RePlastix Innovations

Transforming Plastic Waste into Sustainable Solutions

— A Salesforce-Based Capstone Project —

Submitted by:
Sony Mukkera
SmartBridge Salesforce Capstone Program

2025

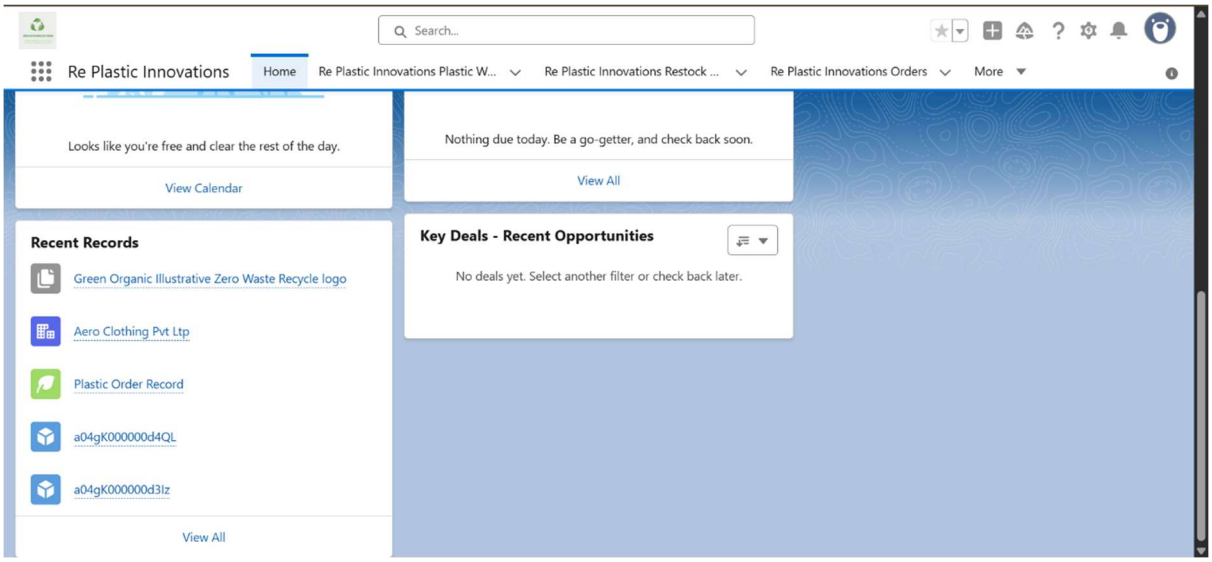
SMARTBRIDGE
Let's Bridge the Gap

salesforce
PARTNER

1.Project Overview:

RePlastix Innovations is a Salesforce-based CRM solution designed to streamline the tracking and processing of plastic waste into eco-friendly recycled products. The system addresses the growing need for sustainable waste management by digitizing and automating the collection, approval, and conversion processes. Key features include custom objects for managing plastic waste, recycling facilities, and final products; automated approval flows based on waste volume; and real-time dashboards to monitor operations. This CRM helps meet the business need for transparency, efficiency, and data-driven decision-making in the recycling industry, ultimately supporting environmental sustainability and smart infrastructure initiatives.

Lightning App: Re Plastic Innovations Home Page



2. Objective:

The primary objective of building the RePlastix CRM on the Salesforce platform is to streamline and automate key business processes such as plastic waste tracking, inventory management, and order fulfilment. By leveraging Salesforce's automation tools, the system aims to reduce manual errors, improve operational efficiency, and ensure timely replenishment of low-stock items. Additionally, the CRM provides role-based data access to enhance security and support cross-department collaboration. These objectives collectively contribute to improved resource planning, faster decision-making, and stronger alignment with the organization's sustainability goals.

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements:

RePlastix faced challenges in tracking stock levels and managing recyclable inventory manually, leading to inefficiencies and communication delays. The project was initiated to automate these processes and ensure timely stock replenishment and transparent task handling across departments.

Defining Project Scope and Objectives:

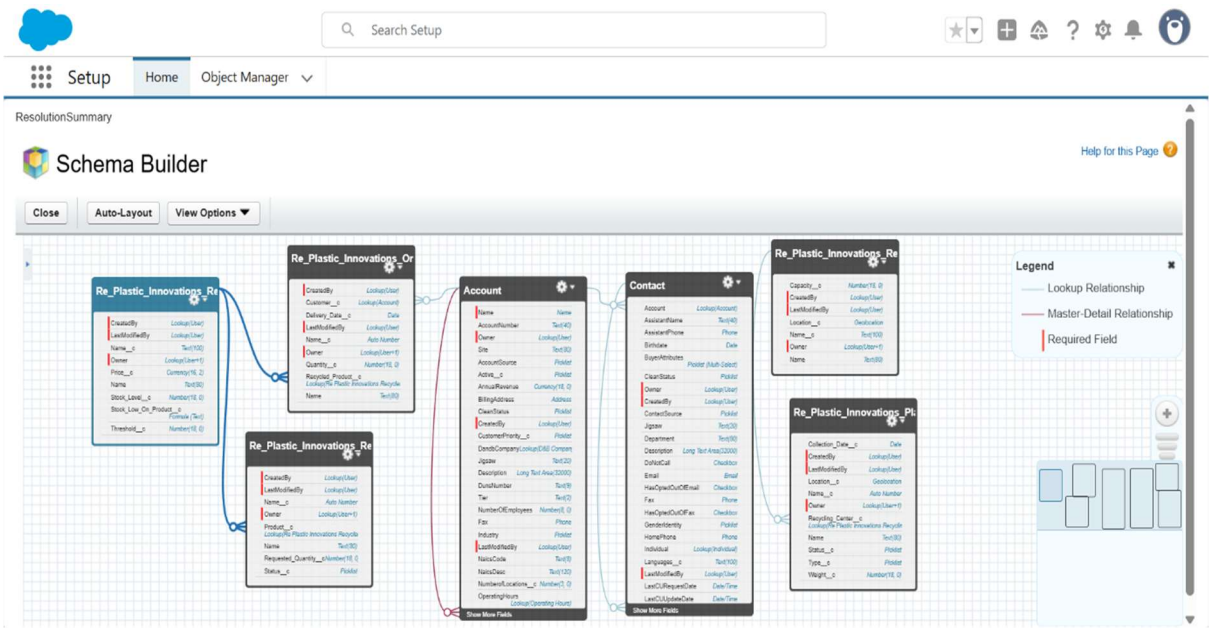
The CRM system was scoped to include:

- Custom Salesforce Objects for Products, Recycled Items, Orders, and Tasks
- Trigger-based alerts for stock monitoring
- Flows and approval processes for managing restock requests
- Secure access through role-based permissions
- Email notifications for inter-departmental updates

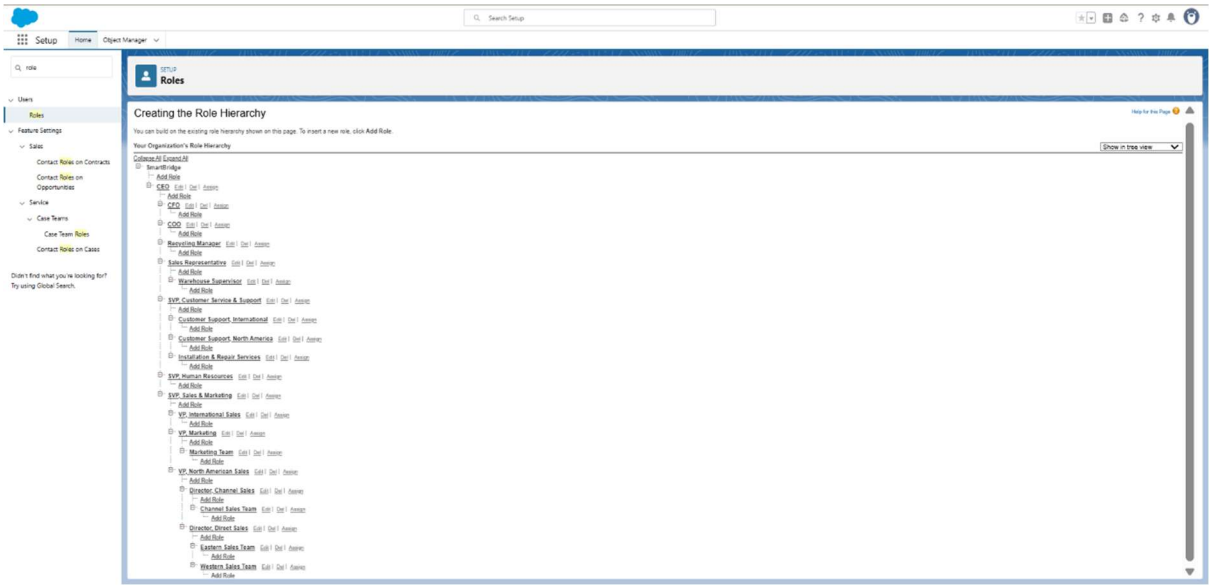
Design Data Model and Security Model:

- Custom objects created with fields for quantity, product type, status, and owner
- Validation Rules added for maintaining data integrity
- Profiles and Roles set up to restrict access according to user responsibilities
- Role hierarchy defined to match organizational workflow
- Permission Sets assigned for managing special feature access

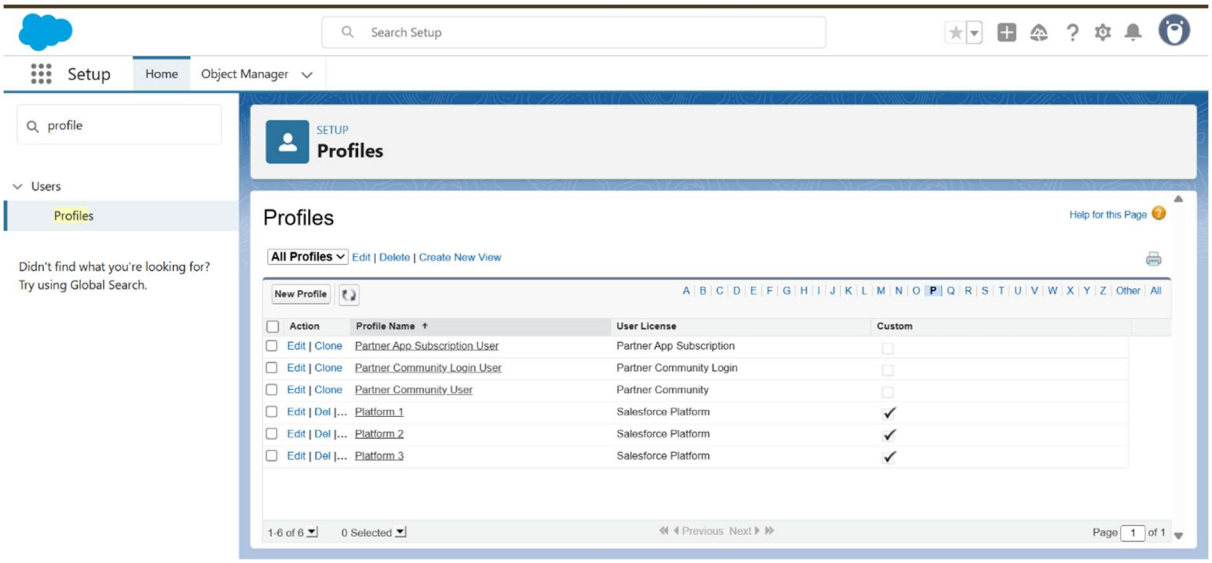
Object Overview:

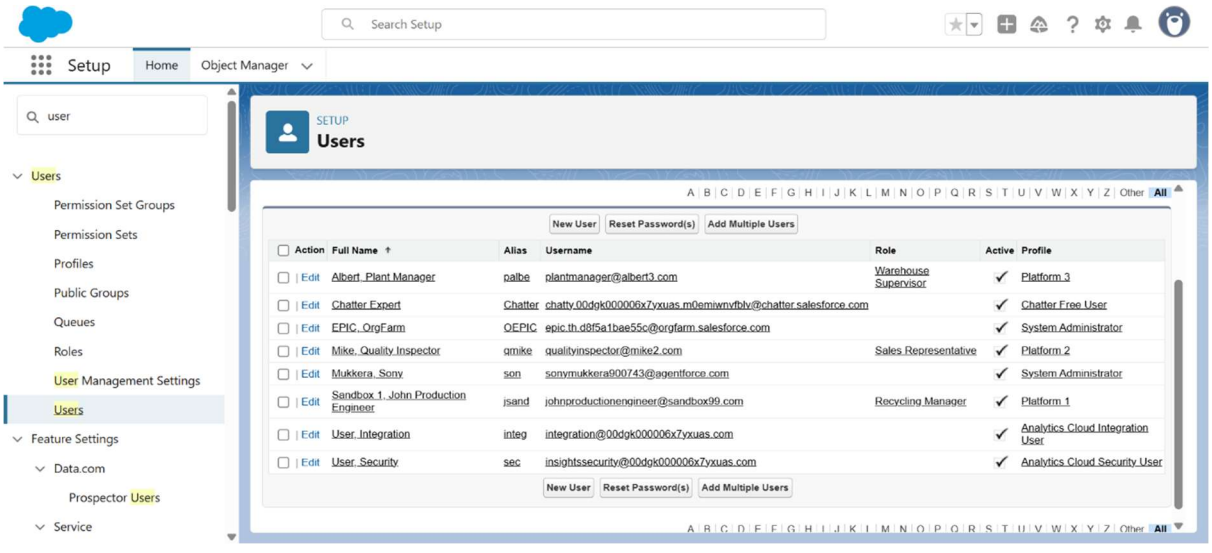


Role hierarchy settings:



Profile and User configuration:

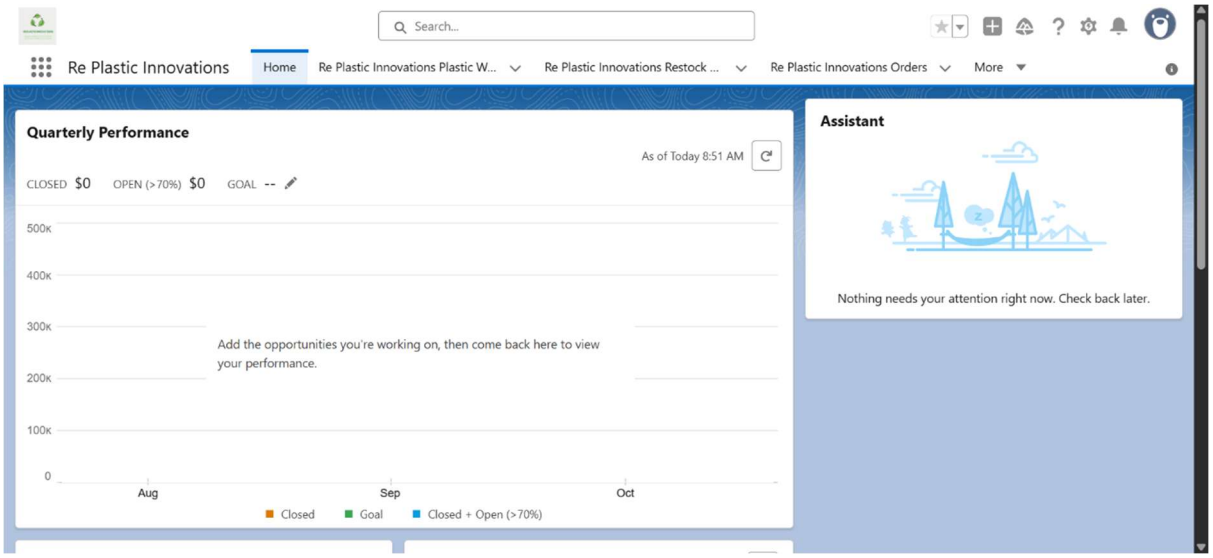




Phase 2: Salesforce Development - Backend & Configurations

Developer Edition Org Used:

Development and testing were carried out in a **Salesforce Developer Edition** org instead of a Sandbox. This setup allowed full access to Salesforce customization features needed for the RePlastix CRM project.



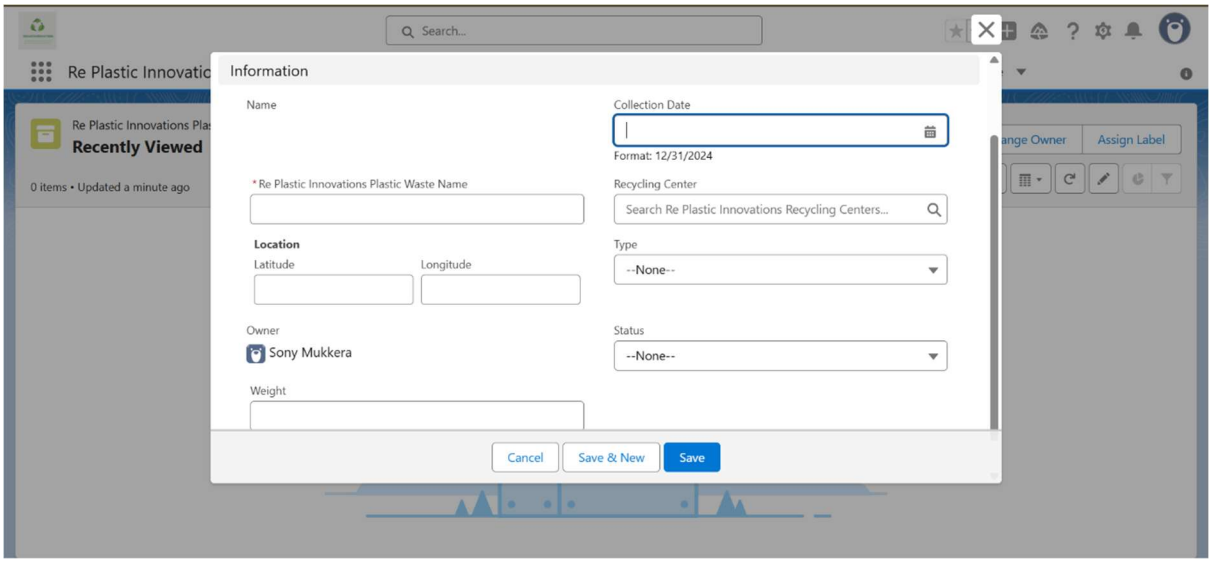
Object Customization:

Created custom objects and Fields:

1: Re Plastic Innovations Plastic Waste (Re_Plastic_Innovations_Plastic_Waste)

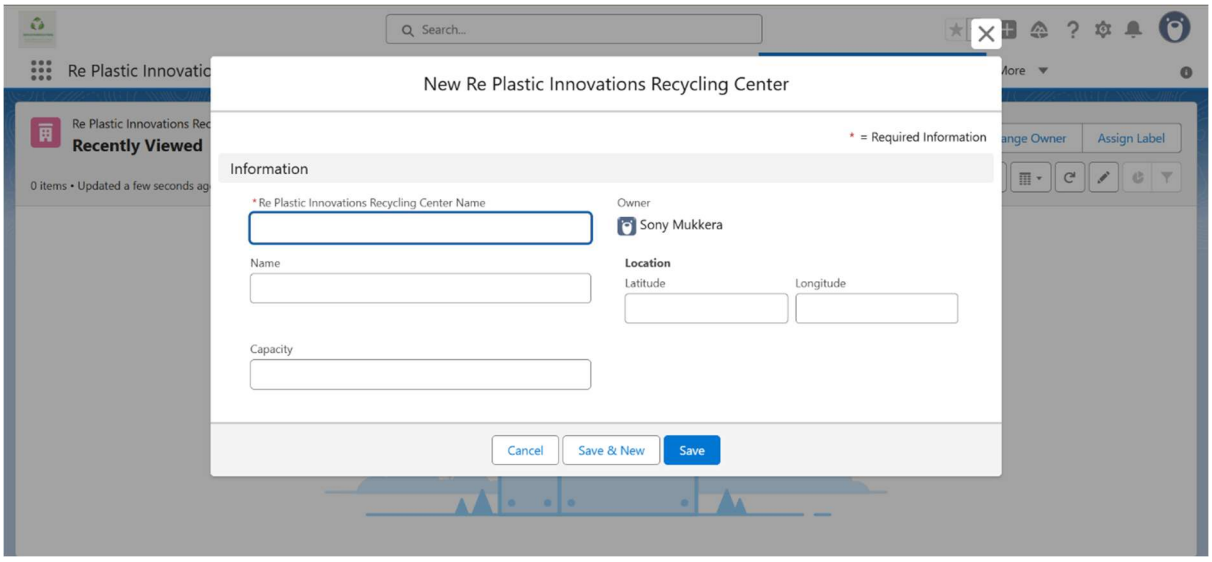
Field API Name	Data Type	Description
Name	Text (Auto Number)	Unique ID for waste records
Weight__c	Number (18,2)	Weight of plastic waste (kg)
Type__c	Picklist	Type of plastic (PET, HDPE, PVC, etc.)
Collection_Date__c	Date	Date waste was collected
Status__c	Picklist	["Collected", "Processing", "Recycled"]
Recycling_Center__c	Lookup (Recycling_Center__c)	Assigned recycling center

Location__c	Geolocation	Waste collection location



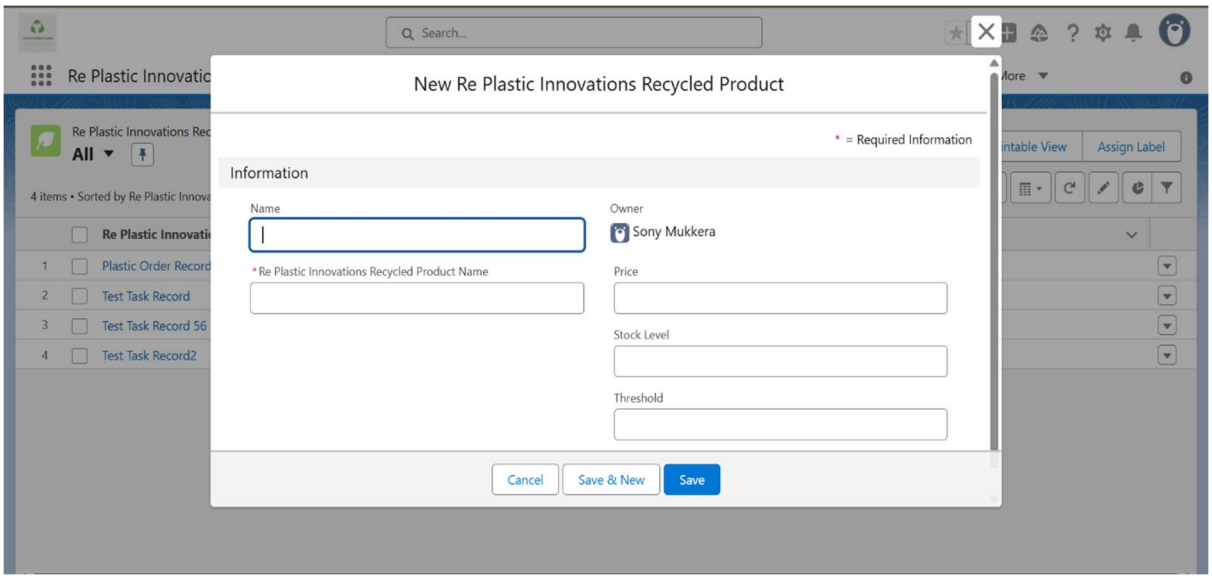
2: Re Plastic Innovations Recycling Center (Re_Plastic_Innovations_Recycling_Center__c)

Field API Name	Data Type	Description
Name	Text	Recycling Center Name
Location__c	Geolocation	Center's location
Capacity__c	Number (18,2)	Max capacity for processing waste



3: Re Plastic Innovations Recycled Product (Re_Plastic_Innovations_Recycled_Product__c)

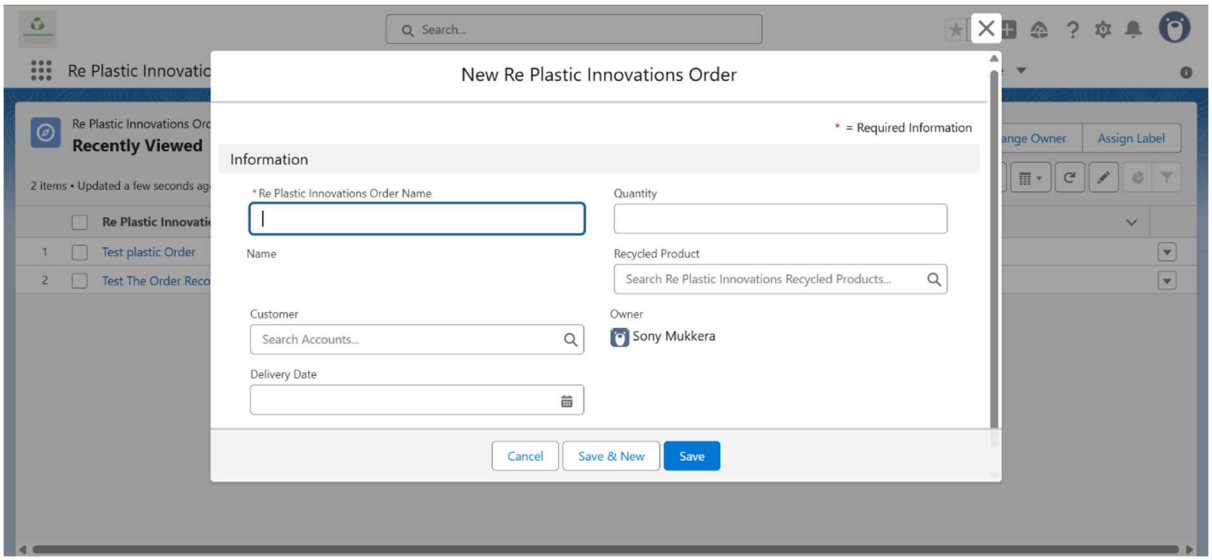
Field API Name	Data Type	Description
Name	Text	Recycled product name
Stock_Level__c	Number	Current stock available
Threshold__c	Number	Minimum stock before restock is triggered
Price__c	Currency	Price per unit



4: Re Plastic Innovations Order (Re_Plastic_Innovations_Order__c)

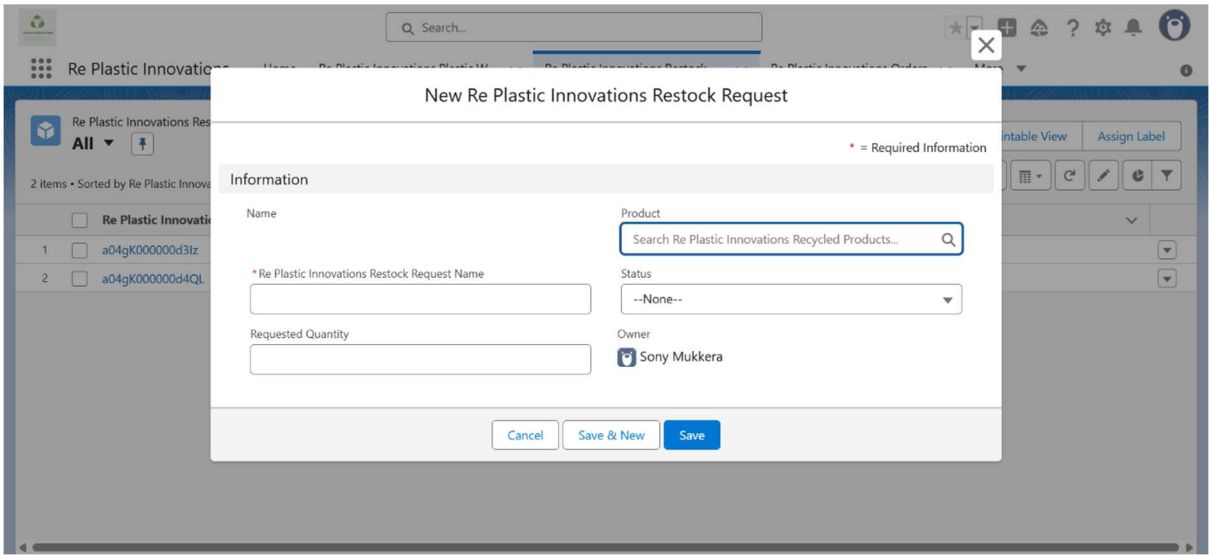
Field API Name	Data Type	Description
Name	Auto Number	Order ID
Customer__c	Lookup (Account)	Customer placing the order
Recycled_Product__c	Lookup (Recycled_Product__c)	Ordered product
Quantity__c	Number	Quantity ordered

Delivery_Date__c	Date	Expected delivery date



5: Re Plastic Innovations Restock Request (Re_Plastic_Innovations_Restock_Request__c)

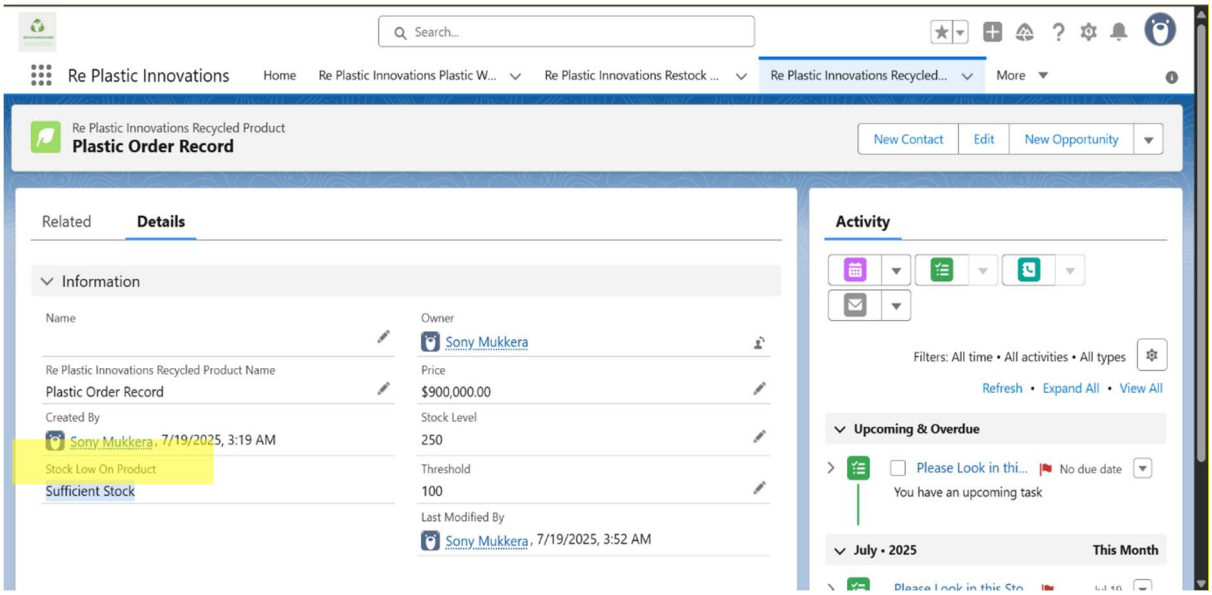
Field API Name	Data Type	Description
Name	Auto Number	Request ID
Product__c	Lookup (Recycled_Product__c)	Product to restock
Requested_Quantity__c	Number	Quantity requested
Status__c	Picklist	["Pending", "Approved", "Completed"]



Formula Field- Stock Low on Product:

Label Name: Stock Low On Product

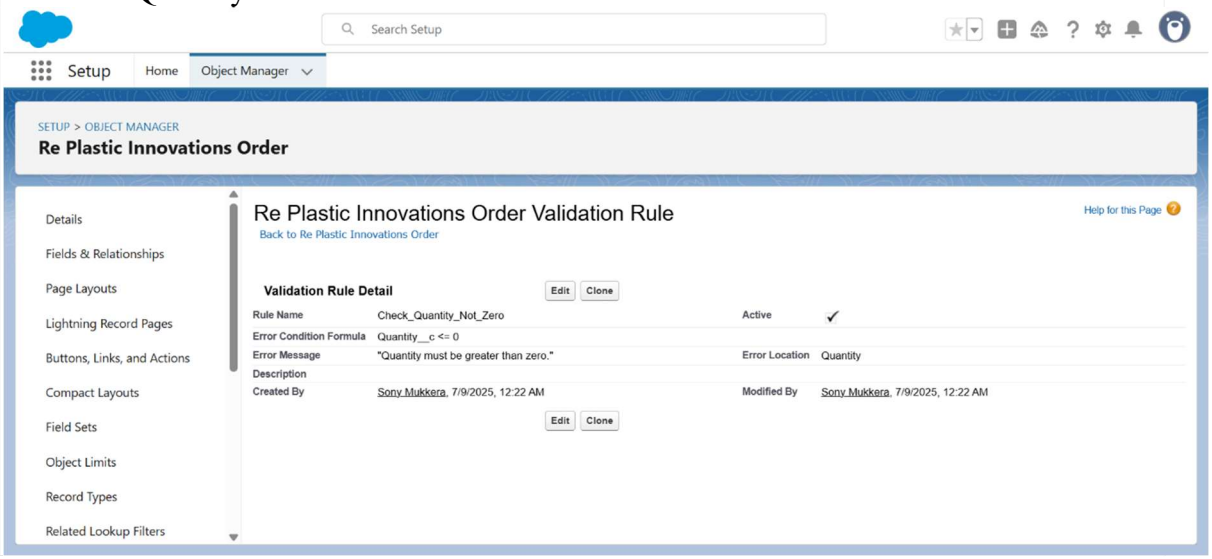
Formula: IF (Stock_Level__c < Threshold__c, "Low Stock - Restock Needed", "Sufficient Stock")



Validation Rules:

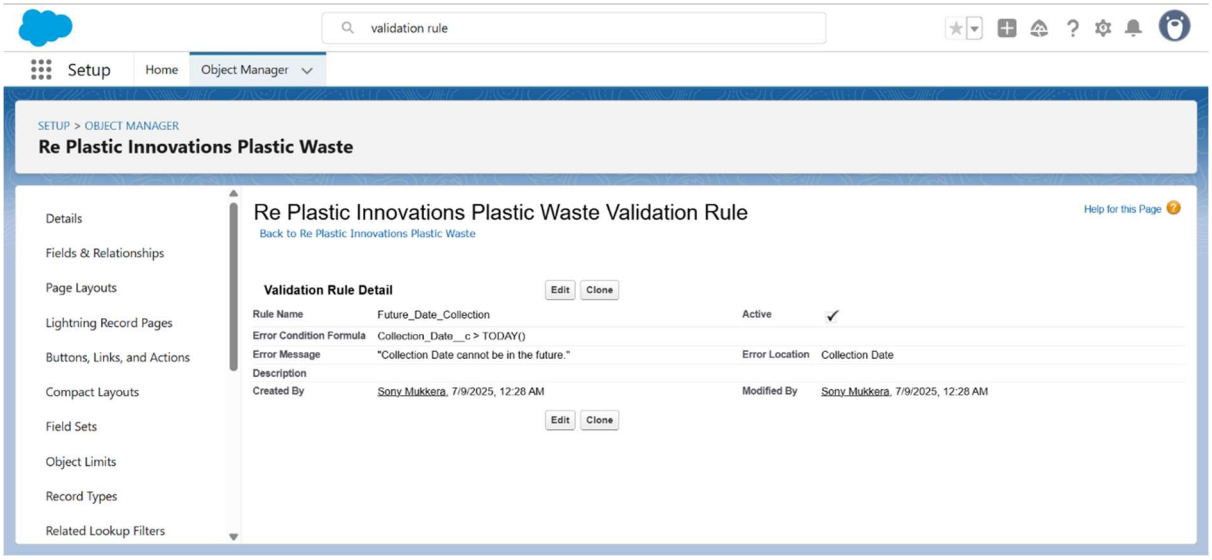
1: Validation rule on Re Plastic Innovations Order Object

- Rule Name: Check_Quantity_Not_Zero
- Active: True
- Formula: Quantity__c <=0
- Error Message: "Quantity must be greater than zero."
- Filed: Quantity



2: Validation on Re Plastic Innovations Plastic Waste Object

- Name: Future_Date_Collection
- Active: True
- Formula: Collection_Date__c > TODAY ()
- Error Message**:** "Collection Date cannot be in the future."
- Filed: Collection_Date__c

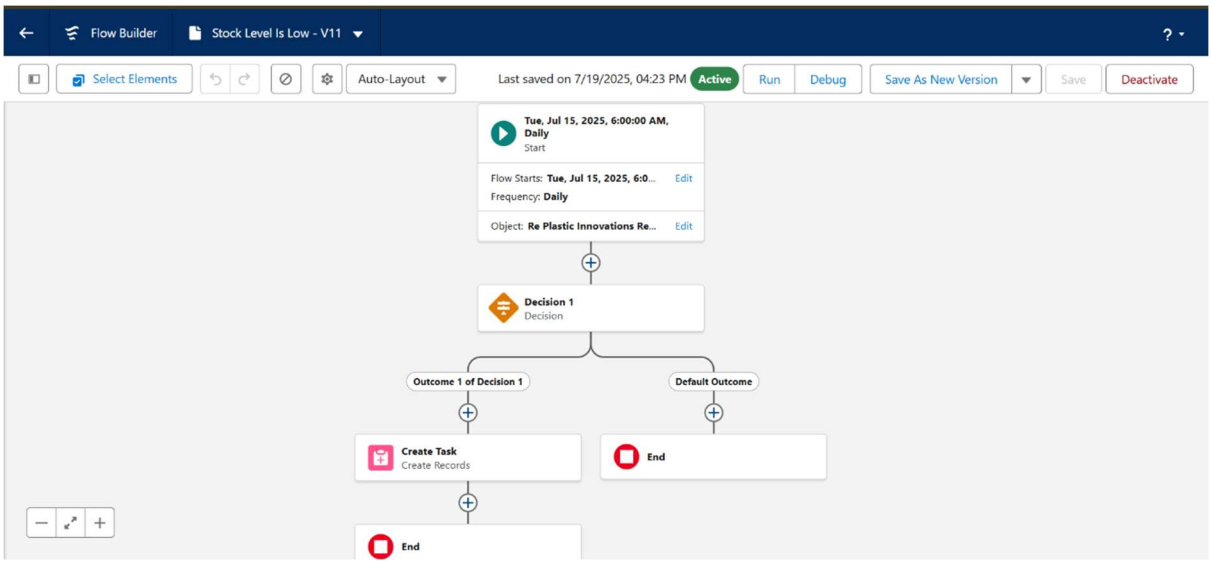


Automation - Flow Builder:

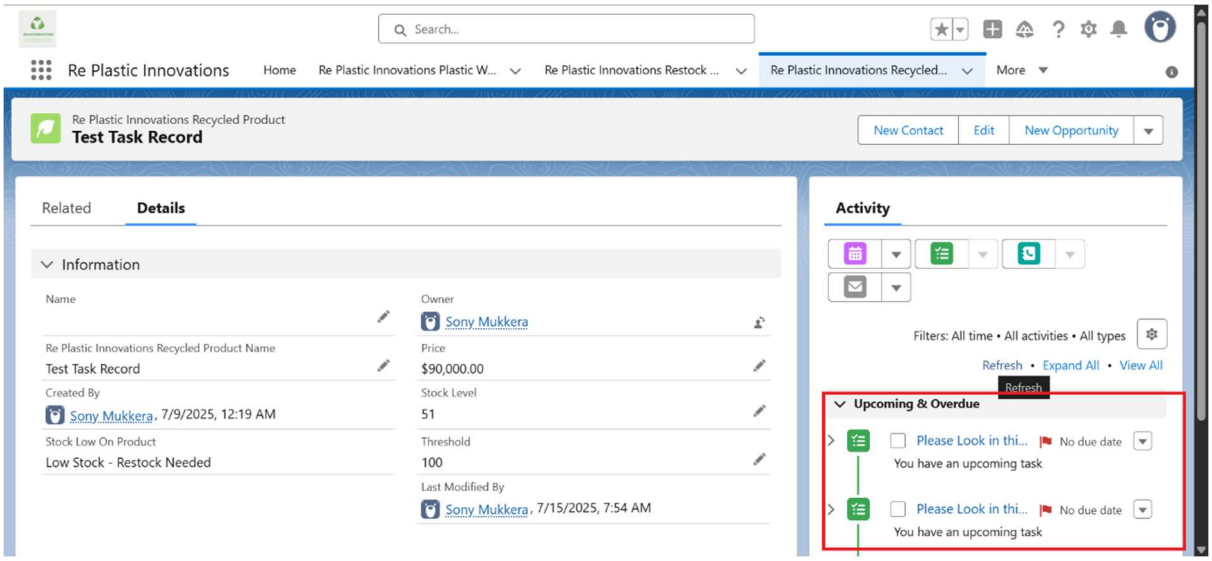
Scheduled-Triggered Flow:

A key component of RePlastix CRM's automation strategy was a Scheduled Flow designed to monitor inventory levels at regular intervals (e.g., daily 6AM).

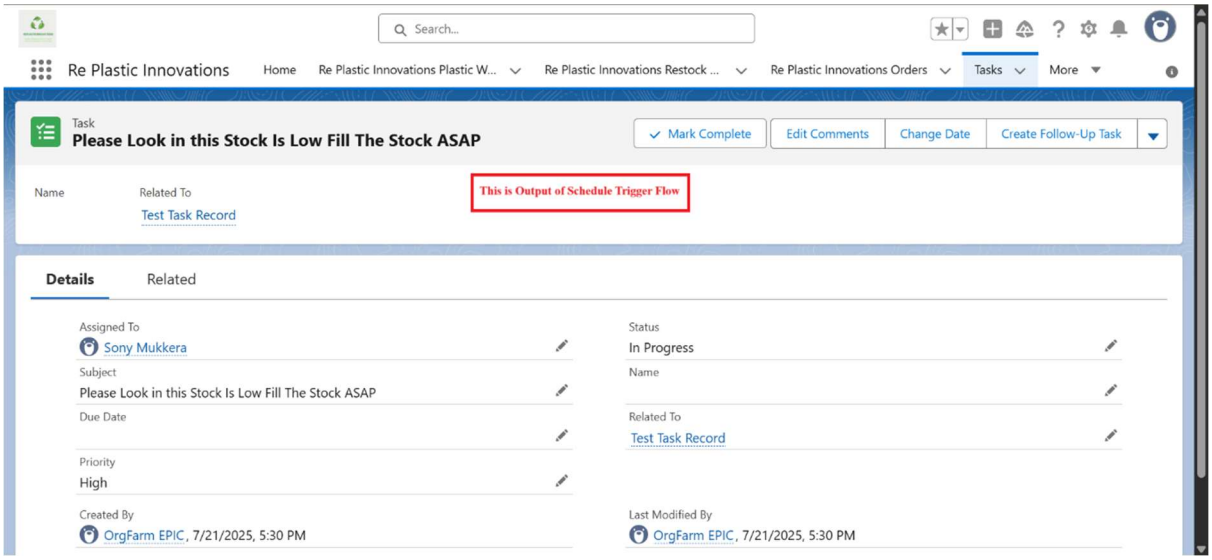
- This flow is scheduled to run every morning at 6:00 AM, ensuring proactive stock monitoring before warehouse dispatch begins.
- This flow runs without manual initiation and checks the stock quantity of all products.
- If any product falls below a predefined threshold, the flow automatically creates a Task.
- The Task is assigned to the record owner for restock action.



Test The Flow Set Time and Create Record and Check It.



Output of Schedule Trigger Flow:



Apex Development:

In RePlastix CRM, Apex was used to automate backend logic such as task creation when inventory drops below threshold and sending out notification emails. The code ensures fast execution, error handling, and alignment with business logic.

Apex Classes in Backend Automation:

Class Name: InventoryManager

The InventoryManager Apex class plays a critical role in automating stock adjustments and managing inventory flow within the RePlastix CRM system. It encapsulates two main methods responsible for maintaining accurate product quantities and ensuring restock operations are handled seamlessly.

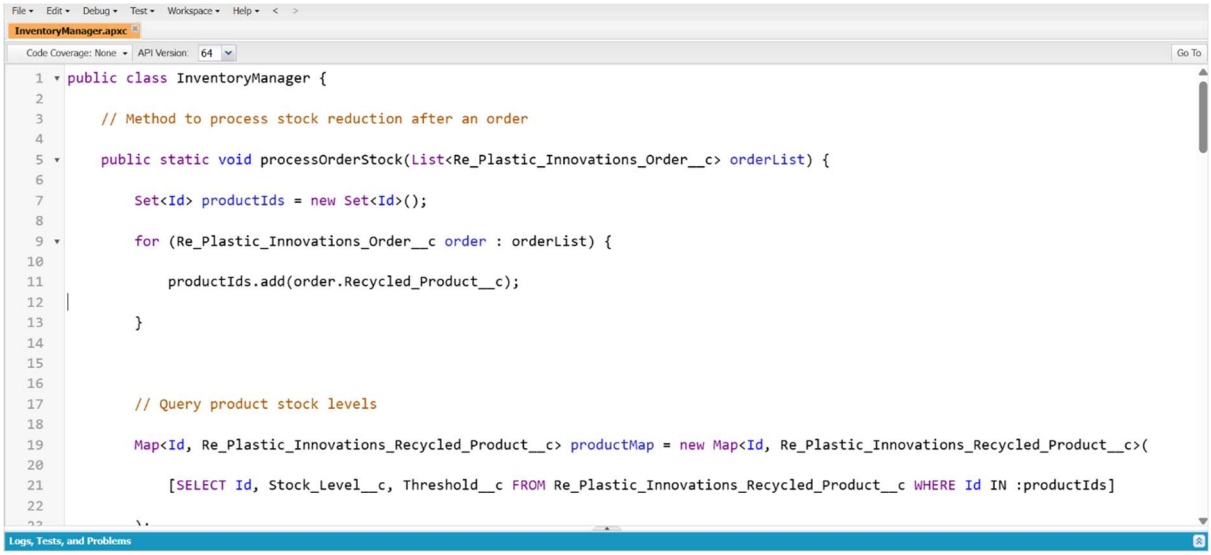
Purpose:

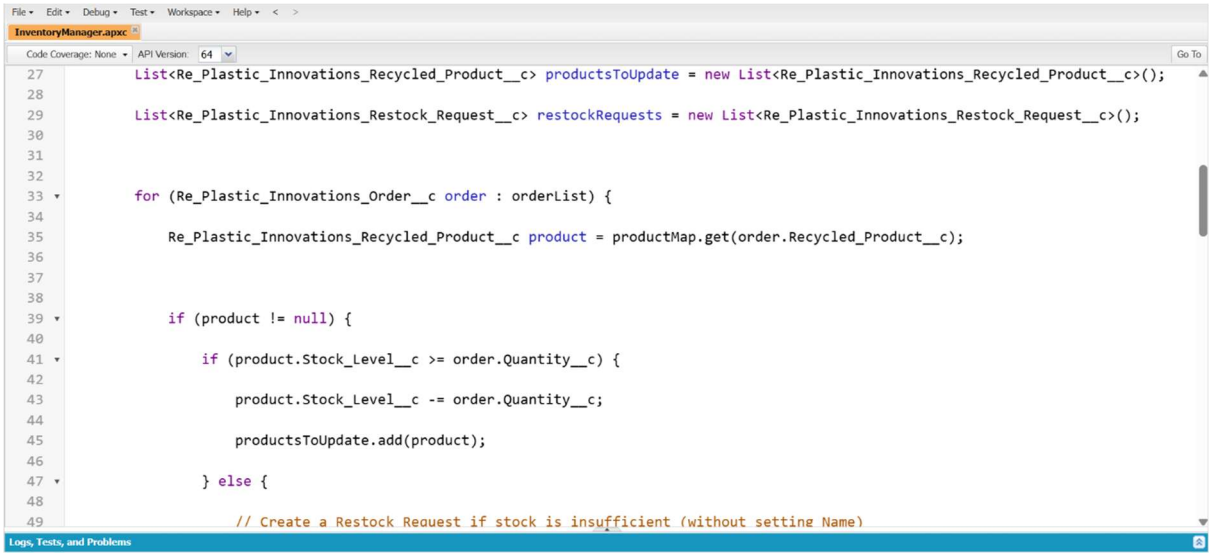
This Apex class automates inventory adjustments in RePlastix CRM. It includes two main methods:

- **processOrderStock**
  Reduces product stock after an order is placed. If stock is insufficient, it creates a Restock_Request__c record for replenishment.
- **processRestockApproval**
  Increases stock when restock requests are approved by updating the product's quantity accordingly.

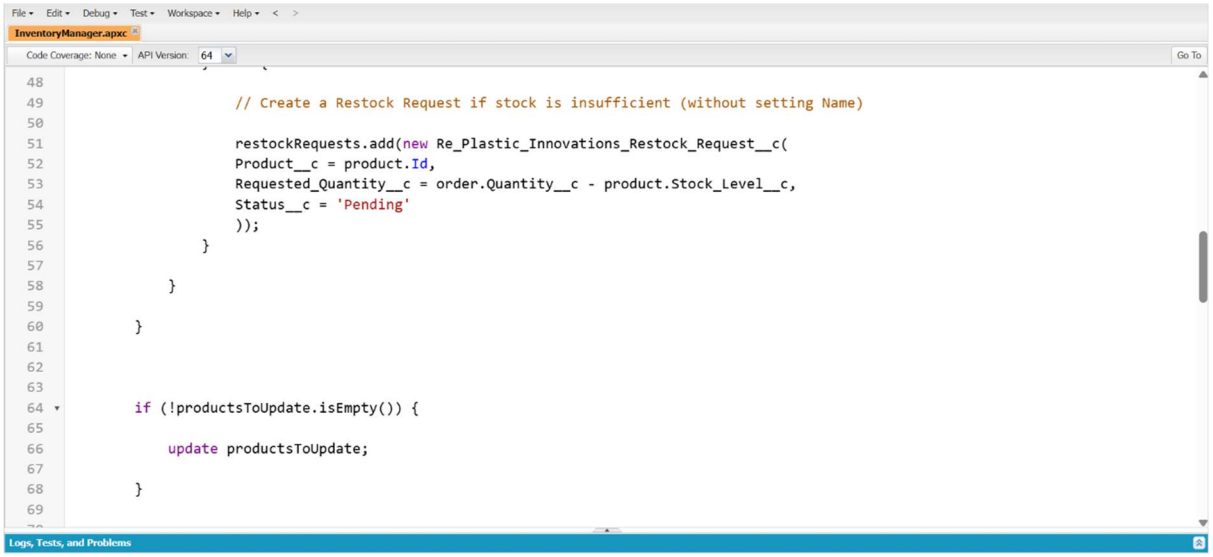
Together, these methods ensure real-time inventory updates and support smooth warehouse operations.

Apex Class - InventoryManager Class Code:



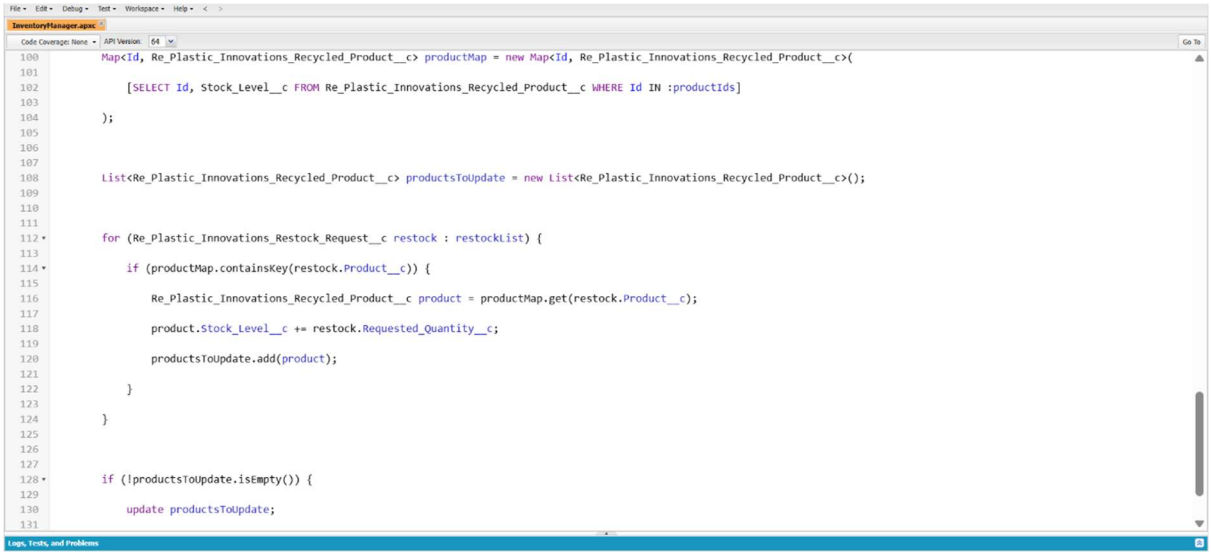


```
List<Re_Plastic_Innovations_Recycled_Product__c> productsToUpdate = new List<Re_Plastic_Innovations_Recycled_Product__c>();

List<Re_Plastic_Innovations_Restock_Request__c> restockRequests = new List<Re_Plastic_Innovations_Restock_Request__c>();



for (Re_Plastic_Innovations_Order__c order : orderList) {

    Re_Plastic_Innovations_Recycled_Product__c product = productMap.get(order.Recycled_Product__c);



    if (product != null) {

        if (product.Stock_Level__c >= order.Quantity__c) {

            product.Stock_Level__c -= order.Quantity__c;

            productsToUpdate.add(product);

        } else {

            // Create a Restock Request if stock is insufficient (without setting Name)
```



```
            // Create a Restock Request if stock is insufficient (without setting Name)

            restockRequests.add(new Re_Plastic_Innovations_Restock_Request__c(
            Product__c = product.Id,
            Requested_Quantity__c = order.Quantity__c - product.Stock_Level__c,
            Status__c = 'Pending'
            ));
        }

    }

}



if (!productsToUpdate.isEmpty()) {

    update productsToUpdate;

}
```

```
    if (!restockRequests.isEmpty()) {

        insert restockRequests;

    }
}



// Method to process stock increase after restock approval

public static void processRestockApproval(List<Re_Plastic_Innovations_Restock_Request__c> restockList) {

    Set<Id> productIds = new Set<Id>();

    for (Re_Plastic_Innovations_Restock_Request__c restock : restockList) {

        if (restock.Status__c == 'Approved') {

            productIds.add(restock.Product__c);

        }

    }
```



```
        Map<Id, Re_Plastic_Innovations_Recycled_Product__c> productMap = new Map<Id, Re_Plastic_Innovations_Recycled_Product__c>(
            [SELECT Id, Stock_Level__c FROM Re_Plastic_Innovations_Recycled_Product__c WHERE Id IN :productIds]
        );

        List<Re_Plastic_Innovations_Recycled_Product__c> productsToUpdate = new List<Re_Plastic_Innovations_Recycled_Product__c>();

        for (Re_Plastic_Innovations_Restock_Request__c restock : restockList) {
            if (productMap.containsKey(restock.Product__c)) {
                Re_Plastic_Innovations_Recycled_Product__c product = productMap.get(restock.Product__c);
                product.Stock_Level__c += restock.Requested_Quantity__c;
                productsToUpdate.add(product);
            }
        }

        if (!productsToUpdate.isEmpty()) {
            update productsToUpdate;
```

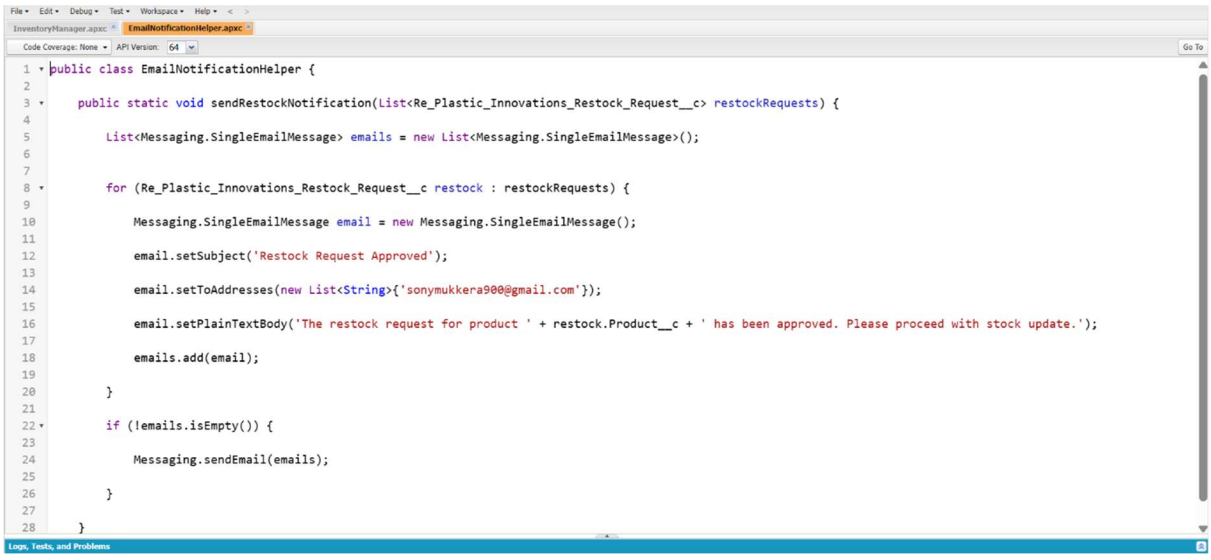
Apex Class Name: EmailNotificationHelper

Purpose:

This class automates email alerts when restock requests are approved. It loops through each request, creates a message with product info, and sends notifications to the warehouse team for immediate action.

Key Features:

- Loops through the approved restock requests
- Generates a clear, subject-labeled email for each product
- Sends all emails in bulk using Messaging.sendEmail()
- Keeps warehouse managers informed to take immediate action



```
public class EmailNotificationHelper {

    public static void sendRestockNotification(List<Re_Plastic_Innovations_Restock_Request__c> restockRequests) {

        List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();

        for (Re_Plastic_Innovations_Restock_Request__c restock : restockRequests) {

            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

            email.setSubject('Restock Request Approved');

            email.setToAddresses(new List<String>{'sonymukkera900@gmail.com'});

            email.setPlainTextBody('The restock request for product ' + restock.Product__c + ' has been approved. Please proceed with stock update.');

            emails.add(email);

        }

        if (!emails.isEmpty()) {

            Messaging.sendEmail(emails);

        }

    }
```

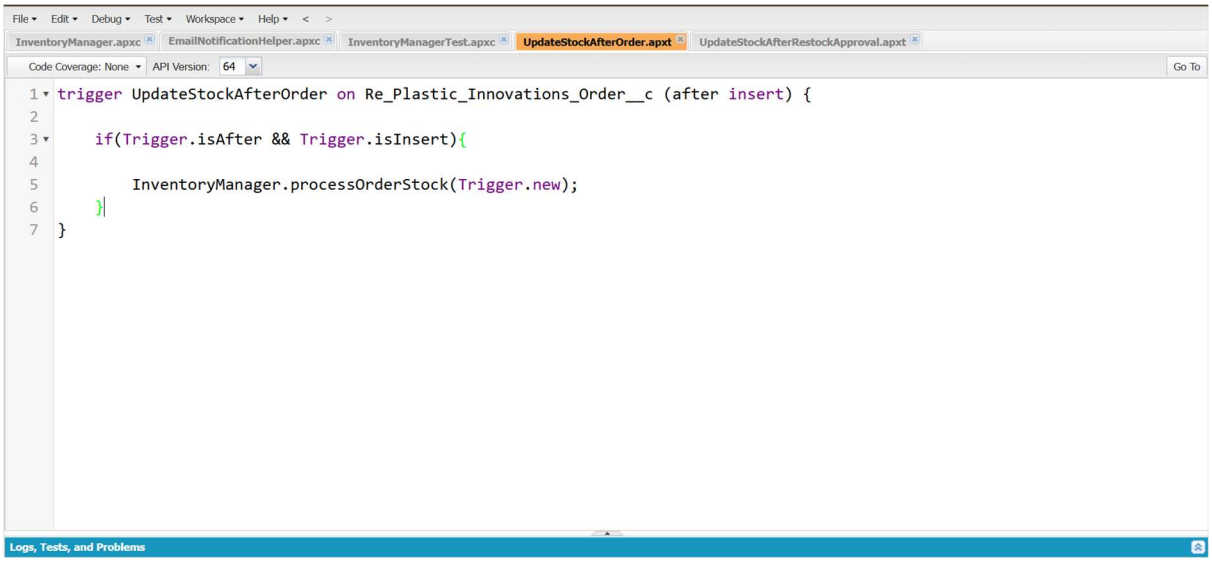
Triggers:

1. Trigger Name: UpdateStockAfterOrder

This trigger runs **after a new order is inserted** in the Re_Plastic_Innovations_Order__c object. Its primary role is to update inventory by invoking the processOrderStock() method from the InventoryManager class.

What It Does:

- Automatically deducts product stock based on the order quantity
- Creates restock requests if inventory is insufficient
- Ensures real-time stock updates for accurate inventory tracking



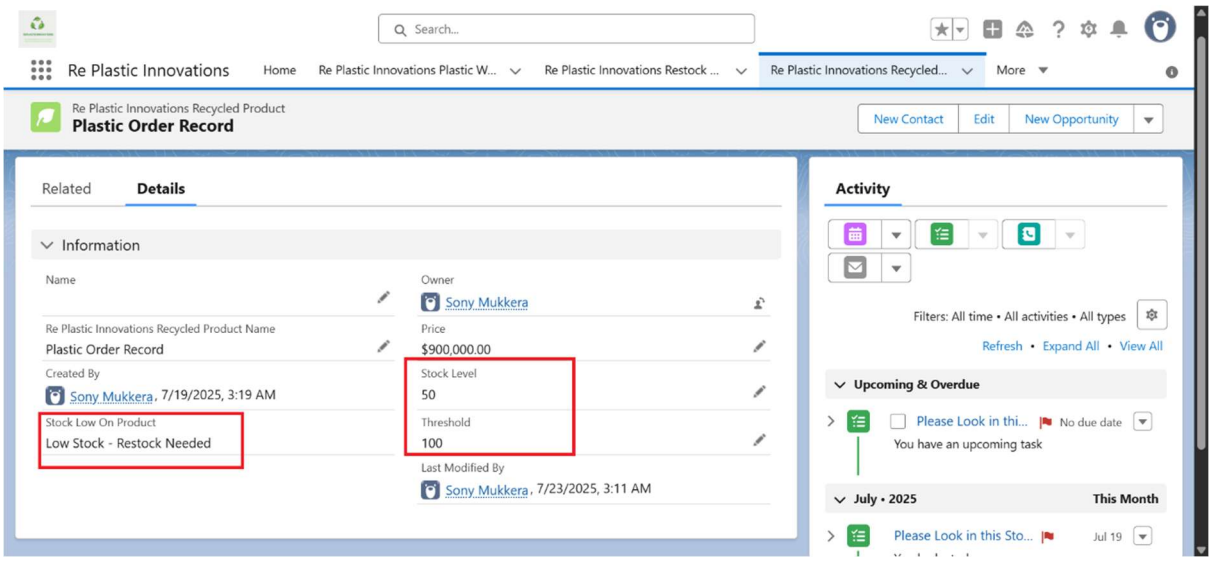
2. Trigger Name: **UpdateStockAfterRestockApproval**

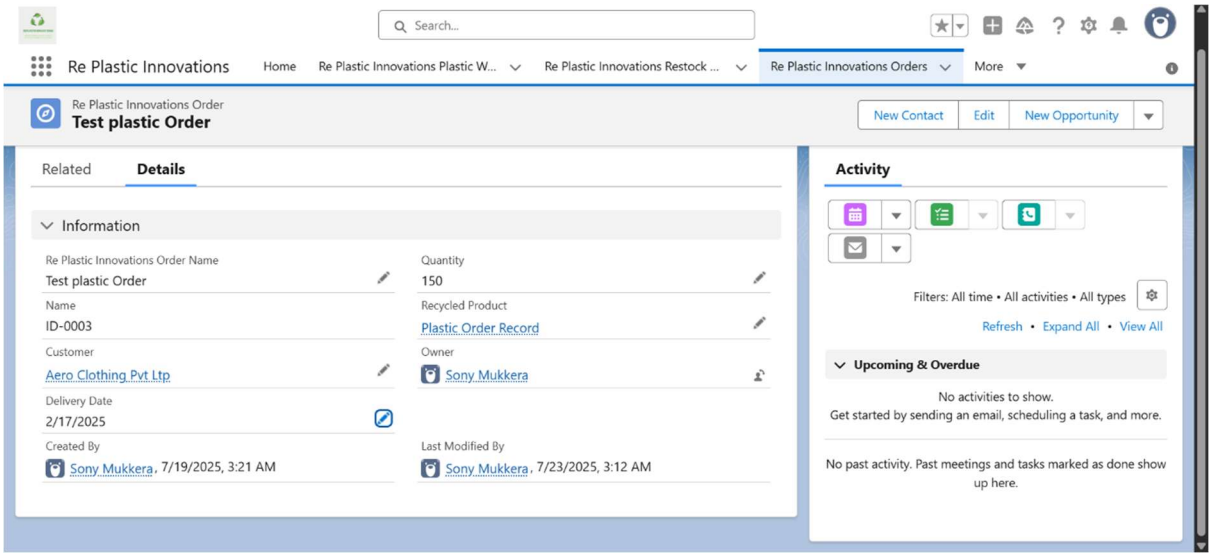
This trigger ensures that when a restock request is marked as **Approved**, the system immediately updates the product's stock and sends an automated email to notify the warehouse team. It keeps inventory accurate and the workflow responsive, all without manual intervention.

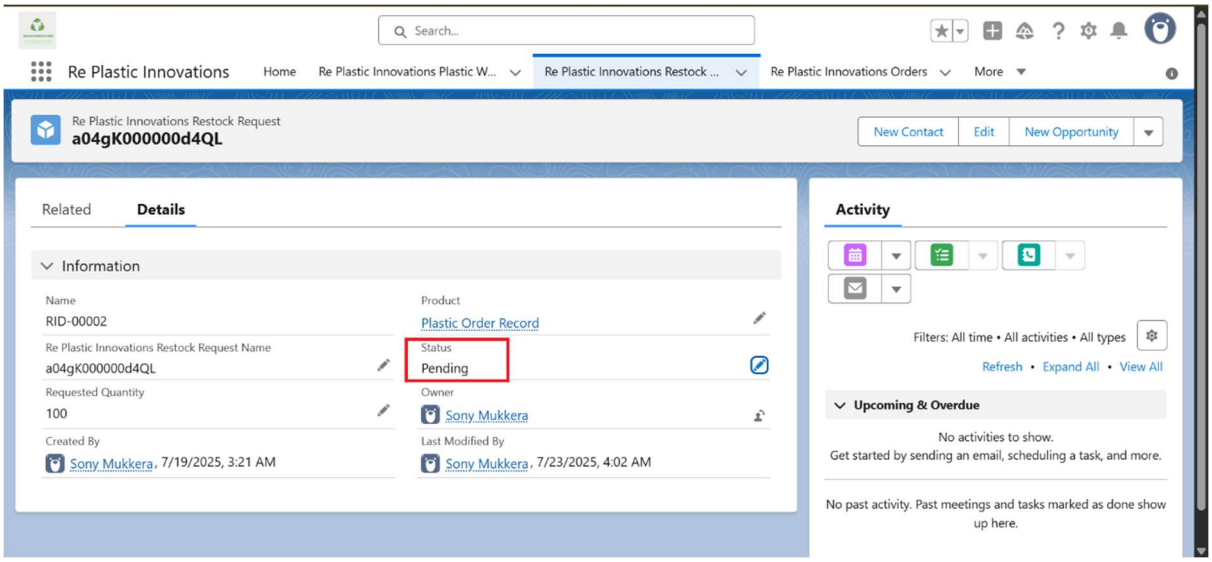
```
trigger UpdateStockAfterRestockApproval on Re_Plastic_Innovations_Restock_Request__c (after update) {

    List<Re_Plastic_Innovations_Restock_Request__c> approvedRestocks = new List<Re_Plastic_Innovations_Restock_Request__c>();

    for (Re_Plastic_Innovations_Restock_Request__c restock : Trigger.new) {

        if (restock.Status__c == 'Approved' &&

            Trigger.oldMap.get(restock.Id).Status__c != 'Approved') {

                approvedRestocks.add(restock);

            }
}

    if (!approvedRestocks.isEmpty()) {

        InventoryManager.processRestockApproval(approvedRestocks);

        EmailNotificationHelper.sendRestockNotification(approvedRestocks);

    }
```

Output:

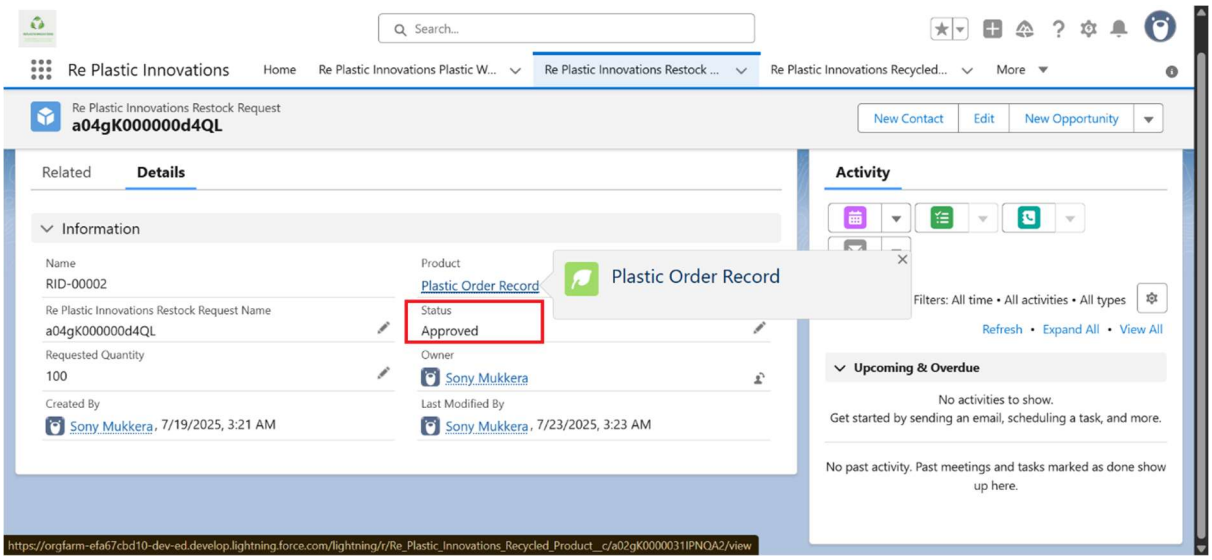
Before Trigger Fire:





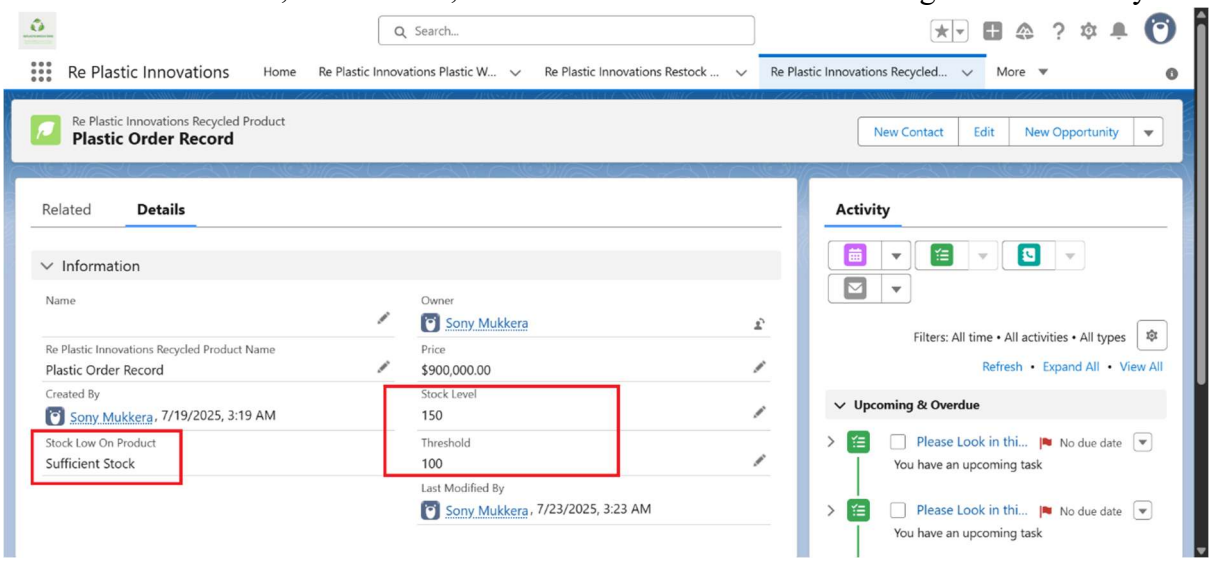


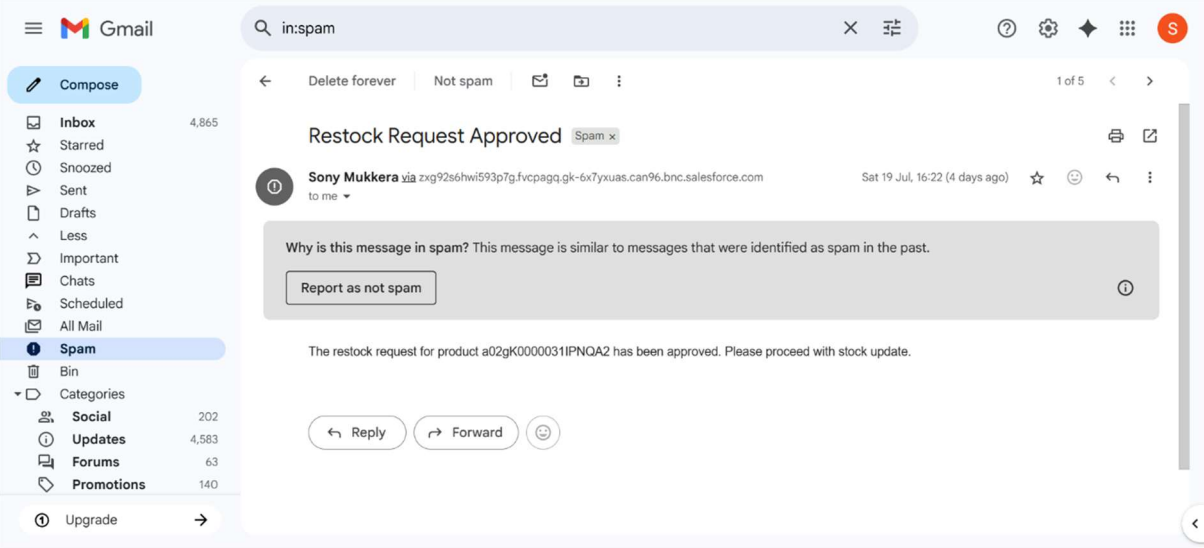
After Trigger Fired:



After changing Pending into Approval:

Stock low on Product, Stock Level, and Threshold those fields are changed automatically





Phase 3: UI/UX Development & Customization:

Lightning App Setup:

A custom **Lightning App** named **Re Plastic Innovation** *was* created using the App Manager.

The Re Plastic Innovation Lightning App was designed to provide users with a clean, role-based interface tailored to the operational needs of Re Plastic Innovation. It consolidates critical business objects, applies custom branding, and enhances usability across roles.

Configuration Overview:

- **Created via App Manager** with a project-branded logo, favicon, and custom colour palette to reflect RePlastic's sustainability theme
- **Tabs configured**: Re Plastic Innovations Plastic, Re Plastic Innovations Recycling Center, Re Plastic Innovations Recycled Product, Re Plastic Innovations Order, Re Plastic Innovations Restock Request**,** Task, Dashboard and etc.
- **Navigation Style:** Standard Lightning for consistent user experience on desktop and mobile devices
- **App Visibility:** Controlled by profile-level access — ensuring users see only relevant tabs based on their roles and permission

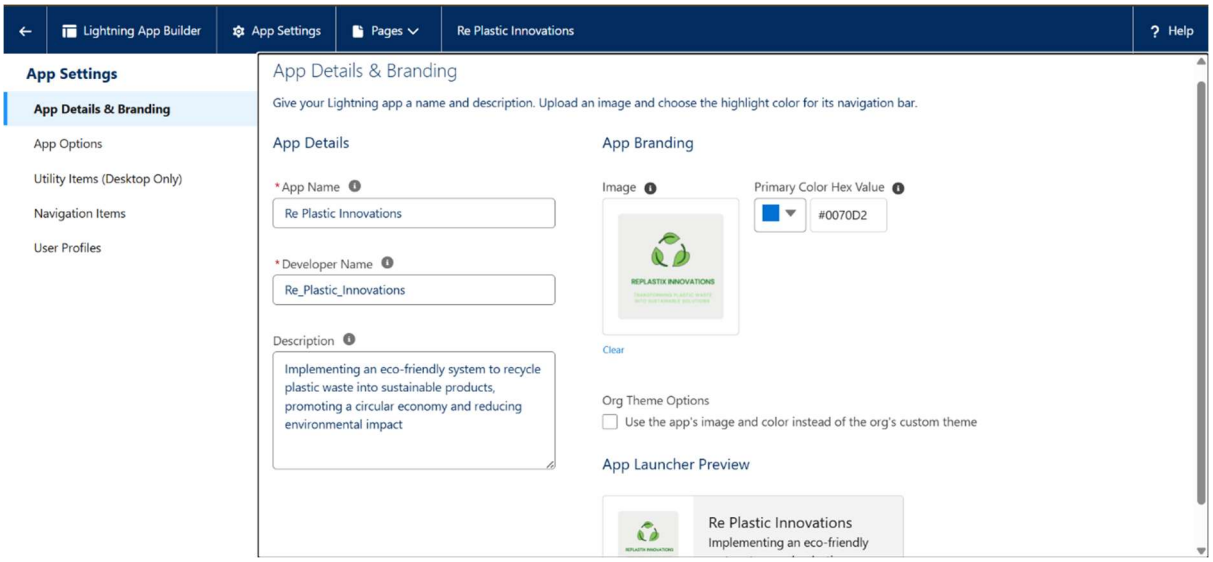
Role-Based Navigation

- Warehouse Supervisor: Focused access to inventory, restock requests, and tasks
- Recycling Manager: Access to recycled product tracking and reports
- Admin: Full configuration and visibility, include dashboards and approvals

Branding Elements

- Custom logo and theme colours reinforce organizational identity

- Home page layout optimized for quick navigation and data visibility



Page Layouts:

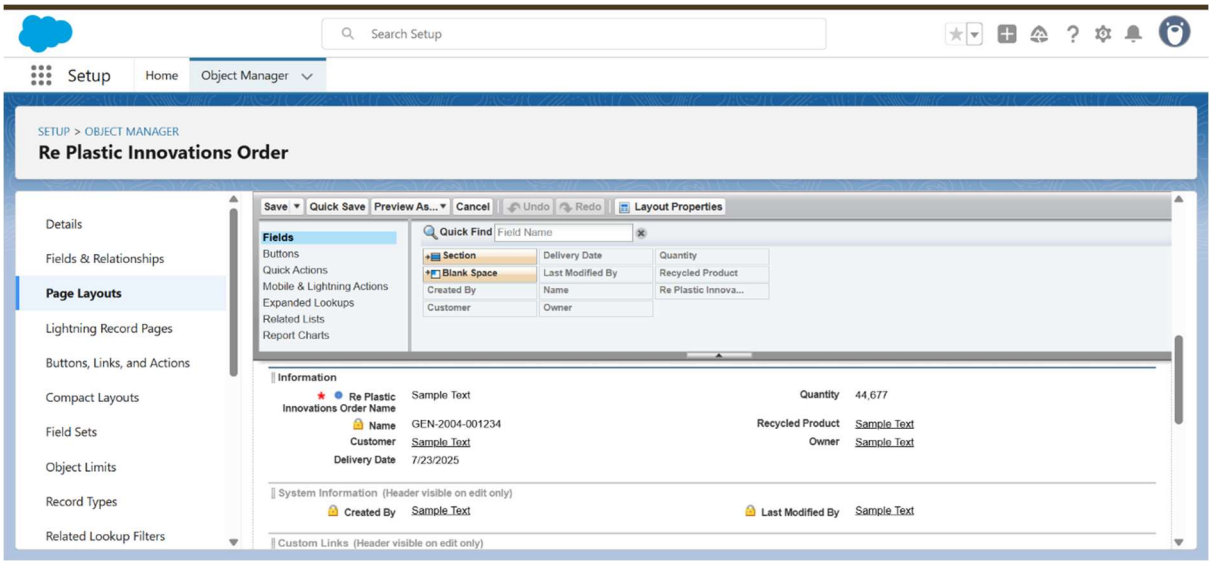
The Page Layouts and Dynamic Forms in *Re Plastic Innovation* were configured to deliver a personalized, role-sensitive data entry experience across key objects such as Recycled_Product__c, Order__c, and Restock_Request__c.

Page Layout Customization

- **Layouts tailored by role** — Admins, Warehouse Supervisor, and Recycling Managers all see only the fields relevant to their function
- **Section grouping** — Fields were categorized into logical groups (Product Info, Inventory Stats, Request Details) for better clarity
- **Related Lists** embedded to display Tasks, Orders, and Restock history inline on record views

UX Impact

- Simplified user interaction with tailored field visibility
- Reduced error rates and improved speed of data entry
- Helped maintain data integrity by limiting edits on sensitive fields



User Experience by Role:

The Re Plastic Innovations interface was designed with precision to align user access with departmental responsibilities. Role-specific configurations enhance operational clarity, data security, and workflow efficiency.

The role hierarchy in Re Plastic Innovation *CRM* reflects the organizational reporting chain and enforces data visibility based on supervisory relationships.

Hierarchy Breakdown:

- CEO (Top-Level Role)
  The CEO oversees all CRM activity, with full visibility and administrative control.

- Sales Representative (Report to CEO)
  This role bridges sales operations with executive oversight. Sales Representatives inherit visibility from the CEO and manage order-related data and reporting.

- Warehouse Supervisor (Report to Sales Representative)
  Warehouse Supervisors handle inventory execution and logistics. They inherit visibility from the Sales Representative, allowing direct collaboration and operational alignment.

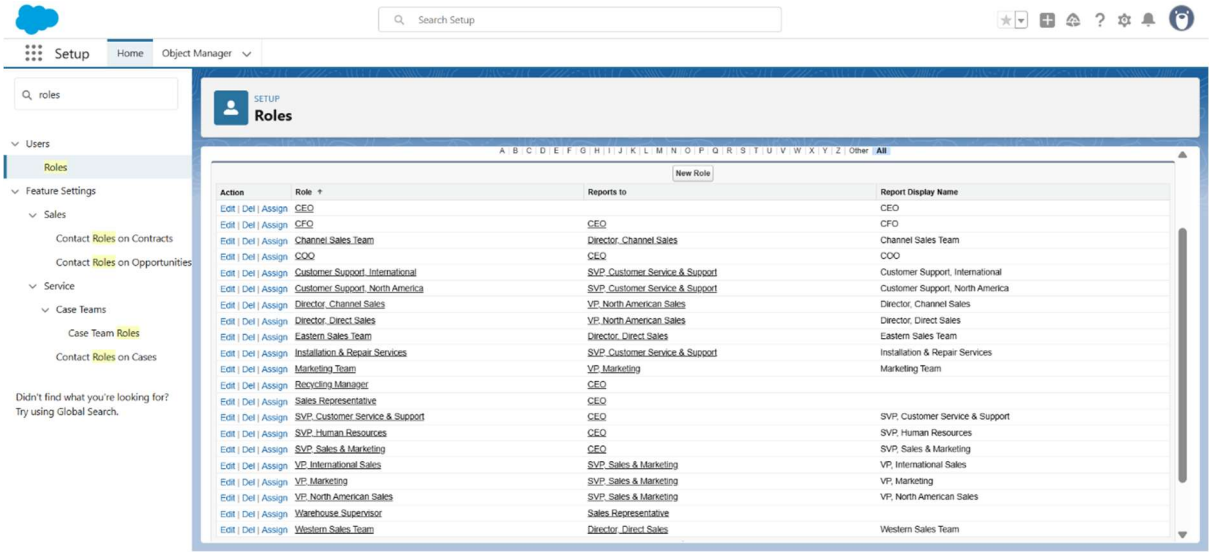
Administrator:

- **Access Scope:** Full-system control and configuration
- **Available Tabs:** All CRM objects and setup components
- **Functional Capabilities:**
- Manage roles, profiles, and sharing rules
- Configure automation, triggers, and workflows

- Unrestricted data visibility and backend control

CRM Implementation

- **Data Access:**
  Lower roles inherit record visibility upward — for example, the CEO can view all data created by the Sales Rep and Warehouse Supervisor.
- **Workflow Transparency:**
  Approval requests and task assignments follow the reporting line, ensuring accountability.
- **Security Enforcement:**
  Role-based visibility controls help prevent unauthorized access to sensitive records, maintaining data integrity and compliance



Phase 4: Data Migration, Testing & Security:

In this phase, essential records — including Products, Orders, Restock Requests, and User profiles — were manually uploaded and validated directly within the production environment. Care was taken to maintain data integrity, enforce field mappings, and verify related object relationships. Functional testing was executed across critical components like Apex triggers, email notifications, and record approval processes. The system's response to bulk operations and edge-case scenarios was monitored to ensure alignment with governor limits and performance expectations.

Security configurations were rigorously implemented, incorporating tailored Permission Sets, Object-level access controls, and Field-Level Security for sensitive metrics. Role hierarchy was structured to support visibility inheritance and accountability across operational units. Access was provisioned to align with defined business roles, ensuring that users interacted only with relevant data and functionality.

Data Migration:

During this phase, critical CRM records — including Products, Orders, Restock Requests, and User details — were manually imported into the production org without the

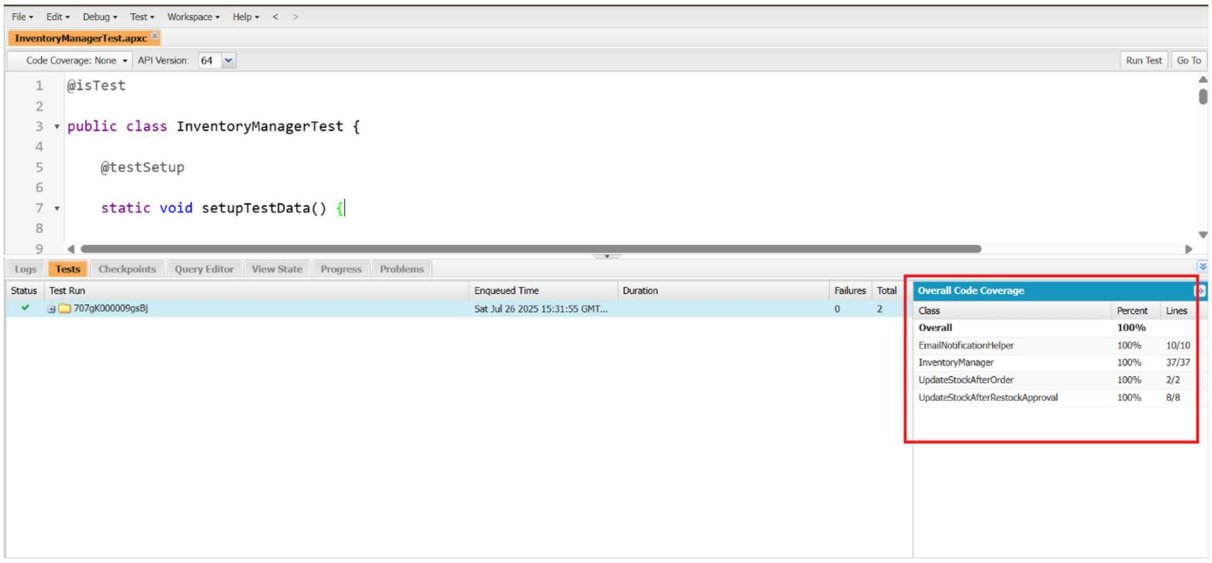
use of a sandbox. Field mappings and validation checks ensured each entry was accurate and consistent with defined schema rules. Lookup relationships and record ownership were reviewed to preserve data integrity and relational accuracy across objects.

To maintain control, metadata and Apex components were versioned via GitHub, enabling change tracking and rollback capability. This direct migration approach prioritized hands-on verification and speed, with meticulous attention given to preventing record duplication and maintaining referential consistency.

Testing:

Functional and behavioural tests were conducted across all custom logic and automated flows to ensure reliability and system consistency. Apex triggers, email notifications, and record approval processes were validated through manual and scenario-based testing within the production environment. Bulk data operations were reviewed to confirm alignment with Salesforce governor limits. Edge-case handling and error prevention mechanisms were assessed for stability under real-use conditions.

Test results confirmed accurate record updates, successful automation execution, and appropriate workflow responses. System logs and record histories were examined to ensure that exceptions were handled gracefully and processes triggered as designed.



Security Configuration:

A layered security strategy was deployed to ensure that sensitive business data remains secure, accessible only to authorized users, and compliant with best practices.

Profiles and Permission Sets:

Each user role was assigned a base profile reflecting job responsibilities (e.g., Inventory Manager, Approver). Permission sets extended functionality—such as access to dashboards, API features, and reporting tools—without altering core role definition

Role Hierarchy:

Built to maintain visibility across levels; senior roles can view subordinate records without granting editing rights, preserving upward transparency and downward confidentiality.

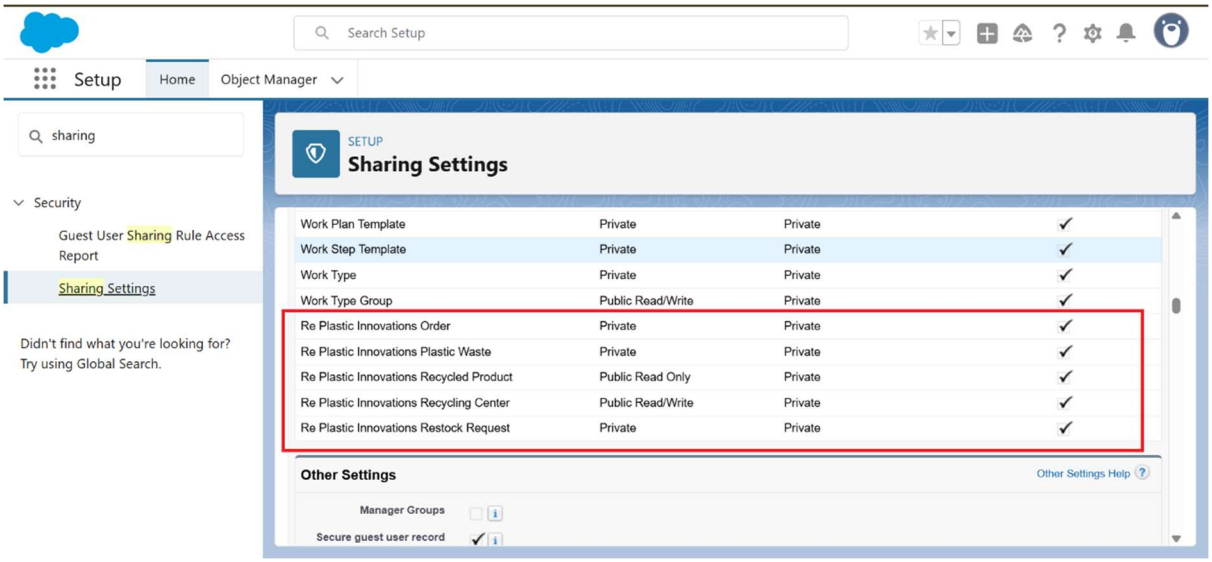
Access Controls:

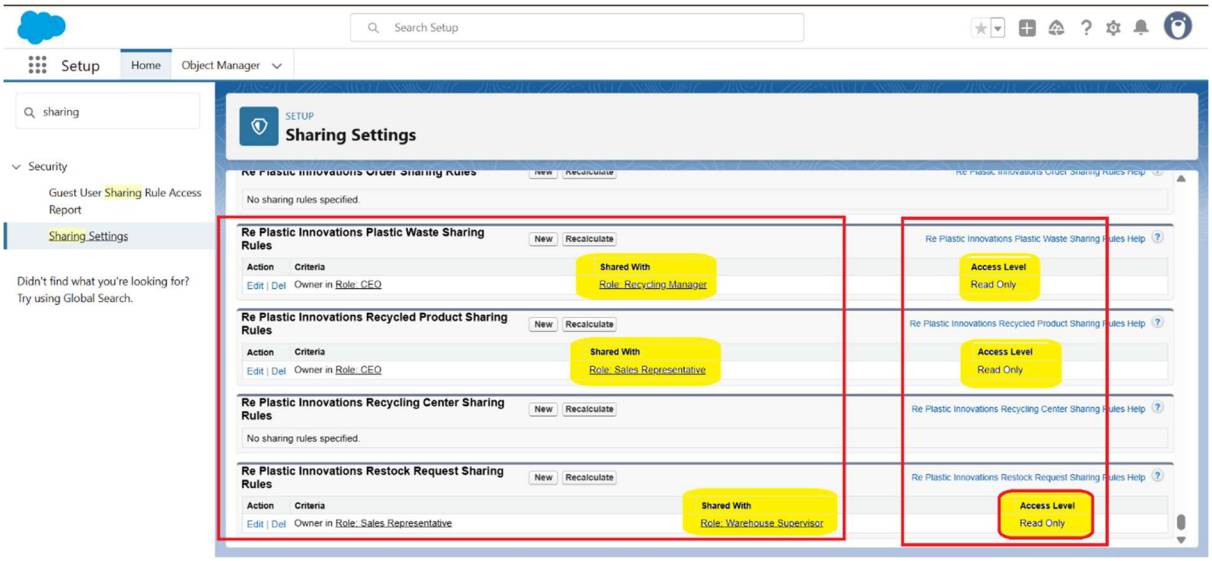
Sharing Rules:

Sharing Rules are automated configurations that extend record access beyond the baseline set by **Organization-Wide Defaults (OWD)**. OWD defaults were configured as “Private” for sensitive records. Specific sharing rules allowed targeted visibility based on ownership, record type, or criteria.

Purpose

- Facilitate secure collaboration across roles and teams
- Maintain data confidentiality while enabling operational efficiency
- Automate exceptions to restrictive OWD settings



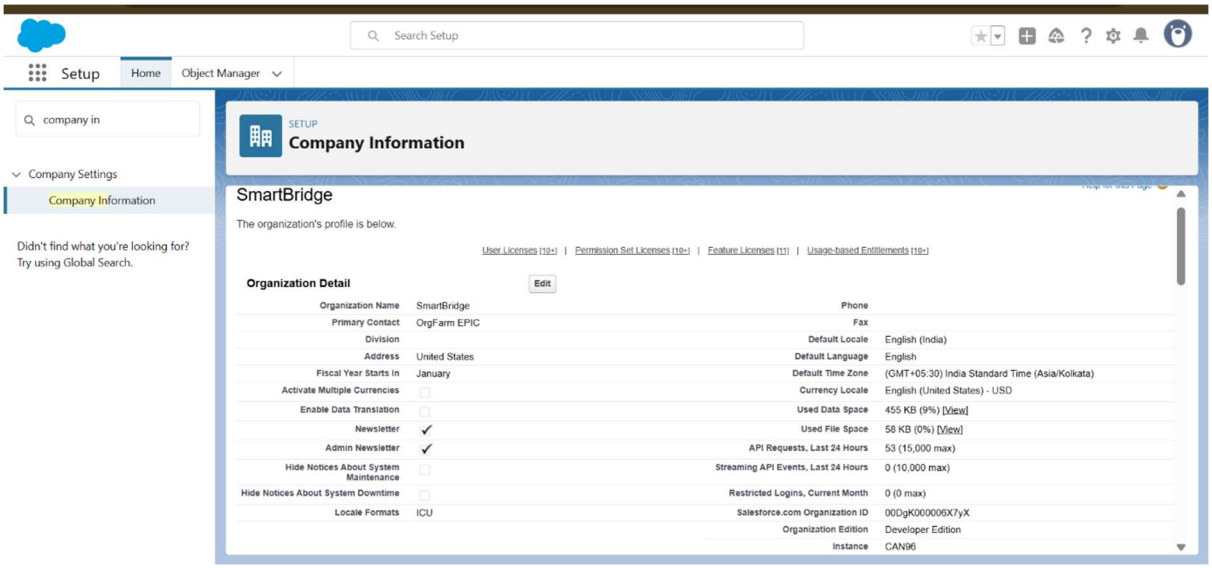


Field-Level Security: Restricted exposure of fields like internal valuation, margin calculations, or confidential notes.

Validation Rules: Prevent incomplete submissions and enforce logical boundaries between interrelated field

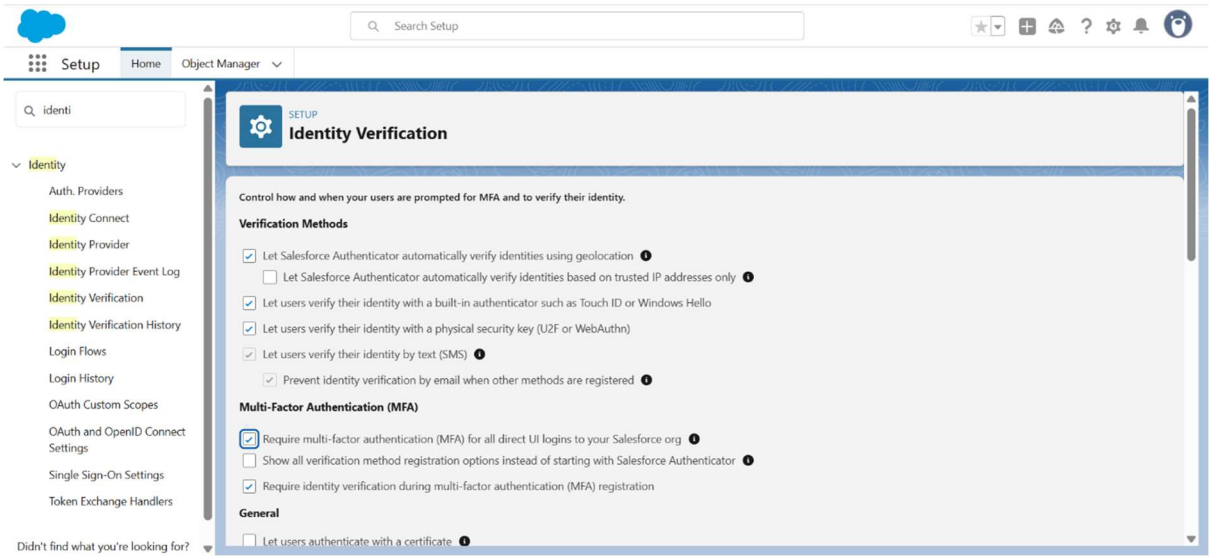
Session & Login Protections:

- **Trusted IP Ranges**: Restricted login access to verified network zones.
- **Login Hours**: Defined per profile to prevent access outside business hours.
- **Session Timeout Settings**: Automatically signed out inactive users to reduce unattended exposure risks



Two-Factor Authentication (2FA):

Two-Factor Authentication (2FA) is a security process that requires users to verify their identity using **two distinct methods** before gaining access to a system or application. This layered approach significantly reduces the risk of unauthorized access, even if one credential is compromised.



Phase 5: Deployment, Documentation & Maintenance *(No Sandbox Used):*

This phase involved launching the CRM solution directly in the production environment, ensuring a real-time operational setup backed by thorough documentation and a well-defined maintenance protocol. Despite the absence of a sandbox, each configuration was meticulously implemented with robust safeguards and traceable change logs to support long-term scalability and confidence.

Deployment:

Configuration and feature setup were executed directly within the production environment to maintain speed and precision. Key workflows — including lead assignment, inventory actions, and approval routing — were validated using live user profiles and record interactions. A manual change management process was followed with rollback planning and version tracking to ensure release integrity.

- **Live Configuration Execution**: All user roles, layouts, approval logic, and automation flows were directly configured within the production org, avoiding the overhead of sandbox environments.
- **Workflow Verification in Production**: User-level testing confirmed data access, record creation flows, and task assignments across Inventory, Sales, and Admin roles.
- **Change Tracking Protocol**: All changes were manually logged, including Apex updates, field-level security configurations, and layout customizations. A rollback strategy using metadata backups and version notes ensured release reliability.

Documentation:

Clear functional summaries and automation flow diagrams were created to support user onboarding and stakeholder visibility. Apex logic was documented with inline comments and stored externally for future reference. Admin-level configuration changes — such as profile setups, page layouts, and field permissions — were systematically logged to maintain transparency and traceability.

- **Functional Module Index**: Comprehensive documentation created for each business module — Inventory, Orders, leads — outlining key objectives and interactions.
- **Technical Asset Registry**: Apex classes and triggers catalogued with method descriptions, logic overviews, and associated record types.
- **Configuration Change Log**: Excel-style tracker maintained with timestamps, admin actions, and purpose notes for every significant change.
- **User Access Matrix**: Role-based visibility and permission structure presented with diagrams for stakeholder clarity.
- **Visual Aids & Flowcharts**: Embedded visuals include workflow diagrams, Lightning Page layouts, and record lifecycle illustrations.

Maintenance:

Ongoing upkeep is supported by scheduled audits for performance, data hygiene, and automation accuracy. A centralized issue-reporting matrix allows users to submit feedback and request enhancements. Governance mechanisms were established to regulate future updates, ensuring consistency and risk control.

- **Data Health Checks**: Routine validations scheduled to inspect orphan records, field population rates, and automation timing.
- **Automation Review Schedule**: Monthly assessment of flows, triggers, and email services to ensure relevance and performance.
- **User Feedback System**: Internal channel setup for reporting issues or suggesting enhancements with ticket tracking.
- **Change Request Governance**: Formal review process defined for future changes, including approval routing, rollback plan, and impact analysis.

Conclusion:

The Salesforce CRM solution implemented for **Re Plastic Innovations** represents a significant milestone in the digital transformation of the plastic waste recycling industry. This project has successfully demonstrated how cloud-based platforms like Salesforce can be harnessed to solve critical business challenges—particularly in areas such as **inventory management, order processing, task automation, and inter-departmental collaboration**.

By designing a secure and scalable data model, along with effective automation through Apex triggers and Flow, the system ensures real-time monitoring of stock levels and reduces the risk of human error. Key functionalities—such as automated restock requests, approval workflows, and email notifications—have streamlined operations and ensured that no aspect of inventory or order management is delayed or overlooked.

Additionally, the integration of role-based access, profile-level security, and field history tracking has enhanced data protection, ensuring compliance and controlled accessibility. The inclusion of dashboards and reports provides management with actionable insights to make informed, data-driven decisions.

Moreover, the CRM system lays a strong foundation for future enhancements, such as integrating Artificial Intelligence for predictive analytics, chatbot support for real-time customer assistance, or mobile optimization for on-the-go accessibility.

This capstone project has not only met the defined business objectives of Re Plastic Innovations but has also helped build a sustainable and technology-driven approach to tackling environmental waste. It showcases how Salesforce can be leveraged as a transformative platform to support both **business growth** and **sustainable development goals**.

Future Enhancements:

As Re Plastic Innovations continues to grow and evolve, the current CRM implementation can be further enhanced with the following upgrades:

1. **Chatbot Integration**: Add a Salesforce-integrated chatbot to guide users in placing restock requests or reporting inventory issues.
2. **AI-Powered Recommendations**: Use Salesforce Einstein Analytics to forecast demand, suggest procurement levels, and analyse recycling trends.
3. **Mobile Optimization**: Develop a lightweight mobile app using Salesforce Mobile SDK for warehouse teams to update stock on the go.
4. **Customer Portal**: Build a Community Cloud portal where external partners or clients can track orders, deliveries, and recycling reports.

These enhancements will not only boost user experience and productivity but also further reinforce the organization's commitment to sustainable innovation.

References:

https://www.salesforce.com/trailblazer/olwc2c4vchcvytuja1

https://github.com/sonymukkera900/Replastix-Innovations-Project

https://www.linkedin.com/in/sony-mukkera-4a5165260