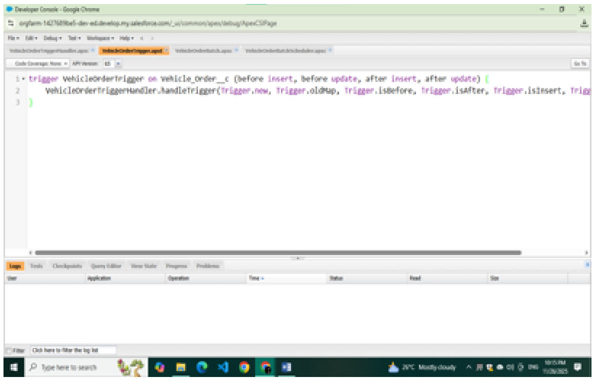
• Documentation

- Included screenshots of Apex classes and triggers as proof of development and for project reference.



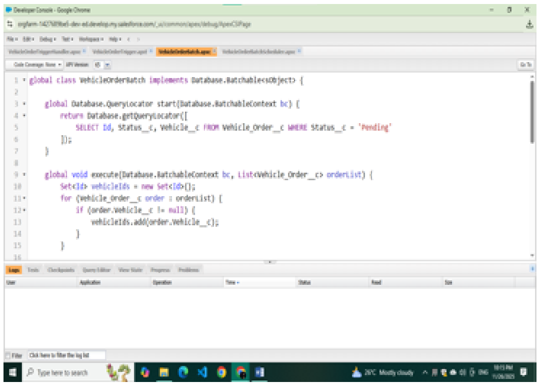
Developer Console - Google Chrome
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list
Type here to search

```
public class VehicleOrderTriggerHandler {

    public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boo
        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>();
```



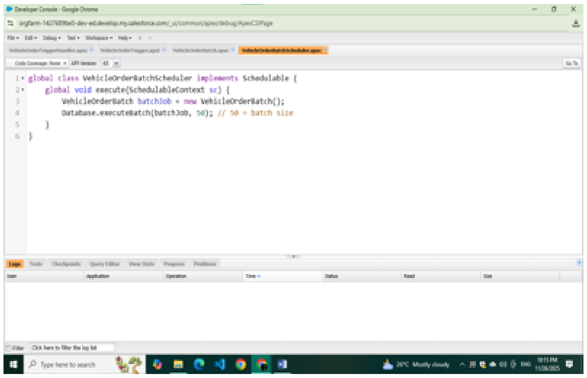
Developer Console - Google Chrome
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list
Type here to search

```
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, Trigg
}
```



Developer Console - Google Chrome
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list
Type here to search

```
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);
            }
        }
```



Developer Console - Google Chrome
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter Click here to filter the log list
Type here to search

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

Phase 3: UI/UX Development & Customization

• Lightning App Setup

- Created a custom Lightning app called **WhatNext Vision Motors** to serve as the main workspace for users.

• Page Layouts and Dynamic Forms

- Organized important fields, buttons, and related lists to make records easy to understand.
- Designed different page layouts for each user role (Vehicle, Dealer, Customer, Order, Test Drive, Service Request) so users only see what's relevant to them.

- Used dynamic forms to show or hide fields based on user actions, making the interface cleaner and easier to use.

• User Management

- Set up profiles, roles, and permission sets to control what each user can view and edit inside the system.

• Reports and Dashboards

- Built real-time dashboards to help track sales, orders, and overall performance.
- Created reports to give users clear and updated insights about customers, vehicles, and business operations.

• LWC Development (if applicable)

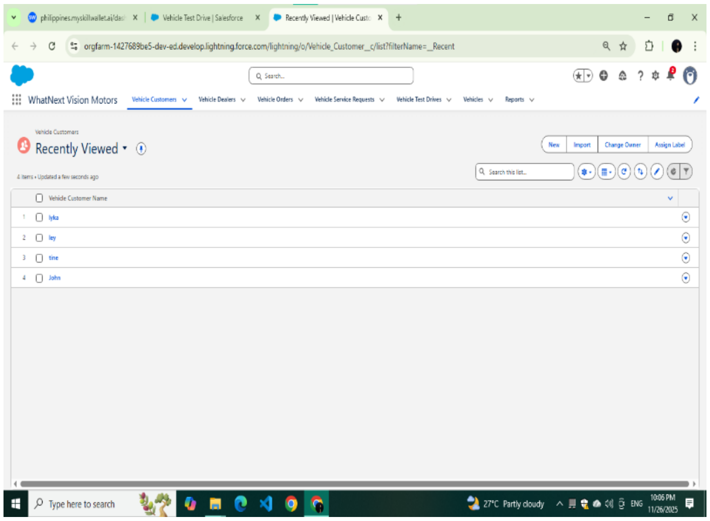
- Developed Lightning Web Components to add richer and more interactive user interface features.

• Lightning Pages Customization

- Customized Lightning Pages to improve navigation and make the system more user-friendly.

• Documentation

- Collected screenshots and notes to document all UI changes and customizations for project reference.



Vehicle Customers

Recently Viewed

4 items • Updated a few seconds ago

	Vehicle Customer Name
1	lyka
2	ley
3	tine
4	John

Phase 4: Data Migration, Testing & Security

• Data Loading Process

- Entered records manually as shown in the demonstration video to ensure accuracy and completeness during the setup.

• Field History Tracking, Duplicate Rules, and Matching Rules

- Enabled history tracking for important fields so changes can be monitored.
- Set up duplicate and matching rules to warn users if a new Lead has the same email or phone number as an existing Contact, helping prevent duplicate records.

• Profiles, Roles, Permission Sets, and Sharing Rules

- Configured security settings to make sure only authorized users can view or edit specific data.

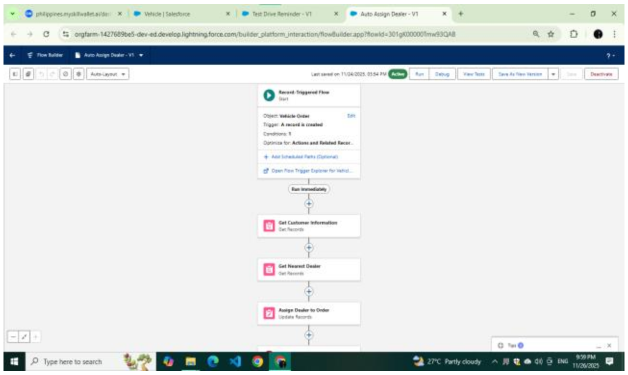
- Ensured each role has the right level of visibility and access based on their responsibilities.

• Creation of Test Classes

- Developed Apex test classes to check if backend code runs correctly and meets Salesforce requirements.

• Documentation

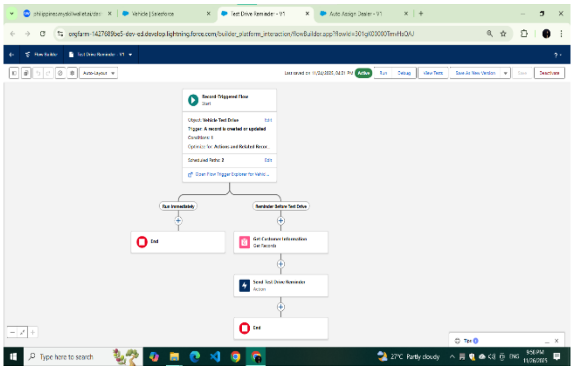
- Included screenshots of automated flows such as **Auto Assign Dealer** and **Test Drive Reminder** for proper recordkeeping and project proof.



orgfarm-1427689be5-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gK000001mw93QAB

Last saved on 11/24/2025, 03:54 PM

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



orgfarm-1427689be5-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gK00000TmvHsQAJ

Last saved on 11/24/2025, 04:21 PM

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

Phase 5: Deployment, Documentation & Maintenance

Deployment Strategy

- Used Salesforce change sets and metadata deployment tools to move all updates from the sandbox to the live environment safely and smoothly.

• System Maintenance & Monitoring

- Performed regular updates, backups, and performance checks to keep the system running well.
- Provided continuous support to users and helped fix any issues they encountered.

• Troubleshooting Documentation

- Prepared step-by-step guides on how to identify and solve common problems, such as reviewing flows and automations, and checking Apex debug logs for errors.

Conclusion

WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence shows how technology can make the vehicle-buying and servicing process easier, faster, and more reliable. With the help of Salesforce, the system brings together all important tasks such as checking vehicle availability, managing orders, scheduling test drives, and handling service requests into one organized platform. This helps both customers and dealers save time and avoid errors. The project proves that using a smart and unified system can greatly improve customer experience and support better decision-making for the business.