Salesforce VIP Implementation Document

Project Title: WhatNext Vision Motors – Salesforce CRM Implementation

# 1. Project Overview / Objective

WhatsNext Vision Motors aimed to transform its customer experience and streamline vehicle ordering and dealership coordination. The key objective was to build an end-to-end Salesforce CRM system that could handle vehicle stock validation, automatic dealer assignment based on customer location, and automated test drive reminders — all while maintaining inventory accuracy through Apex automation.

# 2. Business Requirements / Problems Solved

- Auto-assign orders to the nearest dealer based on customer location.  
- Prevent users from placing orders for out-of-stock vehicles.  
- Automatically update order status after stock availability is confirmed.  
- Send scheduled email reminders for test drives.

# 3. Data Model Design

Custom Objects Created:  
- Vehicle\_\_c (Vehicle Name, Model, Price, Stock, Status)  
- Vehicle\_Dealer\_\_c (Dealer Name, Location, Email, Code)  
- Vehicle\_Customer\_\_c (Customer Name, Contact Info, Preferences)  
- Vehicle\_Order\_\_c (Customer, Vehicle, Dealer, Order Date, Status)  
- Vehicle\_Test\_Drive\_\_c (Customer, Vehicle, Test Drive Date, Status)  
- Vehicle\_Service\_Request\_\_c (Customer, Vehicle, Service Date, Status)  
  
Relationships:  
- Lookup from Order to Customer and Vehicle.  
- Lookup from Vehicle to Dealer.  
- Lookup from Service/Test Drive to Vehicle and Customer.

# 4. Security Model

- Profile used: System Administrator (Full CRUD access for testing).  
- Field-level security enabled for sensitive fields.  
- No public sharing rules configured (Private by default).

# 5. Implementation Steps

- Created 6 custom objects and their fields in Salesforce Object Manager.  
- Built custom tabs for each object.  
- Configured a Lightning App to centralize access to all custom objects.  
- Developed Apex Trigger Handler and Trigger for Order object to check stock and reduce inventory.  
- Created Batch Apex class to confirm pending orders when stock becomes available.  
- Built Scheduler class to invoke batch every night.  
- Designed a Record-Triggered Flow to assign nearest dealer to the order.  
- Created a Scheduled Flow to send email reminders for test drives.

# 6. Apex Code Used

- VehicleOrderTriggerHandler: Validates stock, updates stock.  
- VehicleOrderTrigger: Calls the handler on before/after insert/update.  
- VehicleOrderBatch: Confirms pending orders if stock is available.  
- VehicleOrderBatchScheduler: Schedules the batch class to run daily.

# 7. Flow Automation

- Auto Assign Dealer Flow: Record-triggered on Order creation with Pending status; fetches Customer and assigns nearest Dealer based on location.  
- Test Drive Reminder Flow: Scheduled flow to send email 1 day before scheduled test drive.

# 8. Testing Scenario

Test Case: Create a new Order with a vehicle that is out of stock.  
- Expected: Error is thrown — "This vehicle is out of stock."  
Test Case: Add stock and run batch job.  
- Expected: Status is updated to Confirmed and stock is reduced by 1.

# 9. Deployment and Maintenance

- Deployed manually to the developer org.  
- Scheduled batch job for daily execution using scheduler class.  
- Admin will monitor job logs and update stock or vehicle statuses manually as needed.

# 10. Screenshots

Refer to the GitHub repository for full screenshots and walkthroughs.  
GitHub Repo: https://github.com/codesbysneha/WhatNext-Vision-Motors-salesforce

# 11. Conclusion

This project demonstrates how Salesforce can be used to streamline business processes through automation. With Apex, Flows, and scheduled jobs, WhatNext Vision Motors now has a scalable and intelligent vehicle order management system.