

# Subscripify :- Subscription Box Service Management System

## Project Overview:

This project delivered a specialized Salesforce CRM to manage a subscription box business. The system provides a centralized, secure, and scalable platform to manage the end-to-end subscriber lifecycle.

## Key Features Implemented:

- **Custom Data Model:** Subscription, Invoice, and Inventory objects for a 360-degree customer view.
- **Advanced Security:** A robust security model using Profiles, OWD, and Field-Level Security (FLS) to protect sensitive financial data.
- **Data Integrity:** Custom duplicate and validation rules to prevent bad data entry and ensure accuracy.
- **Analytics:** Real-time operational reports and dashboards to track key metrics like Monthly Recurring Revenue (MRR) and inventory levels.

## Objectives:

The primary goals of this CRM project and their business value are as follows:

- **Centralize Data:** To create a single source of truth that streamlines operations and improves overall customer management.
- **Automate Processes:** To reduce manual errors and increase efficiency across the entire subscription lifecycle, from order to fulfillment.
- **Implement Security:** To safeguard sensitive customer and financial data, thereby increasing customer trust and ensuring compliance.
- **Enable Analytics:** To provide real-time, data-driven insights that allow for proactive decision-making and business growth.

# **Subscripify :- Subscription Box Service Management System**

## **Phase 1: Problem Understanding & Industry Analysis**

### **Requirement Analysis: -**

#### **Subscriber & Subscription Management**

Track subscriber status as Active, Paused, or Canceled and allow admins to create and edit subscription tiers with unique pricing and products.

#### **Automated Billing & Fulfillment**

Automated billing engine to generate invoices and process monthly payments. It also Handle failed payments by notifying customers to update payment info. Real-time inventory tracking with automatic stock deduction to prevent overselling.

#### **Subscriber Self-Service Portal**

Portal for subscribers to update shipping and payment details. Options to pause, resume, or cancel subscriptions. Access to order history, including payments and shipments.

#### **Business Analytics & Reporting**

Dashboard for viewing key business metrics. It also Reports on Monthly Recurring Revenue (MRR), Customer Churn Rate, and Inventory Levels.

### **Stakeholder Analysis :-**

**Subscription Manager:** It Oversees daily operations, manages subscription plans, and monitors business performance dashboards.

**Fulfillment Coordinator:** Manages product inventory and processes paid orders for shipping from a clear fulfillment queue.

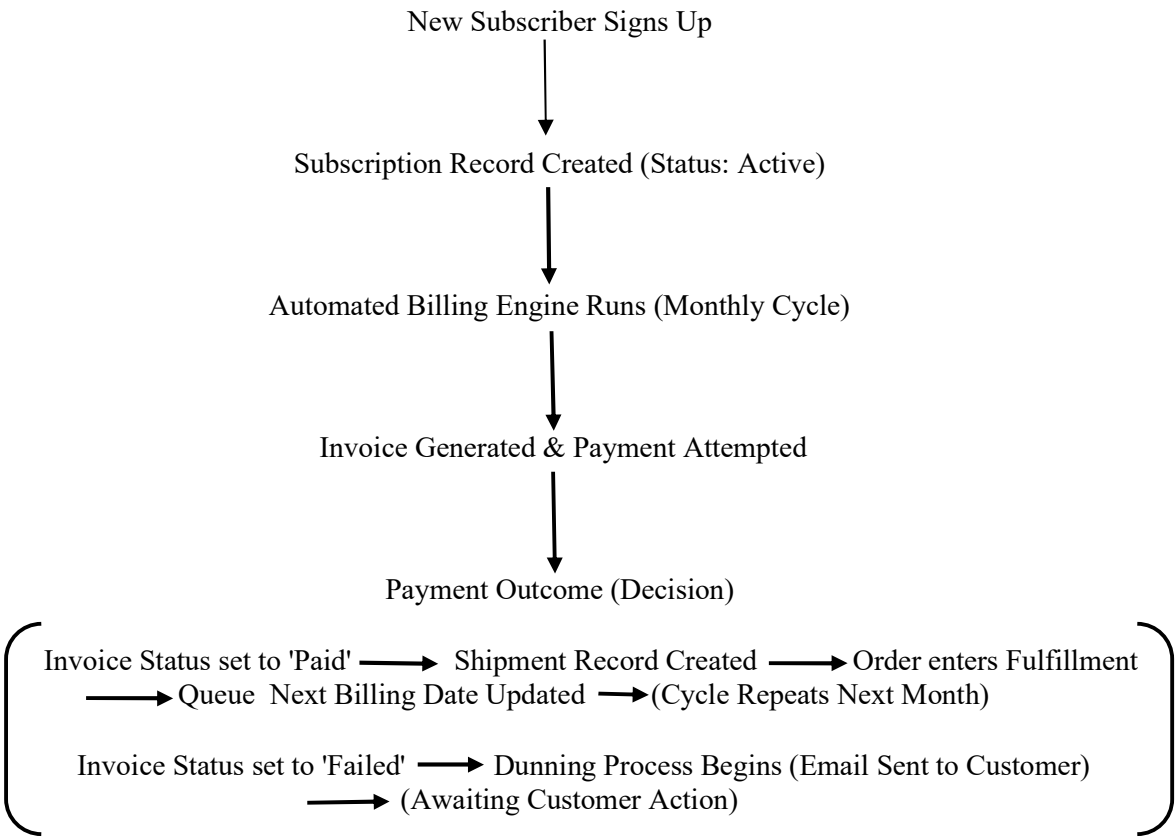
**Finance Manager:** Monitors revenue reports, tracks failed payments, and manages refund approvals.

**Subscriber (External User):** The end customer who uses the self-service portal to manage their account and subscription.

**System Admin:** Manages the system configuration, security, and deployments.

# Subscripify :- Subscription Box Service Management System

## Business Flow Mapping: -



## Industry-specific Use Case Analysis

The Subscripify platform is specifically designed to address the unique challenges of the subscription box industry. Its primary function is to serve as a reliable, automated **recurring revenue engine**, managing the entire monthly cycle of billing and payments. The key goal is to reduce customer cancellations with features designed to keep them, like a flexible "pause subscription" option. Also, because we ship physical goods, the system requires precise, real-time inventory management to prevent selling out-of-stock items and ensure happy customers.

# Subscripify :- Subscription Box Service Management System

## Phase-2 (Org Setup & Configuration)

- **Salesforce Edition:** - Salesforce Developer Edition Org

### Company Profile Setup

- **Company Name:** Subscripify
- **Default Currency:** INR
- **Secondary Currency :** USD
- **Locale & Language:** India (English)
- **Time Zone:** Asia/Kolkata (GMT+5:30)
- **Fiscal Year :** 1<sup>st</sup> April- 31<sup>st</sup> March

Organization Detail		Edit	
Organization Name	Subscripify	Phone	+91 12345
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (United States)
Address	1, Ring Road, Sadar, Jabalpur (M.P) Jabalpur 482005 Madhya Pradesh India	Default Language	English
Fiscal Year Starts In	Custom Fiscal Year	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	<input type="checkbox"/>	Currency Locale	English (India) - INR
Enable Data Translation	<input type="checkbox"/>	Used Data Space	342 KB (7%) <a href="#">View</a>
Newsletter	<input checked="" type="checkbox"/>	Used File Space	17 KB (0%) <a href="#">View</a>
Admin Newsletter	<input checked="" type="checkbox"/>	API Requests, Last 24 Hours	0 (15,000 max)
Hide Notices About System Maintenance	<input type="checkbox"/>	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	<input type="checkbox"/>	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgL00000BT50J
		Organization Edition	Developer Edition
		Instance	CAN98
Created By		OrgFarm EPIC	9/12/2025, 3:00 AM
		Modified By	Akshat Sharma, 9/15/2025, 7:16 PM

### Business Hours & Holidays

- **Business Hours:** 10:00 AM – 5:00 PM (Monday – Saturday).
- **Holidays:** Sunday holiday, Festive Holiday.

### Fiscal Year Settings

- Using the Custom Fiscal Year (1<sup>st</sup> April – 31<sup>st</sup> March).
- Ensures finances can be reported on a yearly basis and can be managed.

### User Setup & Licenses

- **Subscription Admin** – Manages the day-to-day operations and strategy of the subscription service
- **Fulfilment User** – It only has read-only access to orders so they can manage inventory and process shipments, with no access to financial data.
- Finance Manager
- System Administrator

# Subscripify :- Subscription Box Service Management System

## Profiles

- **Subscription Admin Profile** – System access related to subscription which runs the day-to-day operations of the service.
- **Fulfilment User Profile** – It is highly restricted and gives them access only to the tools needed to physically pack and ship customer order.
- **Finance Manager** - Complete system access which runs the day-to-day operations of the service related to finances.
- **System Administrator** - Complete system access which runs the day-to-day operations of the service.

## Roles

- **Director of Operations Role** – Top-level in the business hierarchy, oversees the entire subscription lifecycle and monitors overall business performance.
- **Subscription Manager Role** – Manages the core business functions, including subscription plans, automated billing cycles, and subscriber analytics.
- **Fulfillment Coordinator Role** – Manages the physical logistics of the operation, including inventory levels and the fulfillment of paid orders.
- Finance Manager Role
- System Administration Role

<input type="checkbox"/> Del   Clone	Subscription Admin	Salesforce	✓
<input type="checkbox"/> Del   Clone	Fulfillment Coordinator	Salesforce	✓
<input type="checkbox"/> Del   Clone	Customer	Customer Community	✓
<input type="checkbox"/> Clone	System Administrator	Salesforce	<input type="checkbox"/>

## Permission Sets

- **Billing & Refund Adjustments :-** It Grants special permission to edit paid invoices or process refunds. Assigned to senior managers as needed.
- **Product Catalog Manager :-** It Grants access to create and edit Product records (subscription plans and items), separate from the main Subscription Admin profile.

## OWD (Org-Wide Defaults)

- **Subscription Records :-** Private. This is critical to protect each subscriber's personal and plan information.
- **Invoice Records :-** Private. Protects sensitive customer financial history.
- **Product Inventory :-** Public Read/Write. All internal users can see and manage stock levels.

# **Subscripify :- Subscription Box Service Management System**

## **Sharing Rules**

A criteria-based sharing rule will be created to automatically grant read-only access to Invoice records to the Fulfillment Team public group. This rule will trigger when an invoice's Status is marked as 'Paid', ensuring the fulfillment team can only see orders that are ready to be shipped.

## **Login Access Policies**

The policy of logging in as an administrator will be activated. This crucial setting enables System Administrators to log in as other users (such as the Fulfillment Coordinator or Subscription Manager) in order to effectively support and troubleshoot any user-specific issues without needing the password.

## **Dev Org Setup**

- A Salesforce Developer Edition Org will serve as the primary environment for all configuration, custom development (Apex, LWC), and implementation of the Subscripify platform.
- All unit testing, system validation, and final demonstration preparation will be performed within this dedicated environment.

## **Deployment Basics**

- A modern, professional development lifecycle will be followed to ensure best practices are maintained.
- All new features will be developed and tested in a Developer Sandbox to isolate work-in-progress from the stable main environment.
- Completed features will be deployed from the sandbox to the main Developer Org (acting as 'Production') using Salesforce DX (SFDX) and a source control repository (Git), which is the current industry standard for Salesforce development.

# Subscripify :- Subscription Box Service Management System

## Phase 3: Data Modelling & Relationships

- **Custom Objects & Fields**

- 1. **Subscription (Subscription\_\_c)**

- Fields:**

- Subscription ID (Auto-Number)
- Subscriber (Lookup → Contact)
- Subscription Plan (Lookup → Product2)
- Status (Picklist: Active, Paused, Canceled)
- Start Date (Date)
- Next Billing Date (Date)
- Cancellation Date (Date)
- Currency (Currency)

SETUP > OBJECT MANAGER			
Subscription			
Fields & Relationships			
11 Items, Sorted by Field Label			
Q Quick Find			
New Deleted Fields Field Dependencies Set History Tracking			
Currency	CurrencyIsoCode	Picklist	
Last Modified By	LastModifiedById	Lookup(User)	
Next Billing Date	Next_Billing_Date__c	Date/Time	
Owner	OwnerId	Lookup(User,Group)	✓
Start Date	Start_Date__c	Date/Time	
Status	Status__c	Picklist	
Subscription	Name	Text(80)	✓
Subscription ID	Subscription_ID__c	Auto Number	
Subscription Plans	Subscription_Plans__c	Lookup(Product)	✓

- 2. **Invoice (Invoice\_\_c)**

- Fields:**

- Invoice Number (Auto-Number)
- Subscription (Lookup → Subscription\_\_c)
- Status (Picklist: Draft, Unpaid, Paid, Payment Failed, Refunded)
- Invoice Date (Date)
- Amount (Currency)
- Payment Method (Picklist)

# Subscripify :- Subscription Box Service Management System

SETUP > OBJECT MANAGER

**Invoice**

Details

**Fields & Relationships**  
10 Items, Sorted by Field Label

Q, Quick Find   New   Deleted Fields   Field Dependencies   Set History Tracking

Amount	Amount__c	Currency(18, 0)	
Created By	CreatedById	Lookup(User)	
Currency	CurrencyIsoCode	Picklist	
Invoice Date	Invoice_Date__c	Date/Time	
Invoice__c Name	Name	Text(80)	✓
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User/Group)	✓
Record Type	RecordTypeId	Record Type	✓
Status	Status__c	Picklist	
Subscription	Subscription__c	Lookup(Subscription)	✓

## 3. Product Inventory (Product\_Inventory\_\_c)

### Fields:

- Product (Lookup → Product2)
- SKU (Text)
- Quantity In Stock (Number)
- Re-Order Threshold (Number)
- Warehouse Location (Text)

SETUP > OBJECT MANAGER

**Product Inventory**

Details

**Fields & Relationships**  
9 Items, Sorted by Field Label

Q, Quick Find   New   Deleted Fields   Field Dependencies   Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Currency	CurrencyIsoCode	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Product	Product__c	Lookup(Product)		✓
Product Inventory Name	Name	Text(80)		✓
Quantity In Stock	Quantity_In_Stock__c	Number(18, 0)		
Re-Order Threshold	Re_Order_Threshold__c	Number(18, 0)		
Stock Keeping Unit	Stock_Keeping_Unit__c	Auto Number		

## 4. Shipment (Shipment\_\_c)

### Fields:

- Shipment ID (Auto-Number)
- Related Invoice (Lookup → Invoice\_\_c)
- Subscriber (Lookup → Contact)
- Status (Picklist: Ready to Ship, Shipped, Delivered)



# Subscripify :- Subscription Box Service Management System

- Tracking Number (Text)
- Shipping Address Snapshot (Text Area)

Setup

Home

Object Manager

Search Setup

Star

Grid

Refresh

Help

Settings

Notifications

Profile

SETUP > OBJECT MANAGER

Shipment

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

Fields & Relationships

11 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Currency	CurrencyIsoCode	Picklist		
Inventory Processed	Inventory_Processed__c	Checkbox		
Invoice	Invoice__c	Lookup(Invoice)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Shipment Name	Name	Text(80)		✓
Shipping Address	Shipping_Address__c	Long Text Area(32768)		
Status	Status__c	Picklist		

## Page Layouts

- **Subscription Page Layout** → Show subscriber details, the selected subscription plan, current Status (e.g., Active/Paused), and the Next Billing Date.
- **Invoice Page Layout** → Show the related subscription, the invoice Status (e.g., Paid/Unpaid), the total Amount, and the Invoice Date.
- **Subscriber (Contact) Page Layout** → Show the customer's contact details, their primary shipping address, and a related list to display all of their historical Subscriptions.
- **Shipment Page Layout** → Show the fulfillment Status (e.g., Shipped/Delivered), the Tracking Number, and a link to the paid Invoice.

Setup

Home

Object Manager

Search Setup

Star

Grid

Refresh

Help

Settings

Notifications

Profile

SETUP > OBJECT MANAGER

Subscription

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Page Layouts

1 Items, Sorted by Page Layout Name

Q Quick Find

New

Page Layout Assignment

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Subscription Layout	Akshat Sharma, 9/18/2025, 11:34 AM	Akshat Sharma, 9/25/2025, 10:09 PM

# Subscripify :- Subscription Box Service Management System

SETUP > OBJECT MANAGER

Invoice

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Page Layouts

1 Items, Sorted by Page Layout Name

Q Quick Find

New

Page Layout Assignment

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Invoice Layout	Akshat Sharma, 9/18/2025, 11:52 PM	Akshat Sharma, 9/24/2025, 8:58 AM

Setup

Home

Object Manager

SETUP > OBJECT MANAGER

Contact

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Page Layouts

4 Items, Sorted by Page Layout Name

Q Quick Find

New

Page Layout Assignment

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Contact (Marketing) Layout	OrgFarm EPIC, 9/12/2025, 3:00 AM	Akshat Sharma, 9/23/2025, 11:01 PM
Contact (Sales) Layout	Orgfarm EPIC, 9/12/2025, 3:00 AM	Akshat Sharma, 9/23/2025, 11:01 PM
Contact (Support) Layout	Orgfarm EPIC, 9/12/2025, 3:00 AM	Akshat Sharma, 9/23/2025, 11:01 PM
Contact Layout	Orgfarm EPIC, 9/12/2025, 3:00 AM	Akshat Sharma, 9/23/2025, 11:01 PM

SETUP > OBJECT MANAGER

Shipment

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Page Layouts

1 Items, Sorted by Page Layout Name

Q Quick Find

New

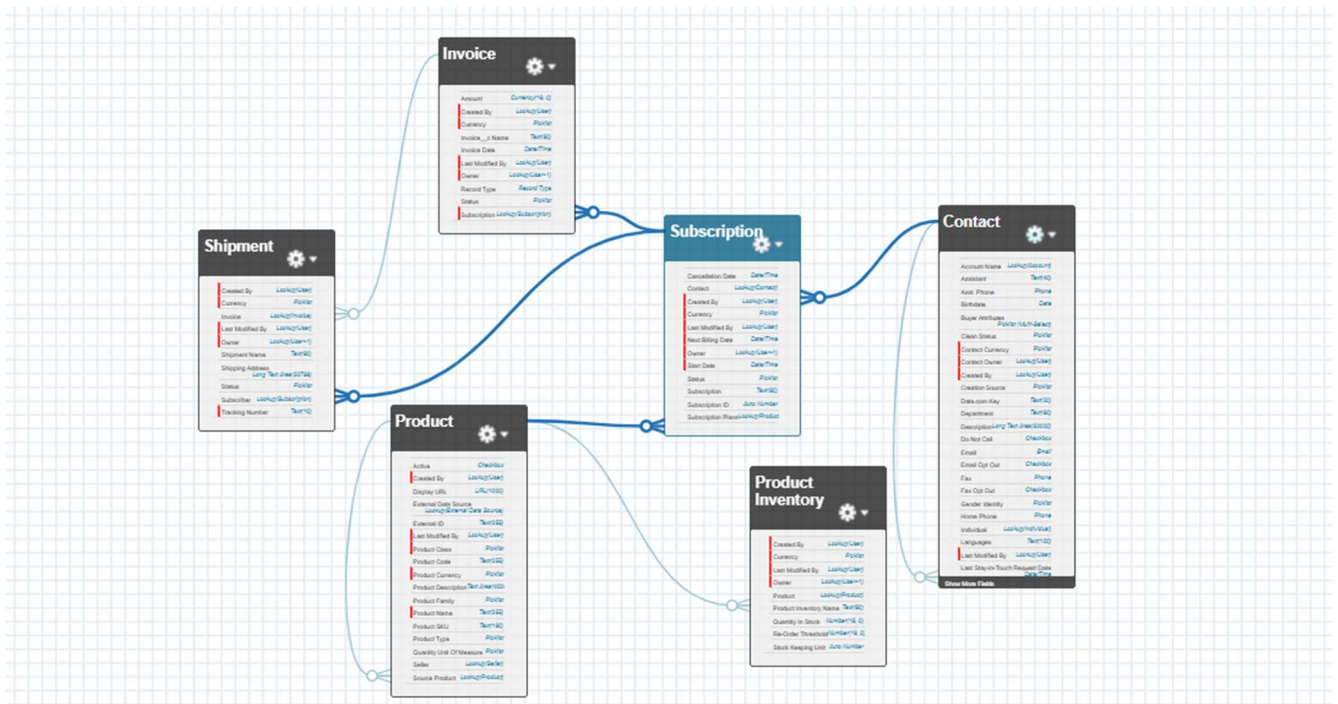
Page Layout Assignment

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Shipment Layout	Akshat Sharma, 9/18/2025, 11:54 PM	Akshat Sharma, 9/24/2025, 8:18 AM

# Subscripify :- Subscription Box Service Management System

## • Schema Builder

- **Customer Transaction Flow:** Contact (Subscriber) → Subscription → Invoice → Shipment
- **Product & Inventory Flow:** Product (Catalog) → Subscription Plan Product (Catalog) → Product Inventory Product (Catalog) → Invoice Line Item → Invoice



## • Relationships

- **Contact (Subscriber) → Subscription:** One subscriber can have multiple subscriptions over their lifetime. (**Lookup**)
- **Subscription → Invoice:** One subscription will generate many invoices over its lifetime. (**Lookup**)
- **Product (Subscription Plan) → Subscription:** One subscription plan (e.g., "Premium Box") can be associated with many different subscriptions. (**Lookup**)
- **Invoice → Invoice Line Item:** One invoice can be made up of multiple line items detailing each charge. (**Master-Detail**)
- **Invoice → Shipment:** One paid invoice results in one shipment to the customer. (**Lookup**)

# Subscripify :- Subscription Box Service Management System

- Rationale

- The separation of **Subscription**, **Invoice**, and **Shipment** objects creates a clear, scalable process. This design accurately models the real-world lifecycle of a subscription and ensures that billing data and fulfillment data are kept distinct and organized.
- Setting the Org-Wide Default for Subscription\_\_c and Invoice\_\_c to **Private** is a critical security decision. It protects sensitive customer and financial data at the most fundamental level, ensuring a trustworthy and secure platform.
- The use of **Scheduled Apex** for the monthly billing engine is the architectural cornerstone of the project. This ensures the entire revenue generation process is automated and scalable, allowing the business to grow without being limited by manual work.s

### Phase 4: Process Automation (Admin)

- Validation Rules

- Subscription

The rule checks the Start\_Date\_\_c and Cancellation\_Date\_\_c fields. If a user tries to save a record where the cancellation date is earlier than the start date, the system blocks the save and shows an error.

### Subscription Validation Rule

[Back to Subscription](#)

Validation Rule Detail		<div>EditClone</div>	
Rule Name	Cancellation_Date	Active	✓
Error Condition Formula	Cancellation_Date__c < Start_Date__c		
Error Message	The cancellation date cannot be before the subscription start date.	Error Location	Top of Page
Description			
Created By	Akshat Sharma, 9/21/2025, 1:44 AM	Modified By	Akshat Sharma, 9/21/2025, 1:44 AM
		<div>EditClone</div>	

- Product Inventory

The rule checks the Quantity\_In\_Stock\_\_c field every time the record is saved. If the number in that field is less than zero, the save action is blocked and an error message appears.

# Subscripify :- Subscription Box Service Management System

## Product Inventory Validation Rule

[Back to Product Inventory](#)

Validation Rule Detail				<div>EditClone</div>	
Rule Name	Cannot_Be_Negative	Active	<input checked="" type="checkbox"/>		
Error Condition Formula	Quantity_In_Stock__c < 0				
Error Message	The quantity in stock cannot be a negative number.	Error Location	Top of Page		
Description	The Product cannot go negative				
Created By	Akshat Sharma, 9/21/2025, 2:04 AM	Modified By	Akshat Sharma, 9/21/2025, 2:04 AM		
				<div>EditClone</div>	

- Invoice**

The rule activates when a user tries to edit an invoice. It checks if the invoice's Status was already 'Paid' *before* the current edit. If it was, and the user is trying to change a key field like Amount\_\_c, the rule stops the save.

## Invoice Validation Rule

[Back to Invoice](#)

Validation Rule Detail				<button>Edit</button>	<button>Clone</button>
Rule Name	Cannot_Edit_Paid_Invoice_Amount			Active	<input checked="" type="checkbox"/>
Error Condition Formula	AND( ISCHANGED(Amount__c), ISPICKVAL(PRIORVALUE(Status__c), "Paid") )				
Error Message	The amount on a paid invoice cannot be modified and is restricted			Error Location	Top of Page
Description	Cannot Edit Paid Invoice Amount				
Created By	Akshat Sharma, 9/21/2025, 2:01 AM			Modified By	Akshat Sharma, 9/21/2025, 2:01 AM
				<button>Edit</button>	<button>Clone</button>

- Shipment**

When a user changes the Status picklist on a shipment to 'Shipped', the rule immediately checks if the Tracking\_Number\_\_c field is empty. If it is, the rule blocks the save and displays an error.

## Shipment Validation Rule

[Back to Shipment](#)

Validation Rule Detail				<div>EditClone</div>	
Rule Name	Share_Tracking_ID		Active	<input checked="" type="checkbox"/>	
Error Condition Formula	AND( ISPICKVAL(Status__c, "Shipped"), ISBLANK(Tracking_Number__c) )				
Error Message	Tracking Number Is Needed For Shipment		Error Location	Top of Page	
Description	Share Tracking Id with the customer				
Created By	Akshat Sharma, 9/21/2025, 2:17 AM		Modified By	Akshat Sharma, 9/21/2025, 2:17 AM	
				<div>EditClone</div>	

# Subscripify :- Subscription Box Service Management System

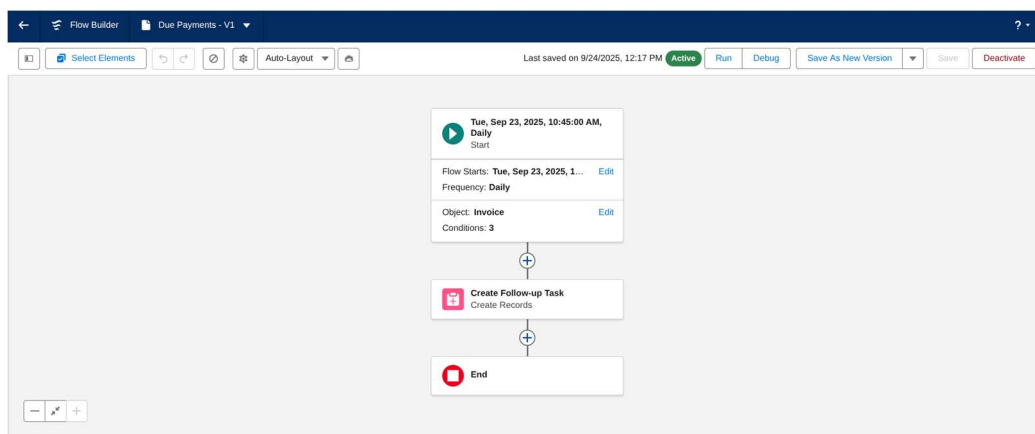
- **Approval Process**

**High-Value Refund Approval:** An approval process is configured on the Invoice\_\_c object. If a user tries to process a refund over a set amount (e.g., ₹500), the record is locked and automatically submitted to a manager for approval before any further action can be taken.

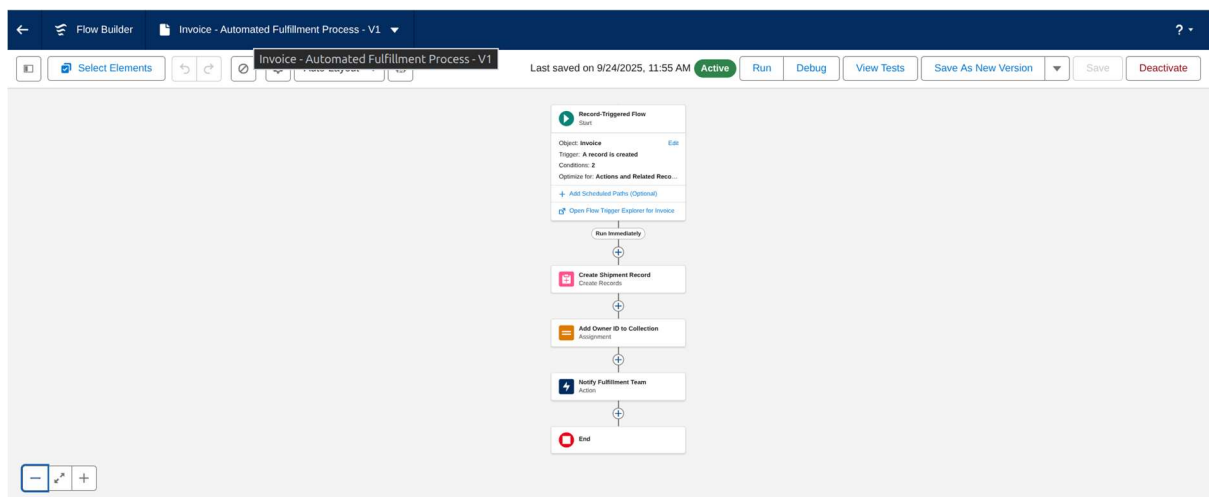
- **Flow Builder**

- **Record-Triggered Flow: Automated Fulfillment Process**

- **Trigger:** This flow automatically runs when an Invoice\_\_c record's Status is updated to 'Paid'.
    - **Actions:** It creates a new Shipment\_\_c record, performs Field Updates to populate the shipment details, sends an Email Alert to the customer, and sends a Custom Notification to the fulfillment team.



- **Scheduled Flow:** A simple flow that runs daily to send an email alert for any high-value Invoice\_\_c that is 5 days overdue.



# Subscripify :- Subscription Box Service Management System

## Phase 5 :- Apex Programming (Developer)

- **Classes & Objects:** A **Trigger Handler Pattern** is used for all triggers, and core business logic is encapsulated in service classes like **BillingEngineService.cls** and **FulfillmentService.cls**.
- **Apex Triggers:** A trigger on **Subscription\_\_c** prevents the cancellation of subscriptions with unpaid invoices, and a trigger on **Shipment\_\_c** automatically stamps the shipping address upon creation.
- **Trigger Design Pattern:** A scalable **handler pattern** is implemented for all triggers to keep logic out of the .trigger files, making the code clean, testable, and reusable.
- **SOQL & SOSL:** **SOQL** is used in all Apex classes and triggers to efficiently query for records such as active subscriptions, unpaid invoices, and product inventory.
- **Collections:** **Collections** (List, Set, Map) are used extensively to process records in bulk, ensuring the code is efficient and avoids hitting Salesforce governor limits.
- **Batch Apex:** A nightly **Batch Apex** job queries all **Shipment\_\_c** records with a status of 'Shipped', aggregates the product data, and efficiently decrements the **Quantity\_In\_Stock\_\_c** from the corresponding **Product\_Inventory\_\_c** records.
- **Scheduled Apex:** A **Scheduled Apex** class serves as the project's core billing engine, running daily to find all **Subscription\_\_c** records where the **Next\_Billing\_Date\_\_c** is today and creating a new **Invoice\_\_c** for each.
- **Queueable Apex:** After new invoices are created by the scheduled job, a **Queueable Apex** job is chained to handle the asynchronous API callout to an external payment gateway.
- **Exception Handling:** All asynchronous jobs and service class methods implement **try-catch blocks** to gracefully handle and log potential runtime errors, making the system more robust.

# Subscripify :- Subscription Box Service Management System

```
force-app > main > default > classes > BillingEngineService.cls > BillingEngineService > generateInvoices() : void
1 public class BillingEngineService {
2     public static void generateInvoices() {
3         try {
4             Date today = Date.today();
5             List<Subscription__c> dueSubscriptions = [
6                 SELECT Id, Subscriber__c, Price Override__c, Next_Billing_Date__c
7                 FROM Subscription__c
8                 WHERE Status__c = 'Active' AND Next_Billing_Date__c = :today
9             ];
10            if (dueSubscriptions.isEmpty()) {
11                return;
12            }
13            List<Invoice__c> newInvoices = new List<Invoice__c>();
14            List<Subscription__c> subscriptionsToUpdate = new List<Subscription__c>();
15            for (Subscription__c sub : dueSubscriptions) {
16                Invoice__c newInvoice = new Invoice__c();
17                newInvoice.Subscription__c = sub.Id;
18                newInvoice.State__c = 'Unpaid';
19                newInvoice.Date__c = today;
20                newInvoice.Due_Date__c = today.addDays(14);
21                newInvoice.Amount__c = (sub.Price_Override__c != null) ? sub.Price_Override__c : 999.00;
22                newInvoices.add(newInvoice);
23                Subscription__c updatedSub = new Subscription__c();
24                updatedSub.Id = sub.Id;
25                updatedSub.Next_Billing_Date__c = sub.Next_Billing_Date__c.addMonths(1);
26                subscriptionsToUpdate.add(updatedSub);
27            }
28            if (newInvoices.isEmpty()) {
29                insert newInvoices;
30            }
31            List<Id> newInvoiceIds = new List<Id>();
32            for (Invoice__c inv : newInvoices) {
33                newInvoiceIds.add(inv.Id);
34            }
35            System.enqueueJob(new ProcessPaymentQueueable(newInvoiceIds));
36            if (subscriptionsToUpdate.isEmpty()) {
37                update subscriptionsToUpdate;
38            }
39        } catch (Exception e) {
40            System.debug('An error occurred in the Billing Engine: ' + e.getMessage() + ' at line ' + e.getLineNumber());
41        }
42    }
43 }
```

```
force-app > main > default > classes > MonthlyBillingScheduler.cls > MonthlyBillingScheduler
1 public class MonthlyBillingScheduler implements Schedulable {
2
3     public void execute(SchedulableContext sc) {
4         BillingEngineService.generateInvoices();
5     }
6 }
```

```
force-app > main > default > triggers > SubscriptionTrigger.trigger > SubscriptionTrigger
1 trigger SubscriptionTrigger on Subscription__c (before update) {
2
3     if (Trigger.isBefore && Trigger.isUpdate) {
4         SubscriptionTriggerHandler.handleBeforeUpdate(Trigger.new, Trigger.oldMap);
5     }
6 }
```

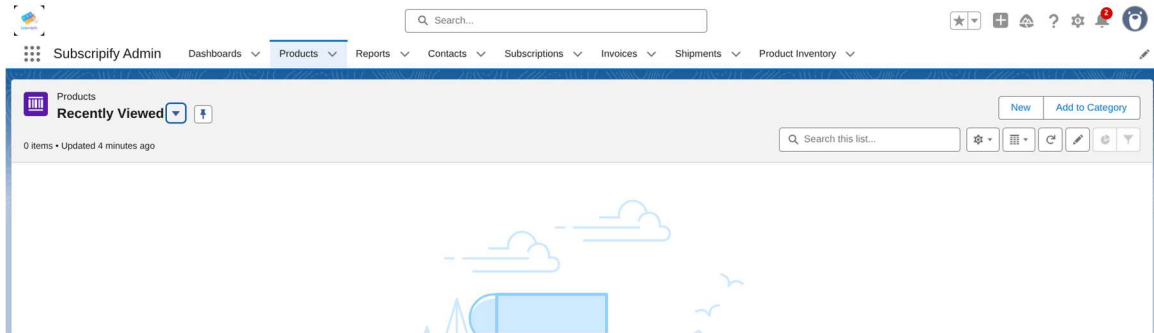
```
force-app > main > default > classes > FulfillmentService.cls > FulfillmentService > updateInventory(List<Shipment__c>) : void
1 public class FulfillmentService {
2     public static void updateInventory(List<Shipment__c> shipments) {
3         Set<Id> invoiceIds = new Set<Id>();
4         for (Shipment__c sh : shipments) {
5             invoiceIds.add(sh.Invoice__c);
6         }
7         Map<Id, Decimal> productQuantities = new Map<Id, Decimal>();
8         for (Invoice_Line__c ill : [
9             SELECT Product__c, Quantity__c
10            FROM Invoice_Line__c
11            WHERE Invoice__c IN :invoiceIds
12        ]) {
13            if (productQuantities.containsKey(ill.Product__c)) {
14                productQuantities.put(ill.Product__c, productQuantities.get(ill.Product__c) + ill.Quantity__c);
15            } else {
16                productQuantities.put(ill.Product__c, ill.Quantity__c);
17            }
18        }
19        if (productQuantities.isEmpty()) {
20            return;
21        }
22        List<Product_Inventory__c> inventoriesToUpdate = [
23            SELECT Id, Quantity_In_Stock__c, Product__c
24            FROM Product_Inventory__c
25            WHERE Product__c IN :productQuantities.keySet()
26        ];
27        for (Product_Inventory__c inv : inventoriesToUpdate) {
28            Decimal soldQuantity = productQuantities.get(inv.Product__c);
29            if (inv.Quantity_In_Stock__c != null) {
30                inv.Quantity_In_Stock__c = inv.Quantity_In_Stock__c - soldQuantity;
31            }
32        }
33        List<Shipment__c> shipmentsToUpdate = new List<Shipment__c>();
34        for (Shipment__c sh : shipments) {
35            shipmentsToUpdate.add(new Shipment__c(Id = sh.Id, Inventory_Processed__c = true));
36        }
37        update inventoriesToUpdate;
38        update shipmentsToUpdate;
39    }
40 }
```



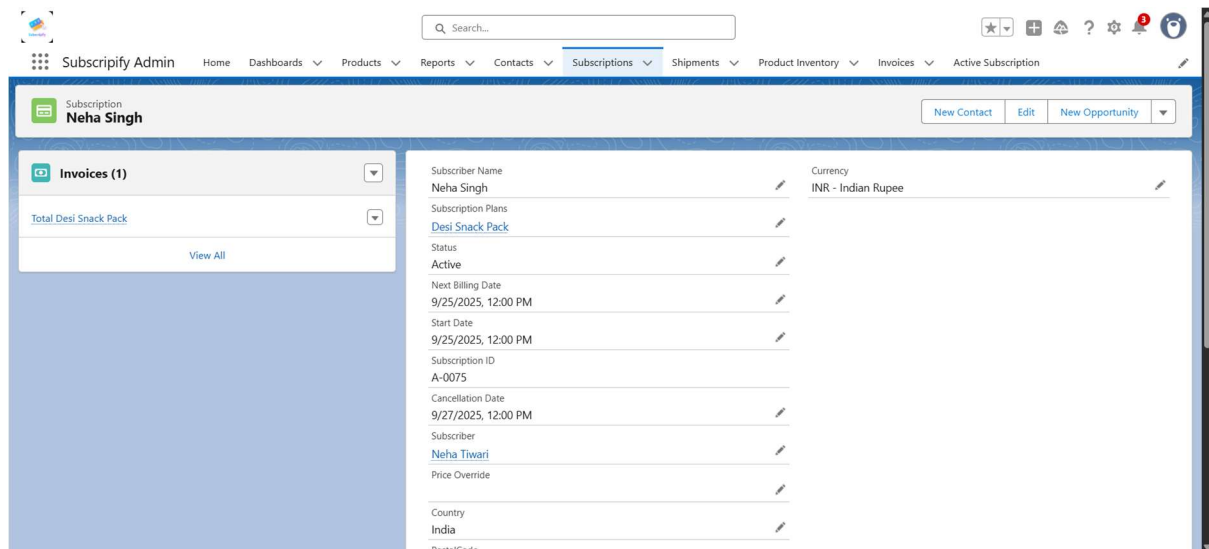
# Subscripify :- Subscription Box Service Management System

## Phase 6 :- User Interface Development

- **Lightning App Builder:** Created the Subscripify Admin Lightning App with tabs for Subscriptions, Invoices, Products, etc.



- **Record Pages:** Customized the Subscription\_\_c record page to include a component showing a history of all related invoices.

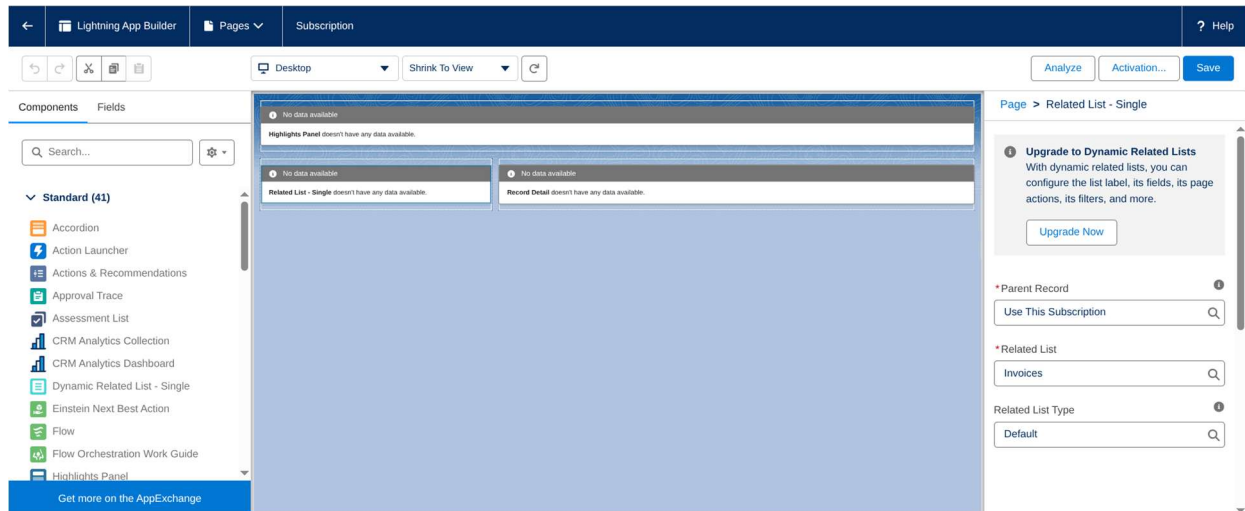


- **LWC (Lightning Web Components)**

- **subscriptionDetails:** This component was built to display core subscription information. It uses a @wire service to fetch data such as status and next billing date from a Salesforce record.
- **subscriptionActions:** This component provides key user functionality. It contains "Pause" and "Cancel" buttons that are wired to specific JavaScript methods.

# Subscripify :- Subscription Box Service Management System

- **addressManager:** This component handles user address information. It fetches and displays the user's shipping address, and includes an "Edit" button that switches the view to an editable form.
- **paymentHistory:** An additional component was created to display a user's past payment records in a clean lightning-datatable, enhancing the portal's functionality.



## • Apex with LWC

- **Data Retrieval:** Methods such as `getSubscriptionDetails`, `getAddress`, and `getPaymentHistory` are marked with `@AuraEnabled(cacheable=true)`. This annotation optimizes performance by caching the results on the client side, reducing server calls for read operations.
- **Data Manipulation:** Methods like `pauseSubscription`, `cancelSubscription`, and `saveAddress` are marked with `@AuraEnabled`. These methods are designed for updates and do not use caching, ensuring that DML operations are always performed on the most current data.

## • Imperative Apex Call

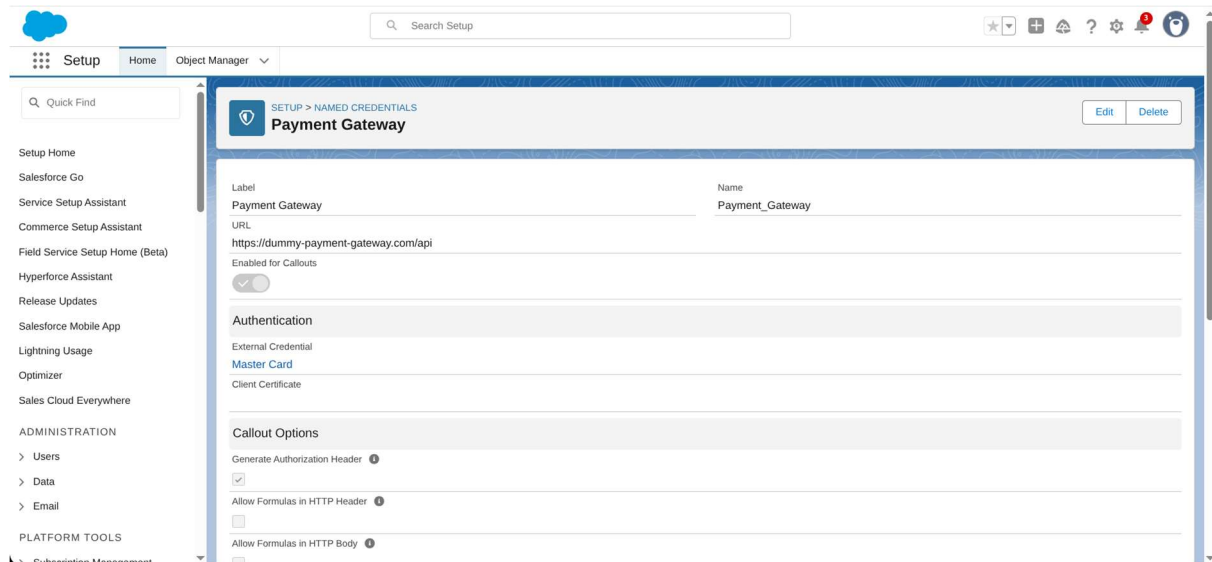
- When a user clicks one of these buttons, a JavaScript method is triggered.
- This method then makes an explicit call to the corresponding Apex method, such as `pauseSubscription` or `cancelSubscription`, passing any necessary parameters.
- This approach provides precise control over when the server-side action is executed, ensuring that the user's interaction directly results in a data update.

# Subscripify :- Subscription Box Service Management System

## Phase 7: Integration & External Access

- Named Credentials

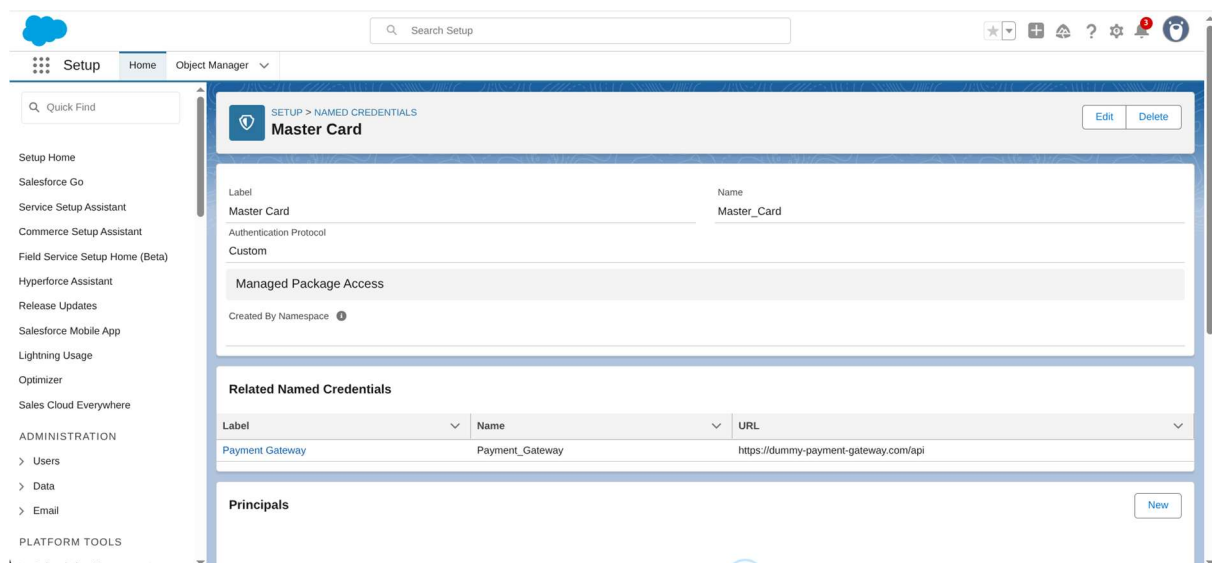
The `My_Payment_Gateway` Named Credential is now configured to reference a secure `External Credential`, keeping your API keys out of your code and maintaining best practices.



The screenshot shows the Salesforce Setup interface for a Named Credential. The left sidebar contains navigation links like 'Setup Home', 'Salesforce Go', and 'Administration'. The main content area is titled 'SETUP > NAMED CREDENTIALS' and 'Payment Gateway'. It includes fields for 'Label' (Payment Gateway), 'Name' (Payment\_Gateway), and 'URL' (https://dummy-payment-gateway.com/api). There are also checkboxes for 'Enabled for Callouts', 'Authentication' (External Credential, Master Card, Client Certificate), and 'Callout Options' (Generate Authorization Header, Allow Formulas in HTTP Header, Allow Formulas in HTTP Body).

- Web Services (REST/SOAP)

`createPaymentCharge`, designed to handle the REST callout to an external payment gateway. This method will be responsible for securely sending invoice details to create a payment charge, laying the groundwork for real-world payment processing.



The screenshot shows the Salesforce Setup interface for a Named Credential. The left sidebar contains navigation links like 'Setup Home', 'Salesforce Go', and 'Administration'. The main content area is titled 'SETUP > NAMED CREDENTIALS' and 'Master Card'. It includes fields for 'Label' (Master Card), 'Name' (Master\_Card), and 'Authentication Protocol' (Custom). There is a section for 'Managed Package Access' and a table for 'Related Named Credentials' with columns 'Label', 'Name', and 'URL'. The table shows a link to 'Payment Gateway' with name 'Payment\_Gateway' and URL 'https://dummy-payment-gateway.com/api'. There is also a 'Principals' section with a 'New' button.

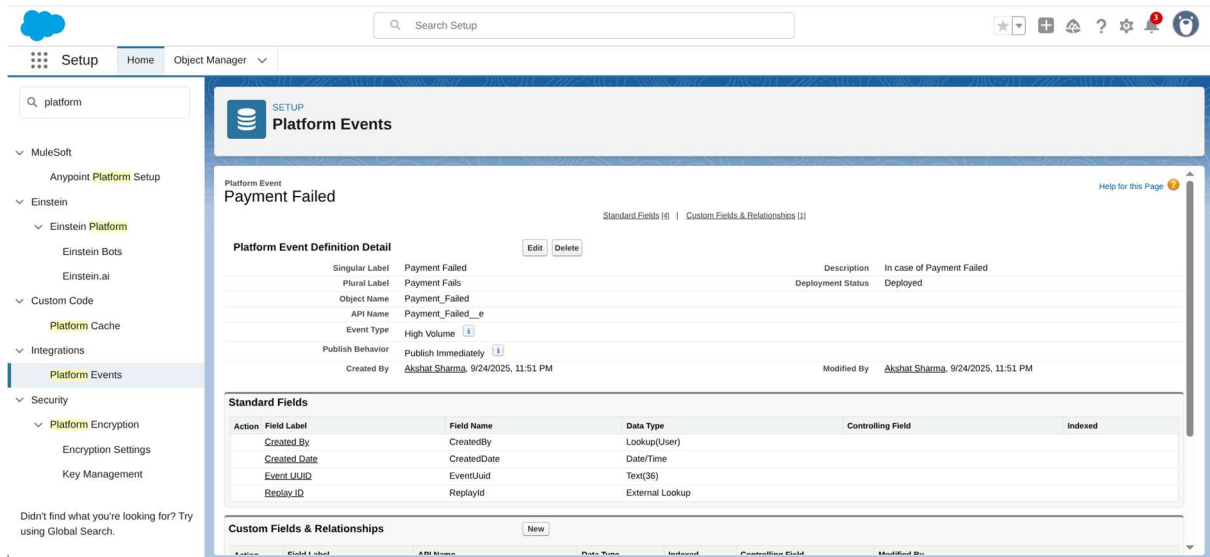
# Subscripify :- Subscription Box Service Management System

- **Callouts**

`@future(callout=true)`, which ensures that the callout runs asynchronously in a separate thread. This prevents your LWC from waiting for a response from the external service and avoids hitting governor limits, maintaining a smooth and responsive user experience.

- **Platform Events**

Created a **Platform Event**, `Payment_Failed__e`, in Salesforce org. This event provides a powerful and scalable way to handle asynchronous notifications. The billing engine will now be able to publish this event when a payment callout fails, allowing other systems, such as a Flow or a support notification system, to subscribe and react to the event without being directly tied to the billing logic.



The screenshot displays the Salesforce Setup interface for Platform Events. The left sidebar shows the navigation menu with 'Platform Events' selected under 'Integrations'. The main content area shows the configuration for the 'Payment Failed' event.

**Platform Event Definition Detail**

Field	Value
Singular Label	Payment Failed
Plural Label	Payment Fails
Object Name	Payment_Failed
API Name	Payment_Failed__e
Event Type	High Volume
Publish Behavior	Publish Immediately
Created By	Akshat Sharma
Modified By	Akshat Sharma

**Standard Fields**

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Event UUID	EventUuid	Text(36)		
	Replay ID	ReplayId	External Lookup		

**Custom Fields & Relationships**

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
--------	-------------	----------	-----------	---------	-------------------	-------------

# Subscripify :- Subscription Box Service Management System

## Phase 8: Data Management & Deployment

- **Data Import Wizard / Data Loader:**

Multiple tools were evaluated for the bulk upload process, including the Salesforce Data Loader (GUI and CLI for Linux), the SFDX Command Line Interface, the native Data Import Wizard, and Dataloader.io. The final import process follows a structured, three-step approach: (1) Import Contacts, (2) Import Products, (3) Use the Salesforce IDs from the first two imports to populate and import the linking Subscription records.

dataloader.io™

NEW TASK

ⓘ UPGRADE

Info

Help Center

akshatleader4174@agentforce.com

Quick find...

AllImportsExportsDeletesScheduled

History

⌵

All

⌵

✓

Subscription Insert

Task Run 103080212: 20 successes, 0 errors

last run: a few seconds ago

created on September 25th, 2025

✓

Product Insert

Task Run 103079310: 7 successes, 0 errors

last run: 39 minutes ago

created on September 25th, 2025

✓

Contact Insert

Task Run 103079213: 20 successes, 0 errors

last run: 42 minutes ago

created on September 25th, 2025

17 successes, 3 errors

4 minutes ago

15 successes, 5 errors

7 minutes ago

0 successes, 20 errors

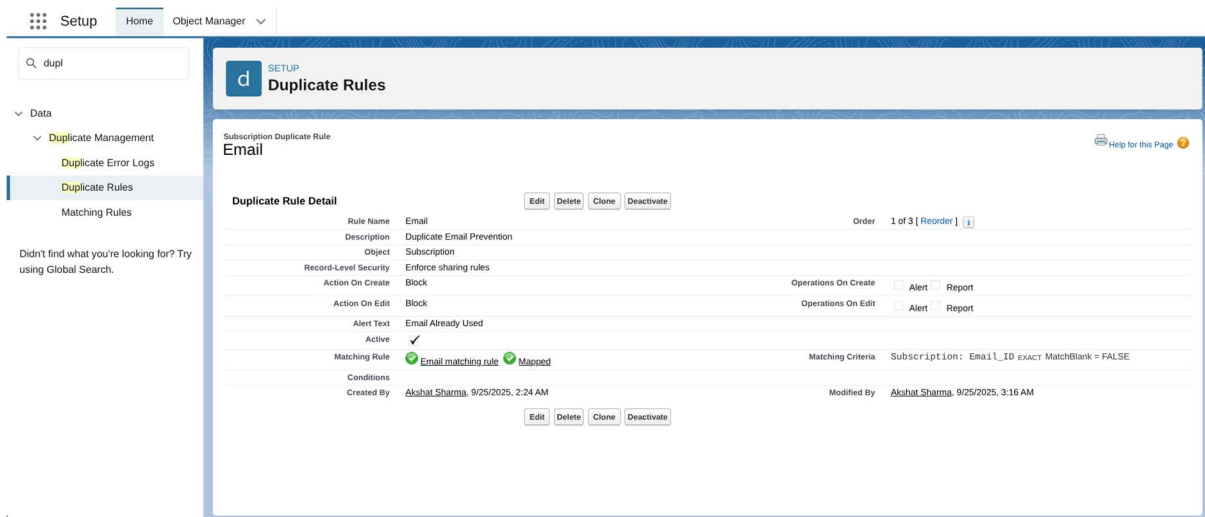
8 minutes ago

0 successes, 20 errors

10 minutes ago

- **Duplicate Rules:**

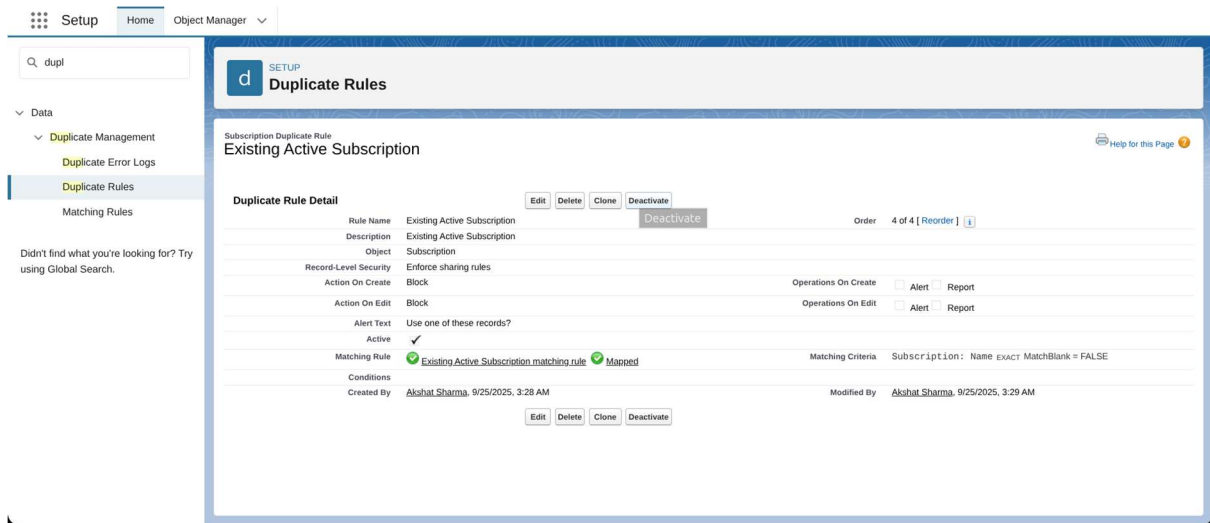
**For the Contact object**, two rules were activated to prevent duplicate subscriber records. The primary rule blocks the creation of a new contact if an exact match on the Email field is found.



Setup		Home	Object Manager
Duplicate Rules			
Subscription Duplicate Rule			
Email			
Duplicate Rule Detail			
Rule Name	Email	Order 1 of 3   Reorder	
Description	Duplicate Email Prevention		
Object	Subscription		
Record-Level Security	Enforce sharing rules		
Action On Create	Block	Operations On Create	Alert Report
Action On Edit	Block	Operations On Edit	Alert Report
Alert Text	Email Already Used		
Active	<input checked="" type="checkbox"/>		
Matching Rule	<input checked="" type="checkbox"/> Email matching rule <input checked="" type="checkbox"/> Mapred	Matching Criteria Subscription: Email_ID exact MatchBlank = FALSE	
Conditions			
Created By	Akshat Sharma, 9/25/2025, 2:24 AM	Modified By Akshat Sharma, 9/25/2025, 3:16 AM	

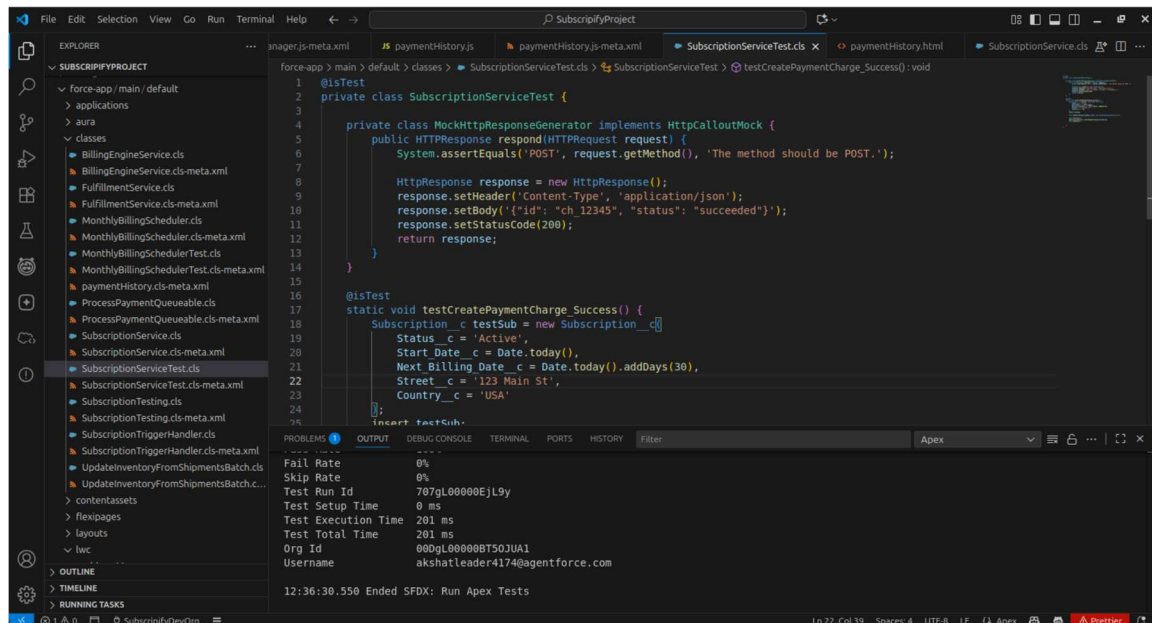
**For the custom Subscription\_\_c object**, a custom duplicate rule was designed to prevent critical business process errors. This rule blocks the creation of a new subscription if a contact already has an Active subscription for the exact same product.

# Subscripify :- Subscription Box Service Management System



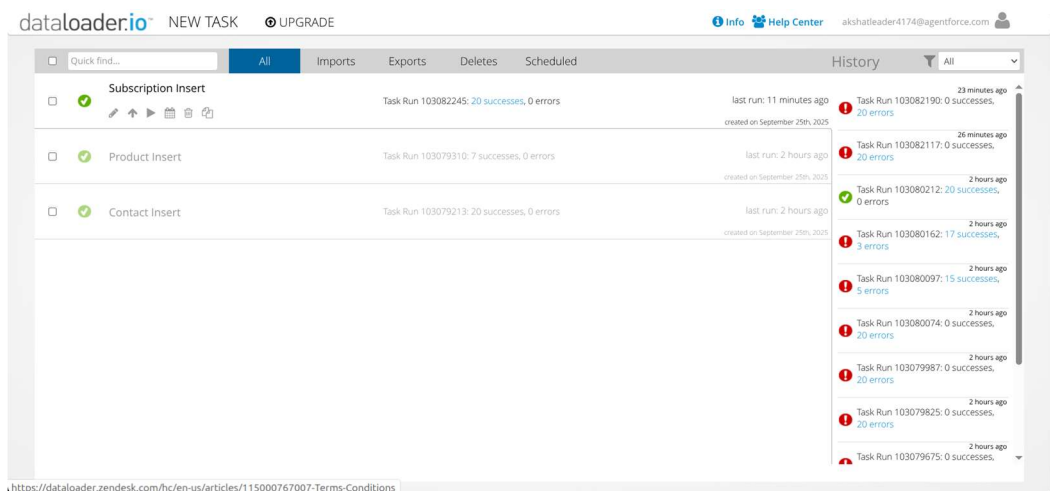
- **VS Code & SFDX**

- **Environment Setup & Stabilization:** A new SFDX project was created locally. Connections to the production (my-prod-org) and sandbox orgs were authorized. Initial command failures led to an environment diagnostic, which identified that the CLI was installed via npm.



# Subscripify :- Subscription Box Service Management System

- **Data Migration & Tooling:** The SFDX CLI was used for the initial data migration tasks. This involved a hands-on process of troubleshooting and resolving several common issues:
  - **File Path Errors:** Corrected data load commands to specify the correct path to CSV files (e.g., `./Downloads/contact.csv`) or by first navigating into the correct directory (`cd Downloads`).
  - **Command Workarounds:** Overcame CLI inconsistencies where the `insert` command failed. A robust workaround was implemented by creating a custom External ID field (`ContactExternalId__c`) on the Contact object. This allowed the use of the `sfdx force:data:bulk:upsert` command to reliably load new records.
  - **Alternative Tool Evaluation:** In response to CLI challenges, GUI-based alternatives were used. The native **Data Import Wizard** and the web-based **Dataloader.io** were successfully



- **Development & Deployment:** All custom metadata, including the `Subscription__c` object and its associated duplicate/matching rules, was developed locally in VS Code. After validation, this metadata was deployed to the production org using the stabilized SFDX CLI, ensuring a controlled and versioned release of the new feature.

# Subscripify :- Subscription Box Service Management System

- Testing

```
23:33:56.814 Starting SFDX: Run Apex Tests

=== Test Results
TEST NAME                                     OUTCOME
MESSAGE
RUNTIME (MS)

LightningSelfRegisterControllerTest.LightningSelfRegisterControllerInstantiation
Pass
32
LightningSelfRegisterControllerTest.testGetExtraFieldsInSelfRegistration
Pass
22
LightningSelfRegisterControllerTest.testGetNonEmptyExtraFields
Pass
12
LightningSelfRegisterControllerTest.testGetNullExtraFields
Pass
10
LightningSelfRegisterControllerTest.testIsValidPassword
Pass
3
LightningSelfRegisterControllerTest.testSelfRegisterWithCreatedAccount
Pass
305
```

```
SubscriptionServiceTest.testCreatePaymentCharge_Success Fail System.DmlException: Insert
CommunitiesLoginControllerTest.testCommunitiesLoginController
Pass
20
ChangePasswordControllerTest.testChangePasswordController
Pass
19
LightningLoginFormControllerTest.LightningLoginFormControllerInstantiation
Pass
26
LightningLoginFormControllerTest.testAuthConfig
Pass
16
LightningLoginFormControllerTest.testGetSelfRegistrationURL
Pass
7
LightningLoginFormControllerTest.testIsSelfRegistrationEnabled
Pass
8
LightningLoginFormControllerTest.testIsUsernamePasswordEnabled
Pass
31
ForgotPasswordControllerTest.testForgotPasswordController
Pass
```

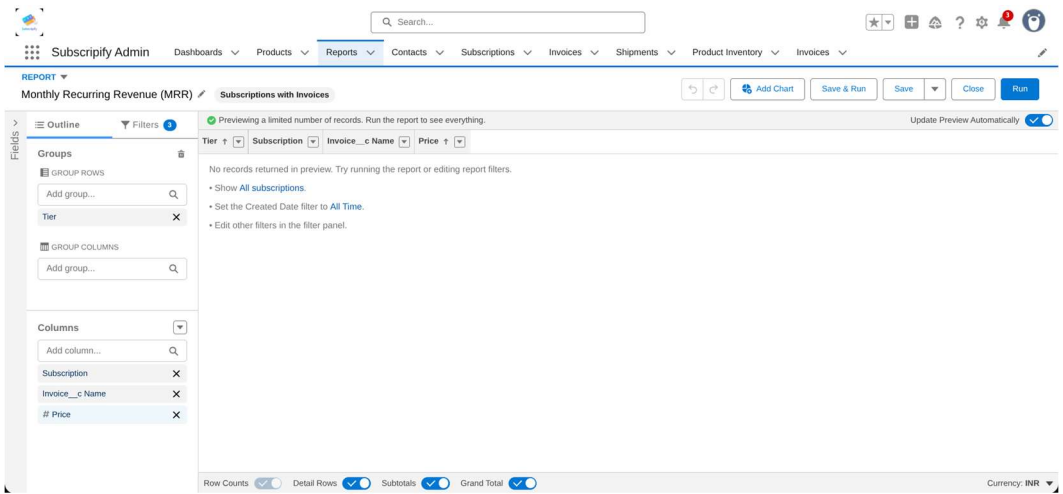


# Subscripify :- Subscription Box Service Management System

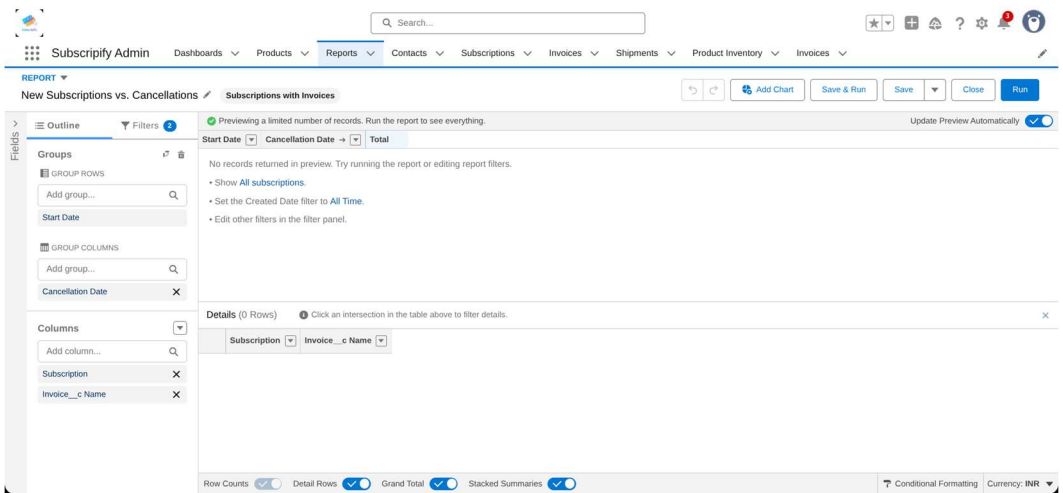
## Phase 9: Reporting, Dashboards & Security Review

### Reports & Dashboards

- Summary Report:** A "Monthly Recurring Revenue (MRR) by Subscription Tier" report was built. This report groups all active subscriptions by their assigned tier (e.g., Gold, Silver) and sums the recurring price, offering a clear, at-a-glance view of the company's core revenue streams.



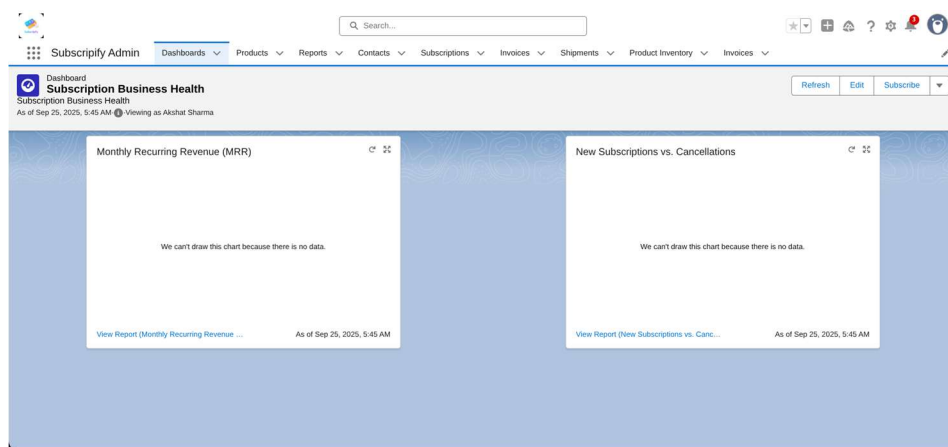
- Matrix Report:** A "New Subscriptions vs. Cancellations by Month" report was configured. This matrix provides a powerful visualization of subscriber churn and growth trends over time by comparing the volume of new subscriptions started against those canceled in any given month.



# Subscripify :- Subscription Box Service Management System

- **Dashboards:**

- A central "Subscription Business Health" dashboard was developed to serve as the primary command center. It contains key performance indicator (KPI) components, including gauges and metrics for total MRR, monthly Churn Rate.
- This dashboard was configured as a Dynamic Dashboard, allowing it to be viewed "As the logged-in user."



- **Sharing Settings & Field Level Security**

- **Organization-Wide Defaults (OWD):** The foundational sharing model was set to be restrictive. The OWD for both the custom `Subscription__c` and `Invoice__c` objects was configured to Private.

The screenshot shows the 'Sharing Settings' page in Salesforce. The page is titled 'Sharing Settings' and is viewed as 'Akshat Sharma' on 'Sep 25, 2025, 5:45 AM'. It displays a table of sharing settings for various objects. The table has three columns: 'Object', 'Sharing Model', and 'Controlled by Parent'. The 'Product Inventory' object is highlighted in blue.

Object	Sharing Model	Controlled by Parent
Tableau Host Mapping	Public Read Only	Private
User Presence	Public Read Only	Private
User Provisioning Request	Private	Private
Waitlist	Private	Private
Web Cart Document	Private	Private
Work Order	Private	Private
Work Plan	Private	Private
Work Plan Template	Private	Private
Work Step Template	Private	Private
Work Type	Private	Private
Work Type Group	Public Read/Write	Private
Invoice	Private	Private
Invoice Line	Controlled by Parent	Controlled by Parent
Payment	Public Read/Write	Private
Product Inventory	Public Read/Write	Private
Shipment	Public Read/Write	Private
Subscription	Private	Private

Below the table, there is a section for 'User Visibility Settings' with a 'Portal User Visibility' checkbox and a 'User Visibility Settings Help' link.

# Subscripify :- Subscription Box Service Management System

- **Login IP Ranges:** To enhance system-level security, logins for internal users have been restricted to a specific corporate IP address range. This ensures that employees and managers can only access the Salesforce environment when they are connected to the company's trusted network, preventing unauthorized access from external locations.

