Salesforce CRM Project Documentation

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

Submitted by:
KATIGANDLA SOWMYA

Institution/Organization:
SMART BRIDGE

Program:
Salesforce Developer with Agentblazer Champion Program

Project Type:

Salesforce CRM Implementation
(Custom Objects, Flows, Apex, Automation, Reports)

Guided by:
Smart Bridge Educational Pvt. Lmt

Project Overview

WhatNext Vision Motors is on a mission to redefine the automotive customer journey by integrating intelligent CRM capabilities into its vehicle sales and service ecosystem. As the industry evolves, customer expectations have grown beyond simple transactions. The company aims to meet this demand by establishing an intelligent and agile digital infrastructure using Salesforce CRM.

This comprehensive CRM initiative enables seamless interaction across departments, empowering both customers and employees with real-time data access and decision-making tools. From intuitive order flows and stock-aware checkouts to intelligent dealer assignment, every phase of customer interaction is reimagined to be faster, more transparent, and fully automated.

Through the introduction of smart workflows, field validation, and scalable automation using Apex and Flows, WhatNext Vision Motors is laying the groundwork for future-ready automotive experiences. These enhancements streamline sales, service, and operations, ensuring that customers experience smooth order placements while the business gains sharper operational control.

The project's success marks a major milestone in the company’s digital journey — one that connects innovation, efficiency, and customer satisfaction within a centralized CRM hub. Salesforce's cloud-based ecosystem was chosen specifically for its reliability, security, and flexibility, making it the ideal platform for a scalable and robust solution in the high-demand vehicle sector.

Project Objectives

The overarching objective of this CRM initiative is to build an end-to-end digital platform that bridges customer needs with operational intelligence in real-time. The CRM implementation supports the company's vision of offering seamless, accurate, and smart vehicle services from inquiry to delivery. The project aims to:

- Design a centralized CRM platform for vehicle inventory, customer orders, test drives, and dealer coordination
- Provide location-based automation to suggest nearest dealers during order placement
- Ensure only valid, in-stock vehicle orders are accepted using real-time stock validation

- Automate order status updates using Apex Scheduler to reflect actual stock levels
- Establish reliable communication channels using automated emails and task reminders
- Use role-based permissions to maintain secure access across departments
- Reduce dependency on manual updates and streamline processes across sales and service departments
- Establish performance tracking using dashboards and insights to help decision-makers analyze KPIs and optimize workflows
- Build an extendable framework that allows future additions like mobile app integration, AI-powered predictions, and advanced reporting

Key Features

- Automated Dealer Assignment: Based on geolocation of customer address using Flow Builder
- Stock Validation Rules: Prevent orders for vehicles marked as 'Out of Stock' or 'Discontinued'
- Real-time Order Status: Orders are auto-tagged as 'Pending' or 'Confirmed' based on availability
- Centralized Object Management: Separate objects for Orders, Dealers, Customers, Vehicles, Service Requests, and Test Drives
- Email Automation: Customers receive reminders about test drives, order confirmations, or changes in availability
- Dashboard Reporting: Sales reps and managers get insight into booking trends, dealership performance, and customer history
- Scheduled Batch Jobs: Apex batch classes run periodically to validate inventory and notify relevant stakeholders
- Validation Logic: Custom logic ensures correct data inputs and prevents duplicate entries

Target Users

- Vehicle Sales Executives: To create, manage, and track customer orders with real-time inventory insights
- Customers: To experience seamless booking and service coordination without stock issues or unnecessary delays
- Dealership Managers: To receive assigned orders and manage regional performance through CRM dashboards

- CRM Developers/Admins: To implement, maintain, and enhance automation, security, and logic
- Business Analysts: To assess CRM performance using data visualization tools and optimize order fulfillment cycles

Technology Stack

- CRM Platform: Salesforce Developer Edition (Lightning Experience)
- Custom Objects:
    - Vehicle__c
    - Vehicle_Dealer__c
    - Vehicle_Order__c
    - Vehicle_Customer__c
    - Vehicle_Test_Drive__c
    - Vehicle_Service_Request__c
- Automation Tools:
    - Record-Triggered Flows (Dealer Assignment, Order Status Updates)
    - Scheduled Flows (Email Notifications)
    - Apex Triggers & Trigger Handlers (Stock Check, Field Validation)
    - Batch Apex (Inventory Updates)
- Security Controls:
    - Profiles & Permission Sets for different user roles
    - Role Hierarchies for organizational visibility
    - Field-Level Security and Validation Rules
- Integration & Tools:
    - Salesforce Data Loader
    - Import Wizard for bulk uploads
    - Report Builder & Dashboards
    - Lightning App Builder for custom UI pages

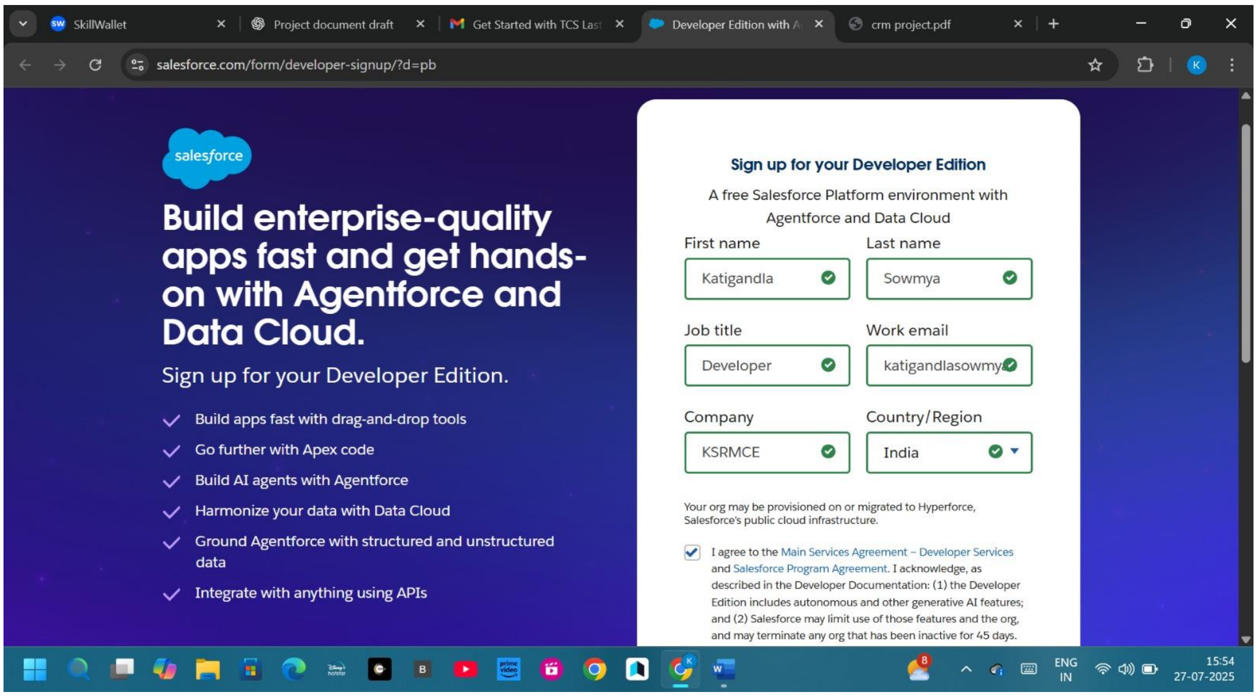
Scope 1: Salesforce Credentials & Environment Setup

1.1 Developer Account Creation

The foundation of the CRM implementation began with the setup of a dedicated Salesforce Developer Edition account. This free environment offers unrestricted access to most platform features, enabling custom development, configuration, and testing.

Users were guided to create their accounts through the Salesforce Developer portal (https://developer.salesforce.com/signup). Upon registration, a verification email was sent, and after confirming the email address, the user gained access to their personal Salesforce environment.

This setup ensures that each developer has isolated access to deploy, test, and experiment without affecting production-level environments. The Developer Edition supports all key modules required for CRM customization including Apex, Visualforce, Lightning components, and App Builder.



Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- Build apps fast with drag-and-drop tools
- Go further with Apex code
- Build AI agents with Agentforce
- Harmonize your data with Data Cloud
- Ground Agentforce with structured and unstructured data
- Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Katigandla	Last name: Sowmya
Job title: Developer	Work email: katigandlasowmy
Company: KSRMCE	Country/Region: India

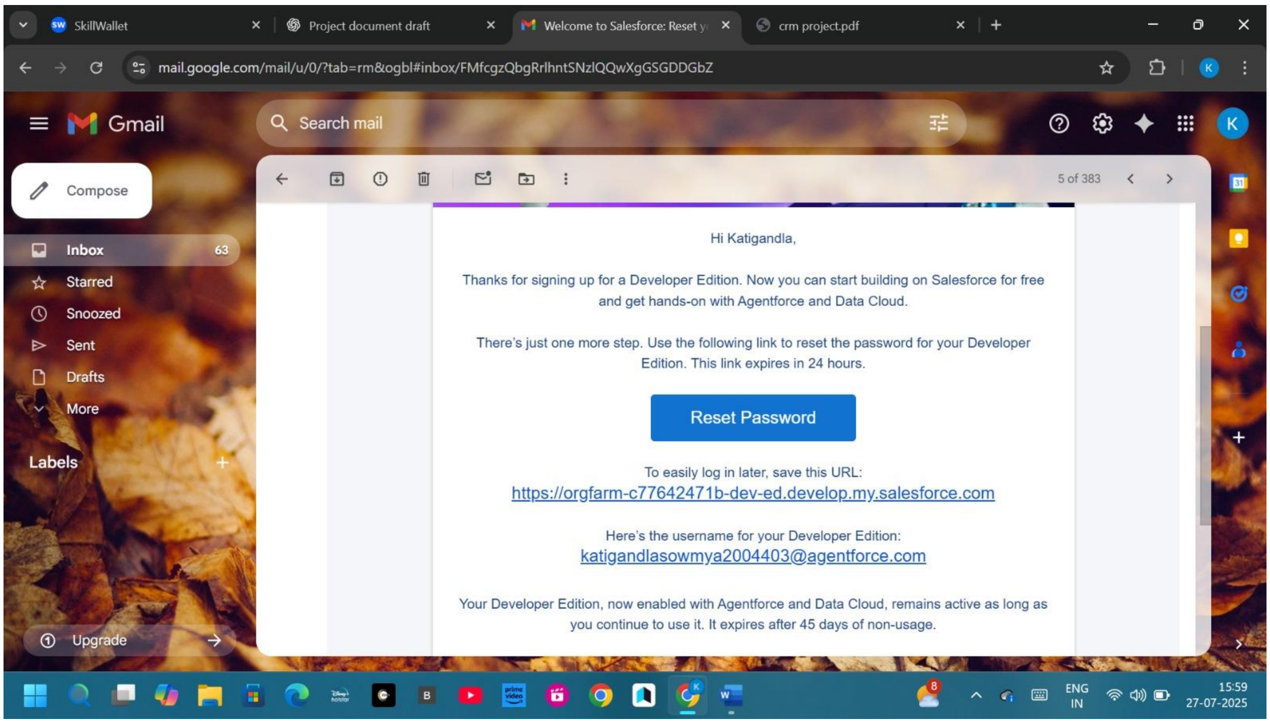
Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

✓ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

1.2 Email Verification and Login Setup

After successful account creation, credentials were configured including a unique username, password, and security token (used for API access and external integrations). These credentials were essential for connecting external tools like Data Loader and ensuring secure system access.

Users were encouraged to enable two-factor authentication (2FA) for improved security and to regularly update their recovery information to prevent account lockouts.


Hi Katigandla,

Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

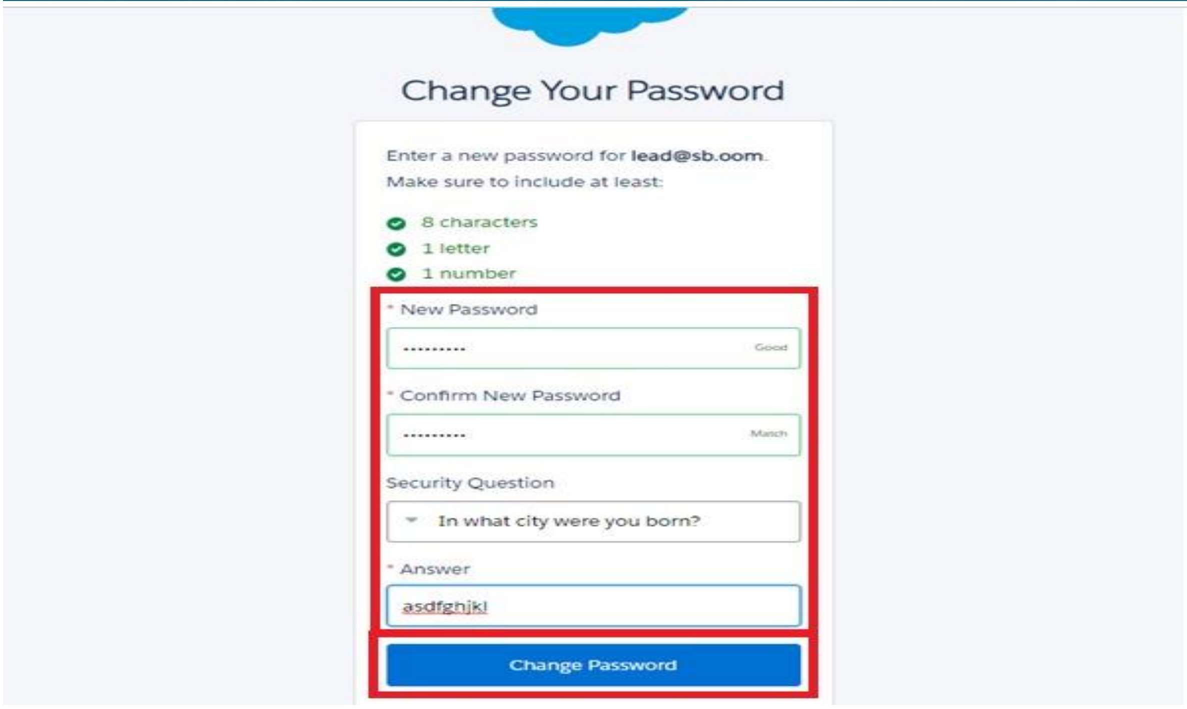
There's just one more step. Use the following link to reset the password for your Developer Edition. This link expires in 24 hours.

Reset Password

To easily log in later, save this URL:
https://orgfarm-c77642471b-dev-ed.develop.my.salesforce.com

Here's the username for your Developer Edition:
katigandlasowmya2004403@agentforce.com

Your Developer Edition, now enabled with Agentforce and Data Cloud, remains active as long as you continue to use it. It expires after 45 days of non-usage.


Change Your Password

Enter a new password for **lead@sb.oom**.
Make sure to include at least:

- ✓ 8 characters
- ✓ 1 letter
- ✓ 1 number

* New Password: ········· Good

* Confirm New Password: ········· Match

Security Question: In what city were you born?

* Answer: asdfghjkl

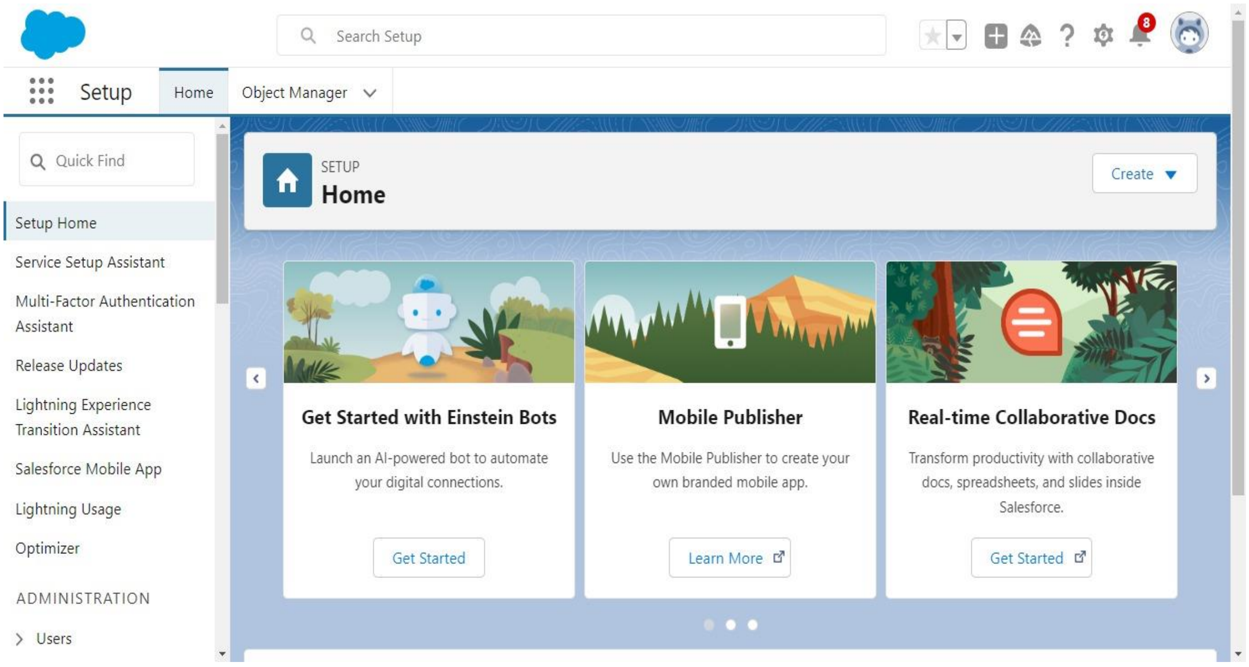
Change Password

1.3 Initial Configuration and Environment Familiarization

Upon login, the Lightning Experience interface was introduced. The setup menu provided access to all administrative tools. Key configurations included changing personal settings, setting up a custom domain (My Domain), and configuring language and locale preferences.

Developers explored the Object Manager, App Manager, and Home tab to get familiar with navigation. This foundational knowledge was critical to understanding how Salesforce data models and apps interact before starting the CRM customization.

1.4 Developer Console and Setup Tools


Search Setup
Setup
Home
Object Manager
Quick Find
Setup Home
Service Setup Assistant
Multi-Factor Authentication Assistant
Release Updates
Lightning Experience Transition Assistant
Salesforce Mobile App
Lightning Usage
Optimizer
ADMINISTRATION
Users
SETUP
Home
Create
Get Started with Einstein Bots
Launch an AI-powered bot to automate your digital connections.
Get Started
Mobile Publisher
Use the Mobile Publisher to create your own branded mobile app.
Learn More
Real-time Collaborative Docs
Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.
Get Started

The Salesforce Developer Console was used for testing Apex code and triggers. It also provided tools for debugging, logs, and script execution. The Setup area allowed developers to begin managing metadata components such as objects, fields, permissions, and flows.

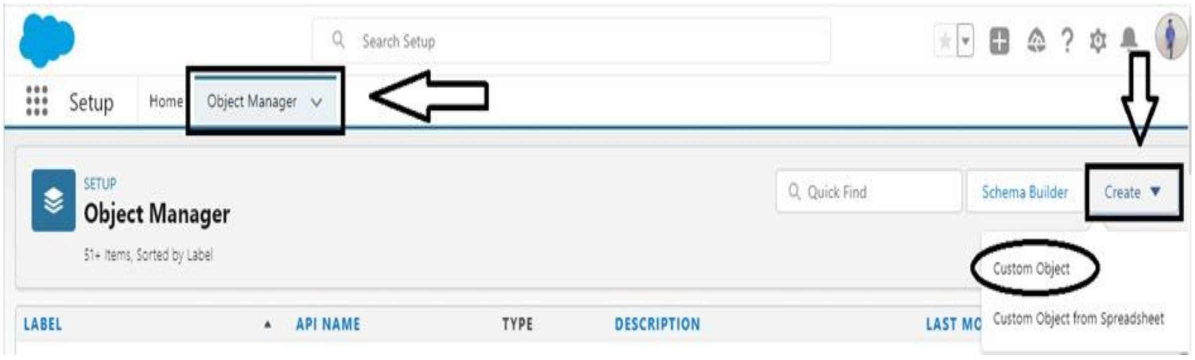
Understanding how to deploy changes from sandbox environments and retrieve component logs formed an important part of initial training.

1.5 Sandbox/Deployment Planning (Optional)

Though this was a developer-only instance, planning for deployment environments (like Full Sandbox, Partial Copy Sandbox, and Production) was documented to support future rollouts of this CRM solution into real-world business use cases.

Scope 2: Data Modeling – Custom Object and Tab Configuration

2.1 Custom Object Creation

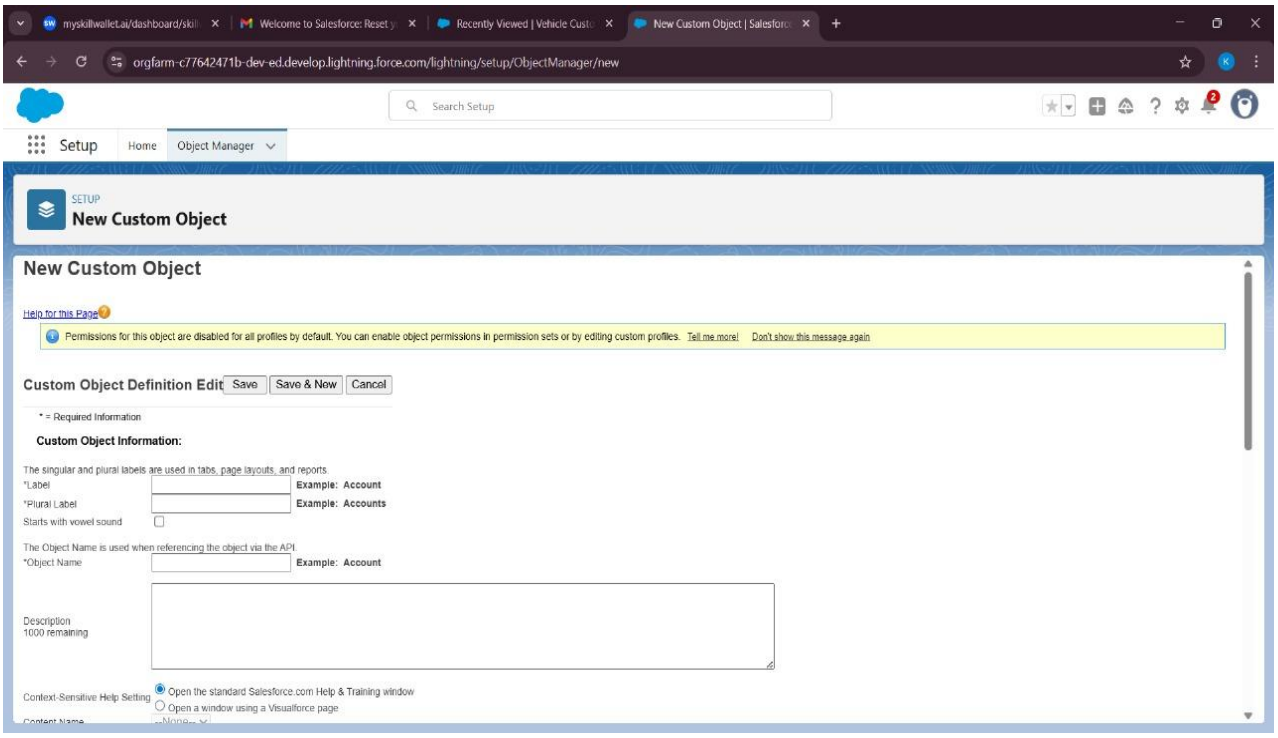


This phase began with the creation of essential custom objects in Salesforce to represent business entities within the automotive ecosystem. Each object was uniquely structured to support data collection, relationships, and reporting relevant to vehicles, customers, dealers, and services.
The following custom objects were created:

- Vehicle__c
- Vehicle_Dealer__c
- Vehicle_Order__c
- Vehicle_Customer__c
- Vehicle_Test_Drive__c
- Vehicle_Service_Request__c

2.2 Object Tab Creation

To ensure easy access and visibility of each object, individual tabs were created. These tabs allowed CRM users to view records, perform CRUD operations, and navigate seamlessly through the Lightning App interface.
Tabs were created for each object using the Lightning App Builder interface and then added to the custom app for WhatNext Vision Motors CRM.


myskillwallet.ai/dashboard/skill
Welcome to Salesforce: Reset y
Recently Viewed | Vehicle Custo
New Custom Object | Salesforce
orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
New Custom Object

New Custom Object

Help for this Page

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. Tell me more! Don't show this message again

Custom Object Definition Edit Save | Save & New | Cancel

* = Required Information

Custom Object Information:

The singular and plural labels are used in tabs, page layouts, and reports.

*Label		Example: Account
*Plural Label		Example: Accounts
Starts with vowel sound	☐	

The Object Name is used when referencing the object via the API.

*Object Name		Example: Account
Description 1000 remaining		
Context-Sensitive Help Setting	[✓ Open the standard Salesforce.com Help & Training window] Open a window using a Visualforce page	
Content Name	--None--	

2.3 Field Definitions and Relationships

Each object was enhanced with key fields relevant to the domain. Field types ranged from text, picklists, currency, dates, to lookups and relationships. Relationships such as Lookup and Master-Detail were created to link records across objects, enabling complex workflows and reporting.
Vehicle__c

- Vehicle_Name__c (Text)

- Vehicle_Model__c (Picklist: Sedan, SUV, EV, etc.)
- Stock_Quantity__c (Number)
- Price__c (Currency)
- Dealer__c (Lookup to Dealer__c)
- Status__c (Picklist: Available, Out of Stock, Discontinued)

Vehicle_Dealer__c

- Dealer_Name__c (Text)
- Dealer_Location__c (Text)
- Dealer_Code__c (Auto Number)
- Phone__c (Phone)
- Email__c (Email)

Vehicle_Order__c

- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Order_Date__c (Date)
- Status__c (Picklist: Pending, Confirmed, Delivered, Canceled)

Vehicle_Customer__c

- Customer_Name__c (Text)
- Email__c (Email)
- Phone__c (Phone)
- Address__c (Text)
- Preferred_Vehicle_Type__c (Picklist: Sedan, SUV, EV, etc.)

Vehicle_Test_Drive__c

- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Test_Drive_Date__c (Date)
- Status__c (Picklist: Scheduled, Completed, Canceled)

Vehicle_Service_Request__c

- Customer__c (Lookup to Customer__c)
- Vehicle__c (Lookup to Vehicle__c)
- Service_Date__c (Date)
- Issue_Description__c (Text)
- Status__c (Picklist: Requested, In Progress, Completed)

2.4 Mapping and Object Relationships

Object relationships were mapped to replicate real-world data interactions:

- A Customer can place multiple Orders (Lookup Relationship)
- Each Order is linked to a Vehicle (Lookup)
- Vehicles belong to Dealers (Lookup)

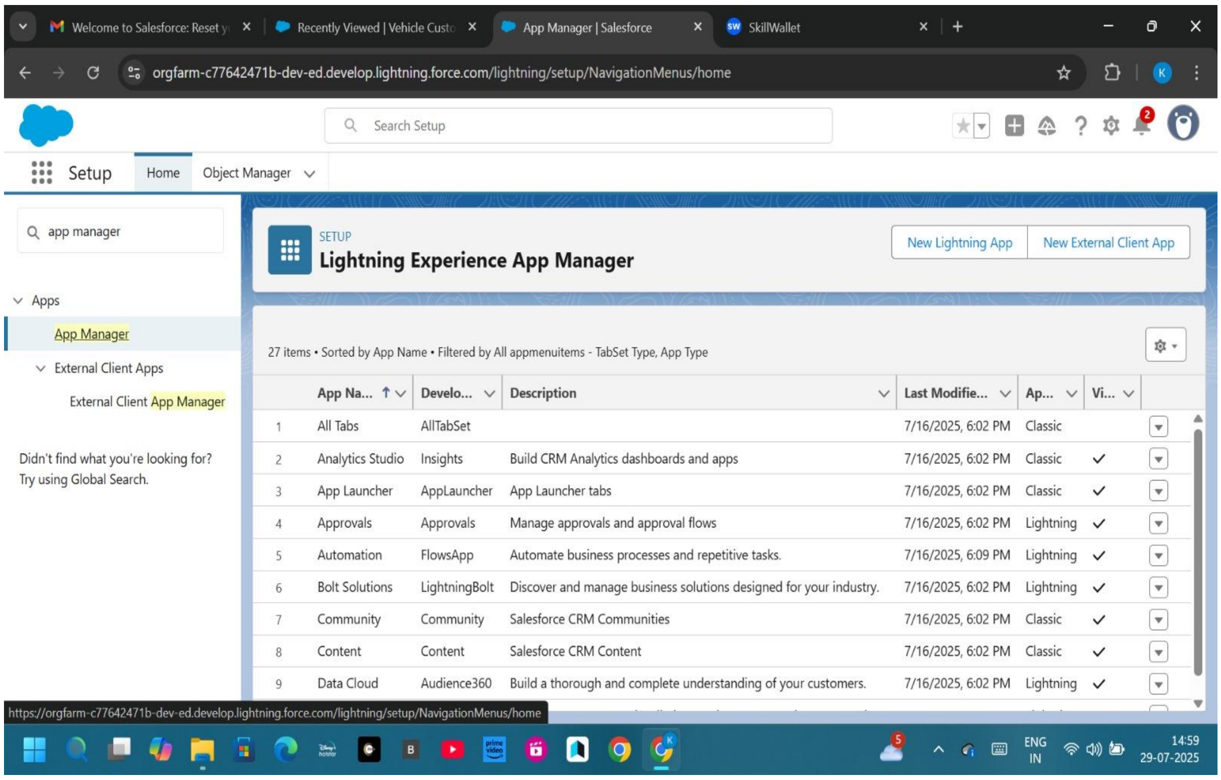
Object Name	Purpose	Relationships
Vehicle__c	Stores vehicle details	Related to Dealer & Orders
Vehicle_Dealer__c	Stores authorized dealer info	Related to Orders
Vehicle_Customer__c	Stores customer details	Related to Orders & Test Drives
Vehicle_Order__c	Tracks vehicle purchases	Related to Customer & Vehicle
Vehicle_Test_Drive__c	Tracks test drive bookings	Related to Customer & Vehicle
Vehicle_Service_Request__c	Tracks vehicle servicing requests	Related to Customer & Vehicle

This mapping ensured that related records could be fetched in reports, flows, and dashboard components, driving dynamic user experiences across the CRM.

Scope 3: Lightning App Customization & Navigation Experience

3.1 App Creation Using Lightning App Builder

With the foundational objects and fields in place, a custom Lightning App was built to serve as the main interface for the WhatNext Vision Motors CRM. Using the Lightning App Builder, the development team created a branded, user-friendly app that centralized all relevant tabs, pages, and tools required for everyday business operations.


orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home

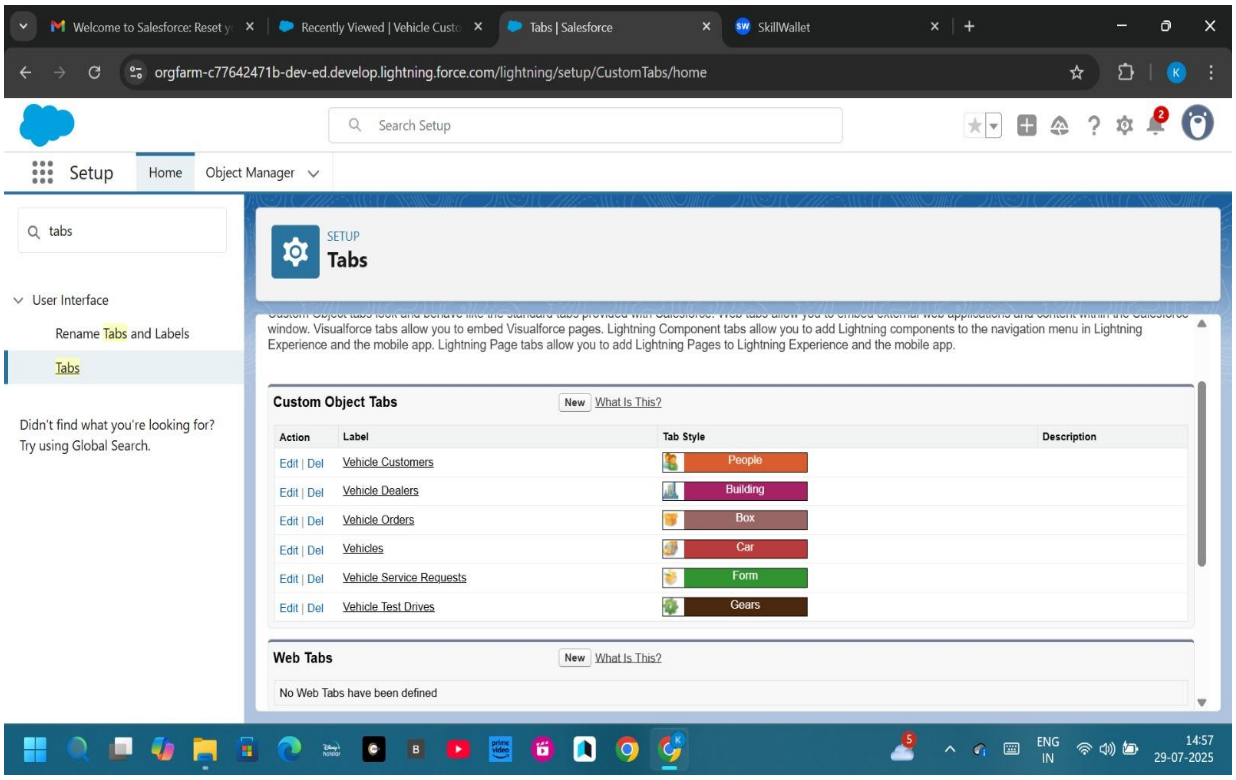
SETUP
Lightning Experience App Manager

27 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type, App Type

	App Na...	Develo...	Description	Last Modifie...	Ap...	Vi...
1	All Tabs	AllTabSet		7/16/2025, 6:02 PM	Classic	
2	Analytics Studio	Insights	Build CRM Analytics dashboards and apps	7/16/2025, 6:02 PM	Classic	✓
3	App Launcher	AppLauncher	App Launcher tabs	7/16/2025, 6:02 PM	Classic	✓
4	Approvals	Approvals	Manage approvals and approval flows	7/16/2025, 6:02 PM	Lightning	✓
5	Automation	FlowsApp	Automate business processes and repetitive tasks.	7/16/2025, 6:09 PM	Lightning	✓
6	Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	7/16/2025, 6:02 PM	Lightning	✓
7	Community	Community	Salesforce CRM Communities	7/16/2025, 6:02 PM	Classic	✓
8	Content	Content	Salesforce CRM Content	7/16/2025, 6:02 PM	Classic	✓
9	Data Cloud	Audience360	Build a thorough and complete understanding of your customers.	7/16/2025, 6:02 PM	Lightning	✓

The app was named "Vision Motors CRM" and included all essential custom objects like Vehicle__c, Dealer__c, Vehicle_Order__c, and more. Navigation items were carefully ordered to reflect actual business workflows, starting from customer onboarding to vehicle assignment and test drive scheduling.

3.2 Tab Management and UI Customization


orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home

SETUP
Tabs

window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

Action	Label	Tab Style	Description
Edit | Del	Vehicle Customers	People	
Edit | Del	Vehicle Dealers	Building	
Edit | Del	Vehicle Orders	Box	
Edit | Del	Vehicles	Car	
Edit | Del	Vehicle Service Requests	Form	
Edit | Del	Vehicle Test Drives	Gears	

Web Tabs

No Web Tabs have been defined

Navigation tabs were organized in a logical sequence to reflect the typical customer journey. Standard objects such as Accounts, Contacts, and Reports were also included where needed for cross-functional access.
The app was enhanced with app branding settings such as a custom logo, color themes, and a personalized utility bar that includes key features like global search, recent items, and quick create actions.

3.3 App-Level Access Control

App visibility was controlled using profiles and permission sets. Only designated users had access to the full Vision Motors CRM app. For example, sales executives could view and create orders, while service agents had limited visibility into the service request tab only.
User roles and app settings were tightly managed to maintain both security and user relevance.

3.4 Layout Configuration and Record Pages

The Lightning App Builder was also used to customize record page layouts for key objects like Vehicle__c and Dealer__c. These pages were configured with dynamic components including:

- Highlights Panel for quick info
- Related List Single for viewing associated orders or test drives
- Tabs and Accordion elements for sectioning content
- Embedded charts or reports for real-time data views

These layouts drastically improved usability and helped each department work with contextual information at a glance.

3.5 Quick Actions and Utility Bar Enhancements

Quick actions were added at the object and app level to allow users to perform frequent tasks (like creating a new order, scheduling a test drive) with fewer clicks. The utility bar provided access to recent items, notes, and global actions improving navigation efficiency and productivity.

Scope 4: Custom Fields & Relationship Architecture

4.1 Object-Specific Field Customization

To ensure that the CRM system meets the unique needs of WhatNext Vision Motors, several custom fields were added to key objects. These fields were thoughtfully selected to represent core data requirements in the automotive domain such as stock quantity, preferred vehicle type, and issue descriptions.

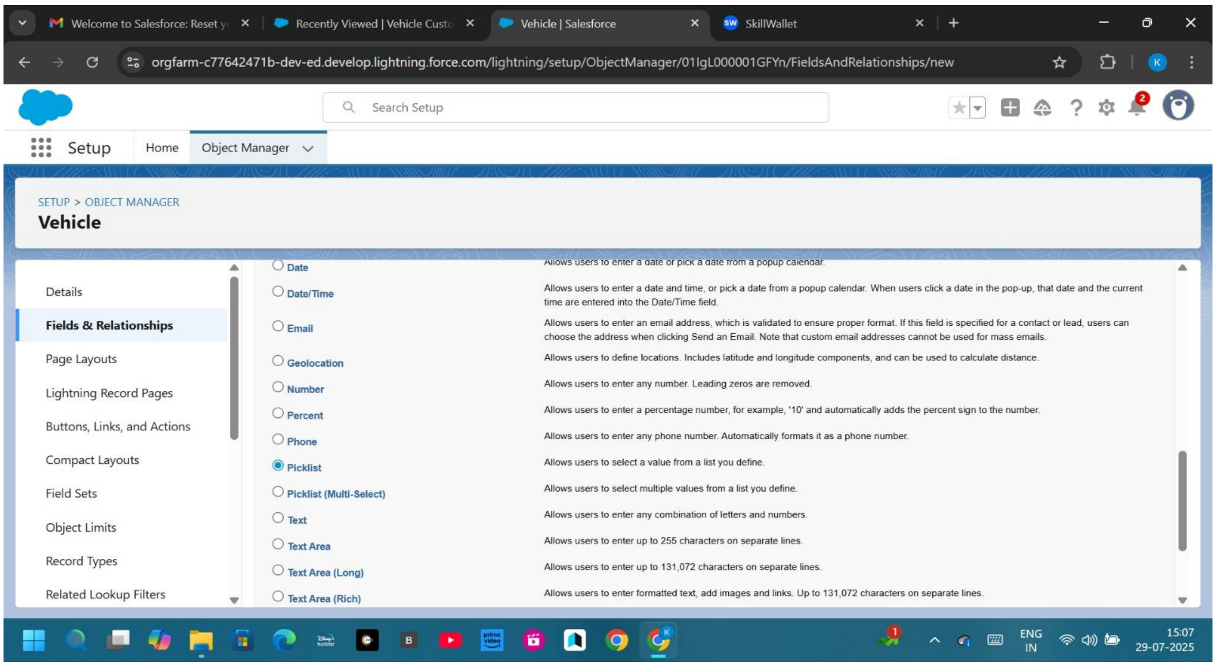
4.2 Field Types and Input Controls

Fields used a combination of input types including Text, Number, Picklist, Currency, Email, Phone, Date, and Lookup. This ensured both structured data capture and enhanced data validation.

Examples include:

- Vehicle_Model__c (Picklist)
- Dealer_Location__c (Text)
- Stock_Quantity__c (Number)
- Status__c (Picklist: Available, Out of Stock, Discontinued)
- Order_Date__c (Date)
- Issue_Description__c (Text)

4.3 Field Relationships and Lookups


orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000001GFYn/FieldsAndRelationships/new

SETUP > OBJECT MANAGER
Vehicle

○ Date	Allows users to enter a date or pick a date from a popup calendar.
○ Date/Time	Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
○ Email	Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
○ Geolocation	Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
○ Number	Allows users to enter any number. Leading zeros are removed.
○ Percent	Allows users to enter a percentage number, for example, '10' and automatically adds the percent sign to the number.
○ Phone	Allows users to enter any phone number. Automatically formats it as a phone number.
● Picklist	Allows users to select a value from a list you define.
○ Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
○ Text	Allows users to enter any combination of letters and numbers.
○ Text Area	Allows users to enter up to 255 characters on separate lines.
○ Text Area (Long)	Allows users to enter up to 131,072 characters on separate lines.
○ Text Area (Rich)	Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.

Lookups were extensively used to establish meaningful relationships between records. For example:

- Dealer__c (Lookup on Vehicle__c)
- Customer__c (Lookup on Vehicle_Order__c, Vehicle_Test_Drive__c, and Service_Request__c)
- Vehicle__c (Lookup on Orders, Test Drives, and Service Requests)

4.4 Field-Level Security and Access Controls

Field-Level Security (FLS) was applied using profiles and permission sets. Only relevant users could access/edit specific fields based on their role. For example, only inventory managers could edit stock-related fields.

Additionally, page layouts were customized for different profiles to show/hide fields as per operational relevance.

4.5 Validation Rules

Validation rules were implemented to maintain data accuracy. Rules included:

- Preventing order placement if vehicle status is 'Out of Stock'
- Ensuring phone numbers are correctly formatted
- Mandatory fields like Dealer Name, Email, and Service Description must be filled before saving

4.6 Summary of Object and Field Mapping

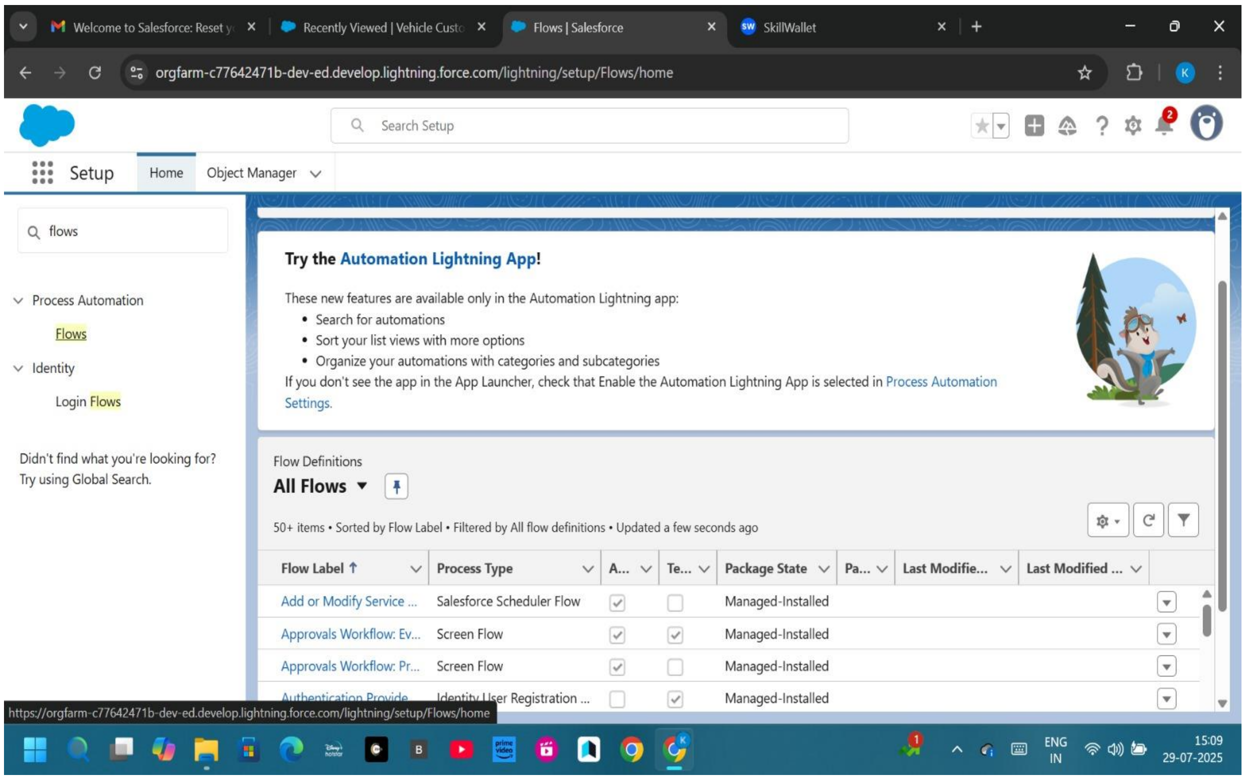
A detailed mapping of objects and fields ensures consistency and aligns with real-world business logic. This structure also facilitates accurate reporting and automation.

Each object supports a specific use case:

- Vehicle__c → Inventory
- Vehicle_Dealer__c → Dealer Management
- Vehicle_Order__c → Customer Purchase Records
- Vehicle_Customer__c → Customer Master Data
- Vehicle_Test_Drive__c → Pre-Sales Engagement
- Vehicle_Service_Request__c → Post-Sales Support

Scope 5: Process Automation Using Flow Builder

5.1 Flow Creation and Automation Strategy



Welcome to Salesforce: Reset y
Recently Viewed | Vehicle Custo
Flows | Salesforce
SkillWallet
orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/Flows/home
Search Setup
Setup
Home
Object Manager
flows
Process Automation
Flows
Identity
Login Flows

Didn't find what you're looking for? Try using Global Search.

Try the Automation Lightning App!

These new features are available only in the Automation Lightning app:

- Search for automations
- Sort your list views with more options
- Organize your automations with categories and subcategories

If you don't see the app in the App Launcher, check that Enable the Automation Lightning App is selected in Process Automation Settings.

Flow Definitions

All Flows

50+ items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

Flow Label ↑	Process Type	A...	Te...	Package State	Pa...	Last Modifie...	Last Modified ...
Add or Modify Service ...	Salesforce Scheduler Flow	✓		Managed-Installed			
Approvals Workflow: Ev...	Screen Flow	✓	✓	Managed-Installed			
Approvals Workflow: Pr...	Screen Flow	✓		Managed-Installed			
Authentication Provide...	Identity User Registration ...		✓	Managed-Installed			

https://orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/Flows/home

Salesforce Flow Builder was utilized extensively to implement automation that replaced manual processes. These automations were central to improving user productivity, eliminating errors, and ensuring seamless business operations. Record-triggered flows, scheduled flows, and decision logic were all employed to create intelligent processes.

5.2 Record-Triggered Flows for Dealer Assignment

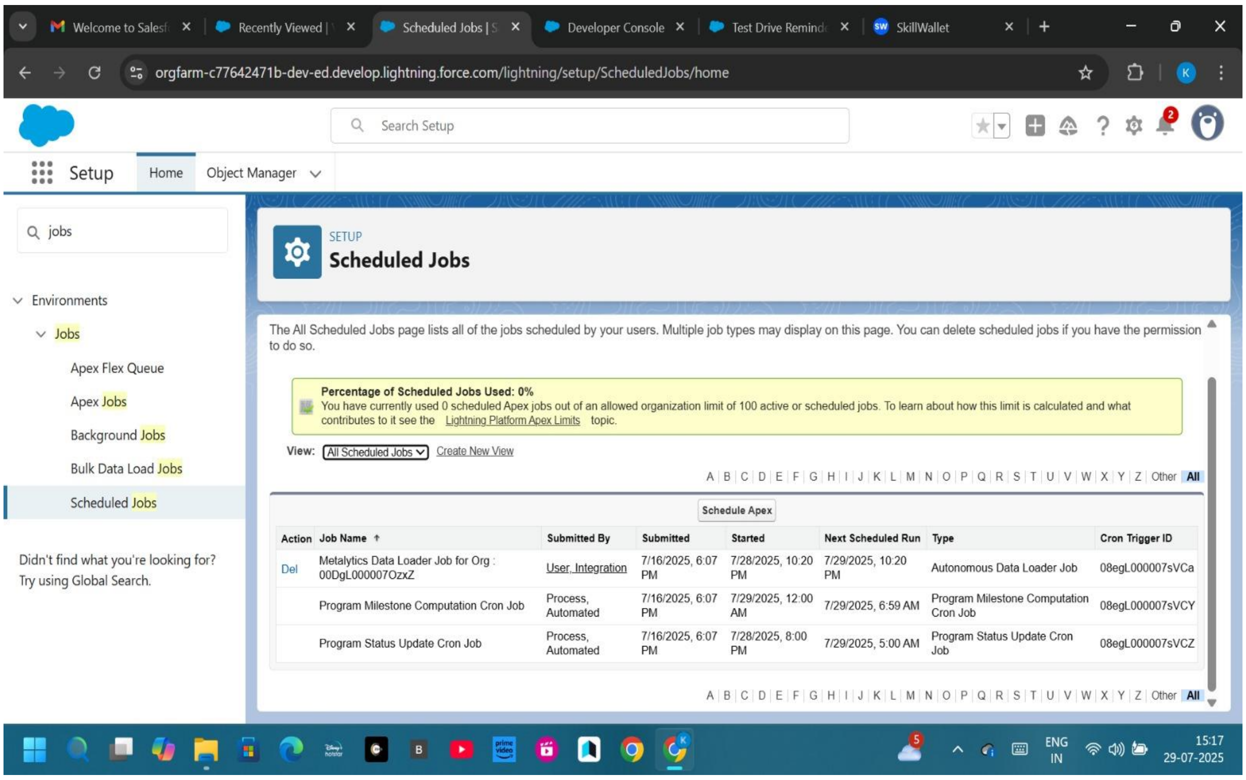
A record-triggered flow was configured on the Vehicle_Order__c object. When a new order was created, the flow would determine the nearest dealer based on the customer's address and assign that dealer to the order.

The flow followed these steps:

- Retrieve customer location
- Match location with nearest Dealer__c record
- Update the Dealer__c lookup on the order
- Send confirmation email to the customer

This real-time automation removed manual dealer matching and ensured orders were routed instantly to the appropriate branch.

5.3 Scheduled Flow for Order Status Management



SETUP
Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

Percentage of Scheduled Jobs Used: 0%
You have currently used 0 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the Lightning Platform Apex Limits topic.

View: All Scheduled Jobs | Create New View

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Del	Metalytics Data Loader Job for Org : 00DgL000007OzxZ	User, Integration	7/16/2025, 6:07 PM	7/28/2025, 10:20 PM	7/29/2025, 10:20 PM	Autonomous Data Loader Job	08egL000007sVCa
	Program Milestone Computation Cron Job	Process, Automated	7/16/2025, 6:07 PM	7/29/2025, 12:00 AM	7/29/2025, 6:59 AM	Program Milestone Computation Cron Job	08egL000007sVCY
	Program Status Update Cron Job	Process, Automated	7/16/2025, 6:07 PM	7/28/2025, 8:00 PM	7/29/2025, 5:00 AM	Program Status Update Cron Job	08egL000007sVCZ

Welcome to Salesf
Recently Viewed |
Scheduled Jobs | S
Developer Console
Test Drive Remind
SkillWallet
orgfarm-c77642471b-dev-ed.develop.lightning.force.com/lightning/setup/ScheduledJobs/home
Search Setup
Setup
Home
Object Manager
jobs
Environments
Jobs
Apex Flex Queue
Apex Jobs
Background Jobs
Bulk Data Load Jobs
Scheduled Jobs
Didn't find what you're looking for? Try using Global Search.
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
Schedule Apex
ENG IN
15:17
29-07-2025

Another critical automation was a scheduled flow that ran at fixed intervals (e.g., every 2 hours) to review all pending Vehicle_Order__c records. Based on the current stock of the requested Vehicle__c, the system would update order statuses:

- If Stock_Quantity__c > 0 → Status = Confirmed

- If Stock_Quantity__c = 0 → Status = Pending

This automation minimized false confirmations and helped customers receive timely status updates, improving trust and communication.

5.4 Test Drive Scheduling Automation

Flows were also used to manage test drive requests. When a Vehicle_Test_Drive__c record was created, an email was automatically sent to both the customer and the nearest dealer, confirming the appointment.

The system checked for conflicts or overlapping schedules and notified users accordingly. This improved coordination and customer satisfaction.

5.5 Email Alerts and Notifications

Multiple email alerts were configured across flows:

- Order placement confirmation
- Test drive scheduling
- Status updates for order processing
- Service request acknowledgment

These communications were triggered automatically based on user actions or field updates, ensuring customers stayed informed throughout the journey.

5.6 Flow Error Handling and Best Practices

Each flow included fault paths and error notifications to prevent incomplete automation runs. Logging flows, using decision nodes effectively, and applying fault connectors helped maintain flow reliability.

Scope 6: Apex Triggers, Batch Jobs & Scheduled Processes

6.1 Order Validation with Apex Triggers

Apex triggers were implemented on the Vehicle_Order__c object to automate business rules related to stock management and dealer assignment. The trigger performs two primary functions:

- Prevent Order Placement: Before insert/update, it checks the stock quantity. If the vehicle is out of stock, the trigger prevents the order from being saved.
- Update Stock: After insert/update, it decreases the stock count of the associated vehicle to reflect real-time inventory levels.

These trigger actions ensure data consistency and improve the reliability of the ordering process.

6.2 Trigger Handler Class

To maintain modularity and follow Salesforce best practices, a trigger handler class (VehicleOrderTriggerHandler) was used. This class separates business logic from the trigger and ensures scalability and easy maintenance.
The structure handles multiple operations and isolates stock validation and stock update logic into dedicated methods like preventOrderIfOutOfStock() and updateStockOnOrderPlacement().

6.3 Batch Apex for Stock Review

A batch class (VehicleOrderBatch) was developed to process pending vehicle orders in bulk. It queries all Vehicle_Order__c records with a status of 'Pending' and checks the availability of the associated vehicle.
If the stock is found available, the batch process updates the order status to 'Confirmed'. This automation is especially useful for managing large datasets and enhances operational efficiency during high-demand periods.

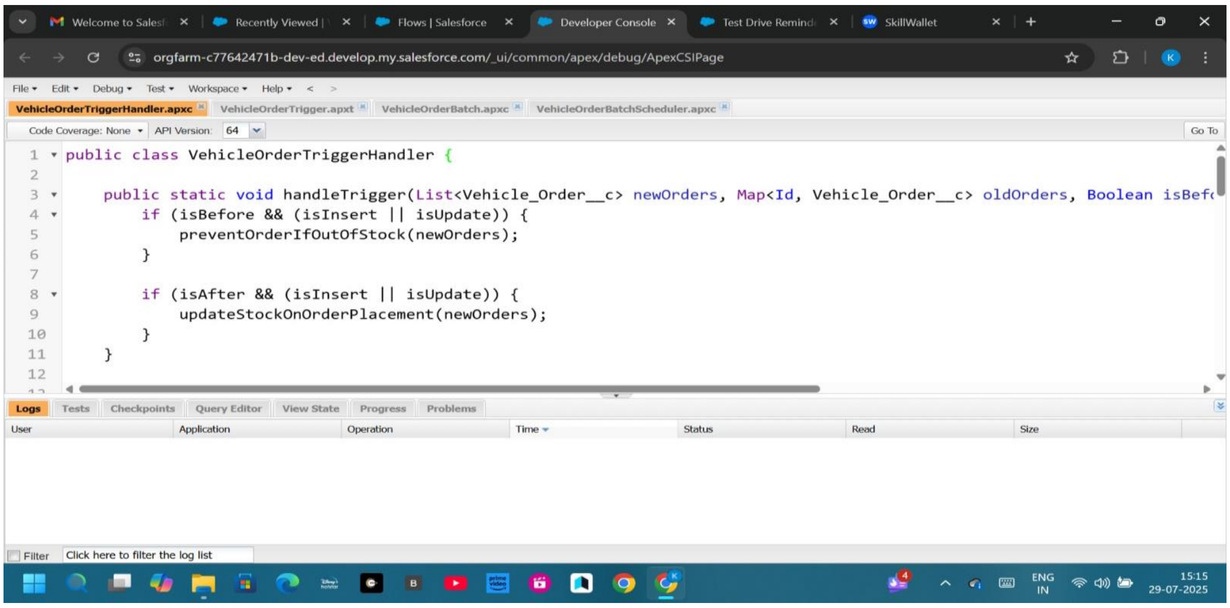
6.4 Scheduled Apex Job

To keep stock status in sync without manual intervention, the batch process was scheduled using the VehicleOrderBatchScheduler class. This class implements the Schedulable interface and executes the batch every defined interval, such as daily or hourly.
This scheduled job ensures regular order validations and stock syncing, supporting smooth automation even during off-hours.

6.5 Visual Proof and Screenshots

Below are screenshots of the implemented logic:

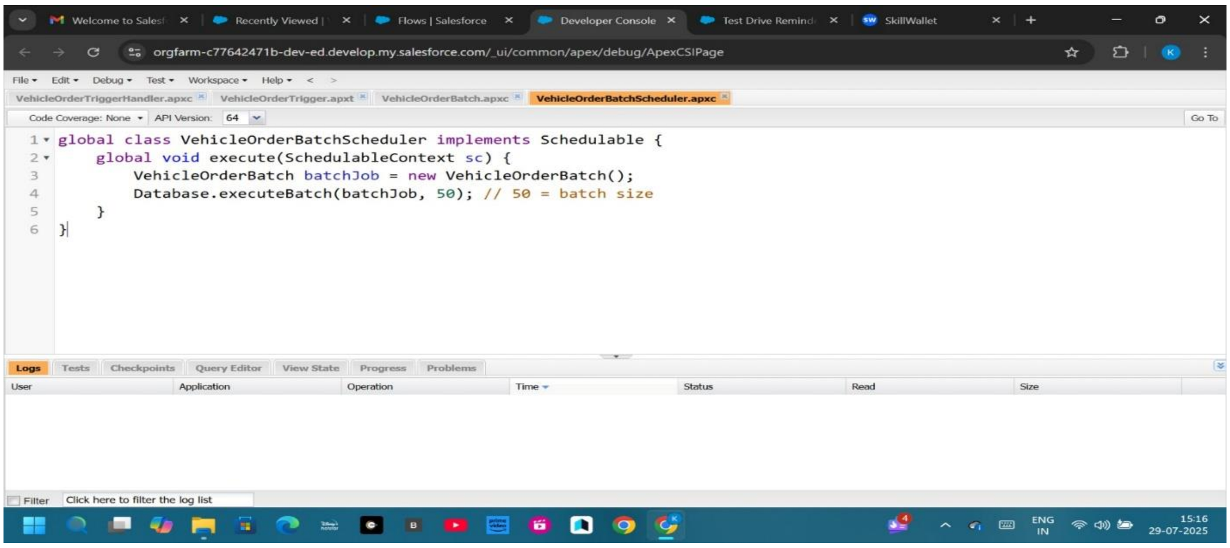
• Trigger Code:



```
public class VehicleOrderTriggerHandler {

    public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBef
        if (isBefore && (isInsert || isUpdate)) {
            preventOrderIfOutOfStock(newOrders);
        }

        if (isAfter && (isInsert || isUpdate)) {
            updateStockOnOrderPlacement(newOrders);
        }
    }

```

Welcome to Salesf
Recently Viewed |
Flows | Salesforce
Developer Console
Test Drive Remind
SkillWallet
orgfarm-c77642471b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 64
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Filter
Click here to filter the log list
ENG IN
15:15
29-07-2025

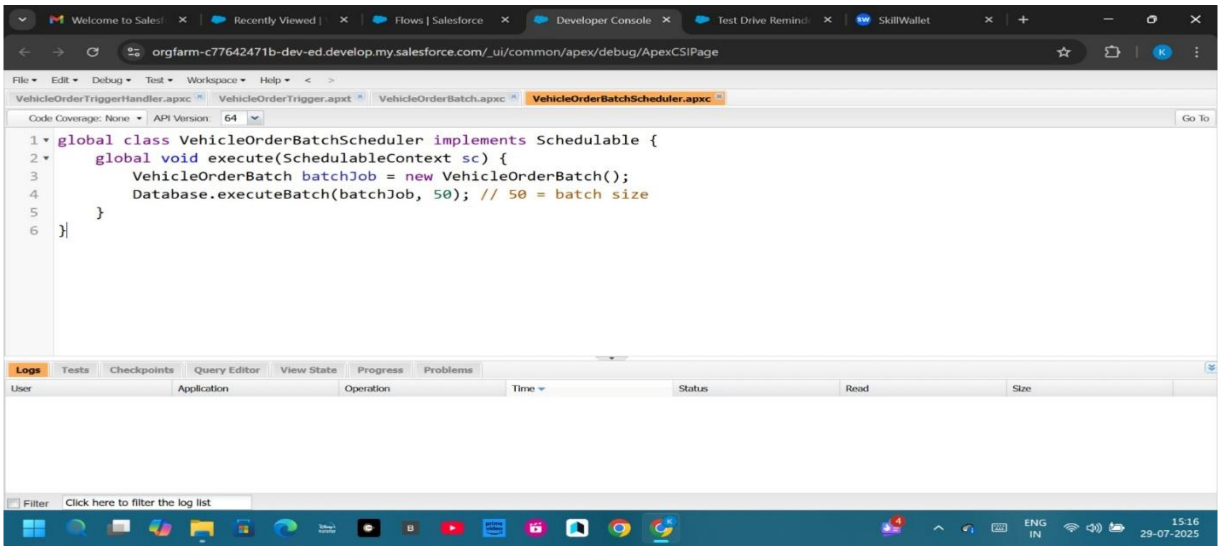
• Batch Class:



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

orgfarm-c77642471b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 64
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

• Scheduled Class:



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

orgfarm-c77642471b-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
VehicleOrderTriggerHandler.apxc
VehicleOrderTrigger.apxt
VehicleOrderBatch.apxc
VehicleOrderBatchScheduler.apxc
Code Coverage: None
API Version: 64
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

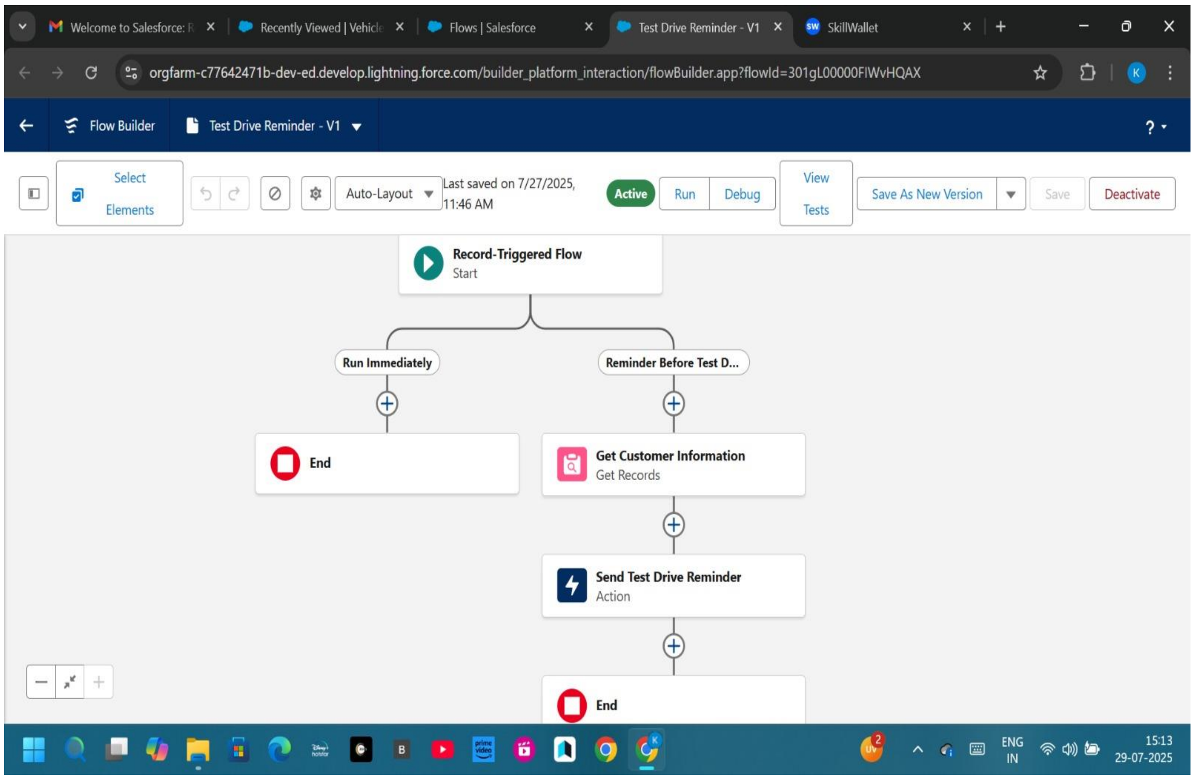
Scope 7: Reports, Dashboards & Visual Automations

7.1 Test Drive Reminder Flow

A record-triggered flow was created to automate reminder notifications for scheduled test drives. The flow triggers based on the Test_Drive_Date__c field in the Vehicle_Test_Drive__c object. Once scheduled, the system sends an automated email reminder to the customer a day prior to the test drive, ensuring timely communication and reducing the likelihood of missed appointments.

This flow uses "Get Records" to fetch the customer's contact information and "Send Email" to dispatch the reminder. It enhances service efficiency, reduces manual follow-ups, and adds a professional touch to customer communication.

7.2 Order Confirmation Flow



A second record-triggered flow handles order confirmation communication. When a Vehicle_Order__c record's status is updated to 'Confirmed', an email confirmation is automatically sent to the customer.
The flow improves user experience by providing real-time updates and builds customer trust by clearly communicating progress on their vehicle order. The email includes vehicle details, estimated delivery timeline, and dealer contact information for convenience.

7.3 Stock Update Notifications

A scheduled flow was created to check vehicle stock levels at regular intervals. If stock is replenished for an out-of-stock vehicle, customers who had pending orders are notified automatically.
This proactive communication helps retain leads and increases conversion chances by informing customers as soon as their preferred vehicle is back in stock.

7.4 Custom Reports & Dashboards

To ensure stakeholders have access to real-time business insights, multiple reports and dashboards were created using Salesforce's Report Builder.
Examples include:

- Vehicle Inventory Status by Model and Dealer
- Order Fulfillment Rate by Week
- Test Drive Conversion Funnel

- Customer Service Requests by Status

Dashboards were shared with team leaders and executives to facilitate data-driven decision-making and improve departmental visibility.

7.5 Future Enhancements & Scalability

This CRM solution was built with scalability in mind. Future roadmap includes:

- Integrating AI for predictive vehicle demand
- Implementing Chatbots for real-time service scheduling
- Adding multilingual support for regional expansion
- Enabling mobile-friendly Lightning components for on-the-go access

Such additions will further empower the business to innovate and evolve as the mobility landscape changes.

7.6 Summary of Process Coverage

From lead capture to service follow-up, this project has implemented end-to-end automation and insights through a Salesforce-native solution. Major accomplishments include:

- Full object model tailored to the automotive industry
- Streamlined dealer assignment and stock validation
- Proactive order and test drive management
- Automated communication across customer journey
- Real-time dashboards for transparency

Conclusion

The implementation of the Salesforce CRM project at WhatNext Vision Motors marks a pivotal transformation in the company's operations, customer engagement, and technological innovation. Through this initiative, the company has effectively digitalized and automated crucial components of its vehicle order management, dealer coordination, and service scheduling processes. The integration of Salesforce capabilities, ranging from custom object design to Apex-based logic, has established a strong foundation for a scalable and intelligent system.

This project has brought forward several high-impact advancements:

- **Enhanced Customer Experience:** With automated dealer assignments, intelligent stock checks, and scheduled reminders for test drives, customers are now experiencing a seamless and transparent interaction throughout their buying journey.
- **Operational Efficiency:** The use of flows, Apex triggers, batch classes, and scheduled jobs hsas reduced the dependency on manual tasks,

enabling staff to focus on strategic functions while ensuring data accuracy and timely updates.

- **Real-Time Inventory and Order Management:** The ability to validate stock before order placement, update order status based on availability, and keep inventory in sync ensures that both customers and dealers have access to real-time, reliable data.
- **Data Transparency and Governance:** Custom fields and relationships, validation rules, and role-based permissions help maintain clean, secure, and well-structured datasets across various functional modules.

Looking ahead, the system is well-equipped to support more advanced functionalities like AI-powered vehicle recommendations, predictive service alerts, and analytics-driven insights to enhance decision-making. As the platform continues to evolve, it holds the potential to drive further excellence in customer satisfaction, operational agility, and business growth.

Ultimately, this project serves as a blueprint for digital innovation in the mobility sector—proving that with the right technological foundation, companies like WhatNext Vision Motors can not only meet but exceed customer expectations in a fast-changing automotive landscape.