WhatsNext Vision Motors – Salesforce CRM Project

Name: Johann Emmanuel B. De Vera
Course/Year: BSIT 4-1

Project Overview

A customer walks into a dealership hoping to buy a vehicle, only to find out that the model they want is unavailable in stock, and the nearest dealer that has it is located far away without their knowledge. WhatsNext Vision Motors is an automotive company aiming to modernize how customers order vehicles, request services, and interact with dealers. The company needs a centralized CRM system that will improve operational efficiency, provide automated solutions, and ensure accurate stock-based ordering. The Salesforce CRM project enhances the customer experience by automatically suggesting the nearest dealer based on customer address and preventing vehicle orders when stock is unavailable. It also manages vehicle inventory, customers, dealers, and test drives while automating notifications and order processing.

Description

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.

At the heart of this project is the improvement of the customer ordering process. The system is designed to automatically suggest the nearest dealer location to customers based on their address. This feature is intended to significantly enhance the convenience and efficiency of the ordering experience, making it more customer-friendly and reducing the time and effort required from the customer's end.

The project also addresses a common issue in the automotive industry: stock availability. The system includes a mechanism that prevents customers from placing orders for vehicles that are out of stock. This proactive approach ensures that customers can only create orders for vehicles that are currently available, thus avoiding potential confusion and disappointment that may arise from stock unavailability. This feature not only enhances customer satisfaction but also improves the accuracy of the company's order fulfillment process.

Furthermore, the project incorporates a scheduled process for updating the status of bulk order records. This automated process is designed to update the order status based on stock availability. If a vehicle is out of stock at the time of order placement, the system will update the order status

to 'Pending.' On the other hand, if the vehicle is in stock, the system will update the status to 'Confirmed.' This ensures that all orders are accurately reflected in terms of their fulfillment status, providing clear and transparent communication to customers regarding the status of their orders.

The implementation of this Salesforce project at WhatsNext Vision Motors is expected to yield several benefits. It aims to create a more efficient ordering system that reduces the potential for errors and improves the overall service provided to customers. By streamlining the ordering process and ensuring accurate stock availability, the company can enhance customer satisfaction and loyalty.

Moreover, the project is expected to contribute to operational efficiency by reducing the administrative burden on staff. By automating certain processes, employees can focus on more strategic tasks that require human intervention and expertise. This not only improves the overall productivity of the company but also allows for a more agile response to market demands and customer needs.

Objectives

The primary objective of this project is to implement a Salesforce-based CRM system that streamlines customer ordering, automates stock validation, and enhances operational workflows for automotive management. The system aims to reduce manual tasks, eliminate order errors caused by unavailable vehicles, automatically assign dealers based on proximity, and increase overall customer satisfaction. By automating business logic and integrating data transparency, the CRM provides scalability, reliability, and improved business decision-making.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements

- Customers should only be able to order vehicles that are available in stock in the vehicle stocks.
- The system must automatically assign the nearest authorized dealer to the customer based on their location.
- The CRM should track vehicles, dealers, customer information, orders, test drives, and service requests.
- Bulk updates must automatically update order or stock statuses depending on availability.

Project Scope

- Manage vehicle and dealer data in Salesforce Platform.
- Maintain customer details and while tracking their orders, create test drives in the platform, and service requests for customers.
- Enforce validation rules to prevent ordering vehicles that are out of stock using apex triggers via. console.
- Automate email reminders for scheduled test drives. Whenever a test drive is set it will automatically be sent on via Gmail.
- Use Apex triggers, batch processes including stocks, and Flow automation for backend processes.

Designed Data Model

Object	Purpose / Stores	Relationships
Vehicle__c	Stores vehicle details (name, price, stock quantity)	Related to Dealer and Vehicle_Order__c
Vehicle_Dealer__c	Stores authorized dealer information	Related to Vehicle_Order__c
Vehicle_Customer__c	Stores customer details	Related to Vehicle_Order__c and Vehicle_Test_Drive__c
Vehicle_Order__c	Tracks vehicle purchases	Related to Vehicle_Customer__c and Vehicle__c
Vehicle_Test_Drive__c	Tracks test drive bookings	Related to Vehicle_Customer__c and Vehicle__c
Vehicle_Service_Request__c	Tracks vehicle servicing requests	Related to Vehicle_Customer__c and Vehicle__c

Security Model

- Profiles define access to objects and records.
- Field-level security hides sensitive fields from unauthorized users.
- Role hierarchy ensures proper visibility across organizational levels.
- Permission sets provide extra access where needed.

Stakeholder Mapping

Stakeholder	Responsibility
Administrator	System configuration and automation
Sales Manager	Dealer management and vehicle allocation
Customer Representative	Customer support and order creation
Management	Reporting and decision-making

Execution Roadmap

1. Requirement Gathering and Data Model Design (with correct object relationships)
2. Backend Configuration and Apex Development
3. UI and Lightning Page Customization
4. Testing, Debugging, and Security Configuration
5. Deployment and Documentation

Phase 2: Salesforce Development – Backend & Configurations

Environment Setup & DevOps Workflow

- Salesforce Developer Org / Sandbox configured for development and testing.
- Version control setup via GitHub repository to track code and configuration changes.
- Deployment workflow established using Salesforce Change Sets or VS Code with Salesforce CLI for migrations between environments.

Customization of Objects, Fields & Validation Rules

- **Objects & Fields:**
    - Vehicle__c – Stores vehicle details (name, price, stock quantity).
    - Vehicle_Dealer__c – Stores authorized dealer information.
    - Vehicle_Customer__c – Stores customer information.
    - Vehicle_Order__c – Tracks vehicle purchases.
    - Vehicle_Test_Drive__c – Tracks test drive bookings.
    - Vehicle_Service_Request__c – Tracks service requests.
- **Validation Rules:**
    - Prevent orders on vehicles with zero stock.
    - Ensure required customer details are filled before order creation.
- **Automation Tools Implemented:**
    - **Workflow Rules / Process Builder / Flows**:

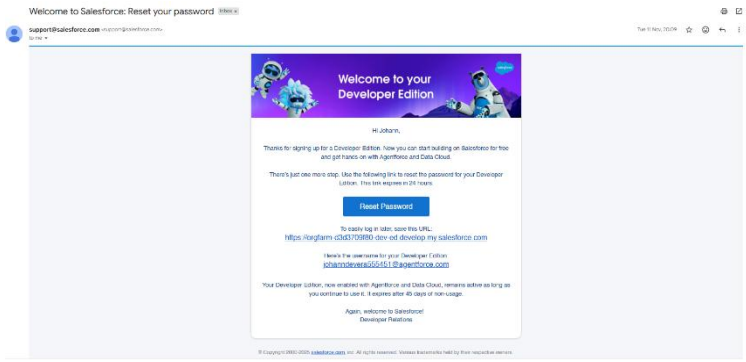
        - Automatic assignment of nearest authorized dealer based on customer location.
        - Sending email reminders for scheduled test drives.
        - Bulk order status updates triggered by stock changes.
    - **Approval Processes:**
        - Optional: High-value orders may require Sales Manager approval.

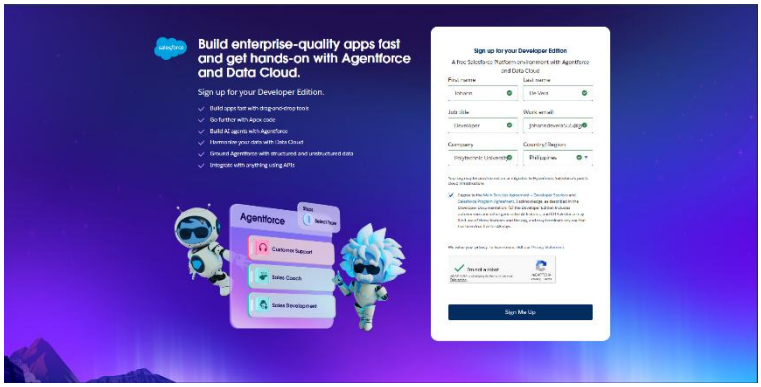
Apex Classes & Triggers

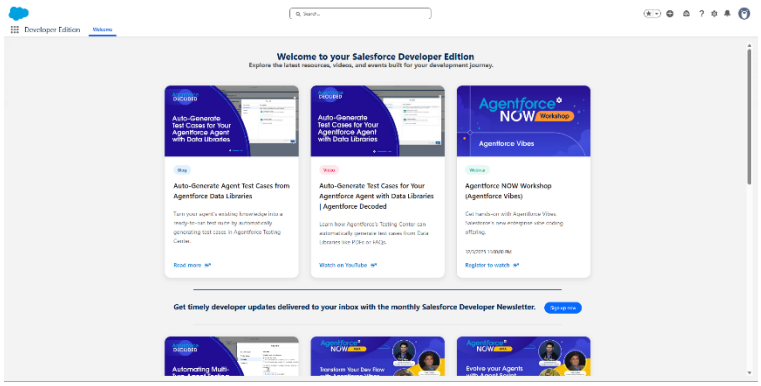
- **Triggers:**
    - Vehicle_Order__c Trigger – Validates stock before creating an order.
    - Vehicle_Test_Drive__c Trigger – Sends email reminders to customers for upcoming test drives.
- **Apex Classes:**
    - DealerAssignmentHandler – Determines nearest dealer for a customer using geolocation fields.
    - BulkOrderUpdater – Batch Apex class that periodically updates order statuses based on stock availability.
- **Asynchronous Apex:**
    - **Batch Apex** for handling bulk order updates efficiently.
    - **Future / Queueable Apex** (if required) for sending mass emails without hitting governor limits.

Screenshots & Documentation

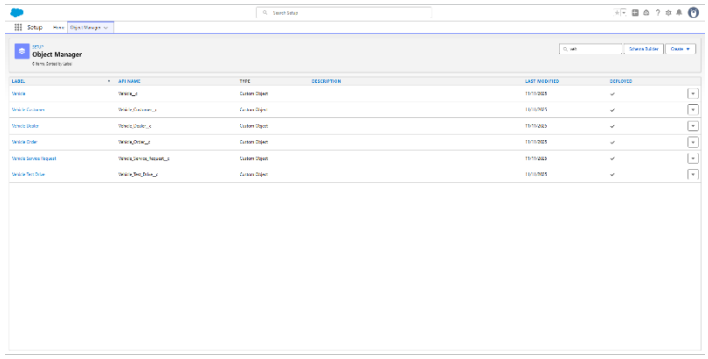
- **Screenshots:**
  - Validation Rules configuration.

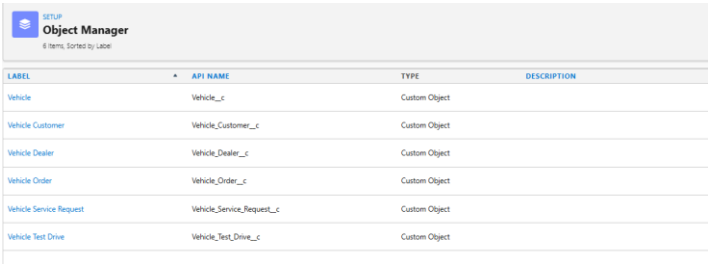






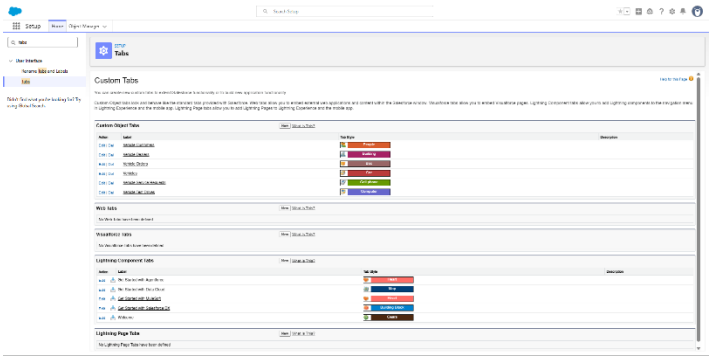
- Custom Objects, Field setup and Tabs.





SETUP
Object Manager
6 Items, Sorted by Label

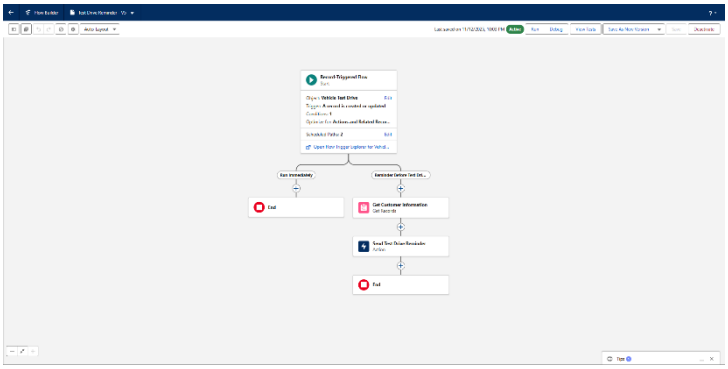
LABEL	API NAME	TYPE	DESCRIPTION
Vehicle	Vehicle__c	Custom Object	
Vehicle Customer	Vehicle_Customer__c	Custom Object	
Vehicle Dealer	Vehicle_Dealer__c	Custom Object	
Vehicle Order	Vehicle_Order__c	Custom Object	
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object	
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object	

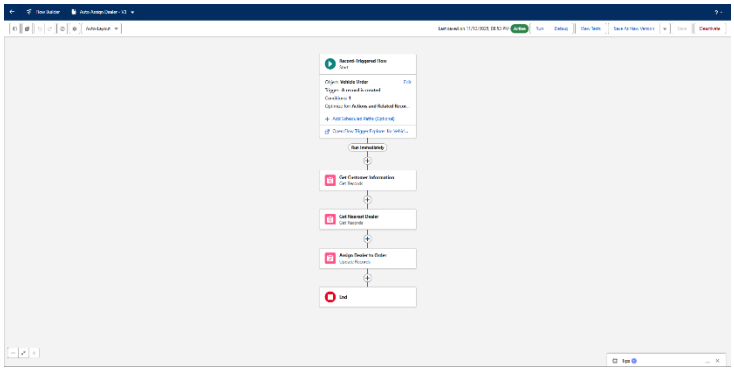


Custom Object Tabs New | What Is This?

Action	Label	Tab Style
Edit | Del	Vehicle Customers	People
Edit | Del	Vehicle Dealers	Building
Edit | Del	Vehicle Orders	Box
Edit | Del	Vehicles	Car
Edit | Del	Vehicle Service Requests	Cell phone
Edit | Del	Vehicle Test Drives	Computer

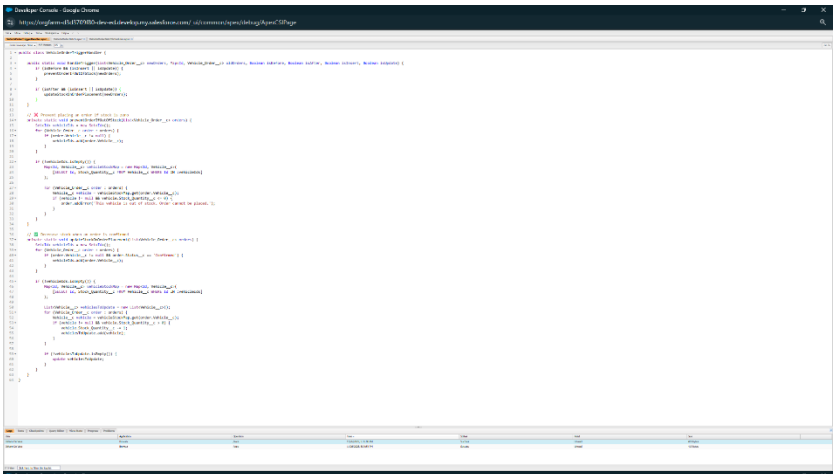
- Flows, Workflow Rules, Process Builders.

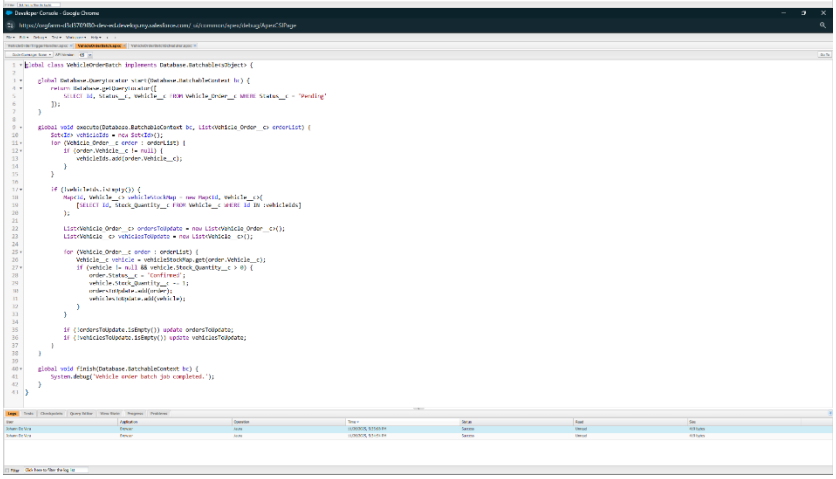




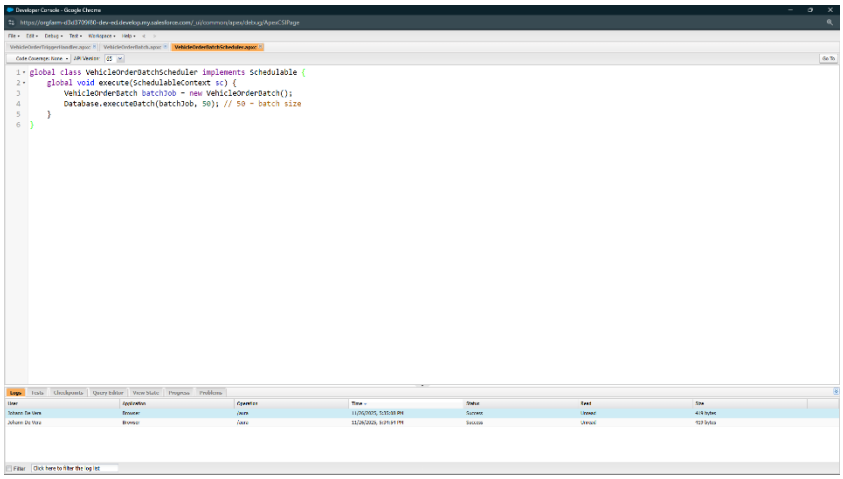
- Sample email notification triggered by system.

- Apex Classes and Trigger code pages.





```
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.');
    }
}
```



```
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** through **App Manager** named WhatsNext Vision Motors CRM.
- Included all relevant custom objects:
    - Vehicle
    - Vehicle Dealer
    - Vehicle Customer
    - Vehicle Order
    - Vehicle Test Drive
    - Vehicle Service Request
- Configured **app navigation**, branding, and tabs for easy access to objects.

Page Layouts & Dynamic Forms

- Designed **custom page layouts** for each object to display relevant fields and related lists.
- Enabled **Dynamic Forms** on Vehicle__c and Vehicle_Order__c pages to show/hide fields based on record criteria.
- Configured **related lists** to show linked records, e.g., Orders related to a Customer, Test Drives linked to Vehicles.

User Management

- Created **Profiles** for different roles (Admin, Sales Manager, Customer Representative).
- Assigned **Permission Sets** for additional access (e.g., for approving high-value orders).
- Implemented **Role Hierarchy** to control data visibility between management, sales, and support teams.

Reports & Dashboards

- Built **custom reports** for:
    - Vehicle stock availability.
    - Pending orders per dealer.
    - Scheduled test drives and service requests.

- Created **dashboards** showing:
    - Total vehicles in stock per dealer.
    - Monthly orders and revenue.
    - Customer engagement via test drives.

Lightning Web Components (LWC) Development (Optional / Bonus)

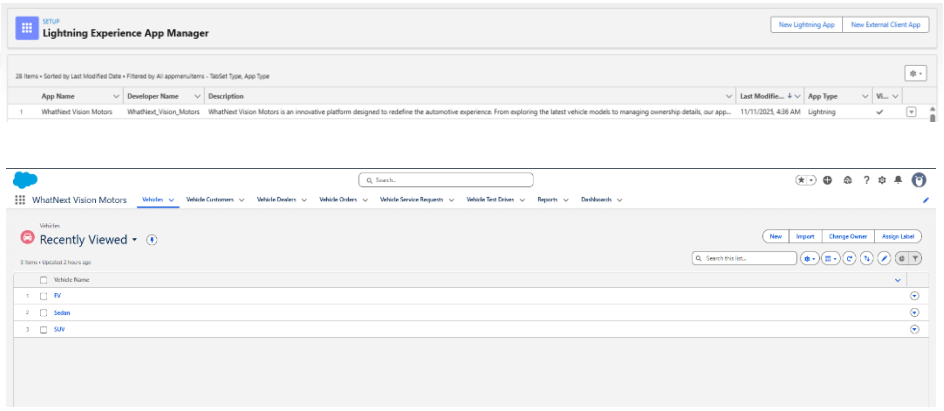
- Developed **LWC components** for enhanced UX:
    - Custom vehicle search component with filters for price, dealer, and stock.
    - Interactive test drive scheduler component.
- Integrated LWC components into Lightning Pages for seamless functionality.

Lightning Pages

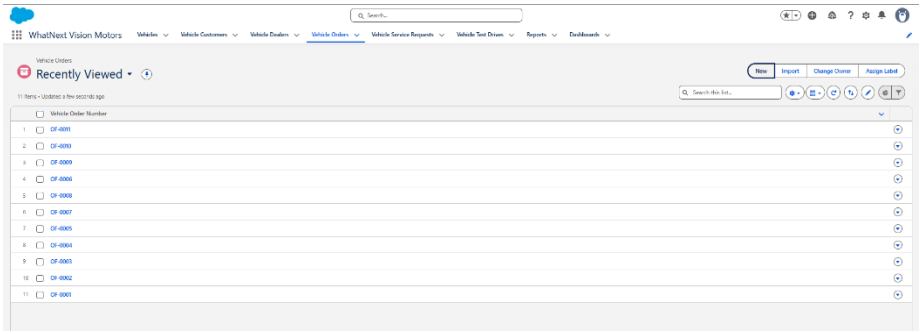
- Configured **record pages** for all objects using Lightning App Builder.
- Added **tabs, related lists, and LWC components** to improve user workflow.
- Optimized layouts for **desktop and mobile users**.

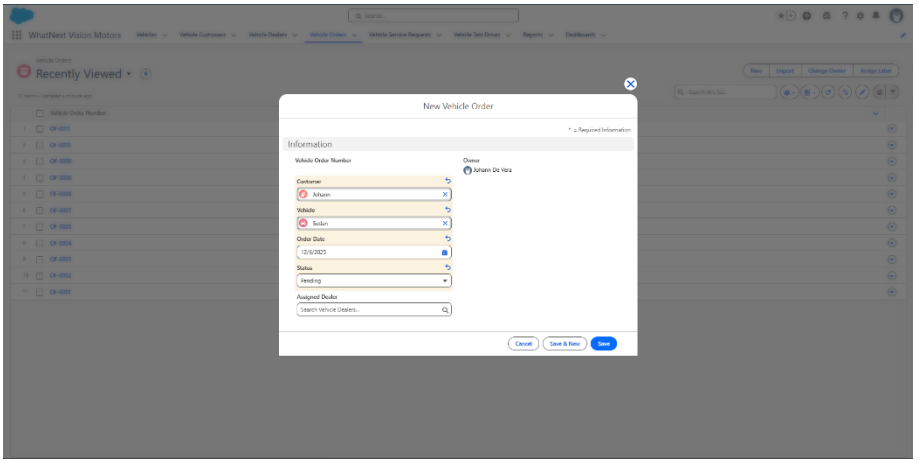
Screenshots & Documentation

- **Mandatory Screenshots:**
    - Lightning App setup in App Manager.

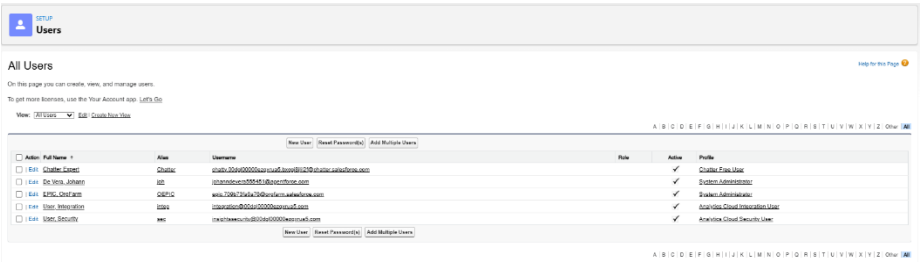


- Page Layouts and Dynamic Forms.

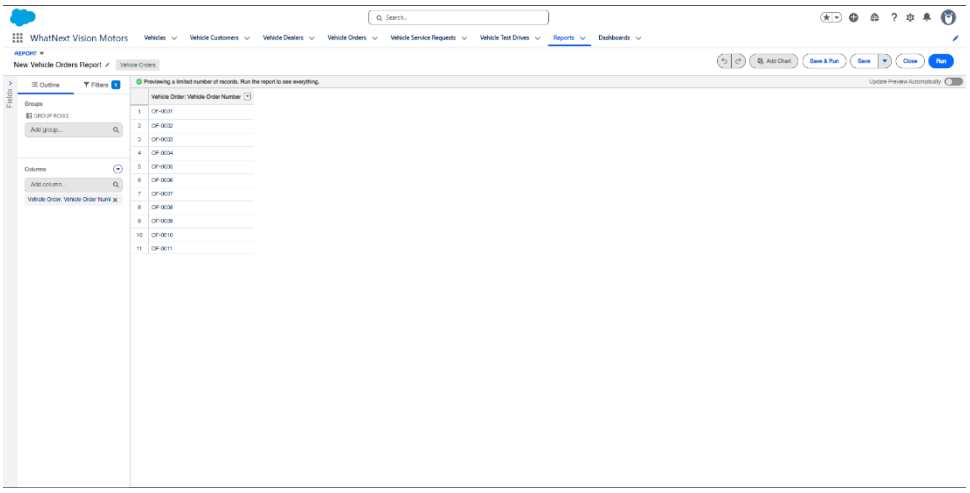




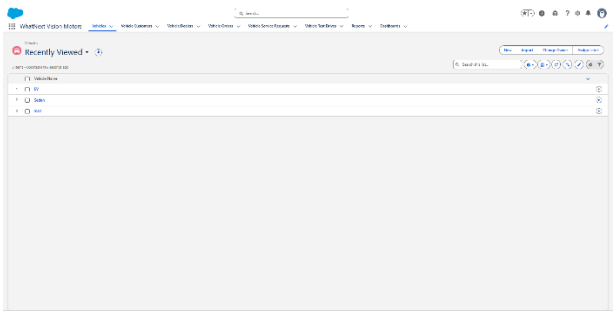
- User profiles, permission sets, and role hierarchy.

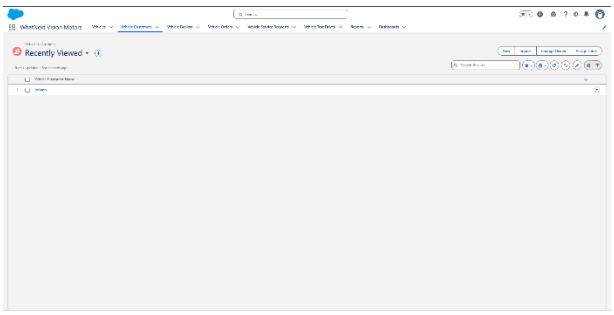


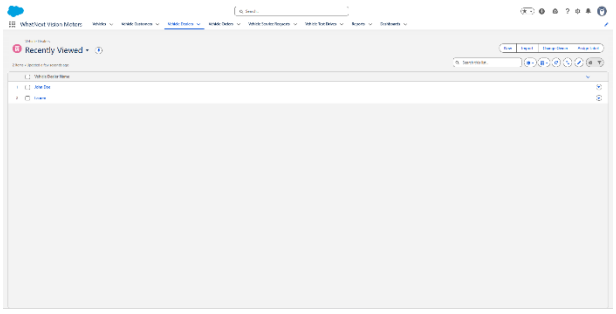
- Reports and dashboards.

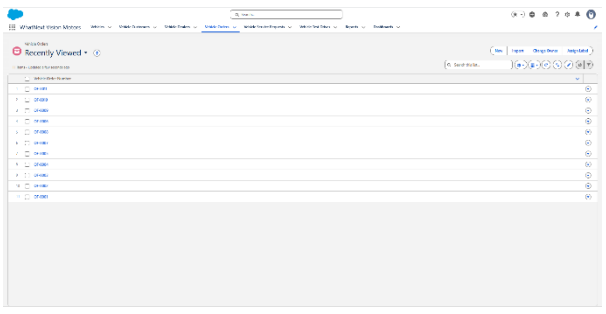


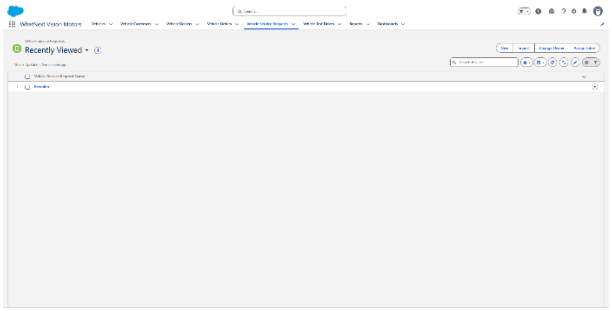
- Lightning Pages with integrated components.

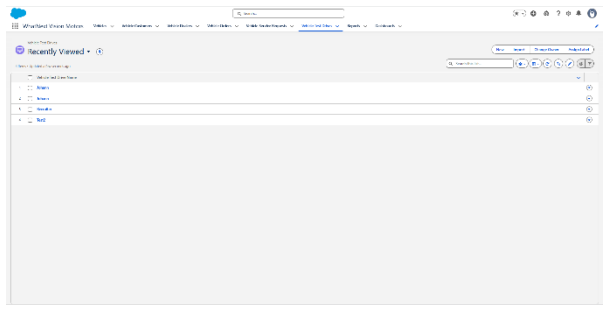












Phase 4: Data Migration, Testing & Security

Data Migration

- **Data Loading Tools Used:**
    - **Data Loader** for bulk upload of Vehicles, Orders, Test Drives, and Service Requests.

Tracking & Data Integrity Controls

- **Field History Tracking**
    - Enabled for critical fields such as Vehicle Stock, Order Status, and Service Request Status.
    - Tracks who made changes and when, supporting audits and compliance.
- **Duplicate Rules & Matching Rules**
    - Configured on Vehicle_Customer__c to prevent duplicate customer records.
    - Matching rules based on **email**.
    - Duplicate rule blocks or warns users when duplicates are detected.

Security Configuration

- **Profiles & Roles**

    Created separate profiles (Admin, Sales Manager, Customer Representative).

- **Permission Sets**

    Additional access such as:
    - Approval permission for high-value orders.
    - Access to restricted reports.
- **Sharing Rules**
    - Implemented sharing rules for Vehicle_Order__c and Vehicle_Service_Request__c.

- Customer Service team gets access to service records; Sales gets access to orders.

Testing & Apex Test Classes

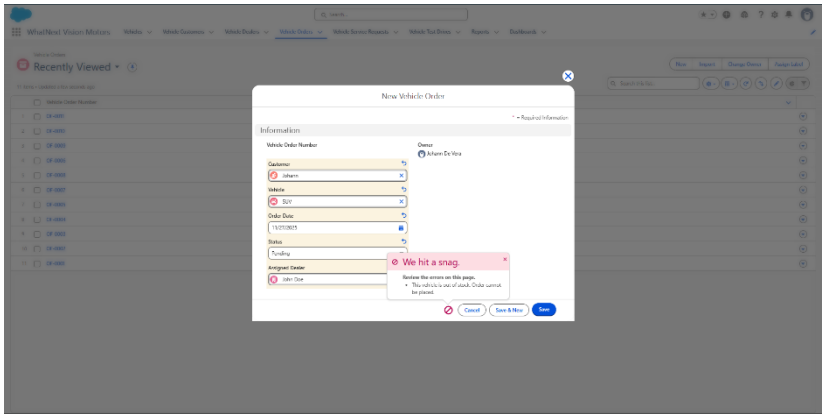
- **Apex Test Classes Created For:**
  - Vehicle_Order__c Trigger (stock validation, dealer auto-assignment).
  - Vehicle_Test_Drive__c Trigger (reminder email logic).

- **Test Class Coverage**
  - Assertions included to verify correct logic execution in triggers, flows, and batch processes.

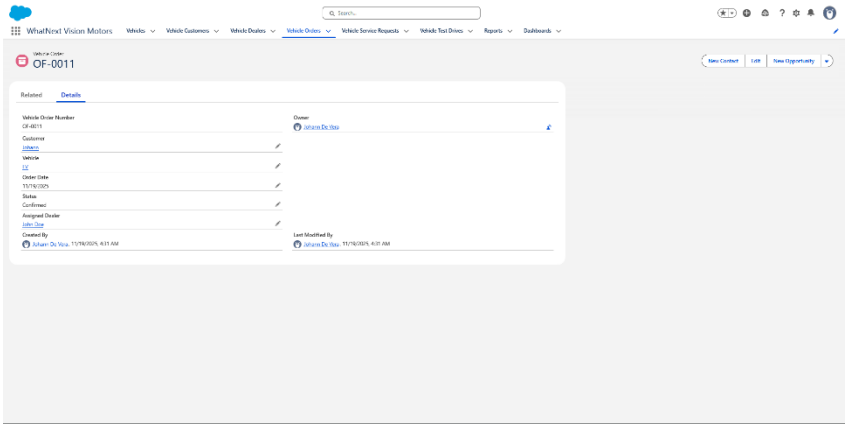
Test Cases & Screenshots

For **each feature**, the following must be documented with **input & output screenshots**:

- Vehicle Order when stock is 0



- Dealer Assignment when the dealer and the customer have the same place



- Email for test drivers

Phase 5: Deployment, Documentation & Maintenance

The Deployment of the system was executed using Salesforce Change Sets that the developer created. All relevant Salesforce components including custom objects, fields, validation rules, automation processes, Lightning pages, and Apex classes were packaged and migrated from the sandbox or the playground environment to production. Each component underwent a validation process to detect dependency or metadata conflicts before deployment. Additionally, naming consistency and configuration alignment were refined during the migration process to ensure seamless execution once deployed since there are error and trouble shoots encountered while creating the system. After successfully passing validation, the change set was deployed to the live environment.

To maintain system reliability after deployment, a structured maintenance and monitoring plan was established by the developer. This includes periodic review of automation processes such as Flows, Approval Processes, and Apex Triggers to verify that they continue to function properly with new and existing records. Debug logs will be observed to identify runtime issues, and data quality will be protected through duplicate rules, field validations, and scheduled cleanup activities. User access and permissions will be regularly reviewed to support organizational changes, while routine system backups will secure operational and historical data.

A comprehensive troubleshooting approach was documented to address potential issues that may arise during system operation. Data-related inconsistencies will be resolved by verifying validation rules and field restrictions. Errors in automations will be diagnosed through the analysis of flow fault logs and Apex debug logs. Issues encountered during data importation will be resolved through accurate field mapping and enforcement of matching rules, while incorrect output values in formulas will be addressed by reviewing logical dependencies and calculations. This structured approach promotes efficient problem detection, resolution, and prevention of recurring system issues.

In conclusion, the implementation of the Salesforce-based CRM solution for WhatsNext Vision Motors significantly enhances and streamlines overall business operations. The platform allows seamless handling of vehicle management, stock tracking, dealer assignment, customer handling, order processing, and approval workflows under a unified system. Through the use of automation tools, Apex logic, and built-in CRM capabilities, the system strengthens data accuracy, minimizes manual effort, and supports faster, data-driven decision-making across departments.

The solution is designed with scalability and user accessibility at its core. Customers are able to request services, create orders, and book test drives more efficiently, while sales representatives and dealers benefit from automated record assignment, stock validation, and real-time visibility of vehicle availability. Managers and administrators are provided with enhanced monitoring tools, reports, dashboards, field-level control, and security settings, ensuring better oversight and performance tracking across the organization.

With its structured maintenance guidelines, documented troubleshooting approach, and secure configuration framework, the system is well-equipped to support long-term operational needs. It enables simplified onboarding of new users, reduces repetitive work, and provides a stable foundation for future functional enhancements such as advanced analytics, mobile optimization, and integration with external automotive systems. Ultimately, this CRM platform evolves into a valuable business tool that creates a smoother and more productive workflow for customers, sales teams, service representatives, and management, contributing to improved service quality, optimized sales processes, and a more efficient end-to-end vehicle management experience.