

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

Project Overview

WhatNext Vision Motors is a future-focused automotive company dedicated to transforming mobility through innovation, smart vehicle solutions, and customer-centric services. This project focuses on the development and implementation of a Salesforce-based Customer Relationship Management (CRM) system to streamline customer interactions, vehicle bookings, service management, and internal operations. The CRM system is designed to centralize customer data, automate business processes, improve service efficiency, and enhance decision-making using real-time analytics. By implementing this system, the organization ensures improved customer engagement, faster service delivery, and data-driven business operations.

Objectives

The main objectives of the WhatNext Vision Motors Salesforce CRM project are as follows:

- To develop a centralized Salesforce CRM system for managing customers, vehicle bookings, payments, and service records efficiently.
- To automate key business processes such as booking approvals, task assignments, and status updates to reduce manual workload.
- To improve customer relationship management by maintaining accurate and organized customer information.

- To enhance operational efficiency through workflow automation, validation rules, and real-time system updates.
- To provide management with real-time reports and dashboards for better decision-making.
- To ensure data security and controlled system access using Salesforce profiles, roles, and permission sets.
- To build a scalable and future-ready CRM system that can support future business expansion and digital transformation.
- To increase customer satisfaction by enabling faster service processing and transparent transaction tracking.

Phase 1: Requirement Analysis & Planning Understanding Business Requirements

- Centralized system required to manage customers, vehicles, bookings, payments, and service records.
- Manual processes cause:
 - Delays in transactions
 - Data duplication
 - Poor tracking of customer history

- Management requires real-time visibility of sales, bookings, and revenue.
- High-level security is required for customer and financial data.

Defining Project Scope and Objectives Implementation of Salesforce CRM for:

- Customer Management
- Vehicle Management
- Booking Processing
- Payment Tracking
- Service Request Management
- Approval Workflows

Phase 2: Salesforce Development – Backend & Configurations Setting Up the Development Environment & Workflow Created a Salesforce sandbox to safely build and test system features.

- Implemented version control and deployment management using tools such as VS Code and Git to track system changes efficiently.

Customizing Objects, Fields, Validation Rules, and Automations

- Developed custom objects including Vehicle, Vehicle Dealer, Vehicle Customer, Vehicle Order, Vehicle Test Drive, and Vehicle Service Request.

Added customized fields to the Vehicle object, such as.

- Vehicle_Name__c – Text
- Vehicle_Model__c – Picklist (Sedan, SUV, EV, etc.)
- Stock_Quantity__c – Number
- Price__c – Currency Dealer__c – Lookup
- (Dealer) Status__c – Picklist (Available, Out of Stock, Discontinued)
- Configured validation rules, workflows, Process Builder, and Flow automations to improve efficiency and reduce errors.

Apex Classes, Triggers, and Asynchronous Processing

- Created Apex classes and triggers to implement backend logic beyond standard configurations.
- Applied asynchronous Apex to manage time-consuming processes without affecting system performance.

Documentation

- Recorded screenshots of Apex classes and triggers as evidence of development and for reference purposes.

```

Developer Console - Google Chrome
File Edit Help Test Debug View Status Progress Problems
orgfarm-59878e6287-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage
VehicleOrderTriggerHandler.apc [VehicleOrderTriggerHandler.apc] - VehicleOrderBatch.apc [VehicleOrderBatch.apc]
Code Coverage None | 0% Uncovered
1 public class VehicleOrderTriggerHandler {
2     ...
3     public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isInsert, Boolean isUpdate) {
4         ...
5         if (isBefore) {
6             ...
7             if (isInsert || isUpdate) {
8                 preventOrderHeldUpStack(newOrders);
9             }
10        }
11    }
12
13    if (isAfter) {
14        ...
15        if (isInsert || isUpdate) {
16            updateStockOnOrderPlacement(newOrders);
17        }
18    }
19}
20
21
22
23
24
25

```

```

Developer Console - Google Chrome
File Edit Help Test Debug View Status Progress Problems
orgfarm-59878e6287-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage
VehicleOrderBatch.apc [VehicleOrderBatch.apc] - VehicleOrderTriggerHandler.apc [VehicleOrderTriggerHandler.apc]
Code Coverage None | 0% Uncovered
1 trigger VehicleOrderTrigger on Vehicle_Order__c (before Insert, before Update, after Insert, after Update) {
2     VehicleOrderBatchHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore, Trigger.isAfter, Trigger.isInsert, Trigger.isUpdate);
3 }

```



```

Developer Console - Google Chrome
File Edit Help Test Debug View Status Progress Problems
orgfarm-59878e6287-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage
VehicleOrderBatch.apc [VehicleOrderBatch.apc] - VehicleOrderTriggerHandler.apc [VehicleOrderTriggerHandler.apc]
Code Coverage None | 0% Uncovered
1 global class VehicleOrderBatch implements Database.Batchable<Object> {
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> vehicleMap = new Map<Id, Vehicle__c>(
19                [SELECT Id, Stock_Quantity__s FROM Vehicle__c WHERE Id IN :vehicleIds]
20            );
21
22            List<Vehicle_Order__c> orderToUpdate = new List<Vehicle_Order__c>();
23            List<Vehicle__c> vehicleToUpdate = new List<Vehicle__c>();
24
25            for (Vehicle_Order__c order : orderList) {

```

```

Developer Console - Google Chrome
File Edit Help Test Debug View Status Progress Problems
orgfarm-59878e6287-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage
VehicleOrderBatch.apc [VehicleOrderBatch.apc] - VehicleOrderTriggerHandler.apc [VehicleOrderTriggerHandler.apc]
Code Coverage None | 0% Uncovered
1 * global class VehicleOrderBatch implements Schedulable {
2     ...
3     global void execute(SchedulableContext sc) {
4         VehicleOrderBatch batchObj = new VehicleOrderBatch();
5         database.executeBatch(batchObj, 50); // 50 = batch size
6     }
7 }

```

Phase 3: UI/UX Development & Customization

Lightning App Configuration

- Built a custom Lightning application **named WhatNext Vision Motors** to act as the main interface for users.

Page Layouts and Dynamic Forms

- Structured layouts to highlight important fields, buttons, and related records clearly.

- Customized layouts based on user roles such as Vehicle, Dealer, Customer, Order, Test Drive, and Service Request.
- Implemented dynamic forms to improve usability by displaying fields only when required.

User Management

- Configured user profiles, roles, and permission sets to control system accessibility.

Reports and Dashboards

- Developed interactive dashboards to monitor sales performance and order trends.
- Created analytical reports to deliver real-time insights on customers, vehicles, and operations.

Lightning Web Components (LWC)

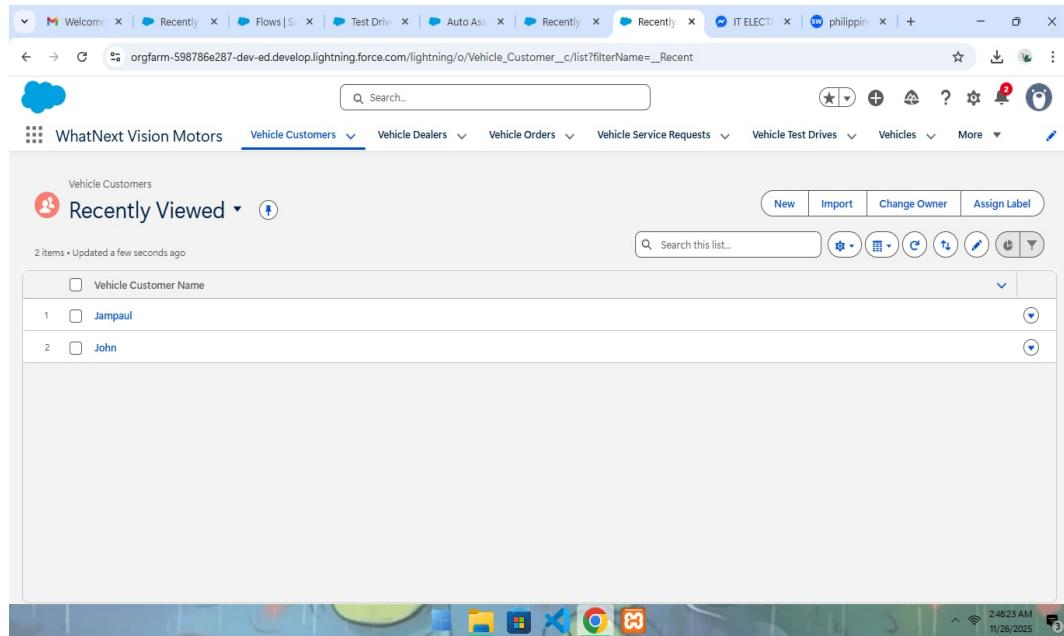
- Implemented LWCs to enhance the visual design and interactivity of the user interface where applicable.

Lightning Page Customization

- Customized Lightning Pages to improve system navigation and user experience.

Documentation

- Compiled screenshots and system notes to document all UI enhancements and custom designs.



Phase 4: Data Migration, Testing & Security

Data Loading Process

- Entered system records manually during demonstration to ensure data accuracy and reliability.

Field History Tracking, Duplicate Rules, and Matching Rules

- Enabled field history tracking to monitor important record changes.
- Configured duplicate and matching rules to prevent records with the same contact details from being created.

Profiles, Roles, Permission Sets, and Sharing Rules

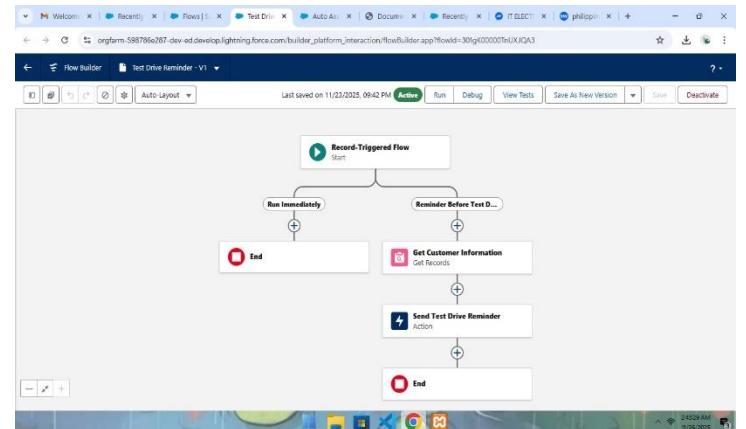
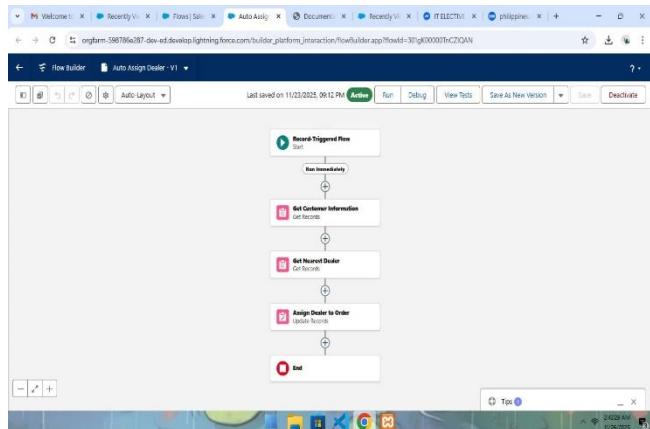
- Applied strict access controls to protect sensitive system data. Ensured each user role received appropriate permissions based on job responsibilities.

Creation of Test Classes

- Developed Apex test classes to validate backend logic and meet Salesforce testing requirements.

Documentation

- Captured screenshots of key automated processes such as Auto Dealer Assignment and Test Drive Reminder for documentation purposes.



Phase 5: Deployment, Documentation & Maintenance

Deployment Strategy

- Used Salesforce Change Sets and metadata deployment tools to safely transition all developments from the sandbox environment to production.

System Maintenance & Monitoring

- Performed regular system updates, data backups, and performance checks.
- Continuously assisted users by resolving technical issues and providing system support.

Troubleshooting Documentation

- Prepared detailed troubleshooting guides for identifying and fixing common system errors using flows and Apex debug logs.

Conclusion

The WhatNext Vision Motors project demonstrates how a Salesforce-based CRM system can significantly transform the vehicle purchasing and servicing process. By integrating vehicle availability, order management, test drive scheduling, and service request handling into a single platform, the system minimizes operational errors and saves valuable time for both customers and dealers. This project highlights how a unified digital solution enhances customer satisfaction while enabling more accurate and efficient business decision-making.

