WhatsNext Vision Motors Salesforce CRM Implementation

Final Project Report

Name: Soma Shekar Rayavarapu

Batch: 1 - TCS LastMile

Mentor: Ms. Mounika Katta

PRAGATI ENGINEERING COLLEGE, SURAMPALEM
Department: Data Science

Phase 1: Problem Understanding & Industry Analysis

Requirement Gathering

- It's done through methods like workshops, interviews, and surveys to understand exactly what the business needs the system to do.
- The goal is to create a detailed list of all functional and technical requirements.

Stakeholder Analysis

- You then analyze their influence, expectations, and how the project will impact them.
- Understanding stakeholders is crucial for managing expectations and ensuring the final product is adopted successfully by its users.

Business Process Mapping

- You map the **"as-is" process** to identify current pain points, bottlenecks, and inefficiencies.
- You then design the **"to-be" process**, which illustrates how the workflow will operate more efficiently with Salesforce automation.

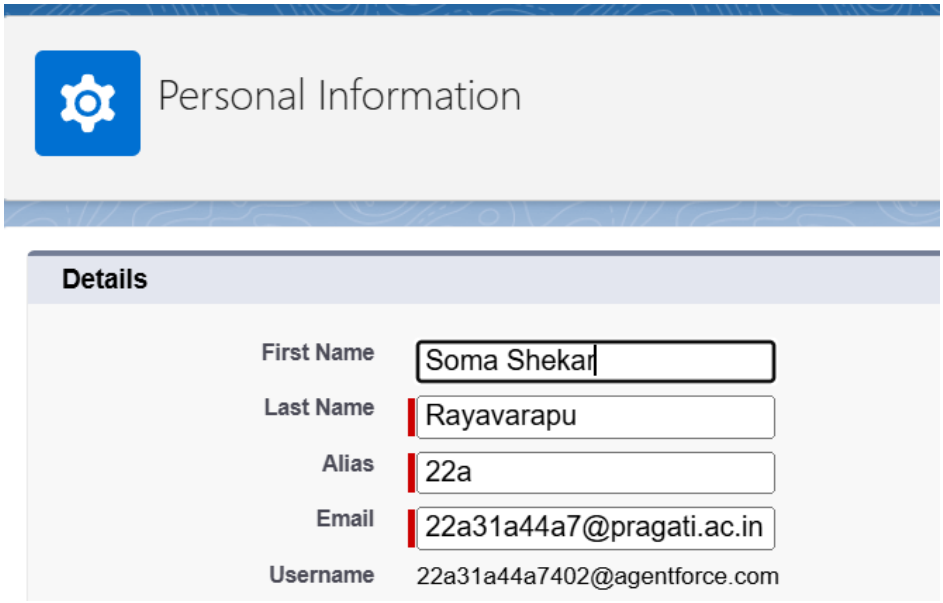
Industry-specific Use Case Analysis

- This step involves analyzing the project's requirements within the specific context of the client's industry.
- It requires focusing on industry-specific challenges, like supply chain issues in the automotive sector or compliance regulations in healthcare.

Phase 2: Org Setup & Configuration

Key activities in this phase:

- Org Setup: Create a Developer Org
- User Setup: Soma Shekar Rayavarapu.
- Sandboxes: Created for testing and UAT before production deployment.
- Deployment Basics: Run and save the apex code in the developer console.



Phase 3: Data Modeling & Relationships

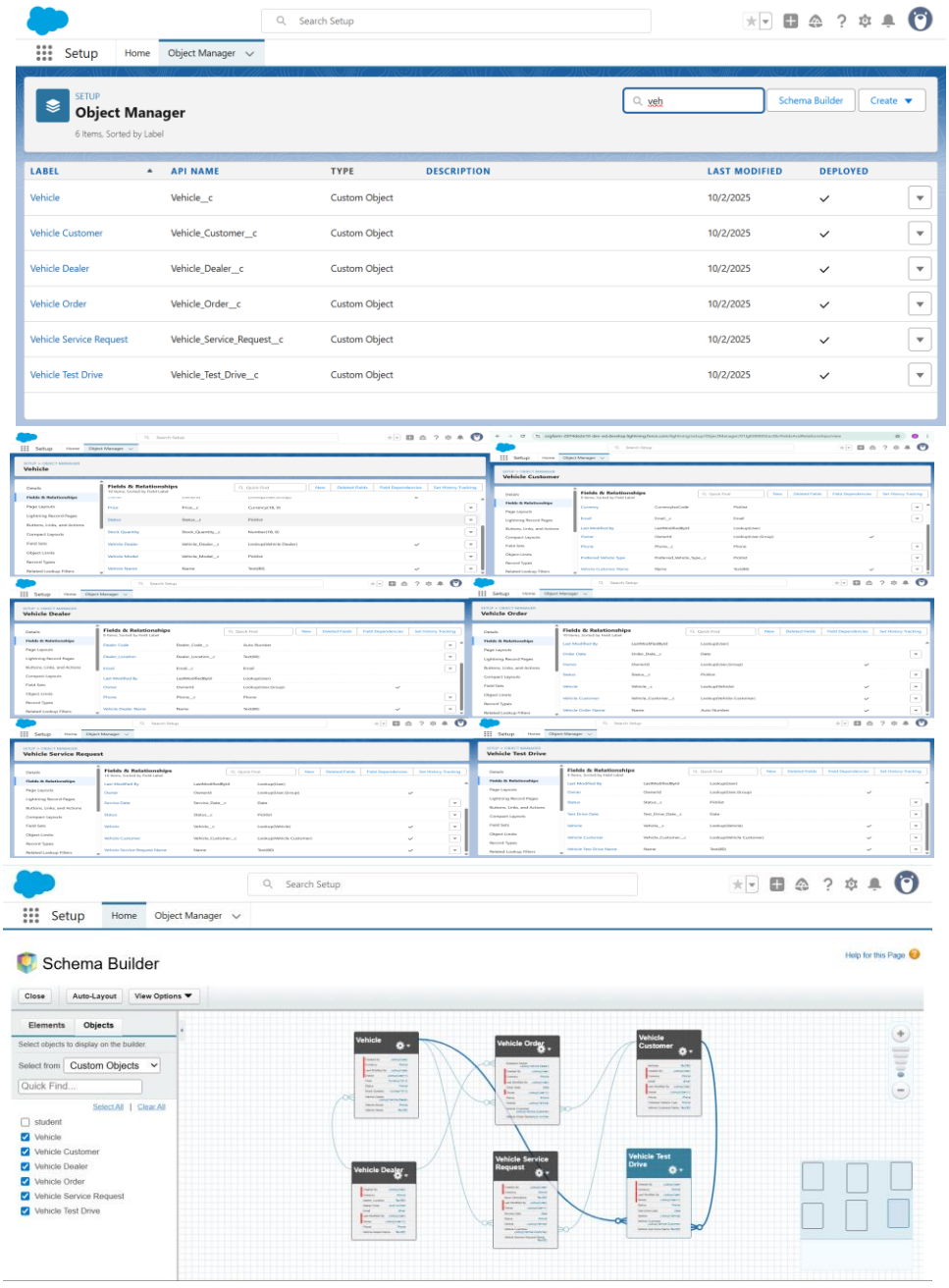
Prescription workflows require custom objects, relationships and fields.

- **Objects**: Vehicle, Vehicle Customer, Vehicle Dealer, Vehicle Test Drive, Vehicle Service Request, Vehicle Order.
- **Fields**: Vehicle Name, Stock Quantity, Price, Dealer, Dealer Location, Dealer Code, Phone, Email, Order Date, Address, Test Drive Date, Service Date, Issue Description.
- **Refill Request:** Vehicle Status (Available, Out of Stock, Discontinued), Vehicle Model (Sedan, SUV, EV, etc.), Order Status (Pending, Confirmed, Delivered, Canceled), Test Drive Status (Scheduled, Completed, Canceled), Service Request Status (Requested, In Progress, Completed).
- **Relationships:**

  Vehicle — Dealer (lookup)

  Vehicle Order — Customer (lookup)

Vehicle Order — Vehicle (lookup)
Vehicle Test Drive — Customer (lookup)
Vehicle Test Drive — Vehicle (lookup)
Vehicle Service Request — Customer (lookup)
Vehicle Service Request — Vehicle (lookup)



Phase 4: Process Automation (Admin)

Flow Builder: The project extensively uses Flow Builder, Salesforce's primary tool for declarative automation, to handle key business processes.

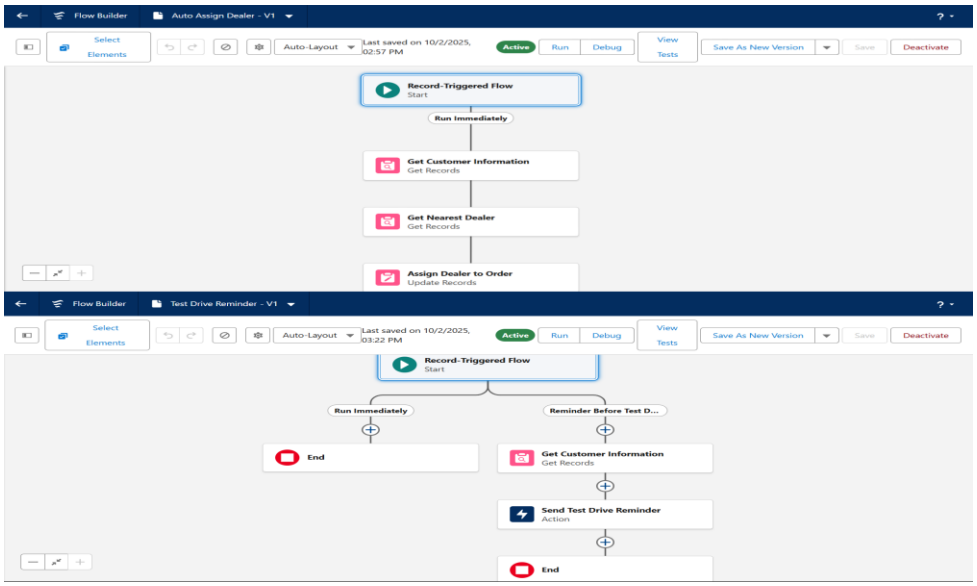
- **Record-Triggered Flow**: An automated flow was built to assign the nearest dealer to a customer's order based on their location.
- **Scheduled Path Flow**: A flow with a scheduled path was created to notify a customer one day before their scheduled test drive.

Email Alerts: Email notifications are a key part of the project's customer communication strategy.

- **Test Drive Reminders**: The project sends automated email reminders to customers about their upcoming test drives.
- **Stock Notifications**: The project includes email notifications to be sent to customers when a previously out-of-stock vehicle is refilled.

Field Updates: This action is used within the flows to modify record data automatically.

- **Dealer Assignment**: The record-triggered flow automatically performs a **field update** by populating the dealer lookup field on the vehicle order record, linking it to the correct dealer



Phase 5: Apex Programming (Developer)

Apex Triggers (before/after insert/update) :

Triggers were used to enforce real-time business logic on the `Vehicle_Order__c` object.

- The project implemented logic for validating stock availability before an order is placed and deducting from the stock count after the order is confirmed.

Classes & Objects :

The project was built using Apex classes to structure the code.

- The documentation mentions that the implementation built Apex triggers, classes, and batch jobs to handle the custom logic.

Trigger Design Pattern :
A handler pattern was used to keep the trigger file itself clean and scalable.

- The trigger's role was to delegate all processing logic to a separate handler class, which is a Salesforce best practice.

SOQL & Collections (List, Set, Map) :

Though not explicitly named in the summary text, the code snippets provided in the PDF report show the use of these fundamental concepts.

- SOQL (Salesforce Object Query Language) was used to retrieve vehicle and order records from the database.
- Collections like `List` and `Set` were used to efficiently handle and process records in bulk to avoid hitting governor limits.

Batch Apex :
This was used for processing a large number of records asynchronously.

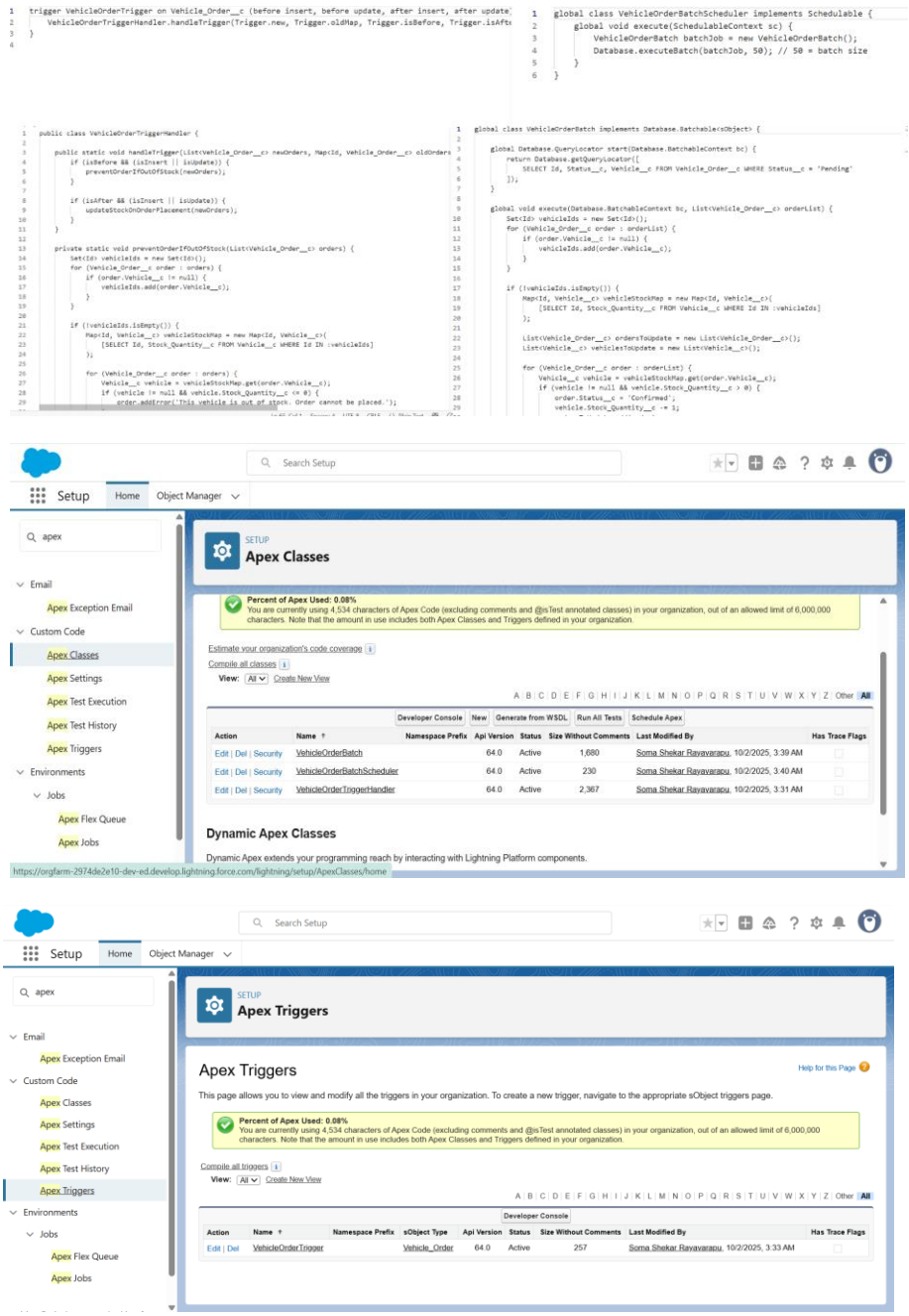
- The project uses a batch job to automate back-order fulfillment by processing all 'Pending' orders in bulk, checking if stock has become available and confirming them.

Scheduled Apex :
This was used to run processes at a specific time.

- A schedulable class was built to **automate the `VehicleOrderBatch` job by putting it on a recurring schedule**, ensuring that pending orders are checked regularly without manual intervention.

Asynchronous Processing :

- The project implemented asynchronous Apex to handle tasks that are too large or long-running for immediate execution.
- The use of both **Batch Apex** and **Scheduled Apex** are the project's methods for handling asynchronous processing.



Phase 6: User Interface Development

Lightning App Builder & Tabs:

- A custom Lightning App was built using the Lightning App Builder to serve as the primary user interface for the project.
- The application includes custom Tabs for each object (e.g., Vehicle Customer, Vehicle Order) to provide users with easy navigation to records.

Record Pages & Home Page Layouts:

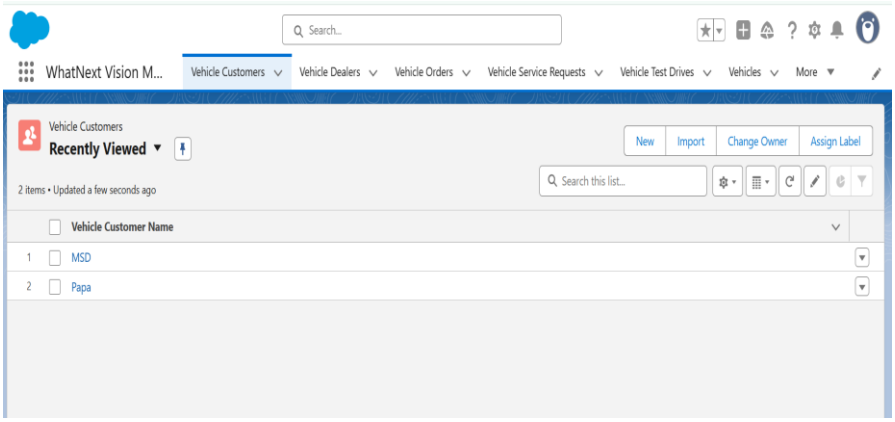
- The project involved building custom record pages and home layouts to ensure data was presented to users in a logical and intuitive manner.

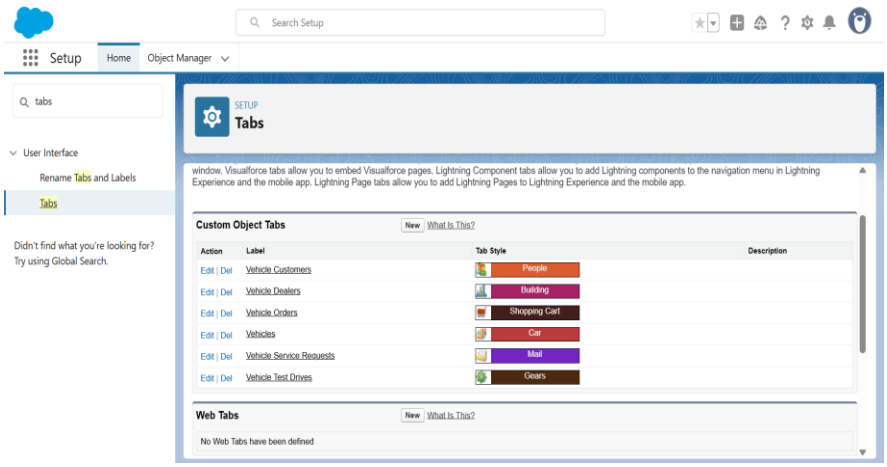
Utility Bar:

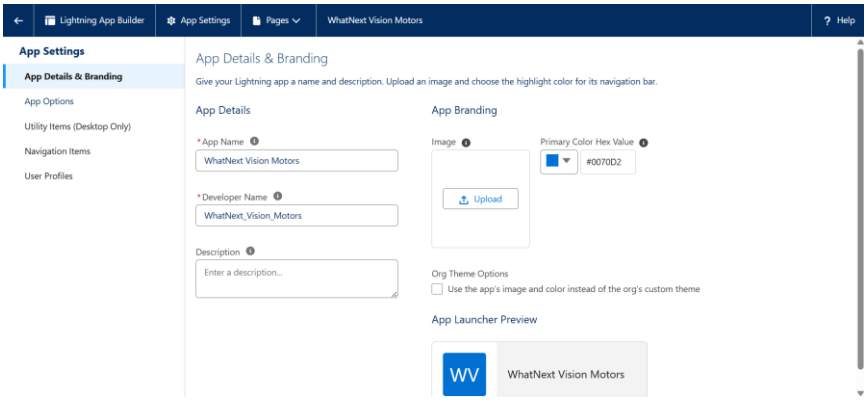
- A utility bar was configured for the custom Lightning App, providing users with quick access to common tools and productivity items from any page.

LWC (Lightning Web Components) & Apex with LWC:

- The project utilized LWC as part of its developer skill set to build the user interface.
- Apex was integrated with LWC to enable the display of dynamic and interactive data, connecting the custom front-end components with back-end business logic.







Phase 7: Integration & External Access

Web Services (REST/SOAP) & Callouts:

The project's functionality relies on communicating with external systems.

- An integration with an external Vehicle Inventory System was planned to check stock availability in real-time.
- This functionality would be implemented by making Apex Callouts from Salesforce to an external Web Service.

OAuth & Authentication
Secure access is a key requirement for the project.

- OAuth was explicitly identified as the authentication method for enabling secure customer login.

Implied Integration Requirements:
The features described above would also require the following concepts to be implemented:

- **Named Credentials or Remote Site Settings**: To make a secure and authenticated callout to the external inventory system, its endpoint URL must be registered in Salesforce using one of these security features.
- **API Limits**: Any integration making callouts from Salesforce is subject to API usage limits, which would need to be monitored and managed

Phase 8: Data Management & Deployment

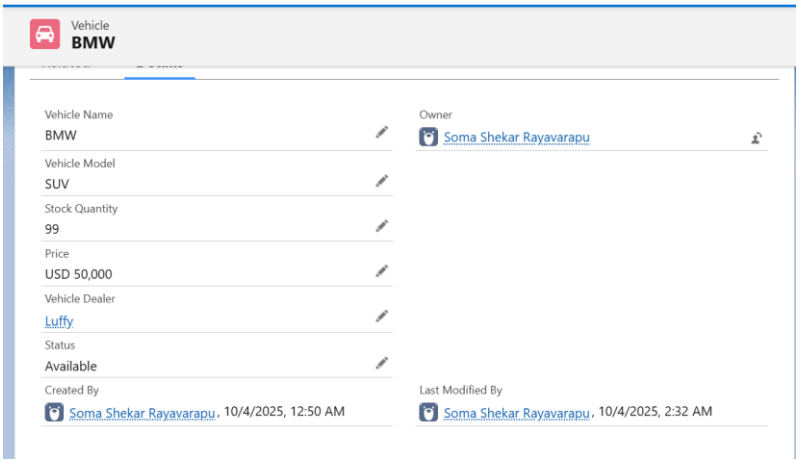
- The creation of custom tabs and a Lightning App directly relates to data management by providing the user interface for end-users to input, view, and interact with the data stored in the custom objects.
- Creating and activating Flows and Apex code.
- Storing Stock quantity, Price of the vehicle. Storing user details, order details, test drive details and service request details.

Data Loader:
The project plan specifies using this tool for data import tasks.

- Data Loader is used to import sample data prepared in CSV files.
- It's used to insert records for parent objects first, followed by junction objects that link them.

Change Sets:
This is the specified tool for migrating customizations between Salesforce environments.

- Change Sets are identified as the method for moving completed components from a Sandbox to the Production org.
- This deployment concept covers all development, including objects, fields, and flows, which are built in a Sandbox first.



Phase 9: Reporting, Dashboards & Security Review

Reports & Report Types:

The project plans involve building several key reports to provide business visibility. The GigHub project also specifies the need to create custom report types to link multiple objects together for more comprehensive analysis.

Examples of planned reports include:

- Sales Performance by Dealer.
- Vehicle Model Popularity.
- Pending Order Aging.
- Service Request Volume.

Dashboards:

Dashboards are used to visualize data from reports, providing at-a-glance insights for different user roles.

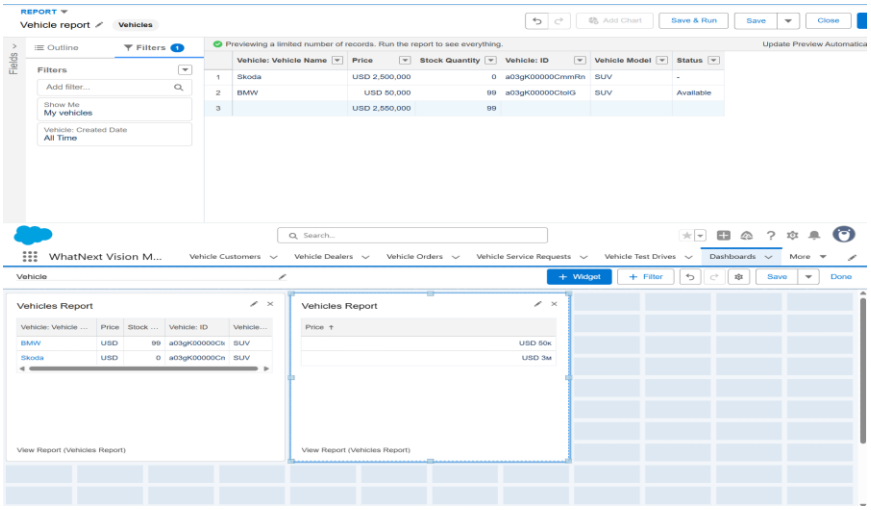
Specific dashboards mentioned in the plans are:

- Sales Manager Dashboard.
- Operations Dashboard.

- A "GigHub Management Dashboard".

Field Level Security:

The security review includes plans to implement Field-Level Security to control data visibility. This is used to hide sensitive fields, such as a freelancer's `Hourly_Rate__c`, from profiles that do not require access to that information.



Phase 10: Final Presentation & Demo Day

Exclusive Summary

- Built a complete Salesforce application to monitor and manage energy usage across devices and facilities.
- Combined **Admin skills** (data modeling, automation, security, dashboards) and **Developer skills** (Apex triggers, batch jobs, LWC)

Demo Flow

1. App Presentation.
2. Working process and details of every object on tabs.
3. Showing objects, fields and relationships in object manager.
4. In Setup, showed the tabs.
5. In Setup, showed the flow creation and uses of it.
6. The Apex Code explanation and live working proof.

Future Scope:

- **Customer Self-Service Portal:** Implement an **Experience Cloud (Community)** portal where customers can log in to track their order status in real-time, view their vehicle service history, schedule test drives or service appointments, and access a knowledge base for common questions.

- **Advanced Vehicle Configurator (CPQ):** Introduce **Salesforce CPQ (Configure, Price, Quote)** to allow customers to build and customize their vehicles online. They could select models, colors, and add-on packages, with the price updating dynamically for a fully personalized purchasing experience.
- **Marketing Automation:** Use **Marketing Cloud Account Engagement (Pardot)** or **Marketing Cloud** to create automated nurture journeys for potential buyers, send targeted promotional offers, and manage post-sale customer satisfaction surveys.