

REPLASTIX INNOVATIONS: TRANSFORMING PLASTIC WASTE INTO SUSTAINABLE SOLUTIONS

Abstract:

RePlastix Innovations is a Salesforce CRM project I developed with the vision of simplifying plastic waste management through automation, real-time tracking, and digital transparency. The entire system is designed around the daily operational needs of organizations that recycle plastic, combining multiple Salesforce features to handle everything from collection to restocking. It automates routine tasks and keeps users informed through alerts and dashboards.

The project uses a mix of declarative tools like Flows, Process Builders, Validation Rules, and backend development through Apex Triggers and Classes. The CRM covers five major custom objects Waste, Recycling Centers, Recycled Products, Orders, and Restock Requests each one tied together to form a seamless workflow. When stock falls, the system takes action without waiting for a manual push.

With clean record structures, scheduled automations, and role based access, the system guarantees data accuracy and security. Email notifications, automated task creation, and dynamic updates make the app responsive and smart. Even as new orders come in, the warehouse stays on top of inventory levels, and the manager doesn't miss a beat.

This report breaks down the journey from requirement analysis to final deployment. It covers object design, flow structures, trigger logic, security models, test classes, and outcome results. Every single step, every configuration it's all documented, tested, and polished to reflect a real-world business implementation.

Project Overview:

Plastic pollution is a growing threat, and managing it smartly is no longer optional it's a responsibility. The Indian recycling sector is evolving fast, supported by national programs and strict environmental rules. This is where RePlastix comes in a CRM designed to support this shift by digitizing every process in plastic waste management using Salesforce.

I wanted the project to support circular economy principles by helping organizations repurpose waste instead of discarding it. RePlastix makes this possible by giving users the ability to track plastic waste in detail, monitor the recycling center's capacity, and maintain a balanced flow of supply and demand with real-time insights.

The system is not just about tracking; it's about making smart moves with data. Warehouse managers get daily updates, alerts, and tasks the moment something needs action. Restocks are triggered automatically, orders are managed based on live stock, and product status is updated instantly using Flows and Apex Triggers.

This solution fits right in with sustainability efforts it's digital, it's responsive, and it's scalable. Whether it's a local recycling unit or a nationwide program, this CRM can be adjusted and extended. It's not just built to work it's built to grow.

Objectives:

The main goal behind RePlastix was to replace the slow and error-prone manual process with a modern, cloud-based system. Everything had to be automated, visible, and manageable in one place. The project was driven by the need for operational clarity from the moment plastic is collected to the second it becomes a recycled product in stock.

I designed the system to cover end to end operations with little to no manual interference. Stock level updates, restock alerts, and order tracking are all automated through daily checks and system triggers.

Managers receive email notifications, while internal users get tasks and alerts to take immediate action.

Security was another key objective. Not every user should have access to everything, so I set up strict roles, profiles, and sharing rules to limit visibility. Each team member whether in sales, collection, or warehouse can only access what's relevant to their role. That ensures data safety and avoids confusion.

Lastly, I wanted this app to be insightful. So, I built reports and dashboards that offer real-time data views. Whether it's low stock alerts or restock request statuses, the dashboards help in making quick, smart decisions. It's not just about showing data it's about showing what to do next.

Phase 1: Requirement Analysis & Planning

The first step was to fully understand the business needs and expectations. I spent time mapping the pain points like stock mismanagement, delayed orders, lack of tracking and translated those into system requirements. This was about designing a CRM that doesn't just look good but solves real issues for the people using it.

I broke the scope into two main layers: functional and technical. Functionally, I had to create five custom objects and define relationships among them. I also needed to ensure the right validations were applied to prevent user errors. Technically, I was looking at scheduled Flows, Apex logic, and email automation that could act without waiting on users.

The security model was crucial too. I mapped out a role hierarchy and assigned specific CRUD rights through profiles and permission sets. Organizational-Wide Defaults were set to Private, so users only saw what they needed. Sharing rules were defined to open up access only where it made sense like a Warehouse Manager seeing all restocks, but not customer orders.

To support the build, I designed the data model with clarity and growth in mind. I kept naming conventions simple, used formula fields for dynamic calculations, and created an ERD to visualize everything before jumping into development. This planning phase laid the groundwork for a project that could be maintained and expanded easily.

Phase 2: Salesforce Development Backend & Configurations

Once the planning was solid, I jumped into backend development. I created all five custom objects: Plastic_Waste__c, Recycling_Center__c, Recycled_Product__c, Order__c, and Restock_Request__c. Each object was equipped with well thought out fields custom picklists, checkboxes, formula fields, and auto-numbering all selected to make data entry intuitive and system logic predictable. Every field served a purpose, and validation rules ensured the quality stayed airtight like blocking future dates, preventing negative values, and ensuring related records existed.

Next, I built the logic layer, and this became the real engine of the CRM. I created a Scheduled Flow that runs every 24 hours to monitor Stock_Level__c in Recycled_Product__c. When a product falls below its defined Threshold__c, the system doesn't wait it fires a task to the Warehouse Supervisor and flags the product as low in stock. This flow is built with error paths and debug assignments to catch every exception, and it uses formula resources to calculate urgency levels for restocks automatically.

On the Apex side, I wrote the InventoryManager class to control how stock is updated after an order is placed or a restock is approved. The UpdateStockAfterOrder trigger ensures that product quantities reduce in real-time as orders are saved. Likewise, the UpdateStockAfterRestockApproval trigger listens for status changes and adds the approved quantity back into stock. I also created a utility class

called EmailNotificationHelper to send beautifully formatted, dynamic emails containing stock names, remaining quantity, and quick links all handled through Apex Messaging.

But I didn't stop at basic logic. I used SOQL and DML best practices like selective queries, bulk safe triggers, and limits awareness to make sure nothing breaks during mass updates. I wrapped logic blocks in try catch for graceful error handling and added detailed logging using System.debug() and custom debug messages for future troubleshooting. I also included helper methods to centralize logic reuse and kept all Apex classes clean, modular, and scalable. The result? A backend that doesn't just respond it thinks ahead.

Phase 3: UI/UX Development & Customization

Even the most powerful backend is useless without a good interface, so I focused heavily on user experience. Using App Manager, I created a Lightning App called "RePlastix Innovations." It has tabs for all five objects, custom icons, and a clean, modern layout. Navigation is smooth, and every object opens exactly where it's needed.

I used dynamic forms to make data entry clean and context-aware. For example, if a status field is marked "Pending," only relevant fields appear; if it's marked "Approved," the layout shifts to show the next steps. This keeps forms clean, removes clutter, and helps users stay focused on what matters most.

For dashboards, I built two primary views: one for stock monitoring and another for restocks. These dashboards show visual insights bar charts, donut charts, grouped summaries that let managers spot trends instantly. Each chart links back to its underlying report, giving full control to the users.

I also made sure to apply permissions at the UI level using visibility filters and custom record pages. This way, each user only sees tabs, fields, and records that match their role. Admins have the full view; reps get what they need no more, no less.

Phase 4: Data Migration, Testing & Security

Data migration was handled using a combination of the Data Import Wizard and Data Loader. I uploaded dummy records for each object and made sure the relationships worked perfectly orders linked to products, restocks linked to orders, and everything linked back to the recycling centers. This was crucial for testing the system in action.

My testing approach was aggressive I didn't just test success paths, I broke the system on purpose. I fed it wrong data, like negative quantities or future collection dates, just to make sure validations fired correctly. I simulated low stock conditions to test whether the scheduled Flows and triggers reacted exactly as expected.

Security testing was done using profile-based login-as sessions. I tested how each user role viewed the system what they could see, what they could edit, and what stayed hidden. I even tested record sharing manually for exceptions where rules needed to be overridden temporarily.

Finally, I turned on Field History Tracking on critical fields like Stock Level and Order Status. I reviewed logs, debugged Flow errors, and monitored email logs to ensure that every automation fired with precision. No ghost tasks, no silent failures everything had to speak when it moved.

Phase 5: Deployment, Documentation & Maintenance

Deployment was done using Change Sets. I grouped related components together objects, flows, fields, validation rules, Apex classes and moved them in a single outbound package. Every Flow was activated, test classes were executed, and security settings were reverified before migration.

I maintained detailed documentation through the whole build. Each object had a data dictionary. Every Flow was mapped out with a diagram. I wrote comments in Apex and summarized trigger logic in inline notes. Nothing was left vague. It wasn't just built it was built to be understood and maintained.

Debug logs were enabled during testing, and I created a checklist for post-deployment QA. This included checking Flow status, confirming profile access, and validating real-time operations like email alerts and task creation. Maintenance tasks were outlined monthly reviews, user training, log analysis, and exception tracking.

To future-proof the app, I designed it modularly. Object names, field names, and logic patterns were clean and consistent. That way, adding new features like supplier modules or chatbot integration later wouldn't mess up what's already there. The goal? Build once, scale forever.

Outcomes:

This project resulted in a fully functioning Salesforce CRM tailored for the plastic recycling industry. It includes five custom objects with a well-integrated backend, complete automation, dynamic forms, role-based access, and real-time dashboards. Every step from plastic collection to stock management is handled through clean, automated flows and Apex logic.

The stock management process, which was earlier manual and prone to error, is now entirely automated. Stock levels are tracked daily. When thresholds are breached, the system sends tasks and email alerts. Orders

update stock in real time, and approvals trigger restocking without anyone having to remember or manually check inventory levels.

Dashboards give managers a high-level overview of operations low-stock alerts, restock statuses, product types in demand, and order frequency. These analytics help users make smart decisions faster and with confidence. With the click of a tab, the entire lifecycle of plastic waste is traceable from collection to product dispatch.

The app was tested for code quality, logic consistency, and user access across multiple roles. Every automation runs clean. Every object is documented. And the best part? It's future-ready built with modularity, clarity, and adaptability in mind. This isn't just a student project; it's a real, scalable, industry-grade solution.

Conclusion:

This project wasn't just about completing an assignment it was about solving a real-world problem using technology. I took a challenge plastic waste and translated it into workflows, fields, dashboards, and logic that actually do something. What started as a simple idea turned into a fully functioning, impactful CRM.

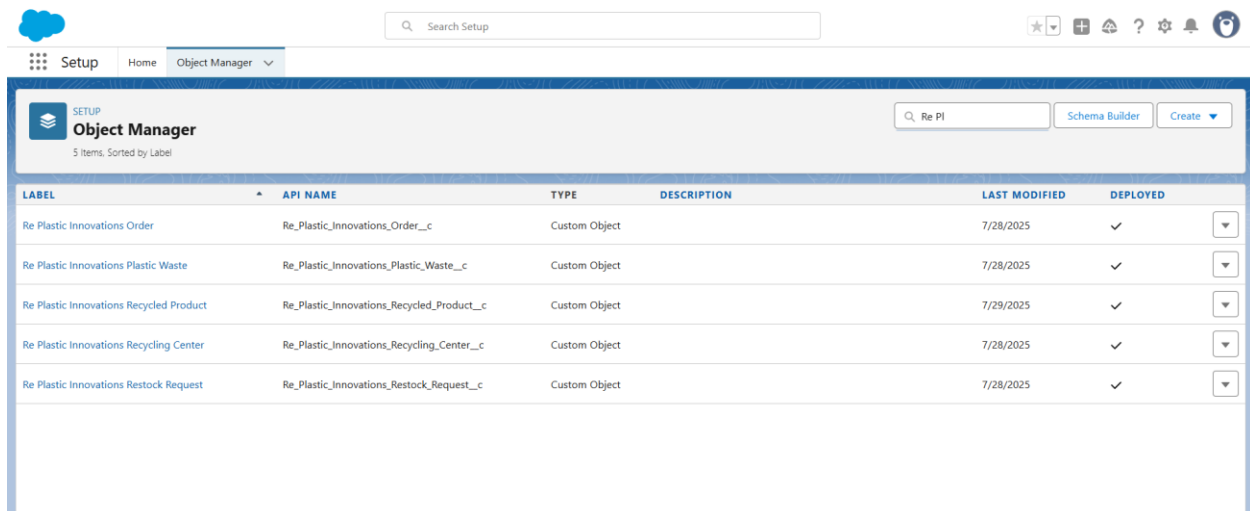
Working with Salesforce gave me the chance to explore both declarative and programmatic tools. I learned to think in relationships, not just fields. I understood the power of Flows, how Apex can be used smartly, and why validation rules aren't just safeguards they're the backbone of data integrity. It wasn't just building it was building right.

I also gained hands-on experience in managing user roles, sharing rules, security models, and object relationships. Understanding how different

departments (warehouse, sales, collection) interact through technology taught me how to design better, more user-centric systems. This experience improved how I approach problem-solving overall.

Most importantly, it reminded me why tech matters not for the flash, but for the function. RePlastix isn't flashy. It's efficient. It works. It makes a dirty, complex process like waste management feel clean, traceable, and easy. And that's the real win.

Appendices:



The screenshot displays the 'Object Manager' interface within a 'Setup' environment. At the top, there is a navigation bar with 'Setup', 'Home', and 'Object Manager' (selected). A search bar labeled 'Search Setup' is present. Below the navigation bar, the 'Object Manager' section shows a search for 'Re PI' and buttons for 'Schema Builder' and 'Create'. The main content is a table with 5 items, sorted by label. The table has columns for LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. Each row represents a custom object, all of which are 'Custom Object' type and were last modified on 7/28/2025. The 'DEPLOYED' column shows a checkmark for each item, and a dropdown arrow is visible for each row.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Re Plastic Innovations Order	Re_Plastic_Innovations_Order__c	Custom Object		7/28/2025	✓
Re Plastic Innovations Plastic Waste	Re_Plastic_Innovations_Plastic_Waste__c	Custom Object		7/28/2025	✓
Re Plastic Innovations Recycled Product	Re_Plastic_Innovations_Recycled_Product__c	Custom Object		7/29/2025	✓
Re Plastic Innovations Recycling Center	Re_Plastic_Innovations_Recycling_Center__c	Custom Object		7/28/2025	✓
Re Plastic Innovations Restock Request	Re_Plastic_Innovations_Restock_Request__c	Custom Object		7/28/2025	✓

The screenshot displays the Salesforce interface for a record titled "Test Task Record" under the "Re Plastic Innovations Recycled Product" object. The interface includes a top navigation bar with a search bar and various menu items like Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, and a dropdown for "Re Plastic Innovations Recycled...". Below the navigation bar, the record details are shown in a table format. The table has two columns: "Related" and "Details". The "Details" column contains the following information:

Related	Details
Re Plastic Innovations Recycled Product Name	Test Task Record
Stock_Level	200
Threshold	100
Price	\$900,000
Stock Low On Product	Sufficient Stock
Created By	thrailokya puvvada, 7/29/2025, 7:48 AM
Last Modified By	thrailokya puvvada, 7/29/2025, 7:51 AM

On the right side of the record details, there is an "Activity" section. It includes a "Filters" dropdown set to "All time", "All activities", and "All types". Below the filters, there is a section titled "Upcoming & Overdue" which shows a task: "Please Look in this Stock Is Low ...". The task is marked as "No due date" and "You have an upcoming task". At the bottom of the activity section, it states "No past activity. Past meetings and tasks marked as done show up here."

The screenshot displays the Salesforce interface for a record titled "Plastic Order Record" under the "Re Plastic Innovations Recycled Product" object. The interface is similar to the one above, with a top navigation bar and a record details table. The "Details" column contains the following information:

Related	Details
Re Plastic Innovations Recycled Product Name	Plastic Order Record
Stock_Level	150
Threshold	100
Price	\$900,000
Stock Low On Product	Sufficient Stock
Created By	thrailokya puvvada, 7/30/2025, 3:47 AM
Last Modified By	thrailokya puvvada, 7/30/2025, 5:03 AM

The "Activity" section on the right is also present, with filters set to "All time", "All activities", and "All types". The "Upcoming & Overdue" section shows "No activities to show." and a message: "Get started by sending an email, scheduling a task, and more." At the bottom, it states "No past activity. Past meetings and tasks marked as done show up here."

Future Enhancements:

Moving forward, there are several enhancements that can make the system even more powerful. The first is adding AI-based recommendations using Salesforce Einstein Analytics so the system doesn't just respond to low stock but predicts when it might go low, based on past orders and usage patterns.

Second, I plan to integrate the Salesforce Mobile App for on-the-ground agents. Field workers could use their phones to log waste collections instantly, attach images, and sync data with the cloud in real time. This would reduce manual data entry and improve collection traceability.

Another enhancement would be IoT sensor integration. If recycling centers have sensors tracking capacity or weight, that data can flow directly into Salesforce to update stock or trigger workflows. This would make inventory truly live and eliminate the guesswork entirely.

Finally, the system can be expanded to include multi-language support, chatbot assistance for customer queries, supplier modules, and integration with third-party logistics tools. The base is solid now it's ready to grow, evolve, and become even smarter and more useful.