Ramos, Ron Gabriel P.

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

Project Overview:

WhatsNext Vision Motors, a pioneering force in the automobile industry, is dedicated to transforming mobility through breakthrough technologies that promote consumer comfort and operational efficiency. To increase dealer collaboration, expedite customer order processing, and automate key business operations, the organization has started a thorough Salesforce CRM deployment. By providing intelligent dealer assignment, enforcing stock-based order validations, and automating scheduled changes for bulk order records, the CRM aims to improve the customer experience. The solution uses batch processes, Apex automation, and connected workflows to build an order lifecycle that is quicker, error-free, and customer-focused.

Objectives:

The main goal of developing the WhatsNext Vision Motors CRM is to improve end-to-end client order management through automation and data-driven procedures. By automatically recommending the closest dealer based on the customer's location, avoiding the development of orders for automobiles that are out of stock, and providing customers with current order statuses, the system seeks to increase customer happiness. These enhancements reduce human labor, minimize order errors, speed up decision-making, and facilitate a smooth customer journey from inquiry to delivery, all of which provide substantial business value.

Phase 1: Requirement Analysis & Planning:

The first phase of the project involved analyzing the business processes of WhatsNext Vision Motors and transforming them into functional requirements for the Salesforce CRM. The goal was to design a system that efficiently manages vehicle inventory, customer orders, dealer assignments, and post-sales activities, while minimizing manual effort.

Key Business Requirements:

The CRM system should be able to:

- Maintain a centralized record of all vehicles, dealers, and customers.

- Verify vehicle stock availability at the time of order creation.
- Automatically assign orders to the nearest dealer based on customer location.
- Record and manage test drives and service requests.
- Automate repetitive tasks to reduce manual work and errors.

Project Scope and Objectives

To address the identified business requirements, the CRM system was designed with features that ensure efficient management of vehicles, orders, dealers, and customer interactions. The main objectives were to automate key processes, maintain accurate stock records, and streamline dealer assignment, while providing a scalable foundation for future growth.

Key Components and Functionalities:

- Custom objects for Vehicles, Orders, Customers, Dealers, Test Drives, and Service Requests to structure all essential data.
- Record-triggered Flows to automatically assign the nearest dealer and send email reminders for test drives or service updates.
- Apex Triggers to validate stock levels and keep inventory records up to date during order processing.
- Batch Apex processes to handle pending orders and update statuses efficiently on a scheduled basis

Data Model

To effectively represent the business structure of the WhatsNext Vision Motors CRM, six custom objects were created. Each object was designed to capture and manage specific data relevant to vehicle management, orders, and customer interactions.

Custom Objects and Their Purpose:

Object Name	Purpose
Vehicle__c	Stores detailed information about vehicles and stock levels.
Vehicle_Dealer__c	Maintains dealer information and locations.
Vehicle_Customer__c	Stores customer details and contact information.
Vehicle_Order__c	Tracks all vehicle orders, including status and fulfillment.
Vehicle_Test_Drive__c	Schedules and records test drives for customers.
Vehicle_Service_Request__c	Manages service history, maintenance requests, and issues.

This data model ensures that all key business entities are captured in Salesforce, enabling automation, reporting, and smooth integration between sales, service, and inventory management.

Security Model

To ensure secure and controlled access to data within the WhatsNext Vision Motors CRM, a combination of standard Salesforce features and custom configurations was implemented. The security model was designed to protect sensitive information while allowing users to perform their tasks efficiently.

Key Security Configurations:

- Standard Salesforce profiles were used, and **Permission Sets** were added to grant appropriate access to custom objects.

- **Field-Level Security** and the **Role Hierarchy** were applied to ensure users can only view or edit records relevant to their role.

- **Field History Tracking** was enabled for critical fields such as Stock_Quantity__c (Vehicle) and Status__c (Order) to support auditing and monitor changes over time.

This approach ensures that the system maintains data integrity, complies with security requirements, and provides transparency for administrative monitoring.

Phase 2: Salesforce Development – Backend & Configuration

Setup Environment & DevOps Workflow

At the start of the project, a dedicated Salesforce Developer Org was prepared to build, configure, and test all custom features and automation. The environment was set up to support a smooth development process and eventual deployment to production.

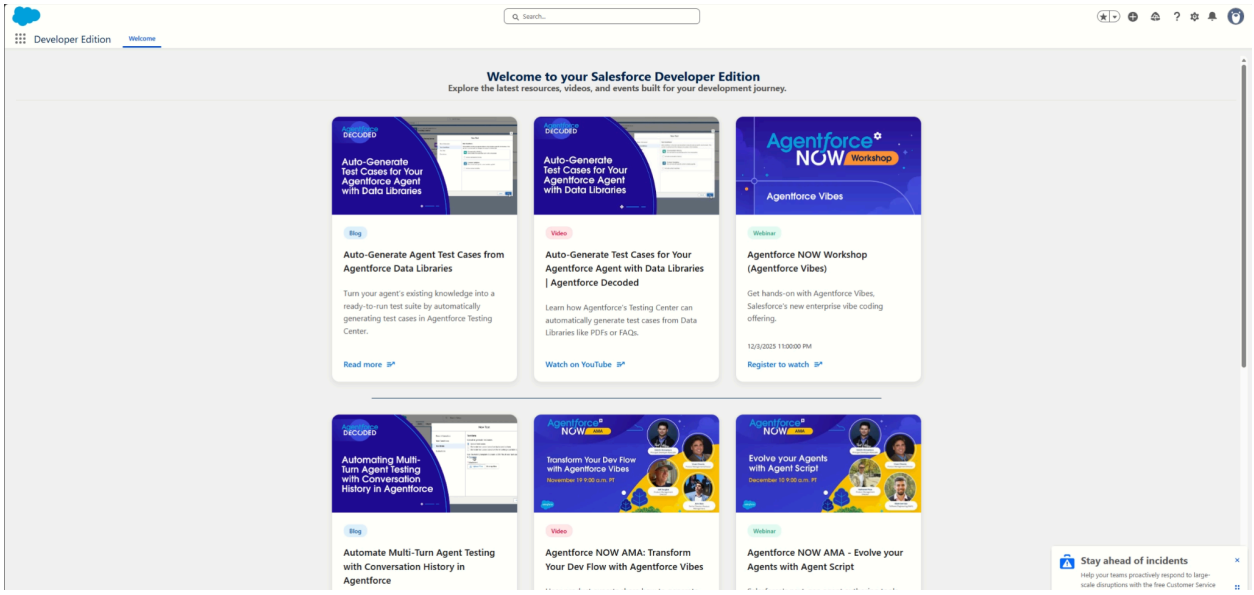
Key Setup Details:

- **Environment:** Salesforce Lightning Experience (Developer Edition) was used for development and testing.

- **User Profiles & Roles:** Standard Salesforce profiles were utilized for testing purposes; no custom profiles were created.

- **Deployment Method:** All metadata and configurations were deployed from the sandbox to production using **Change Sets**, ensuring a controlled and reliable deployment process.

This setup provided a stable environment for development, testing, and deployment, while maintaining proper user access and system integrity.



Customization of Objects, Fields, Validation Rules, and Automation

To support the business processes of WhatsNext Vision Motors, several custom objects and fields were created and configured in Salesforce. These objects capture all relevant information about vehicles, dealers, customers, and orders, while maintaining the relationships required for automation and reporting.

Custom Objects and Fields:

- **Vehicle** – Stores vehicle details such as name, model, and stock quantity.
- **Dealer** – Captures dealer information, including location and available vehicles.
- **Customer** – Maintains customer data, including contact details and address.
- **Order** – Tracks vehicle orders along with their status and fulfillment details.

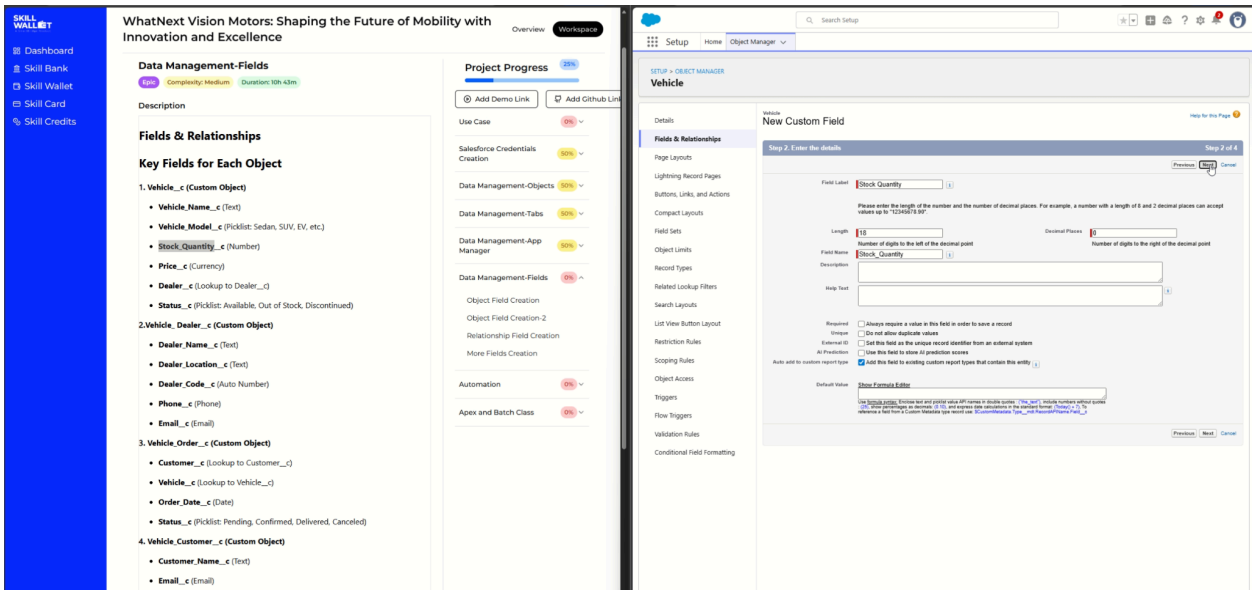
Relationships Between Objects:

- **Order → Vehicle:** Lookup relationship to associate orders with specific vehicles.

- **Order → Dealer:** Lookup relationship to link orders to dealers.

- **Order → Customer:** Lookup or Master-Detail relationship (depending on implementation) to connect orders to customers.

These configurations establish the foundation for further automation, such as validation rules, flows, and Apex triggers, ensuring data integrity and supporting streamlined business processes.



Validation Rules, Automation, and Apex Classes/Triggers

To ensure data integrity and automate critical business processes, several validation rules, workflows, flows, and Apex components were implemented in the CRM. These measures help prevent errors, streamline operations, and maintain accurate records.

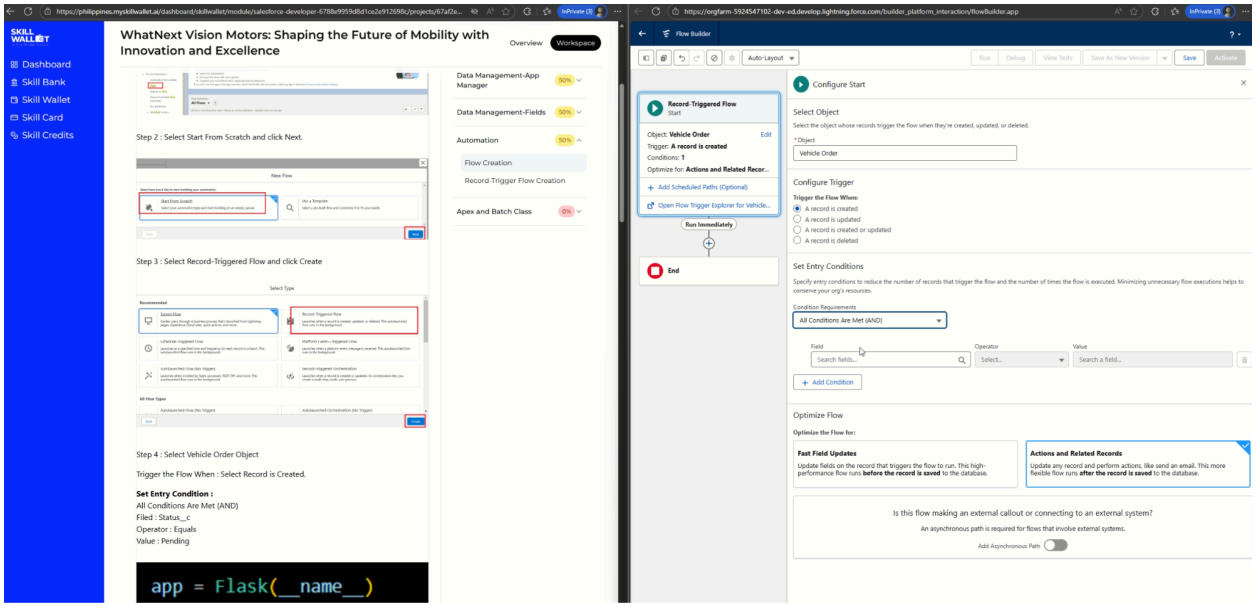
Validation Rules:

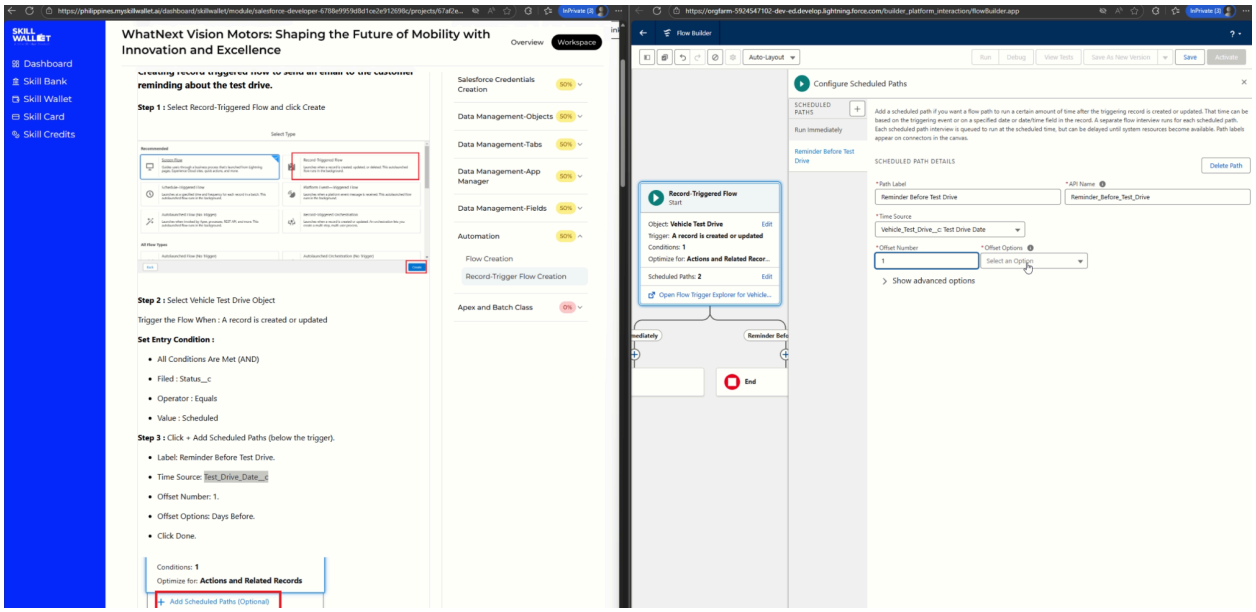
- **Out-of-Stock Order Blocker:** Prevents users from creating an order when the selected vehicle has zero stock, ensuring accurate inventory management.

Automation: Workflow and Flows

- **Record-Triggered Flow (Order Object):** Automatically assigns the nearest dealer based on the customer's address, improving order efficiency.

- **Scheduled Flows:** Sends automated reminders for upcoming test drives, enhancing customer engagement and service.





Apex Classes and Triggers:

- **VehicleOrderTriggerHandler:** Organizes trigger logic to handle stock validation and order updates.

- **VehicleOrderBatch:** Periodically checks pending orders and confirms them when stock becomes available.

- **VehicleOrderBatchScheduler:** Schedules the batch job to run daily at 12 PM, ensuring timely processing of pending orders.

These customizations collectively reduce manual effort, enforce business rules, and ensure smooth, automated order and dealer management within the CRM.

Apex Trigger on Order Object

An Apex Trigger was implemented on the **Order** object to enforce key business rules and automate processes that could not be fully handled by declarative tools. The trigger ensures accurate stock management, proper dealer assignment, and order status updates while following Salesforce best practices through a dedicated Trigger Handler.

Key Functions of the Trigger:

- **Stock Validation:** Checks vehicle stock availability before allowing an order to be processed.

- **Dealer Assignment:** Automatically assigns the nearest dealer if the task is not handled by a Flow.

- **Order Status Updates:** Updates the order status to **Pending** or **Confirmed** based on stock availability.

- **Trigger Handler:** Encapsulates logic for modularity, maintainability, and adherence to best practices.

This approach ensures that all orders are processed accurately, stock levels remain consistent, and automation is efficient and reliable.

SKILL WALLET

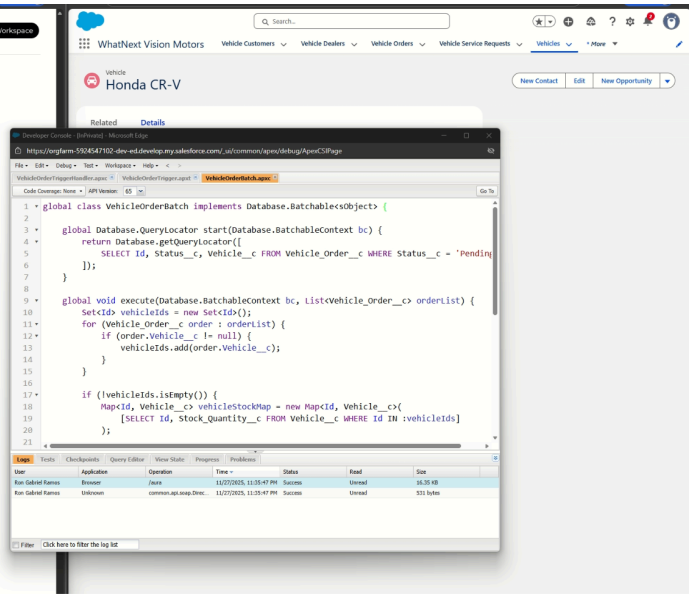
- Dashboard
- Skill Bank
- Skill Wallet
- Skill Card
- Skill Credits

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

Overview | Workspace

```
if (vehicleStockMap.containsKey(order. Vehicle__c)) {
Vehicle_c vehicle = vehicleStockMap.get(order. Vehicle__c);
if (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 (!vehicles ToUpdate.isEmpty()) {
update vehiclesToUpdate;
}
}
global void finish(Database.BatchableContext bc) {
System.debug('Vehicle order batch job completed.');
}
}
```

Step 9 : Create Schedule Class then revoked Batch Class in Schedule Class




```
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 = 'Pendin
        ]);
    }

    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]
            );
```


SKILL WALLET

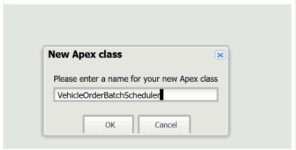
- Dashboard
- Skill Bank
- Skill Wallet
- Skill Card
- Skill Credits

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

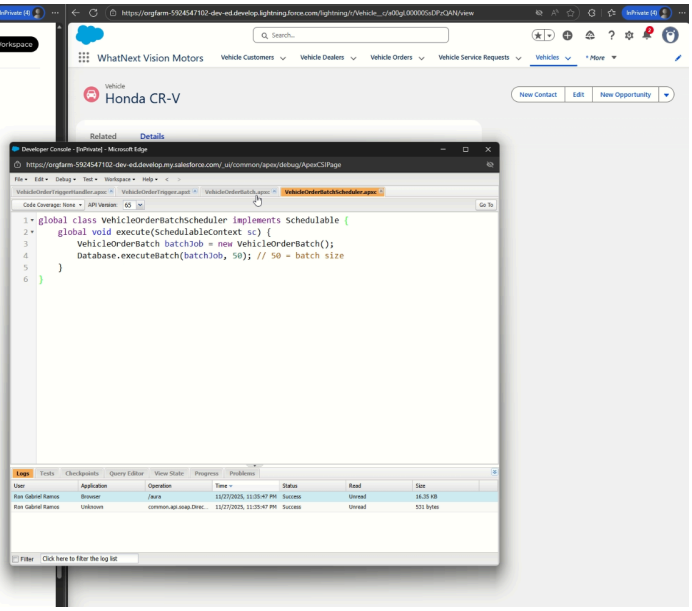
Overview | Workspace

```
if (!ordersToUpdate.isEmpty()) {
update ordersToUpdate;
}
if (!vehicles ToUpdate.isEmpty()) {
update vehiclesToUpdate;
}
}
global void finish(Database.BatchableContext bc) {
System.debug('Vehicle order batch job completed.');
}
}
```

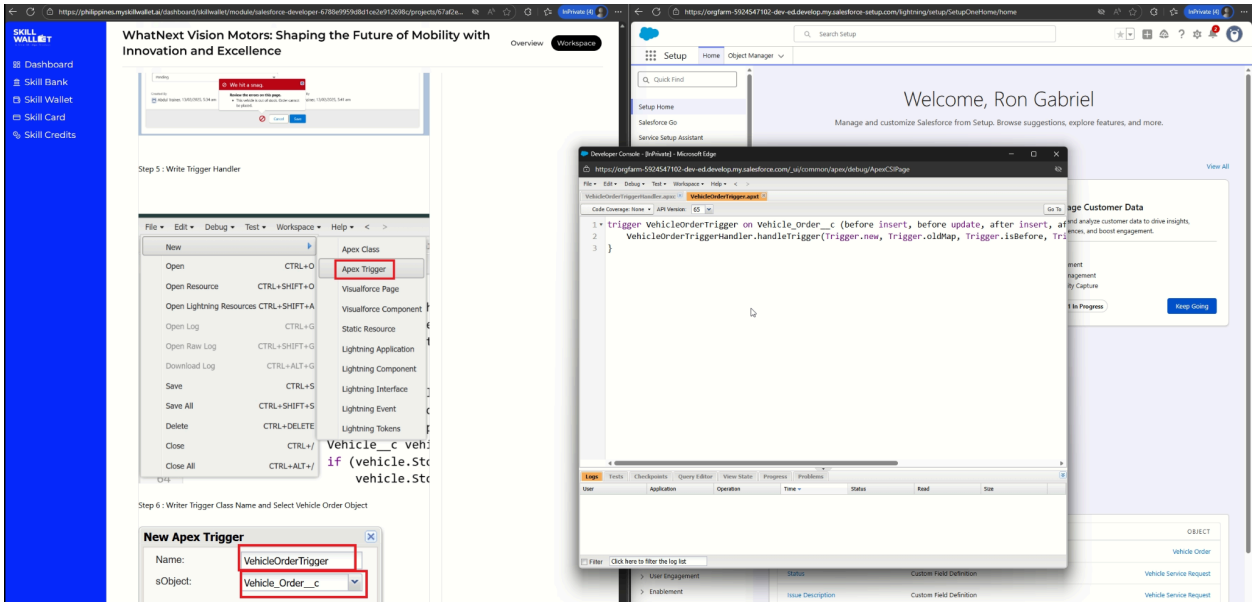
Step 9 : Create Schedule Class then revoked Batch Class in Schedule Class

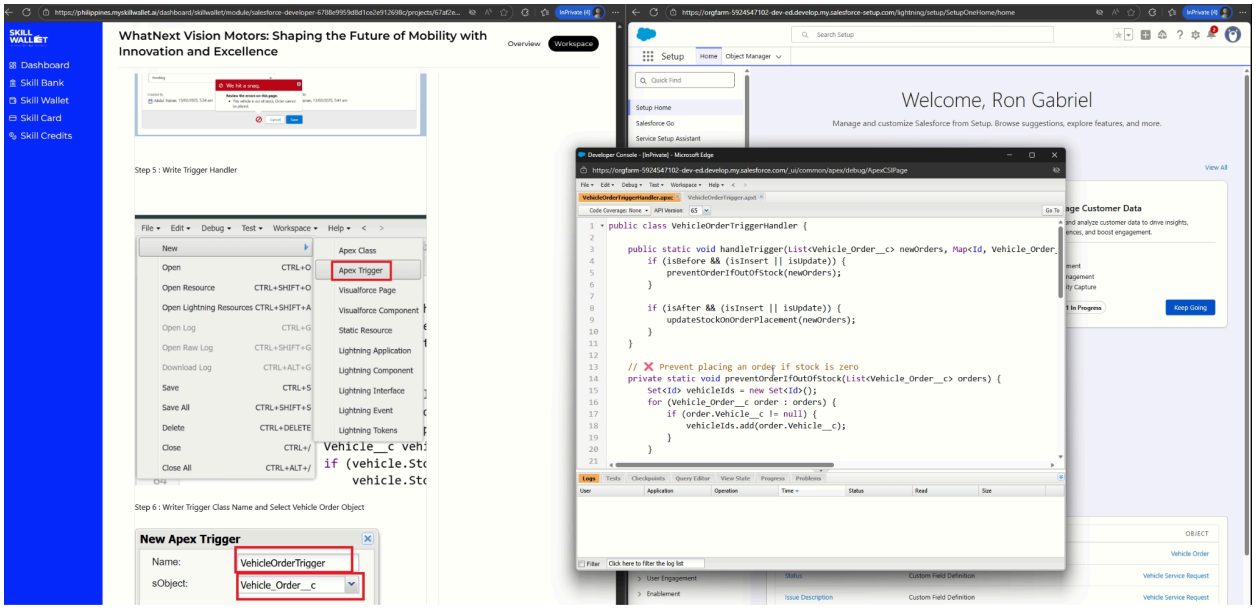


```
global class VehicleOrderBatchScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        VehicleOrderBatch batchJob = new VehicleOrderBatch();
        Database.executeBatch(batchJob, 50);
    }
}
```




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





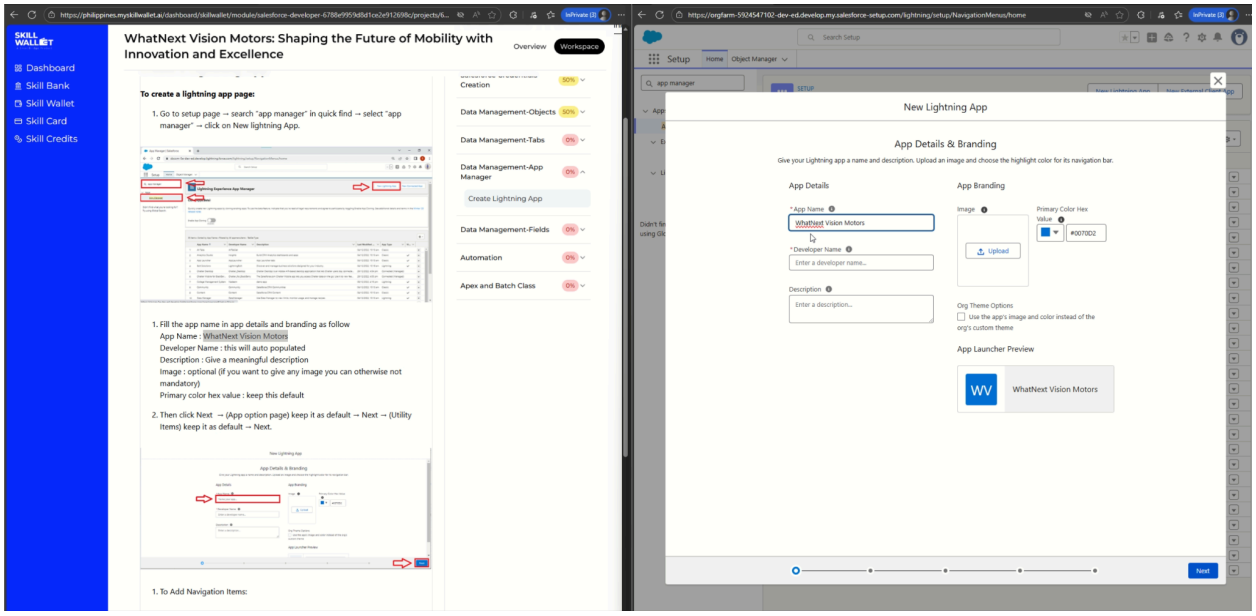
Phase 3: UI/UX Development & Customization

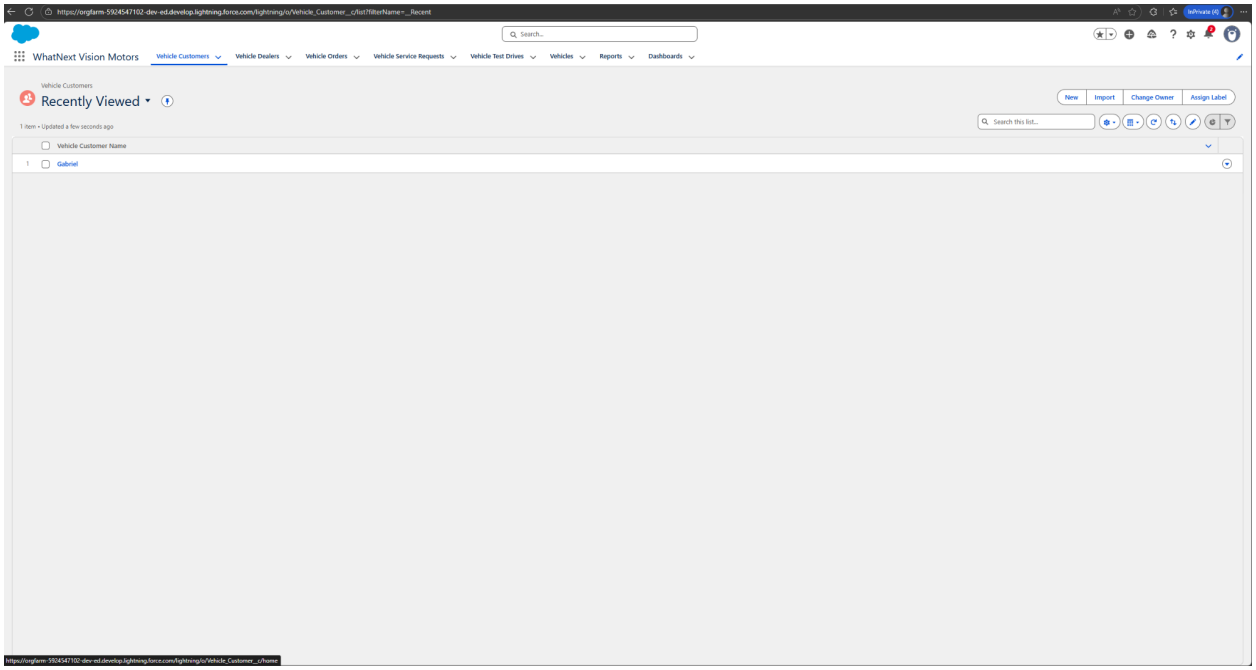
The UI/UX phase focused on creating a user-friendly interface for the WhatsNext Vision Motors CRM using Salesforce Lightning. The goal was to make navigation intuitive, display relevant information efficiently, and enhance the overall user experience.

Lightning App Setup:

- App Name: WhatNext Vision Motors (created via App Manager)
- Tabs Included: Vehicles, Vehicle Dealers, Vehicle Customers, Vehicle Orders, Vehicle Test Drives, Services
- Dynamic Forms: Fields are displayed conditionally based on record status and availability, improving clarity and usability.
- Lightning Pages Enhancements: Highlight panels and related lists were added to provide quick access to key information and streamline workflows.

This setup ensures that users can efficiently navigate the CRM, access relevant data quickly, and perform tasks with minimal effort.





CONCLUSION:

WhatsNext Vision Motors' primary business requirements, such as car management, dealer assignments, customer monitoring, and post-sales support, are effectively met by the Salesforce CRM solution. The solution offers an effective, precise, and user-friendly platform for handling orders and inventory by integrating custom objects, automation flows, validation rules, Apex triggers, batch operations, and Lightning App customizations.

The initiative has produced a number of significant advantages:

Better Customer Experience: Smooth and transparent customer interactions are ensured by automated dealer assignment, stock checking, and test-drive reminders.

Operational Efficiency: By eliminating manual labor, automation frees up employees to concentrate on important duties.

Data Accuracy and Security: Sensitive information is protected while maintaining data integrity with audit tracking, role-based access, and field-level security.

Scalability: Future expansion and changing business procedures are supported by the system's design.