**TCS LAST MILE CAPSTONE PROJECT**

**Project Title: Smart Customer Retention & Loyalty Management System**

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

**🎯 Requirement Gathering**

**Objective:**  
To identify customer retention and loyalty challenges and specify the system features required for addressing them.

**Approach:**

* Conducted stakeholder interviews and discussions (sales, marketing, support, finance) to capture current issues in retention and loyalty management.
* Reviewed historical data such as churn rates, customer lifetime value, and loyalty adoption.
* Identified essential features: churn prediction, loyalty tiering, personalized outreach, campaign management, and ROI measurement.

**Deliverables:**

* Requirement log with business, user, and system requirements.
* Workflow diagrams showing current vs. proposed processes.
* Use case scenarios for key user roles (manager, agent, customer).

**Observation:**  
Major gaps were identified in delayed churn detection and manual customer outreach. Automation and predictive analytics are necessary improvements.

**🧑‍🤝‍🧑 Stakeholder Analysis**

**Objective:**  
To identify stakeholders, their interests, and their roles in customer retention and loyalty management.

**Method:**

* Internal stakeholders: CSMs, Sales Leads, Marketing Team, Data Analytics, IT, Executives.
* External stakeholders: Customers, integration partners, and loyalty platforms.
* Stakeholder needs: CSMs expect proactive churn alerts, executives focus on ROI, customers expect transparent and rewarding programs.
* RACI chart planned to define responsibilities.

**Deliverables:**

* Stakeholder matrix with influence and engagement levels.
* RACI chart for project activities.

**Observation:**  
Different stakeholder groups prioritize different outcomes. Customers focus on rewards, while managers and executives expect measurable business results.

**🗺️ Business Process Mapping**

**Objective:**  
To analyze current customer retention and loyalty workflows and identify areas for improvement.

**Tasks:**

* Mapped customer journey from onboarding to churn.
* Documented loyalty program processes: enrollment, earning, redemption, and targeting.
* Identified issues: manual outreach, lack of segmentation, delayed churn response.
* Designed improved workflows including automation, AI, and real-time insights.

**Deliverables:**

* Swimlane diagrams comparing current vs. improved workflows.
* Process documentation with steps, owners, and data hand-offs.

**Observation:**  
Most inefficiencies were linked to manual processes and lack of proactive mechanisms. Automation can streamline and improve retention.

**📚 Industry-Specific Use Case Analysis**

**Objective:**  
To benchmark customer retention and loyalty practices across industries.

**Method:**

* Reviewed case studies from e-commerce, telecom, SaaS, and retail.
* Best practices identified: predictive churn scoring, gamification, tiered loyalty rewards, and proactive engagement.
* Features mapped to system components.
* Differentiated features for MVP and future expansion.

**Deliverables:**

* Literature review of benchmark programs.
* Best-practice mapping to system features.
* Recommendations for MVP vs. future scope.

**Observation:**  
Gamification and personalization are common success factors in loyalty programs and should be considered core features.

**🛍️ AppExchange Exploration**

**Objective:**  
To explore Salesforce AppExchange solutions relevant to retention and loyalty management.

**Tasks:**

* Researched available apps for churn prediction, loyalty management, and marketing automation.
* Evaluated tools such as Einstein Analytics and survey platforms.
* Determined areas suitable for reuse versus custom development.
* Planned sandbox testing and integration requirements.

**Deliverables:**

* AppExchange feature evaluation matrix.
* Recommendation report for adoption/customization.
* Integration plan with APIs or connectors.

**Observation:**  
Salesforce ecosystem already provides many accelerators that reduce the need for complete custom development.

**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 2: Salesforce Org Setup & Configuration**

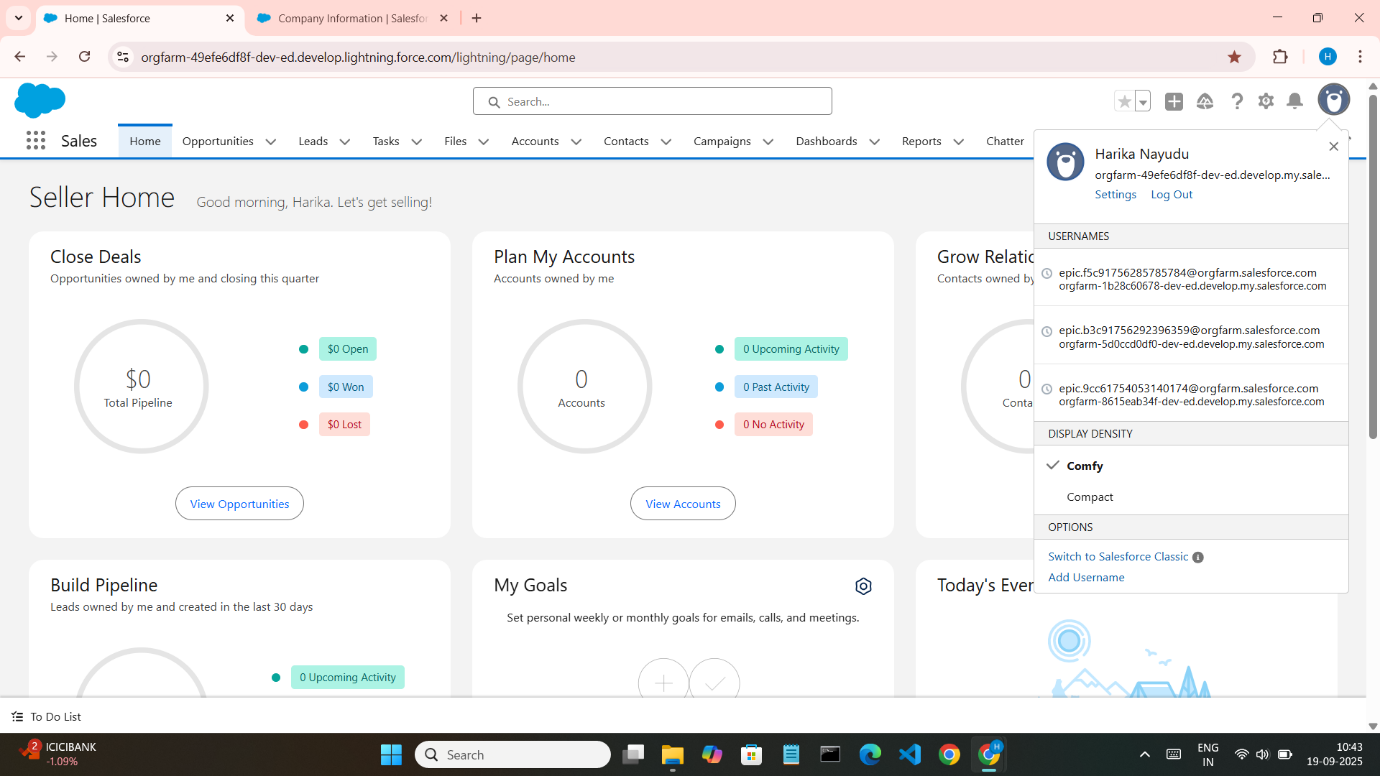
**Step 1: Sign Up for Salesforce Developer Edition**

1. Go to 👉 <https://developer.salesforce.com/signup>.
2. Fill the form:
   * **First Name / Last Name** → Enter your details
   * **Email** → Valid email (you will get verification)
   * **Job Title** → Developer / Student
   * **Company** → College name
   * **Country** → Select your country
3. Click **Sign Me Up**.
4. Check your email → Click the **verification link**.

**Step 2: Log in to Salesforce Developer Org**

1. Once verified, go to:  
    🔗 https://login.salesforce.com
2. Enter your registered email and the password you set.

You will be redirected to the **Salesforce Lightning Experience Dashboard**

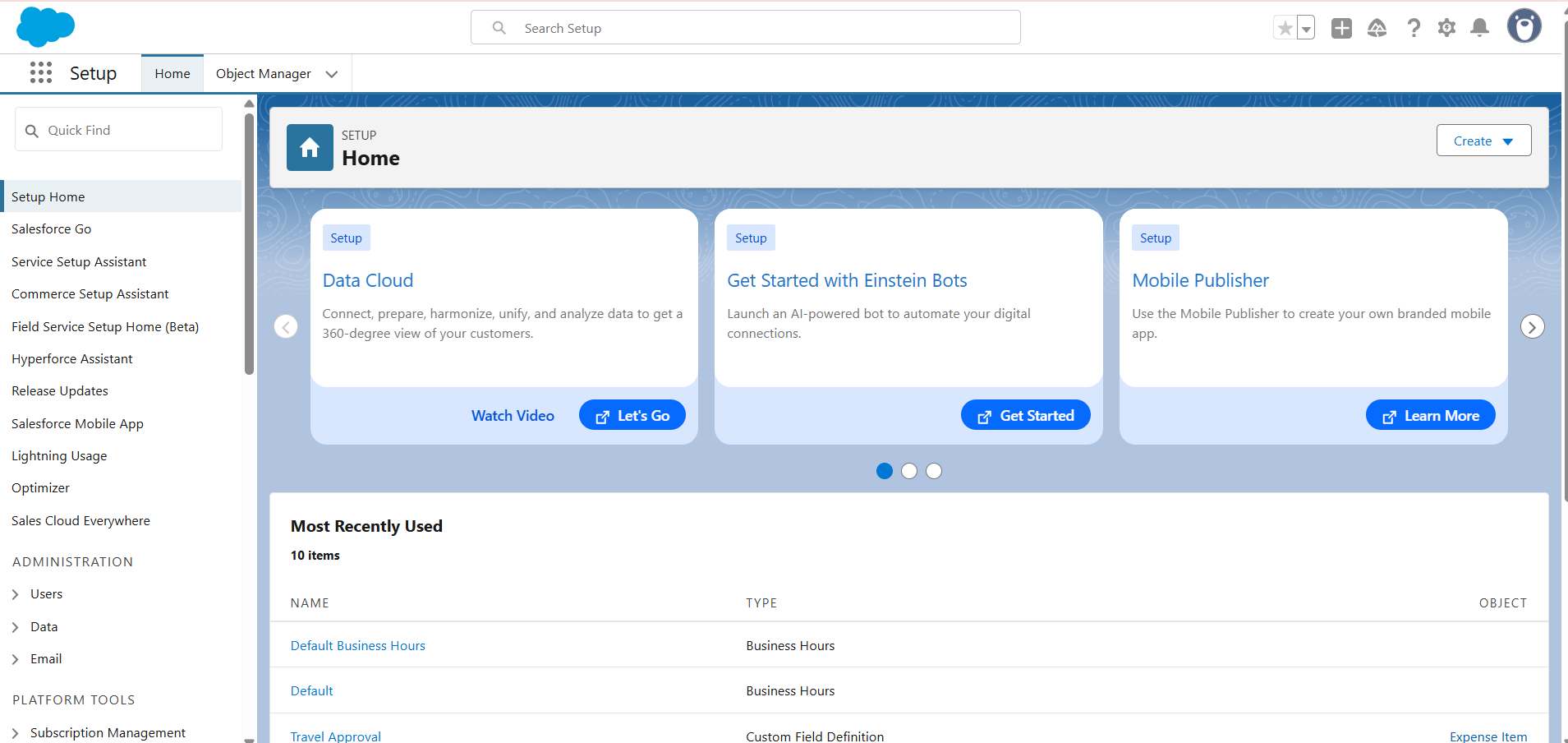


**Step 3: Explore the Org Setup**

Now that you're inside Salesforce:

1. Click on the **gear icon (⚙️)** on the top-right corner.
2. Select **“Setup”** from the dropdown.

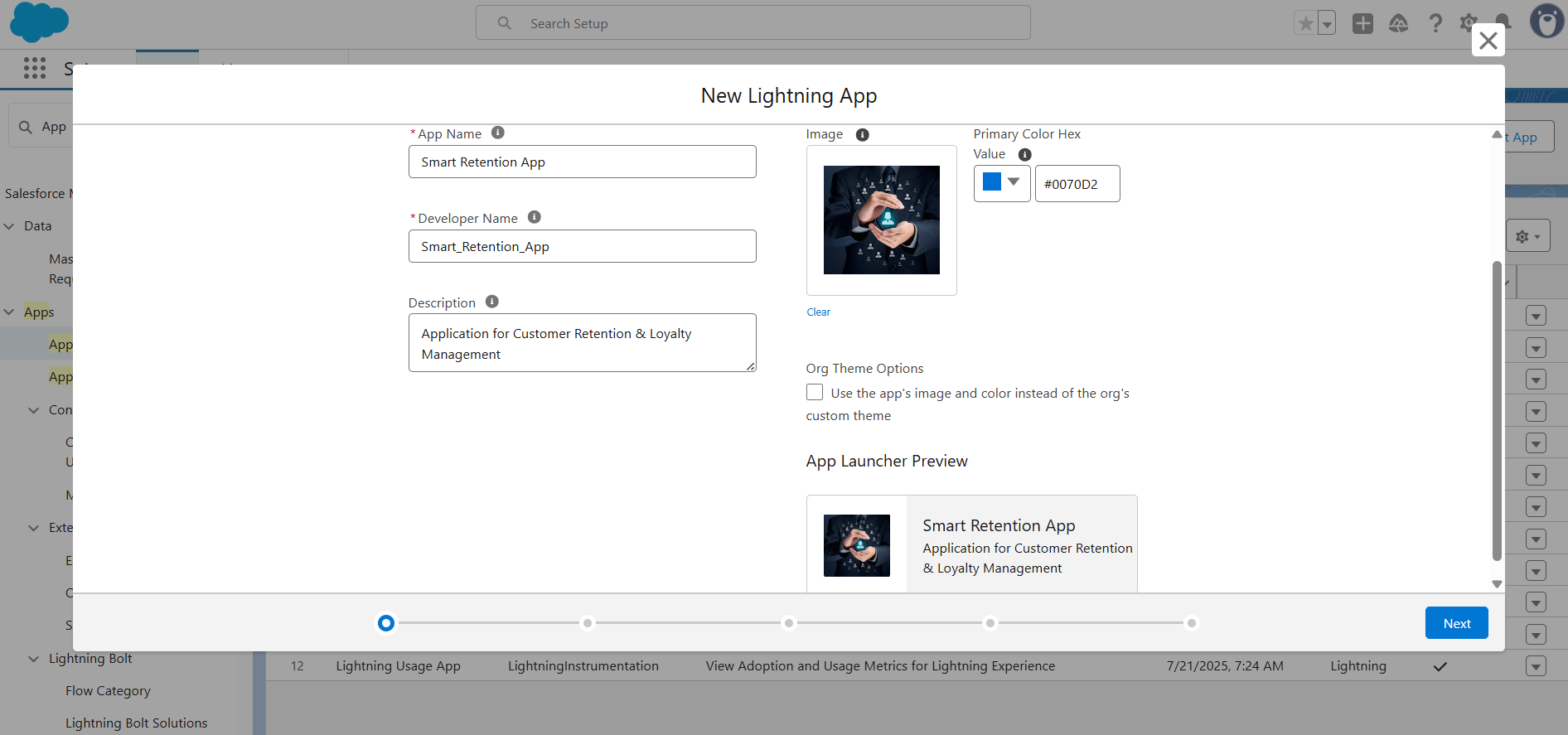
This will take you to the **Setup Home**, where you can manage and configure your org.

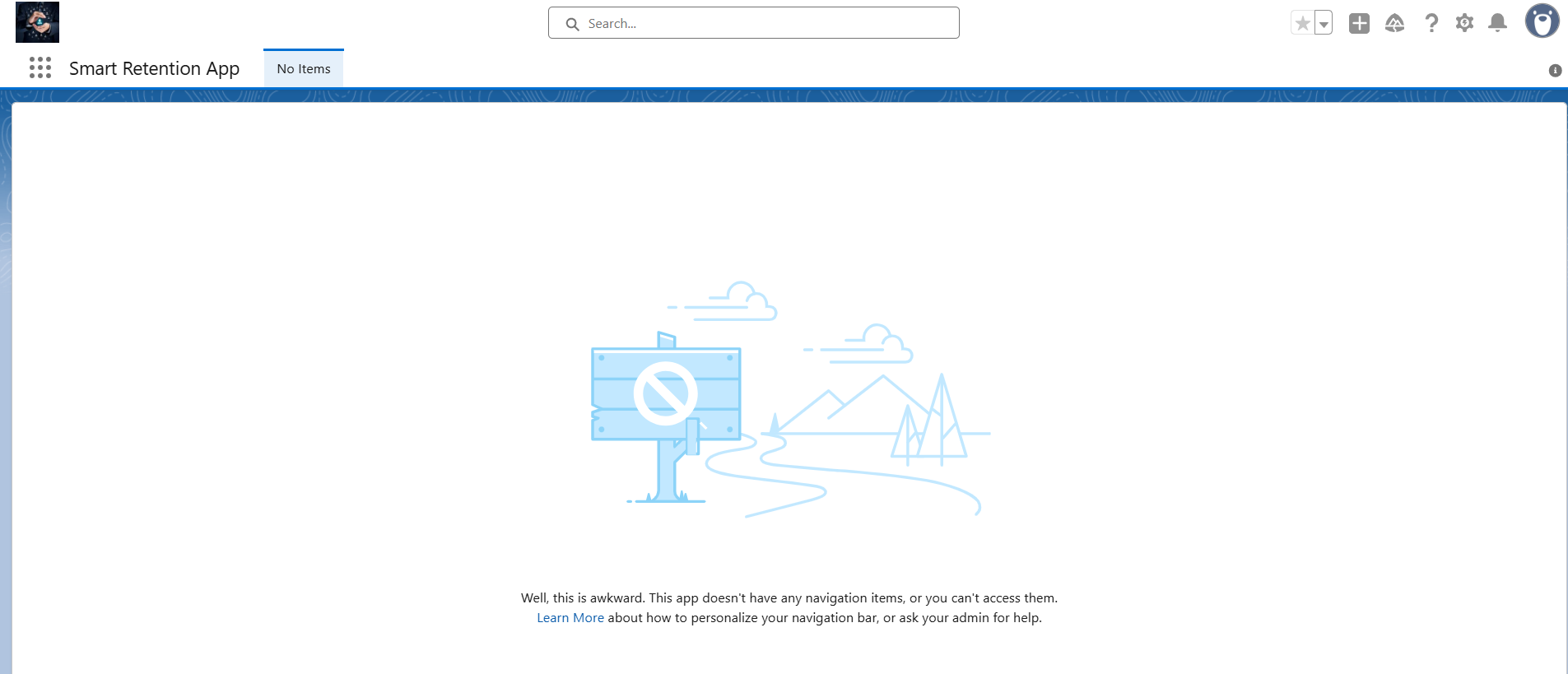


**Step 4: Perform Configuration Steps**

**4.1 Create a New Lightning App**

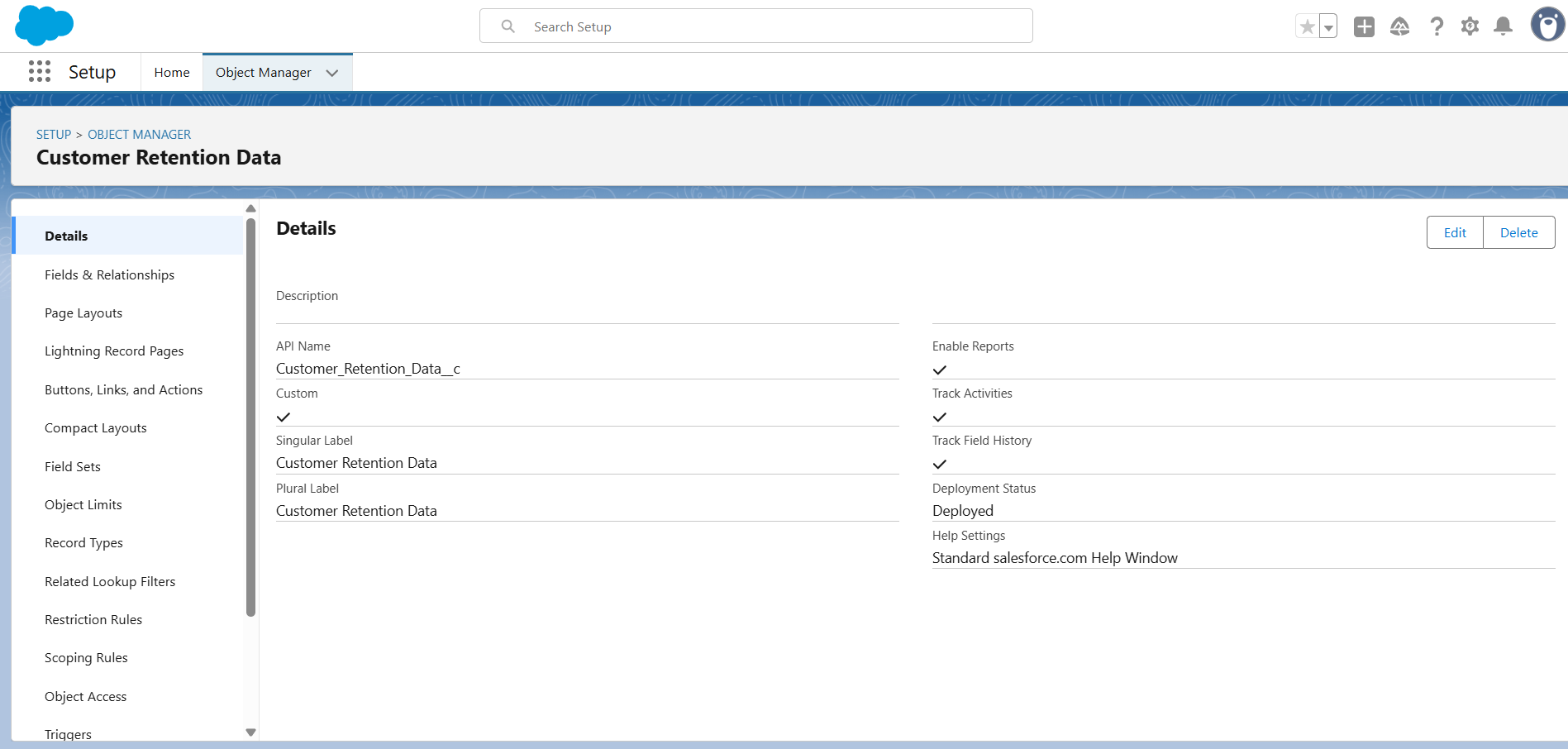
1. In the Setup, left sidebar **Quick Find** search box, type **App Manager** → click it.
2. On the App Manager page, click **New Lightning App**.
3. Fill details:
   * **App Name:** Smart Retention App
   * **Developer Name:** auto-fills
   * **Description:** Application for Customer Retention & Loyalty Management
4. Click **Next**.
5. Keep **Standard Navigation** selected → Click **Next**.
6. Choose a logo/icon (optional) → Click **Next**.
7. Assign the app to the **System Administrator profile** (for now).
8. Click **Save & Finish**.





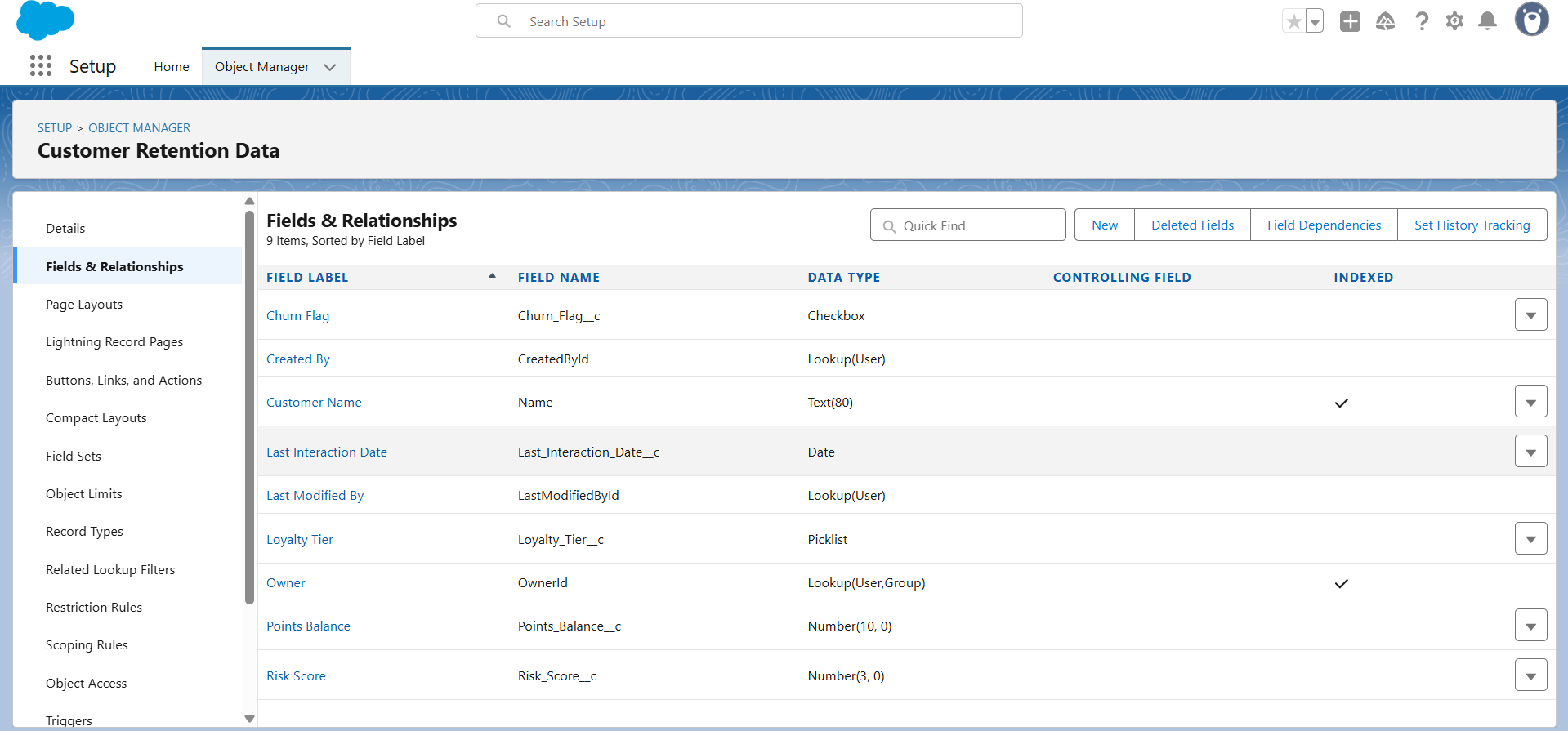
**4.2 Create a Custom Object**

1. In Setup → Quick Find → type **Object Manager** → click it.
2. Click **Create** (top-right) → **Custom Object**.
3. Enter:
   * **Label:** Customer Retention Data
   * **Plural Label:** Customer Retention Data
   * **Object Name:** auto-fills
   * **Record Name:** Customer Name (Data Type: Text)
   * Check these boxes:
     + Allow Reports
     + Allow Activities
     + Track Field History (optional)
4. Click **Save**.



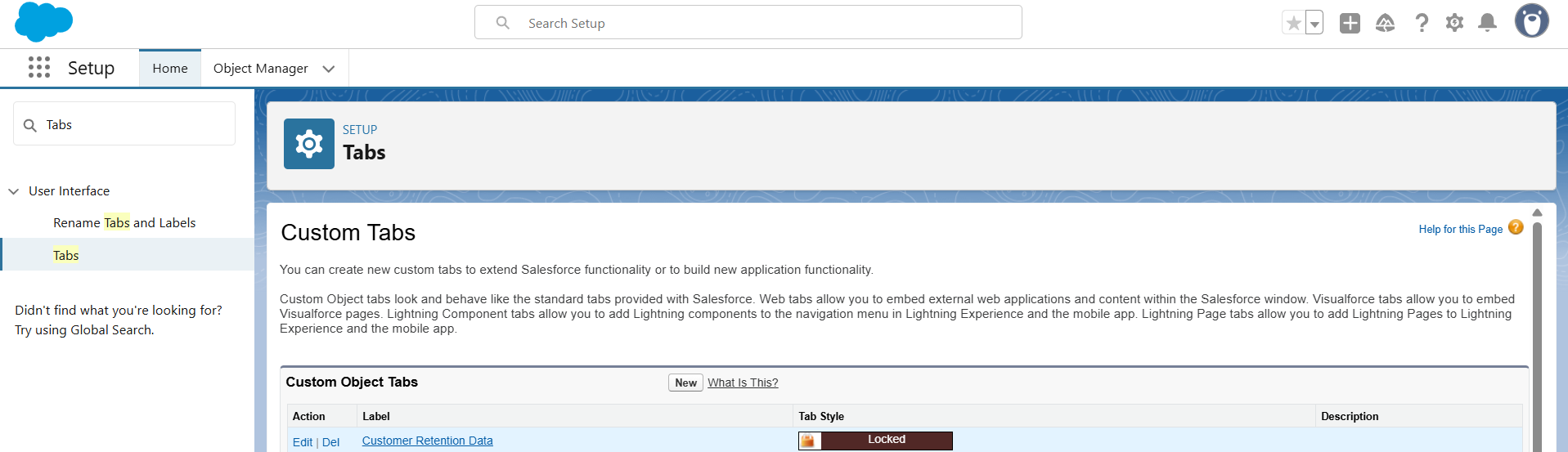
**4.3 Add Custom Fields**

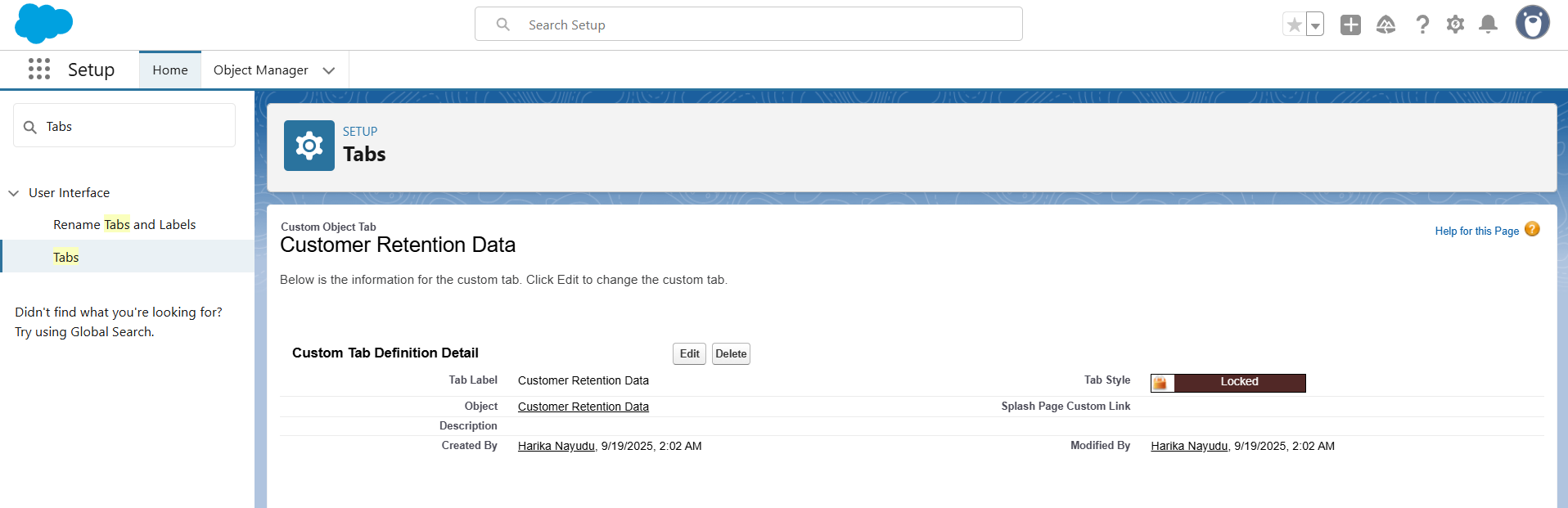
1. From Object Manager → click **Customer Retention Data** → **Fields & Relationships** → **New**.
2. Create these fields one by one:
   * **Risk Score** → Data Type: Number → Length: 3, Decimal: 0
   * **Loyalty Tier** → Data Type: Picklist → Values: Bronze, Silver, Gold, Platinum
   * **Points Balance** → Data Type: Number → Length: 10, Decimal: 0
   * **Last Interaction Date** → Data Type: Date
   * **Churn Flag** → Data Type: Checkbox → Default Value: Unchecked
3. Save after each field.



**4.4 Create a Tab for Custom Object**

1. Setup → Quick Find → type **Tabs** → click it.
2. Under **Custom Object Tabs**, click **New**.
3. Select **Customer Retention Data** from the dropdown.
4. Choose an icon (any).
5. Click **Next** → Select **Smart Retention App** → Save.

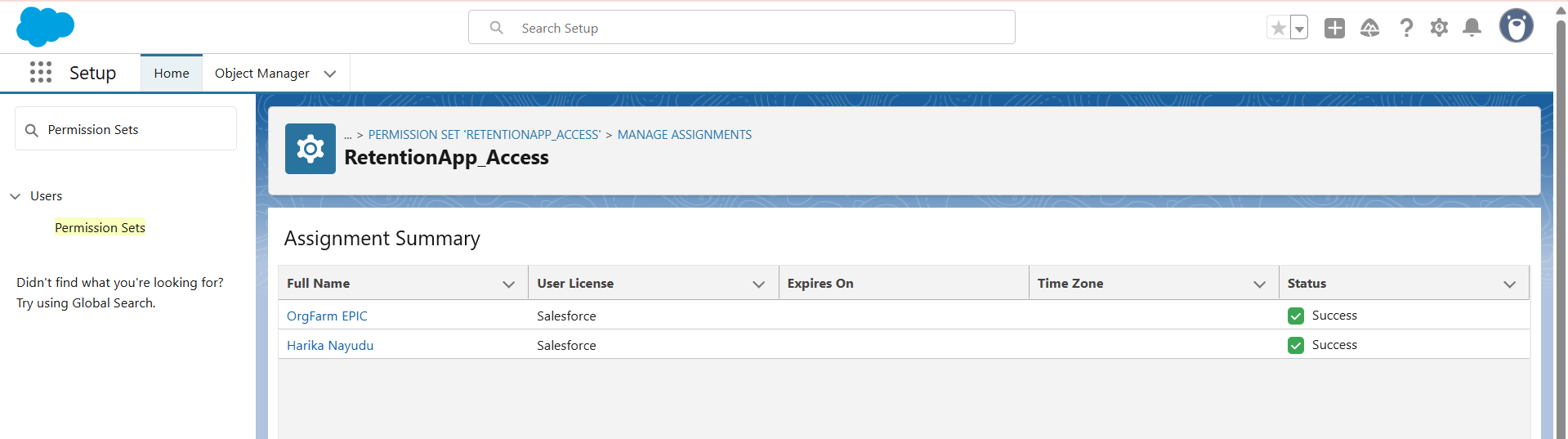




**4.5 Assign Profiles/Permission Sets**

1. Setup → Quick Find → type **Permission Sets** → New.
   * Label: RetentionApp\_Access
   * Save.
2. Open **RetentionApp\_Access** → Manage Assignments → Add Assignments.
3. Select your test users (e.g., Customer Success Manager) → Save.





**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 3: Data Modeling & Relationships**

**Step 1: Standard & Custom Objects**

**Standard Objects (already available)**

* **Account** → represents a company or organization.
* **Contact** → represents individual customers belonging to Accounts.

👉 You don’t need to create these; just know you will link them later.

**Create New Custom Objects**

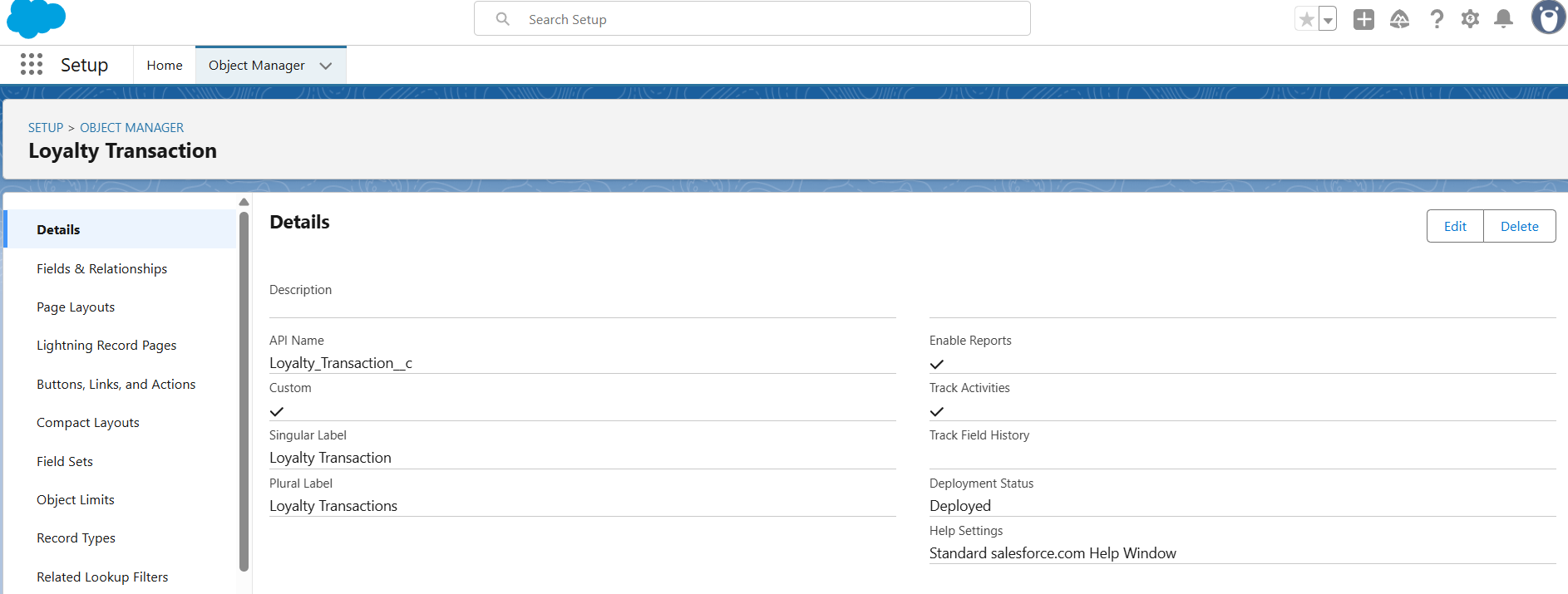
**A. Loyalty Transaction**

1. Click **⚙ Setup** → **Object Manager** → **Create** → **Custom Object**.
2. Fill details:
   * Label: Loyalty Transaction
   * Plural Label: Loyalty Transactions
   * API Name: Loyalty\_Transaction\_\_c
   * Record Name: Transaction ID → Data Type: **Auto Number**
   * Format: TX-{0000}
   * Allow Reports: ✅ Allow Activities: ✅ Track Field History: ✅
3. Save.

**B. Engagement Activity**

1. Setup → Object Manager → New Custom Object.
2. Fill details:
   * Label: Engagement Activity
   * Plural Label: Engagement Activities
   * API Name: Engagement\_Activity\_\_c
   * Record Name: Activity Name (Text).
   * Allow Reports: ✅ Allow Activities: ✅
3. Save.

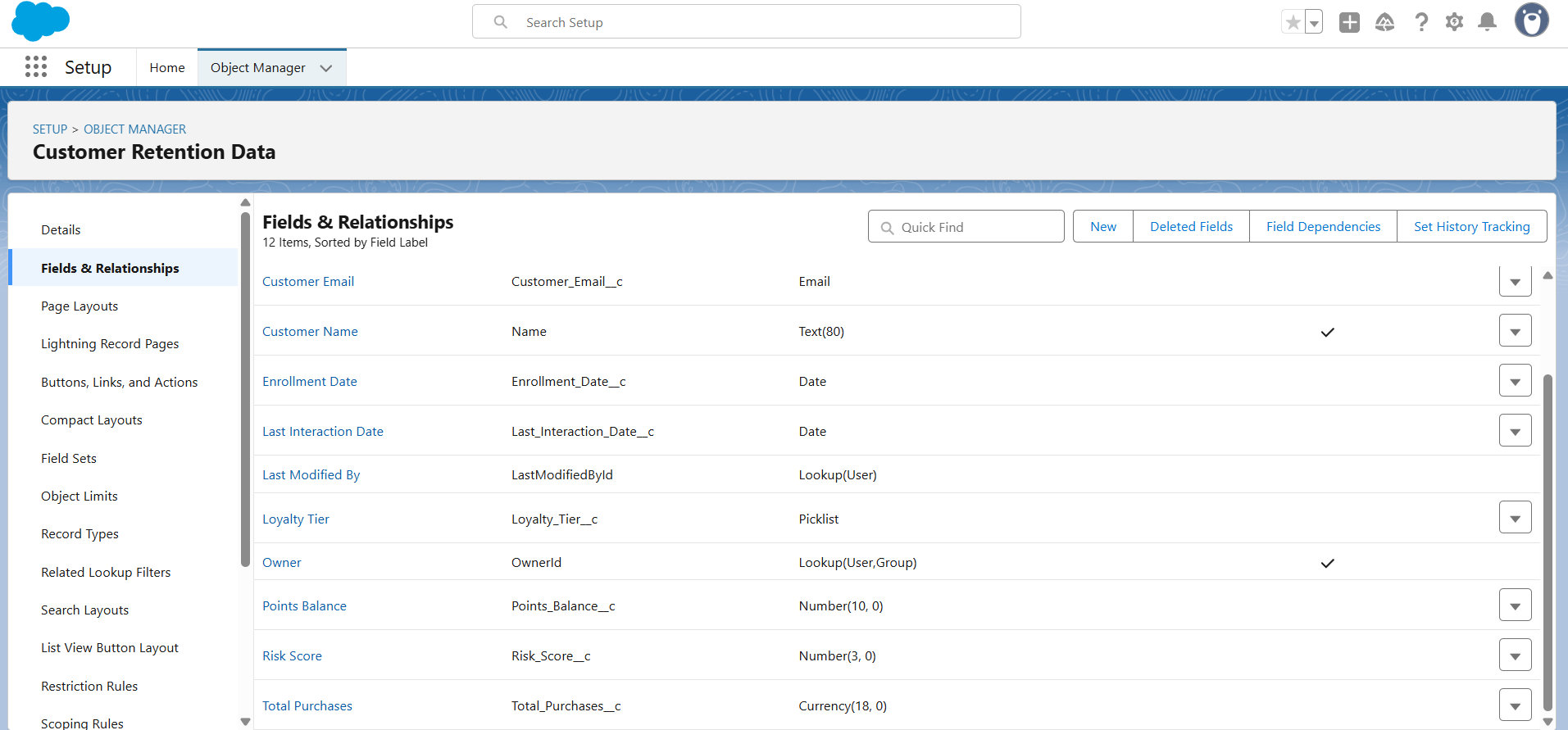




**Step 2: Fields**

**A. On Customer Retention Data**

1. Go to **Object Manager → Customer Retention Data → Fields & Relationships → New**.
2. Create these fields:
   * **Customer Email** → Data Type: Email.
   * **Enrollment Date** → Data Type: Date.
   * **Total Purchases** → Data Type: Currency.



**B. On Loyalty Transaction**

1. Object Manager → Loyalty Transaction → Fields & Relationships → New.
2. Create:
   * **Transaction Type** → Picklist (Credit, Debit).
   * **Points Changed** → Number.
   * **Transaction Date** → Date/Time.



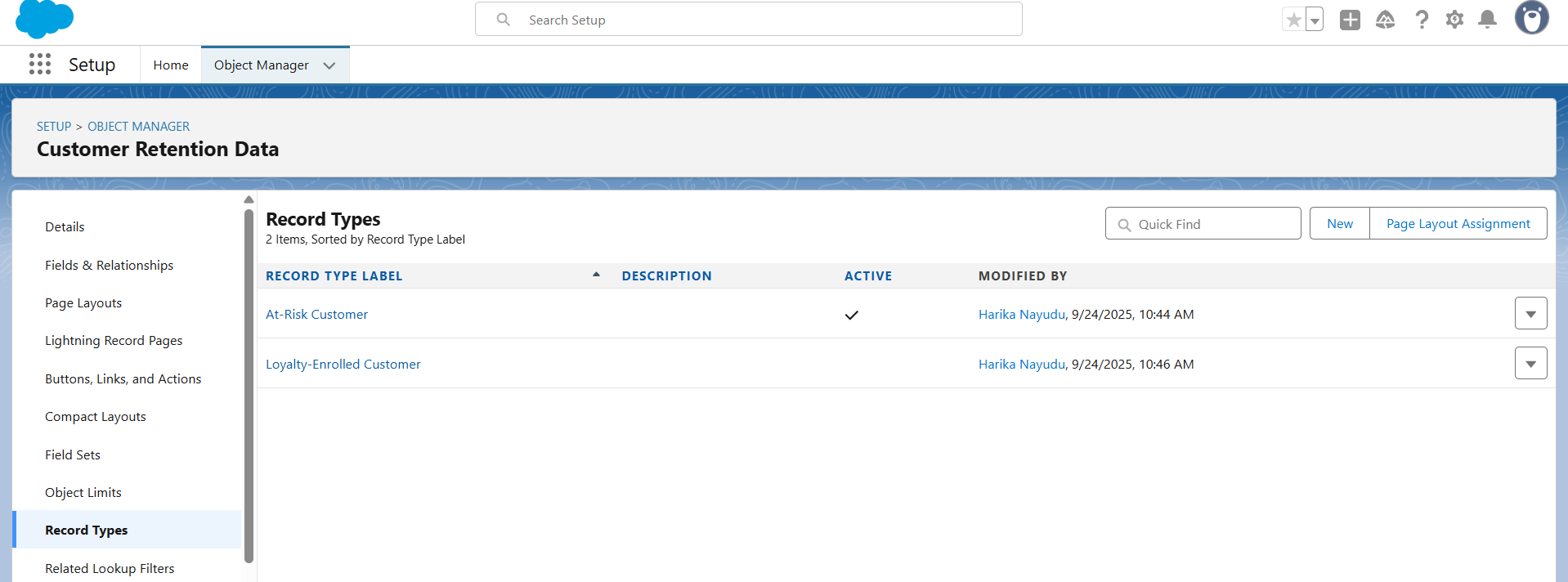
**C. On Engagement Activity**

1. Object Manager → Engagement Activity → Fields & Relationships → New.
2. Create:
   * **Activity Type** → Picklist (Email, Call, Meeting, Survey).
   * **Activity Date** → Date/Time.
   * **Notes** → Long Text Area.



**Step 3: Record Types**

1. Setup → Object Manager → Customer Retention Data → **Record Types → New**.
2. Create:
   * Record Type Label: At-Risk Customer → API Name: At\_Risk\_Customer.
   * Assign to profiles (System Admin, CSM).
3. Create another:
   * Label: Loyalty-Enrolled Customer → API Name: Loyalty\_Enrolled\_Customer.
   * Assign to profiles.

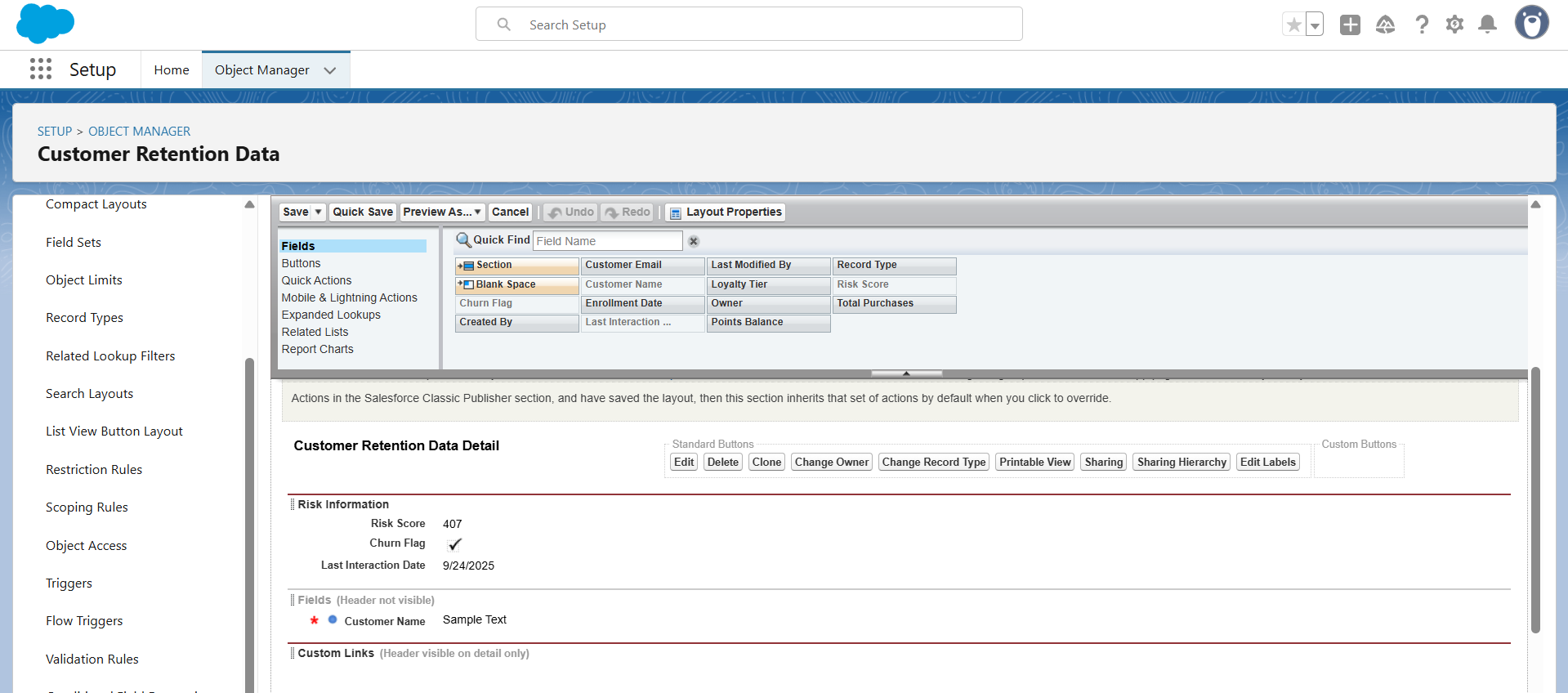


**🔹 Step 4: Page Layouts**

1. Go to **Object Manager → Customer Retention Data → Page Layouts**.
2. Create two layouts:

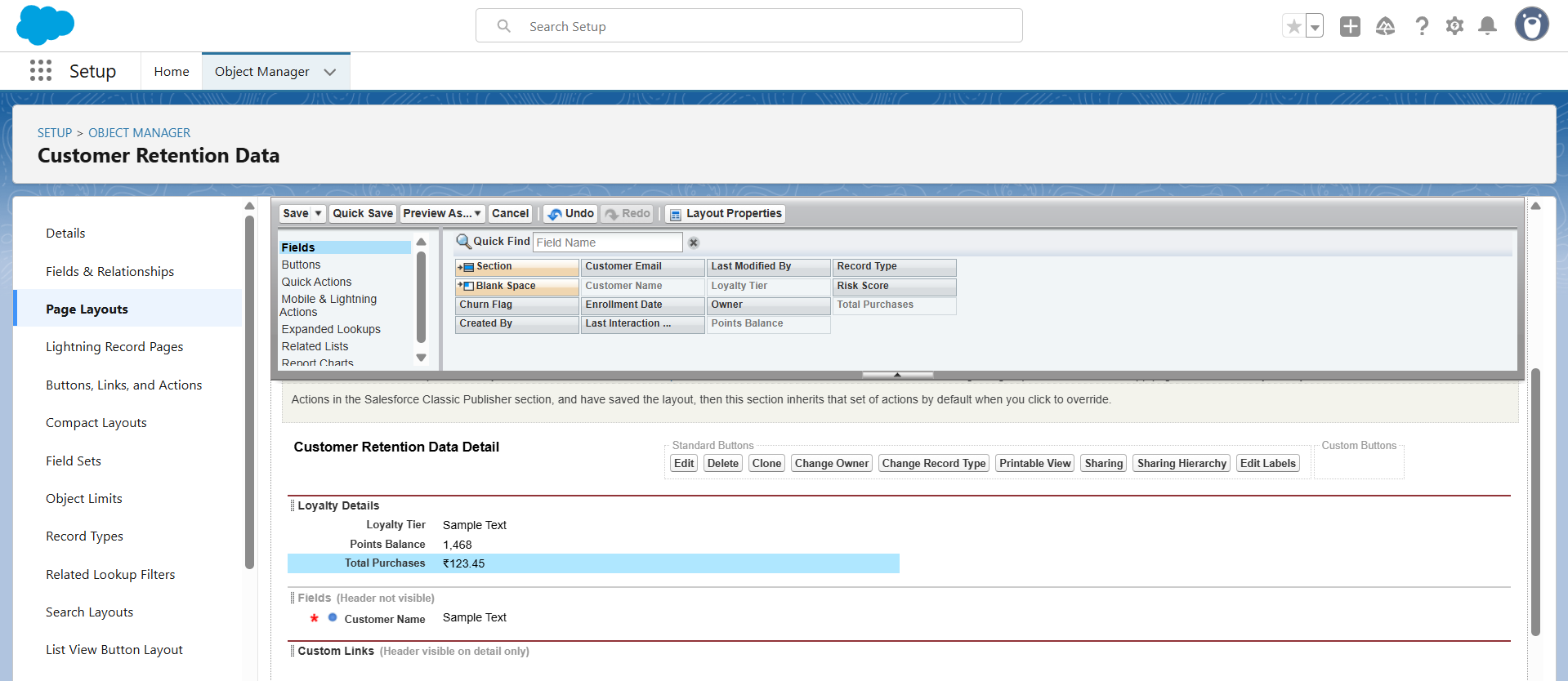
**At-Risk Layout**

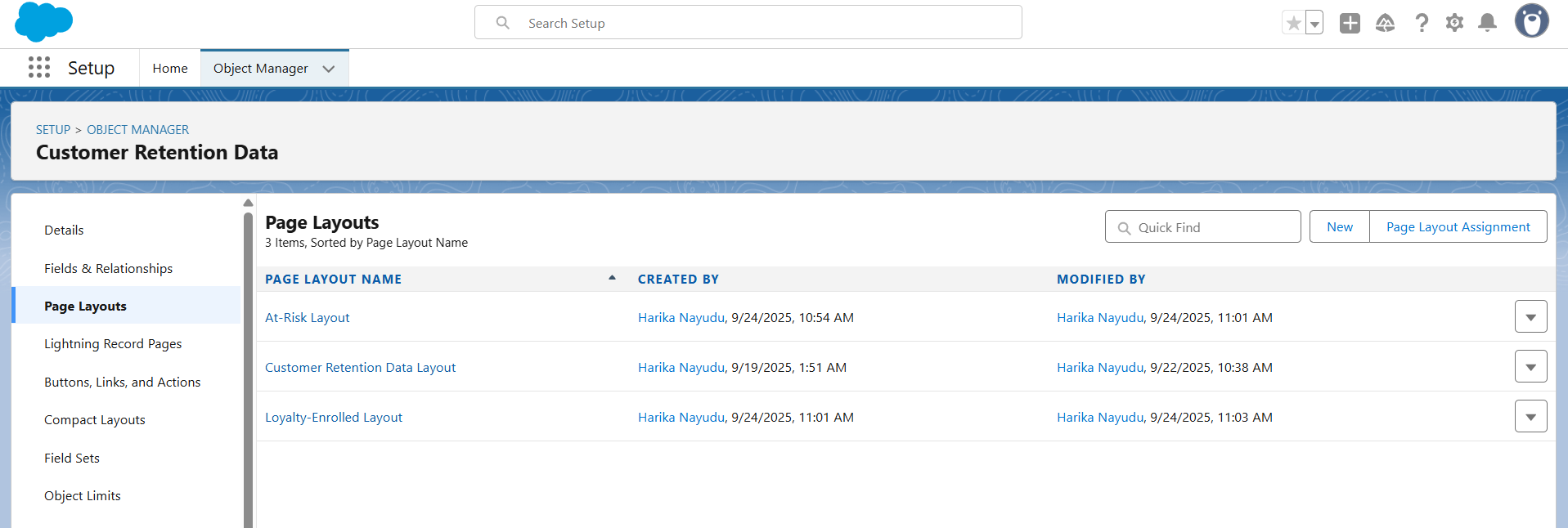
1. Click **New Page Layout**
2. Name it: At-Risk Layout.
3. In the layout editor:
   * Drag fields like **Risk Score**, **Churn Flag**, and **Last Interaction Date** into the section called **Risk Information**.
4. Save.



**Loyalty-Enrolled Layout**

1. Again, click **New Page Layout** → Name it: Loyalty-Enrolled Layout.
2. In the editor:
   * Drag fields like **Loyalty Tier**, **Points Balance**, and **Total Purchases** into a section called **Loyalty Details**.
3. Save.



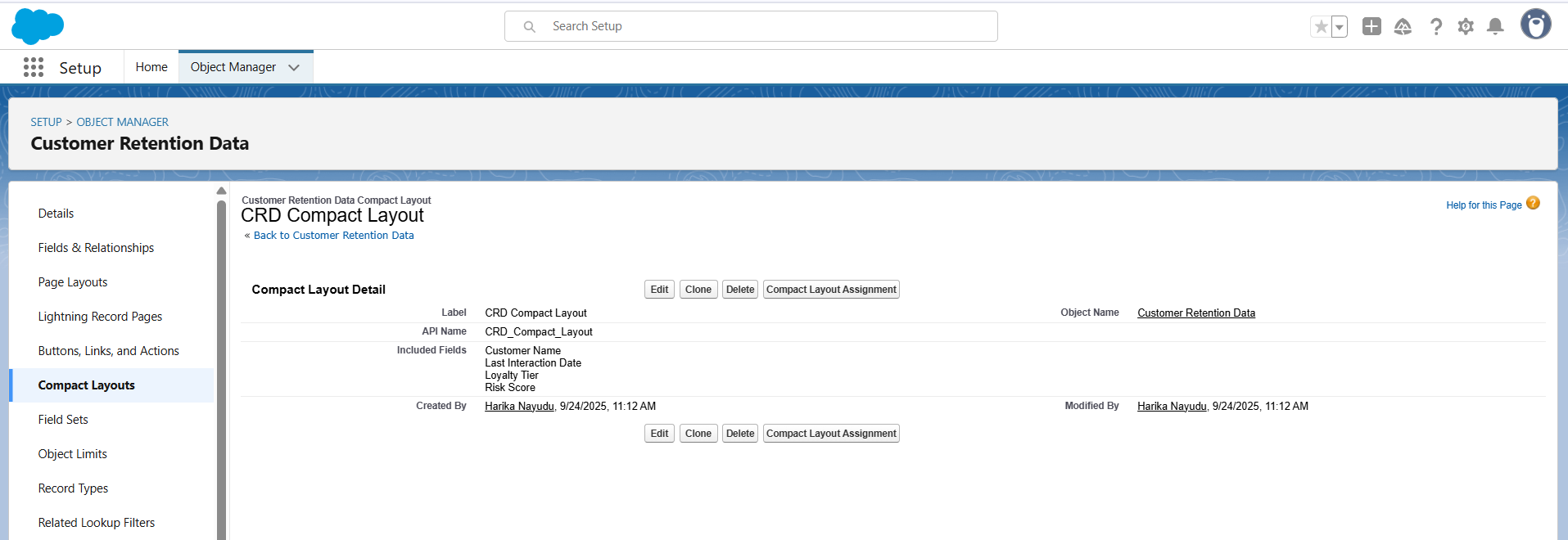


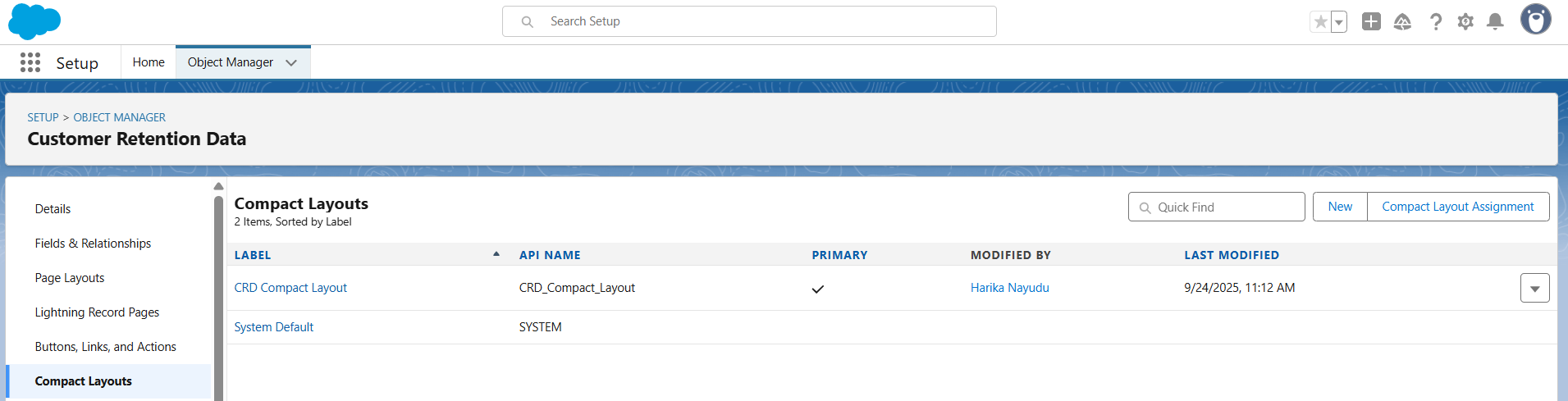
**Assign Layouts to Record Types**

1. Still inside **Page Layouts**, click the button **Page Layout Assignment** (top right).
2. Click it → then click **Edit Assignment**.
3. For:
   * **At-Risk Customer Record Type** → assign the **At-Risk Layout**.
   * **Loyalty-Enrolled Customer Record Type** → assign the **Loyalty-Enrolled Layout**.
4. Save.

**Step 5: Compact Layouts**

1. Go to Object Manager → Customer Retention Data → **Compact Layouts → New**.
2. Name: CRD Compact Layout.
3. Select fields: Customer Name, Loyalty Tier, Risk Score, Last Interaction Date.
4. Save → Set as **Primary Compact Layout**.





**🔹 Step 6: Schema Builder**

1. Setup → Quick Find → **Schema Builder**.
2. From left panel, select objects:
   * Customer Retention Data
   * Loyalty Transaction
   * Engagement Activity
   * Account
3. Arrange them to clearly show relationships.

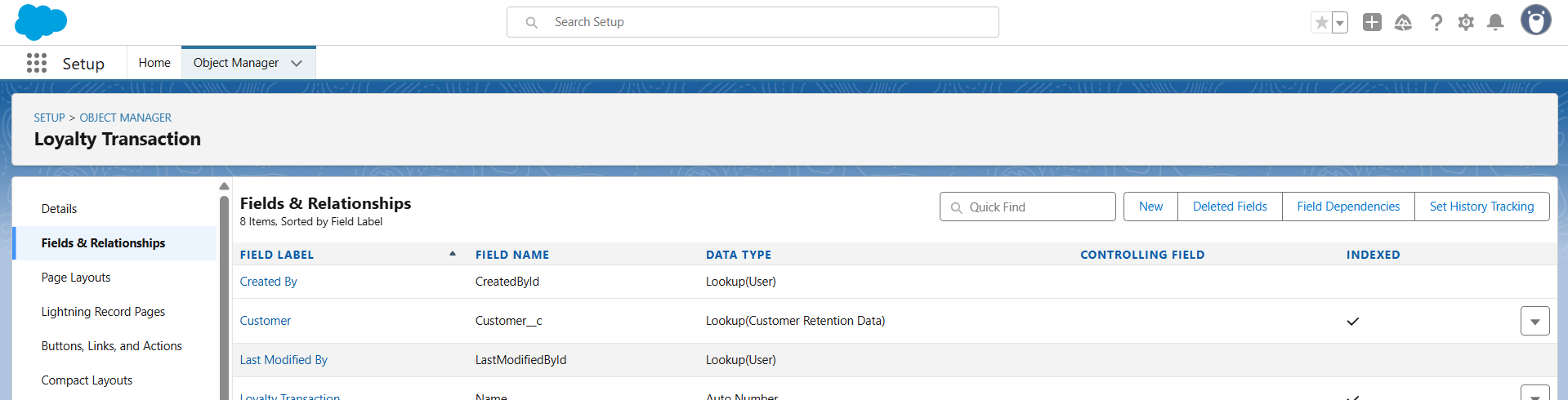
Loyalty Transaction → Customer Retention Data (Lookup).

Engagement Activity → Customer Retention Data (Master-Detail).

Customer Retention Data → Account (Lookup).

**Step 7: Relationships**

* **Lookup Relationship**
  + Loyalty Transaction → Customer Retention Data
  + (Steps: Loyalty Transaction → Fields & Relationships → New → Lookup → Related To Customer Retention Data).



* **Master-Detail Relationship**
  + Engagement Activity → Customer Retention Data
  + (Steps: Engagement Activity → Fields & Relationships → New → Master-Detail → Related To Customer Retention Data).



* **Hierarchical Relationship**
  + Only available on User object.
  + Example: A Customer Success Manager user reports to Sales Manager user.

**Step 8: Junction Objects**

1. Create new custom object: **Customer Loyalty Enrollment**.
   * Record Name: Auto Number (ENR-{0000}).
2. Add two Master-Detail fields:
   * Customer Retention Data.
   * Loyalty Program (create Loyalty Program object first).
3. This junction object now allows many-to-many between customers and loyalty programs.

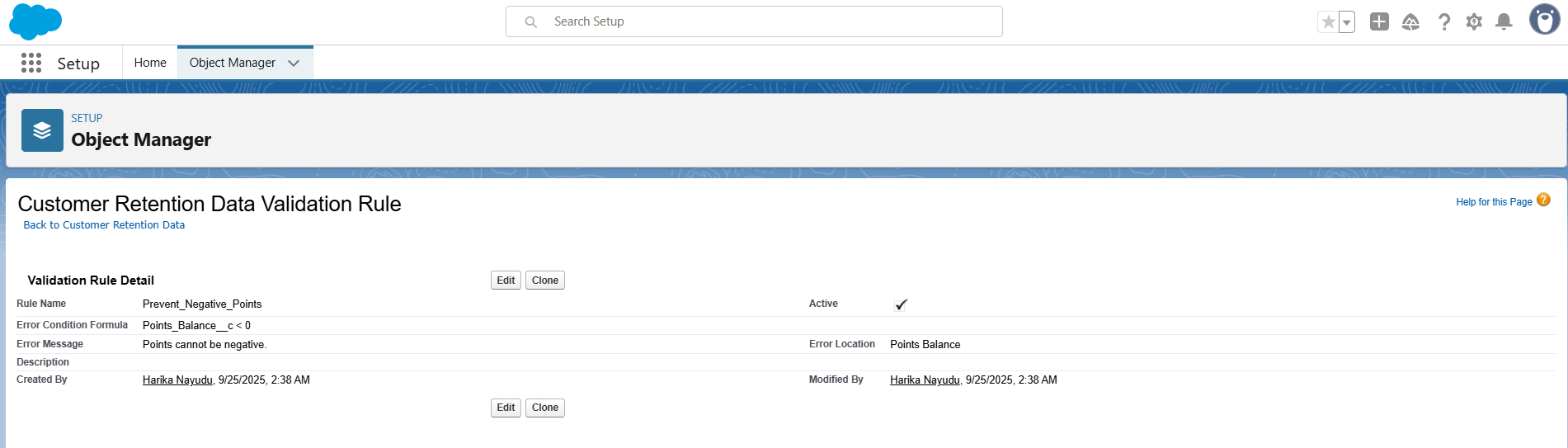
****

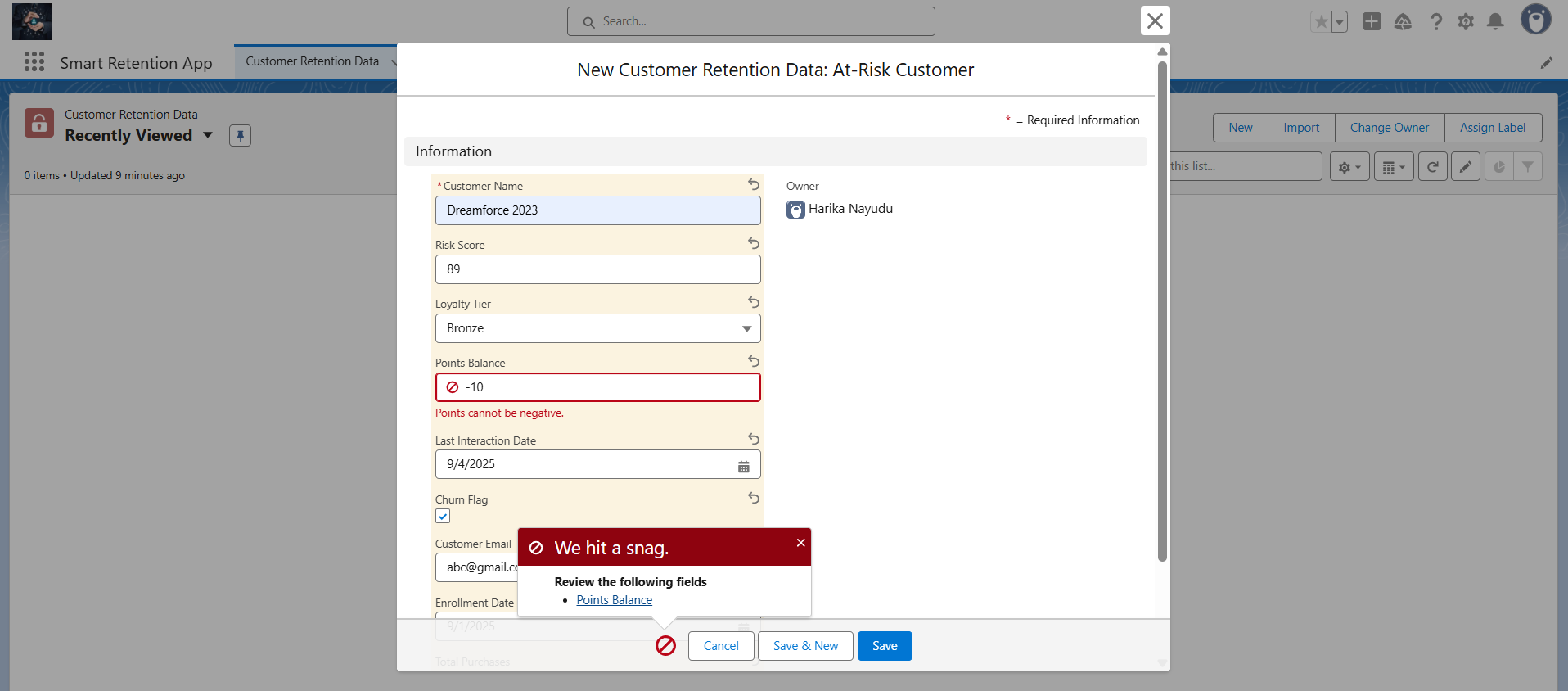
**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 4: Process Automation**

**Step 1: Validation Rules**

1. Go to **Setup → Object Manager**.
2. Choose *Customer Retention Data* object.
3. Click **Validation Rules → New**.
4. Enter:
   * **Rule Name:** Prevent\_Negative\_Points
   * **Error Condition Formula:** Points\_Balance\_\_c < 0
   * **Error Message:** Points cannot be negative.
   * **Error Location:** Points\_Balance\_\_c field.
5. Save → Activate.
6. Test by trying to enter -10 points → it should block save.





**2. Workflow Rules**

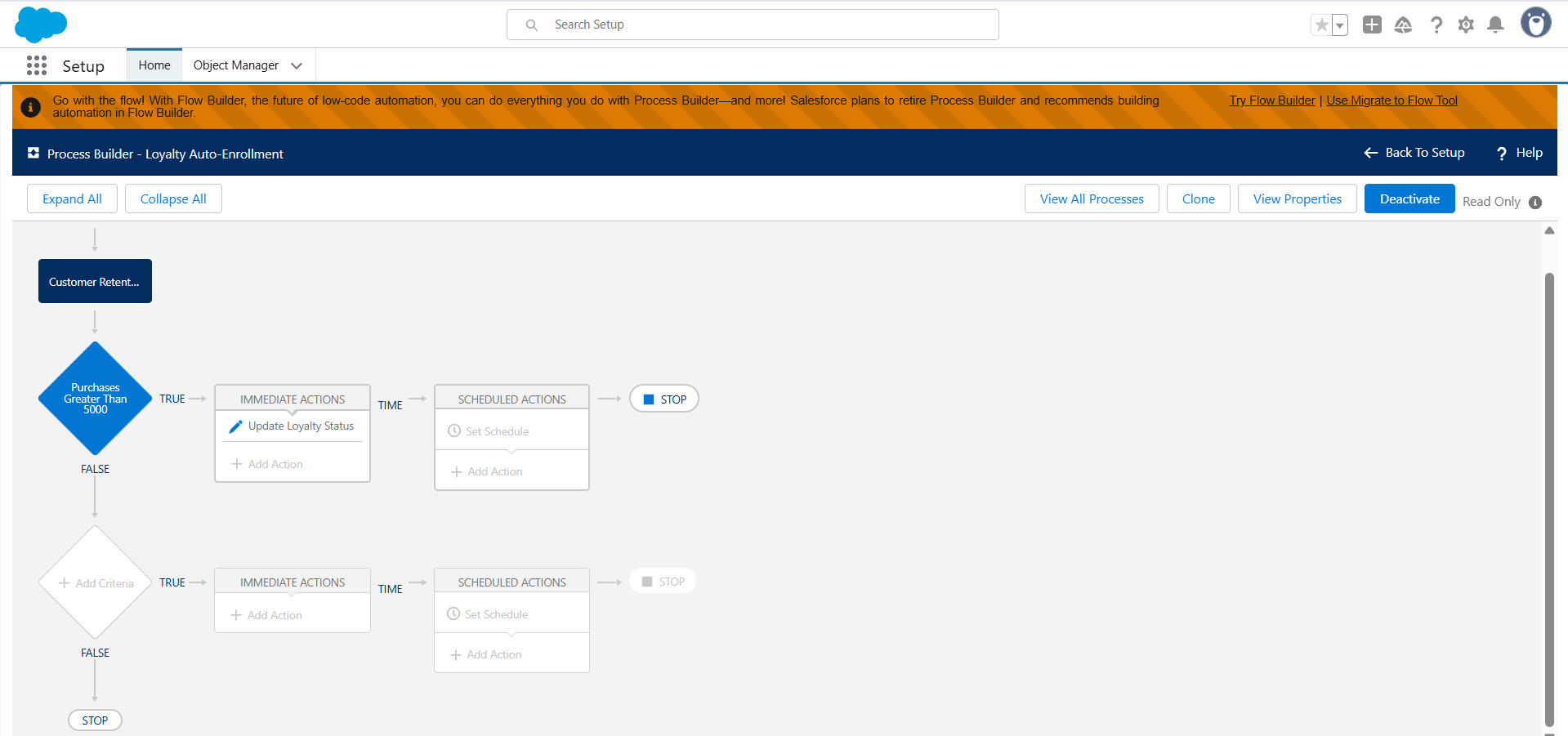
1. Navigate to **Setup → Workflow Rules → New Rule**.
2. Select object: **Customer Retention Data**.
3. Define criteria: Churn\_Flag\_\_c = TRUE.
4. Add actions:
   * Send **Email Alert** to Customer Success Manager.
   * Create **Follow-up Task** for churn handling.
5. Save and activate workflow.

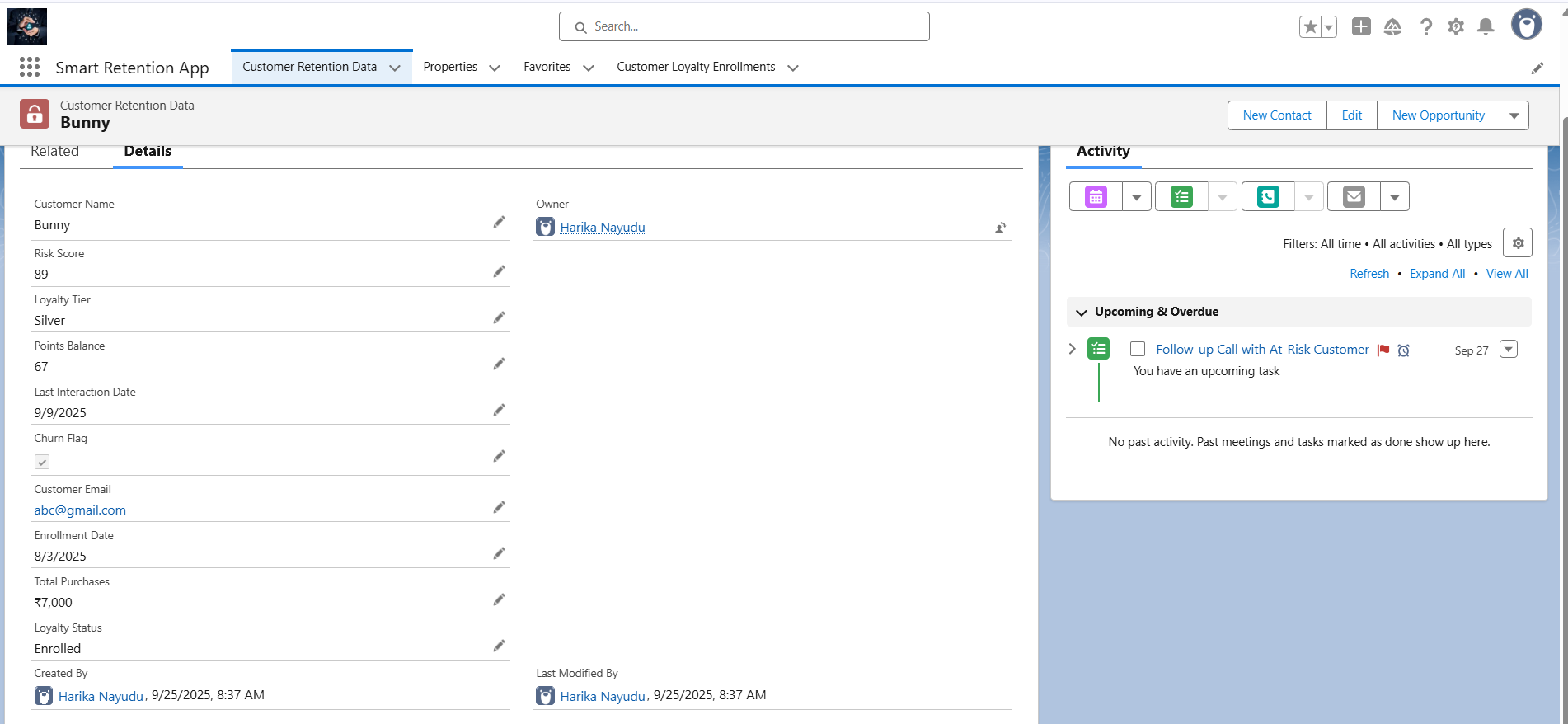




**3. Process Builder**

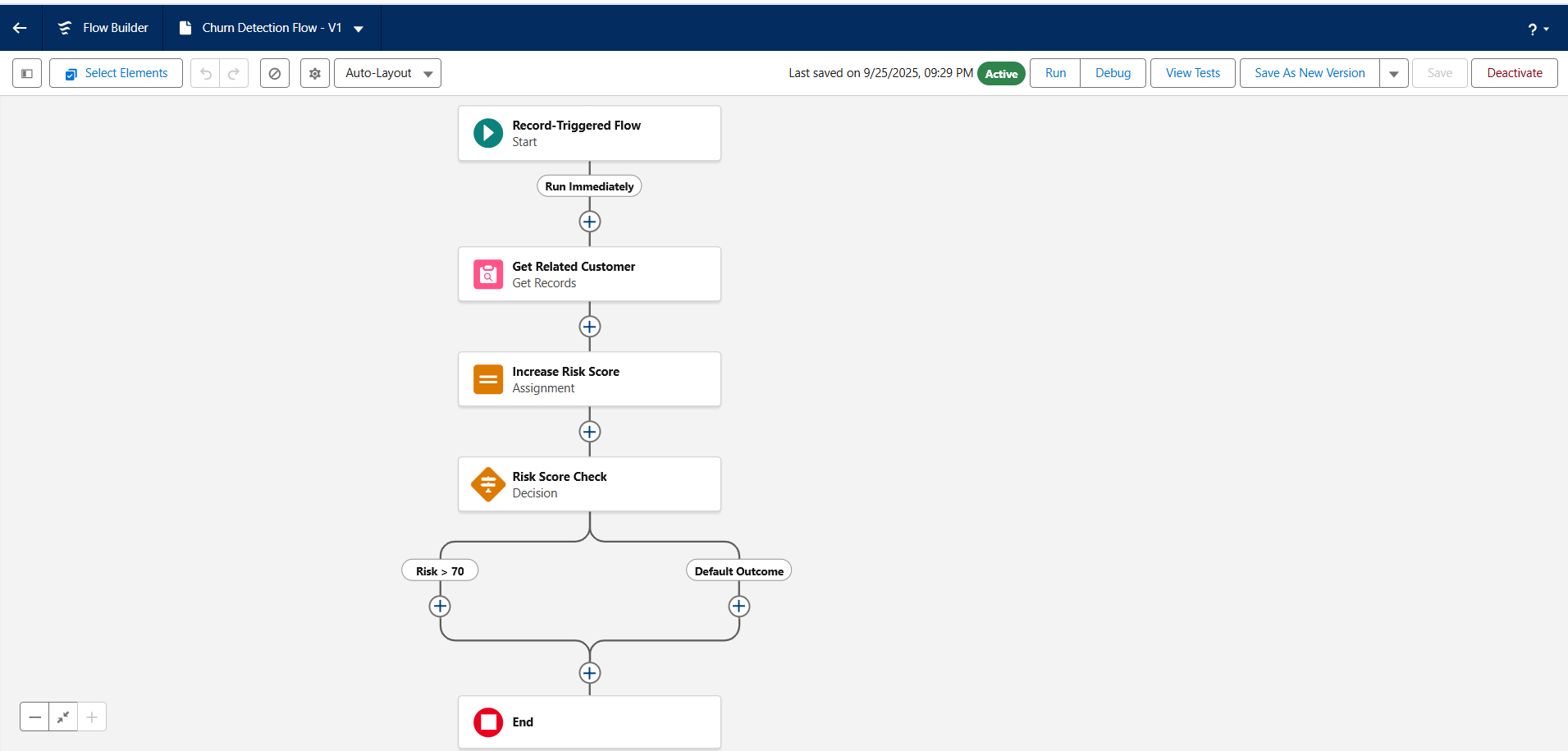
1. Open **Process Builder → New Process**.
2. Select object: **Customer Retention Data**.
3. Define criteria: Total\_Purchases\_\_c > 5000.
4. Action: Update field Loyalty\_Status\_\_c = Enrolled.
5. Save and activate the process.
6. Test: (Smart Retention App → Open a Customer Retention Data record → Update Total Purchases to a value greater than 5000 (e.g., 6000) → Save the record → Check if Loyalty Status automatically changes to Enrolled)





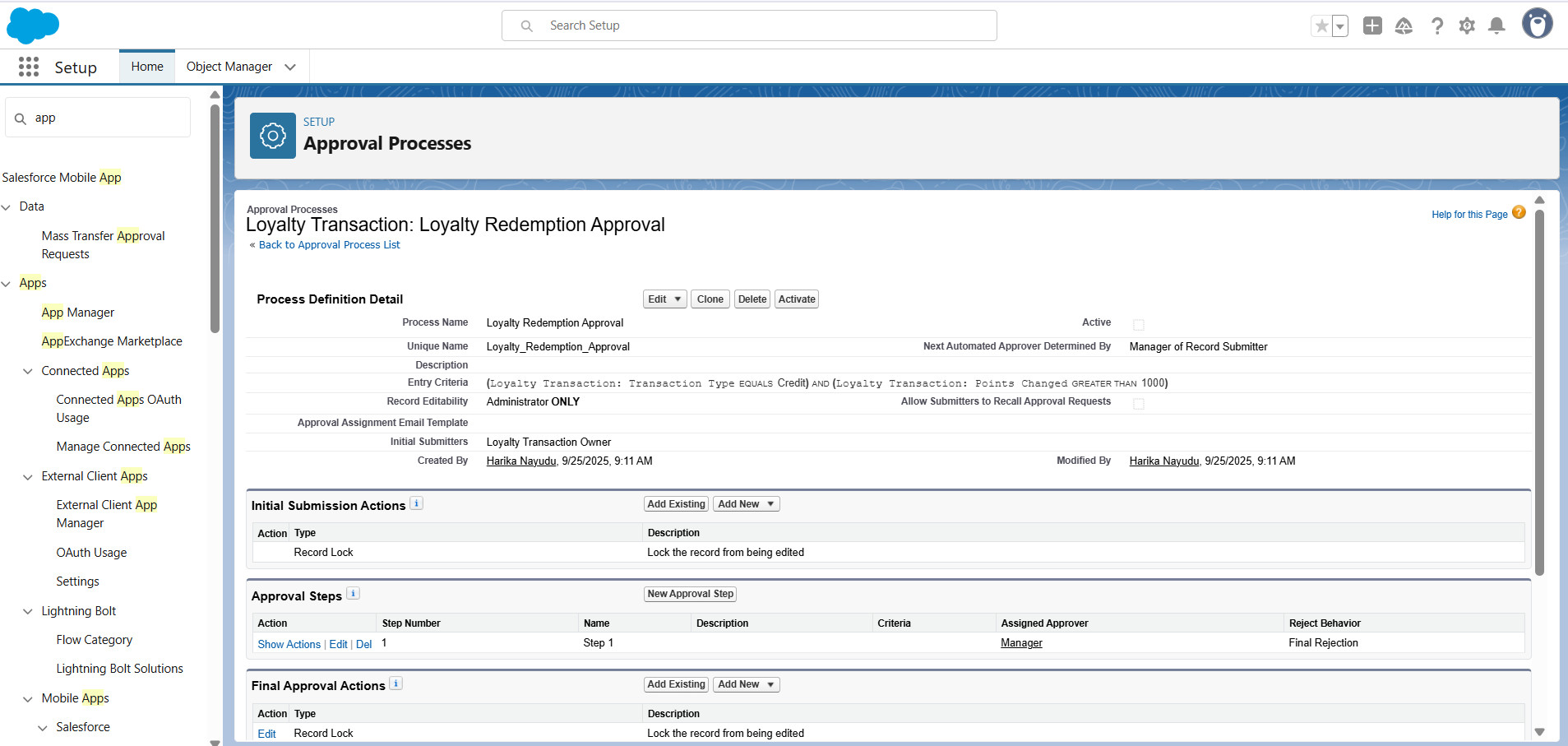
**4. Flow Builder**

1. **Churn Detection Flow**
   * Type: Record-Triggered Flow (Engagement Activity object).
   * Actions:
     1. Get related Customer record.
     2. Increase Risk Score.
     3. If Risk Score > 70 → Mark At\_Risk\_\_c = TRUE and send alert email.
2. **Loyalty Points Flow**
   * Type: Record-Triggered Flow (Loyalty Transaction object).
   * Actions:
     1. Fetch related Customer.
     2. Add Points\_Changed\_\_c to Points\_Balance\_\_c.
     3. Update Customer record.
3. **Auto-Enrollment Flow**
   * Type: Record-Triggered Flow (Customer Retention Data).
   * Condition: Total\_Purchases\_\_c > 5000 AND Loyalty\_Status\_\_c ≠ "Enrolled".
   * Action: Update Loyalty\_Status\_\_c = Enrolled.

****

**5. Approval Process**

1. Go to **Setup → Approval Processes → New Approval Process**.
2. Select object: **Loyalty Transaction**.
3. Define criteria: Transaction\_Type\_\_c = "Redemption" AND Points\_\_c > 1000.
4. Assign approver: Manager.
5. Define outcomes:
   * On Approval → Status = Approved.
   * On Rejection → Status = Rejected.



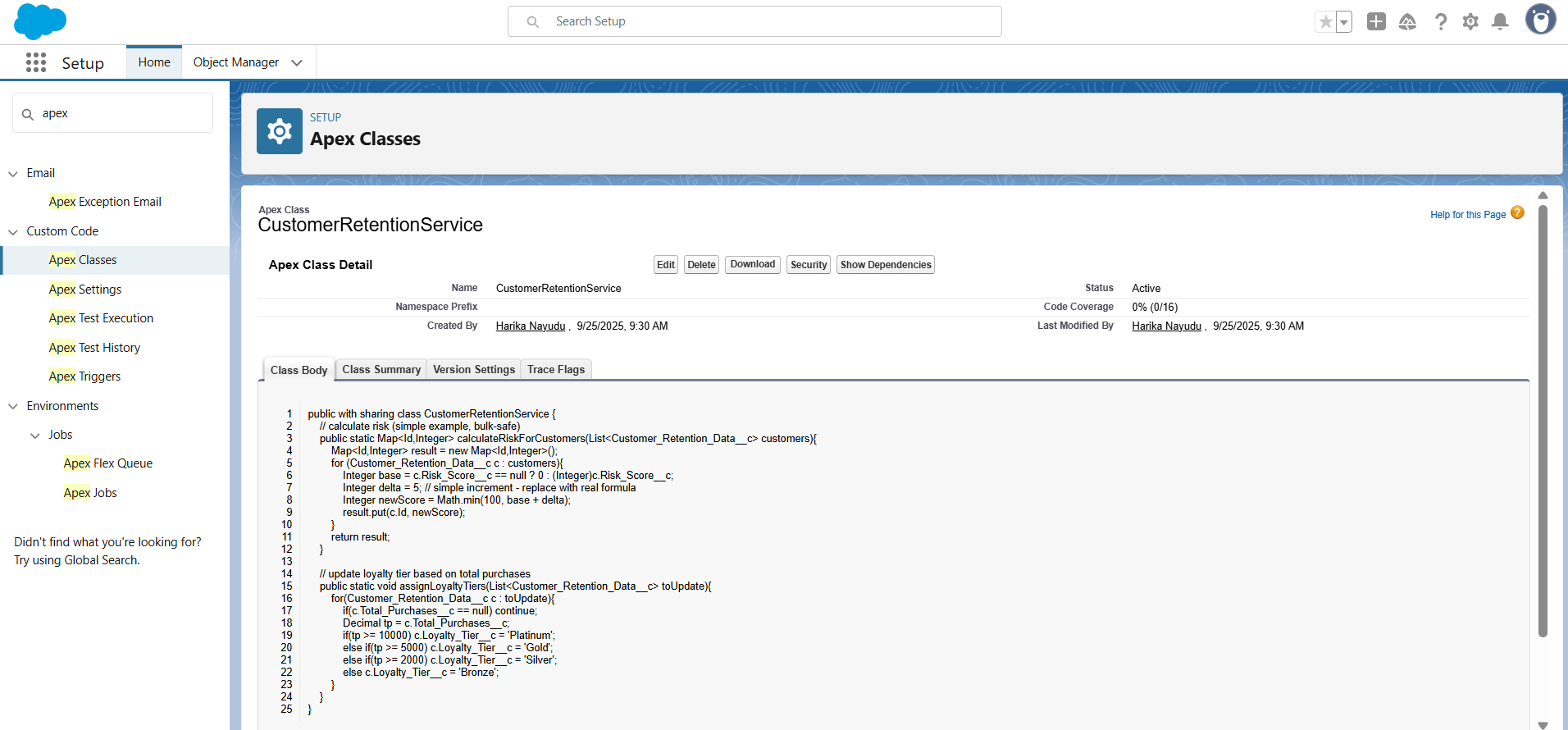
**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 5: Apex Programming (Developer)**

**1. Classes & Objects (Utility Class)**

**Steps**

1. Setup → Quick Find → **Apex Classes** → **New**.
2. Paste code → Save.



**2. Apex Trigger (before/after insert/update/delete)**

**Steps**

1. Setup → Quick Find → **Object Manager** → choose **Engagement Activity** object.
2. Scroll down → **Triggers** → **New**.
3. Paste code → Save.

**Code**

trigger EngagementActivityTrigger on Engagement\_Activity\_\_c (

before insert, before update, before delete,

after insert, after update, after delete

) {

if (Trigger.isBefore) {

if (Trigger.isInsert) EngagementActivityTriggerHandler.beforeInsert(Trigger.new);

if (Trigger.isUpdate) EngagementActivityTriggerHandler.beforeUpdate(Trigger.new, Trigger.oldMap);

if (Trigger.isDelete) EngagementActivityTriggerHandler.beforeDelete(Trigger.old);

}

if (Trigger.isAfter) {

if (Trigger.isInsert) EngagementActivityTriggerHandler.afterInsert(Trigger.newMap);

if (Trigger.isUpdate) EngagementActivityTriggerHandler.afterUpdate(Trigger.newMap, Trigger.oldMap);

if (Trigger.isDelete) EngagementActivityTriggerHandler.afterDelete(Trigger.oldMap);

}

}

**3. Trigger Design Pattern (Handler Class)**

**Steps**

1. Setup → Apex Classes → **New**.
2. Paste code → Save.

**Code**

public with sharing class EngagementActivityTriggerHandler {

public static void beforeInsert(List<Engagement\_Activity\_\_c> newList){

for(Engagement\_Activity\_\_c ea : newList){

if(ea.Activity\_Type\_\_c == null) ea.Activity\_Type\_\_c = 'Email';

}

}

public static void afterInsert(Map<Id, Engagement\_Activity\_\_c> newMap){

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

for(Engagement\_Activity\_\_c ea : newMap.values()){

if(ea.Customer\_Retention\_Data\_\_c != null) custIds.add(ea.Customer\_Retention\_Data\_\_c);

}

if(!custIds.isEmpty()){

System.enqueueJob(new UpdateRiskQueueable(custIds));

}

}

// stubs for other events

public static void beforeUpdate(List<Engagement\_Activity\_\_c> newList, Map<Id, Engagement\_Activity\_\_c> oldMap){}

public static void beforeDelete(List<Engagement\_Activity\_\_c> oldList){}

public static void afterUpdate(Map<Id, Engagement\_Activity\_\_c> newMap, Map<Id, Engagement\_Activity\_\_c> oldMap){}

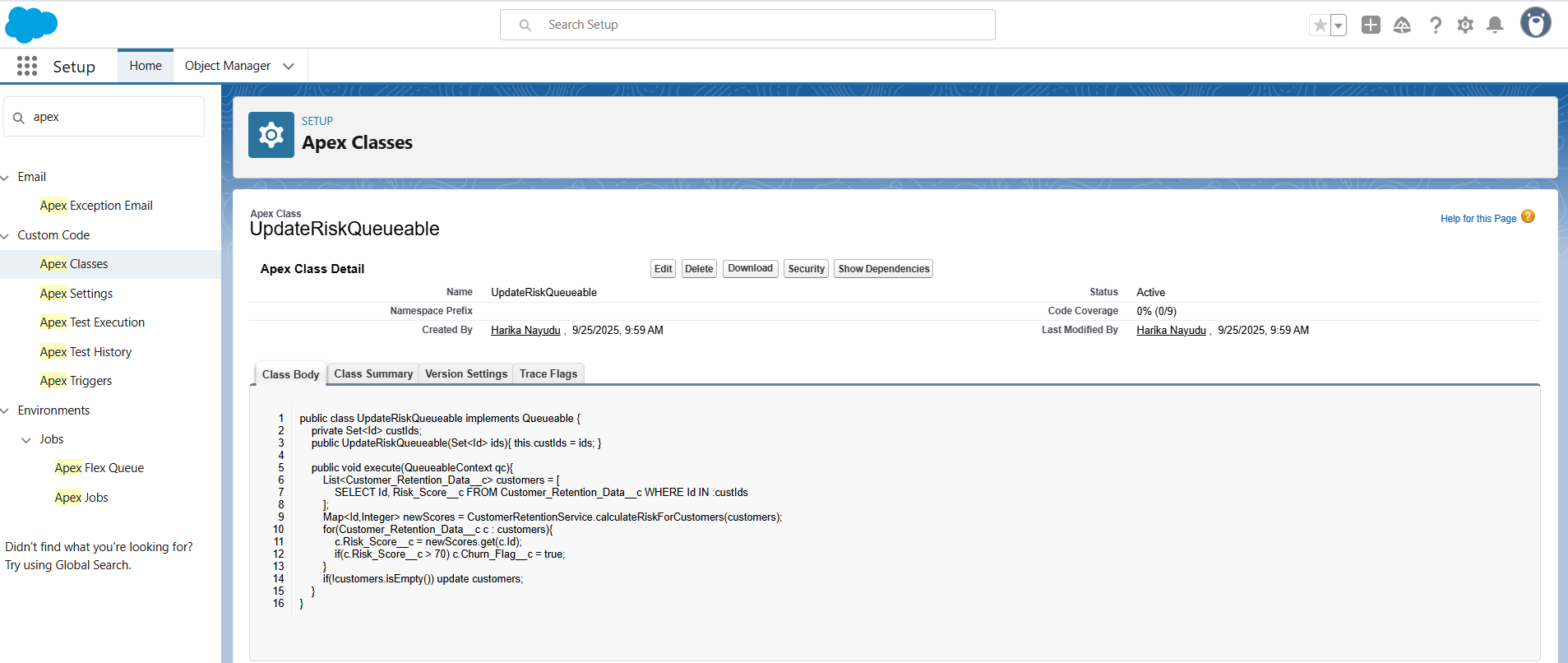
public static void afterDelete(Map<Id, Engagement\_Activity\_\_c> oldMap){}

}

**4. Queueable Apex**

**Steps**

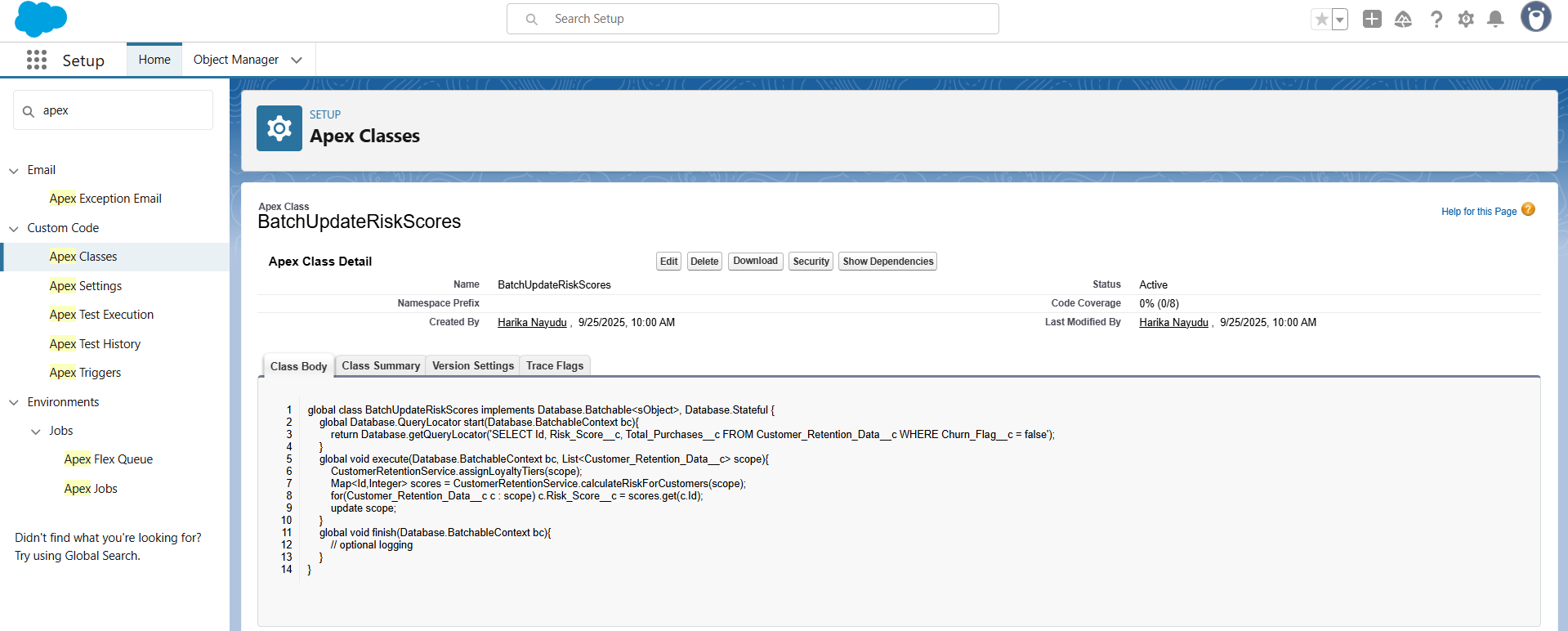
1. Setup → Apex Classes → **New**.
2. Paste code → Save.

****

**5. Batch Apex**

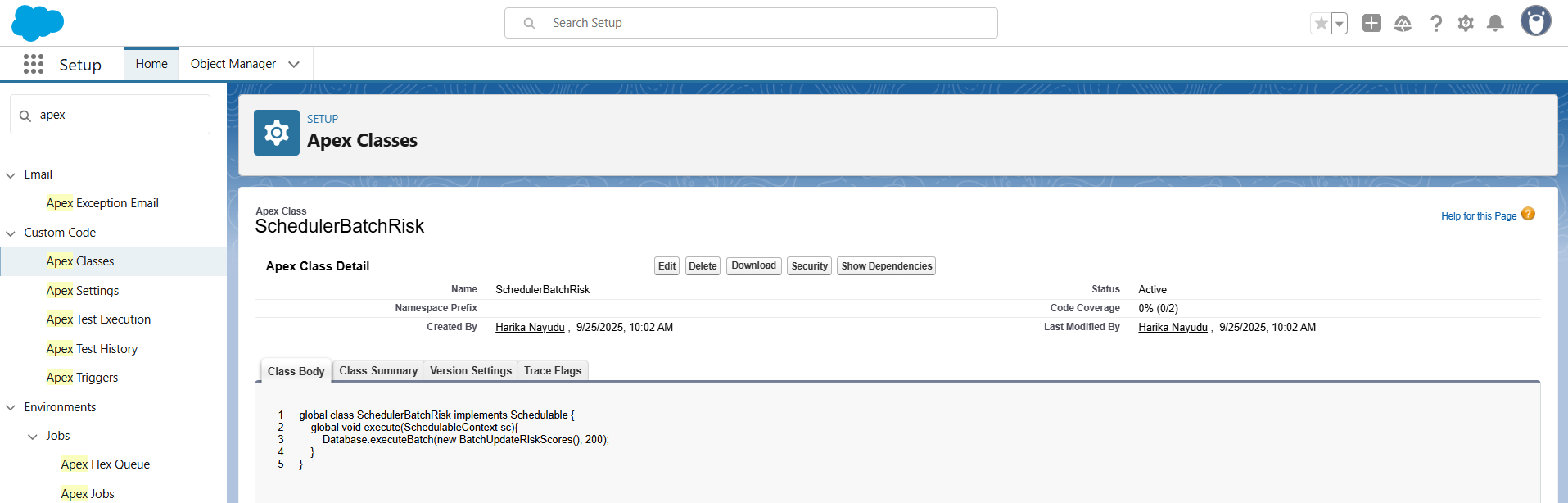
**Steps**

1. Setup → Apex Classes → **New**.
2. Paste code → Save.

**6. Scheduled Apex**

**Steps**

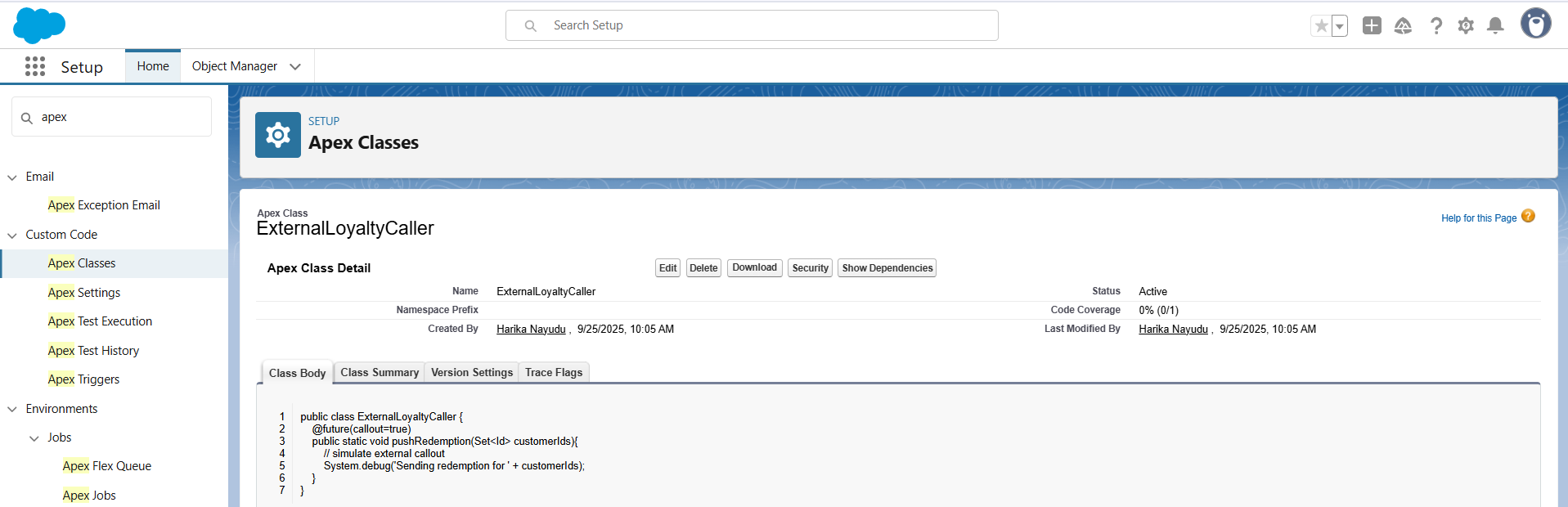
1. Setup → Apex Classes → **New**.
2. Paste code → Save.
3. Setup → Quick Find → **Apex Classes → Schedule Apex** → choose this class.



**7. Future Method (Example)**

**Steps**

1. Setup → Apex Classes → **New**.
2. Paste code → Save.



**8. Test Class**

**Steps**

1. Setup → Apex Classes → **New**.
2. Paste code → Save.

**Code**

@isTest

private class CustomerRetentionTests {

@isTest static void testRiskUpdate(){

Customer\_Retention\_Data\_\_c c = new Customer\_Retention\_Data\_\_c(Name='T1', Total\_Purchases\_\_c=500);

insert c;

Engagement\_Activity\_\_c ea = new Engagement\_Activity\_\_c(Name='E1', Customer\_Retention\_Data\_\_c=c.Id, Activity\_Type\_\_c='Call');

Test.startTest();

insert ea;

Test.stopTest();

Customer\_Retention\_Data\_\_c c2 = [SELECT Risk\_Score\_\_c FROM Customer\_Retention\_Data\_\_c WHERE Id=:c.Id];

System.assertNotEquals(null, c2.Risk\_Score\_\_c);

}

}

**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 6: User Interface Development**

**1. Lightning App Builder (Record Page)**

**Steps**

1. Setup → **Lightning App Builder** → **New** → Record Page.
2. Choose **Customer\_Retention\_Data\_\_c** object.
3. Add **Record Detail** + custom LWC (created below).
4. Save → Activate.

**2. Create LWC**

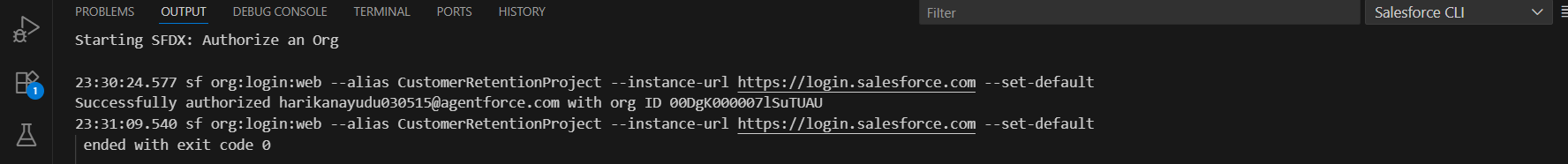
**Create a New Salesforce DX Project**

1. Open **VS Code** → **View → Command Palette** (or Ctrl+Shift+P).
2. Type **SFDX: Create Project** → Enter.
3. Choose **Standard**.
4. Enter a project name, e.g., CustomerRetentionProject.
5. Choose a location on your computer → Click **Create**.

After this, VS Code will create a project folder with this structure:

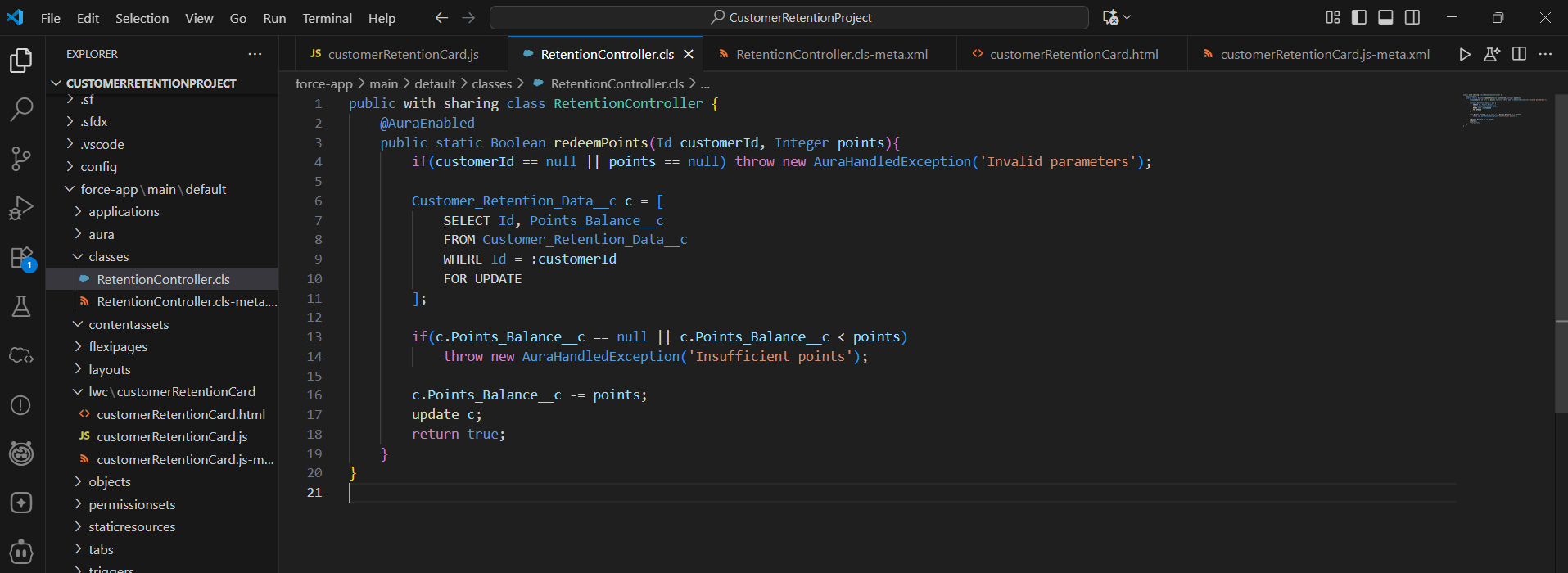
**Connect to Your Salesforce Org**

1. Open **Command Palette → SFDX: Authorize an Org**.
2. Choose **Production or Sandbox**.
3. Log in to your Salesforce org via the browser.
4. After successful login, the CLI will link the org to your project.

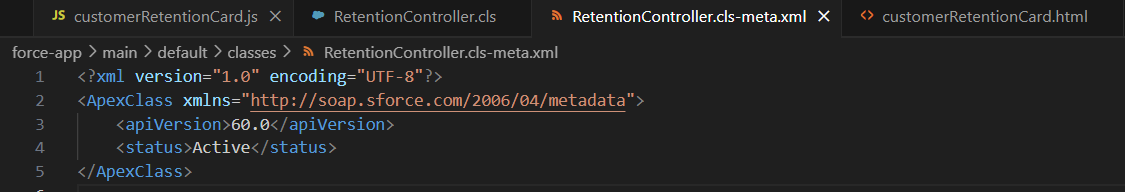
****

**1️⃣ Create the Apex Class First**

1. In **VS Code → Explorer**, go to:
2. force-app/main/default/classes/
3. **Create new file**: RetentionController.cls

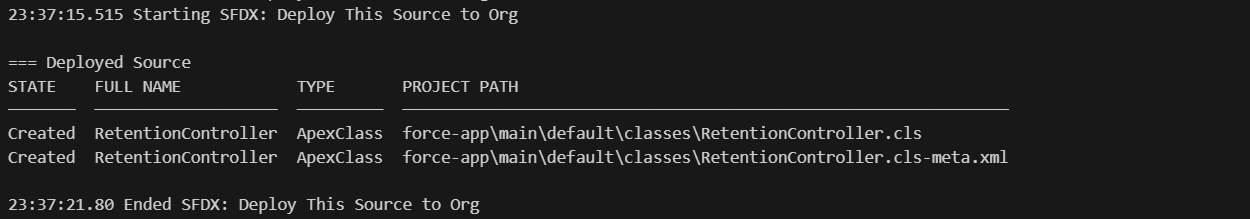


1. Create the **metadata file** next to it: RetentionController.cls-meta.xml:



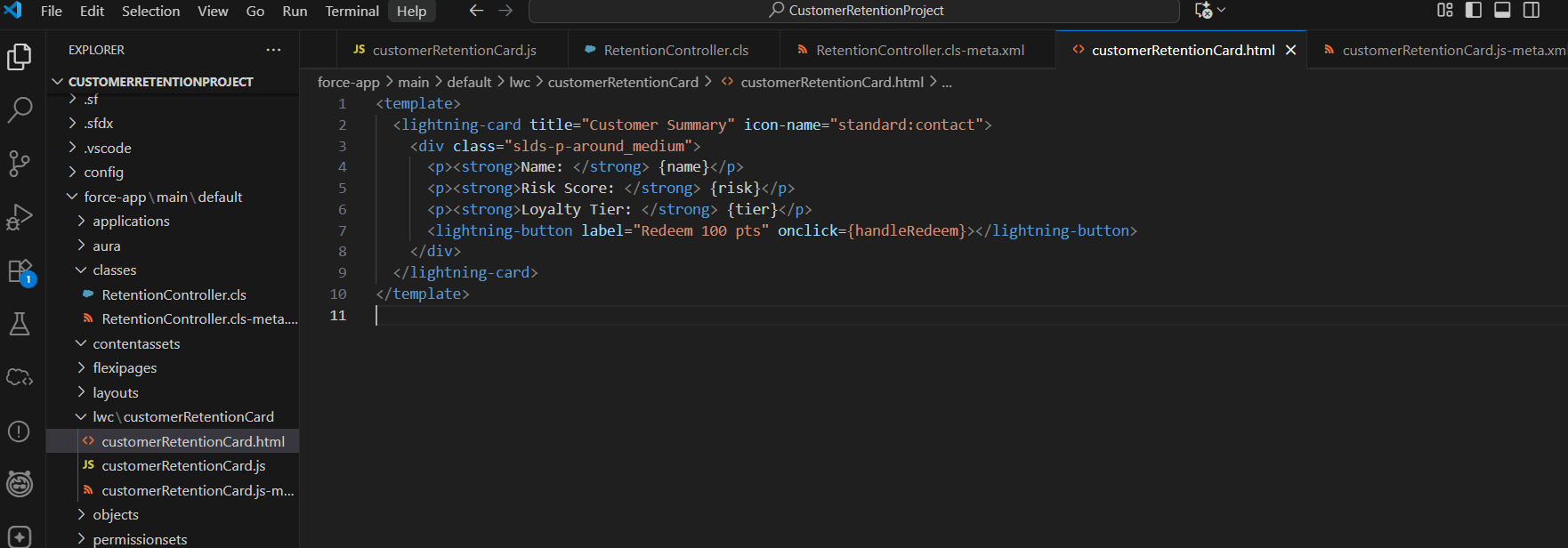
**2️⃣ Deploy the Apex Class**

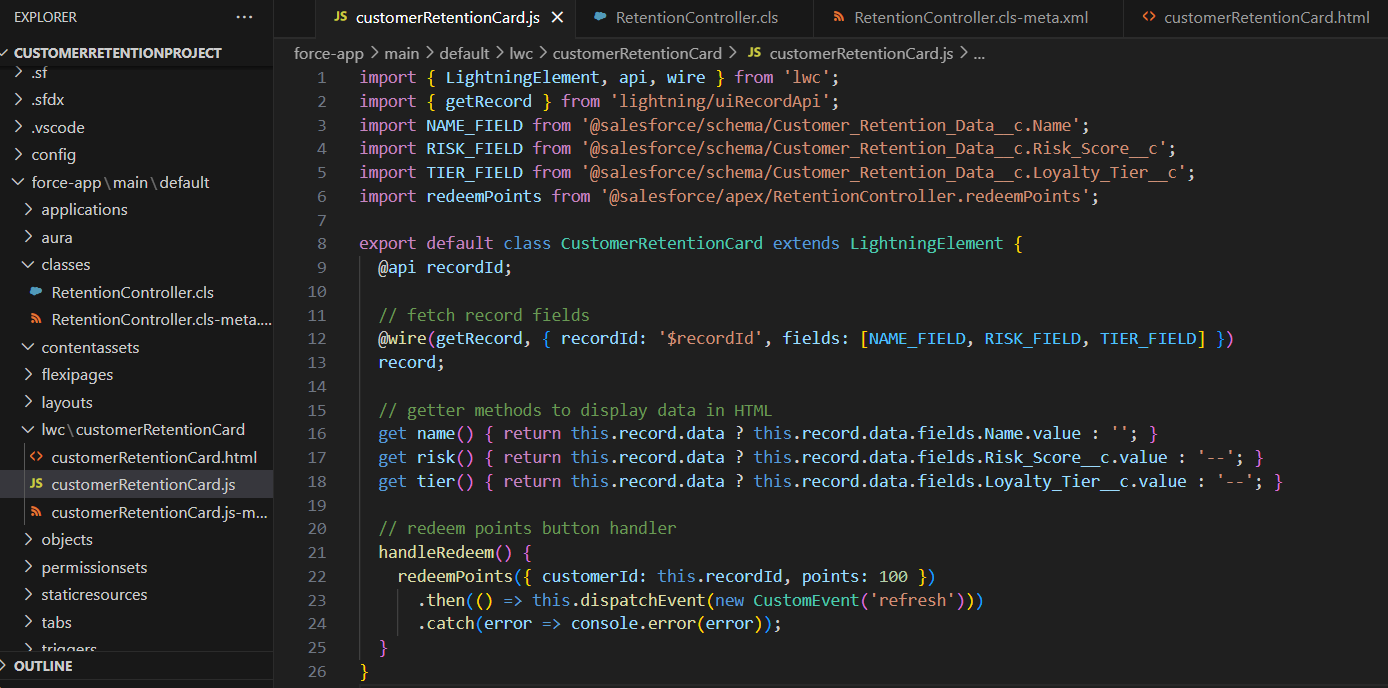
1. Right-click the RetentionController.cls file → **SFDX: Deploy Source to Org**.
2. Wait for **“Deployed Source to Org”** confirmation in the Output panel.

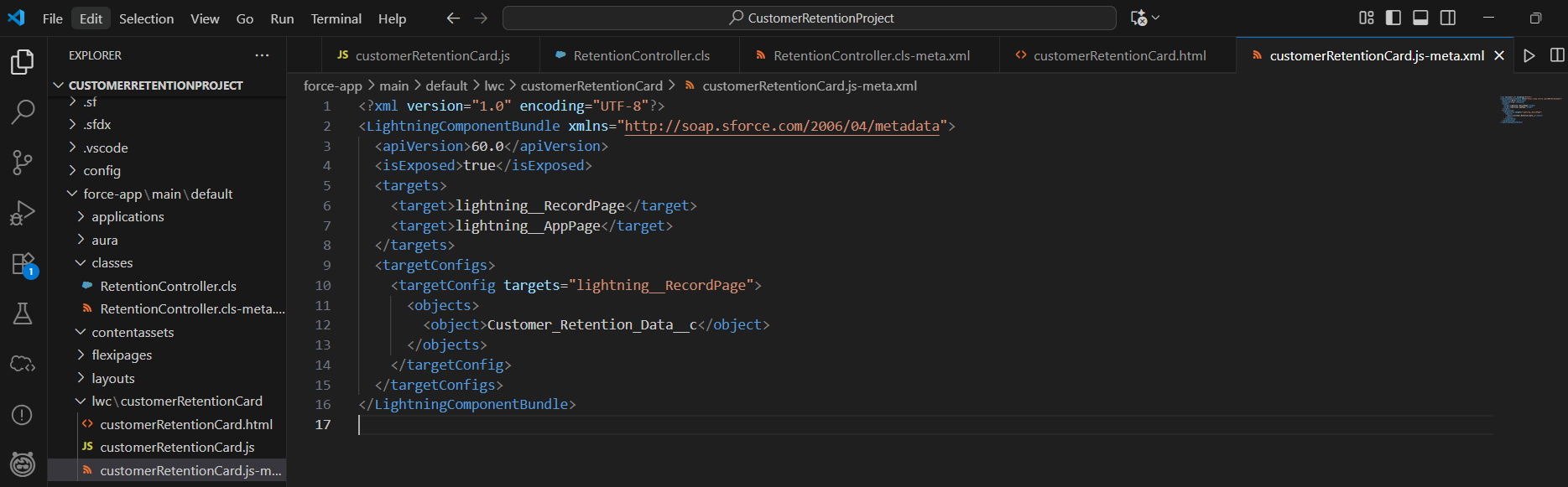


**3️⃣ Deploy the LWC**

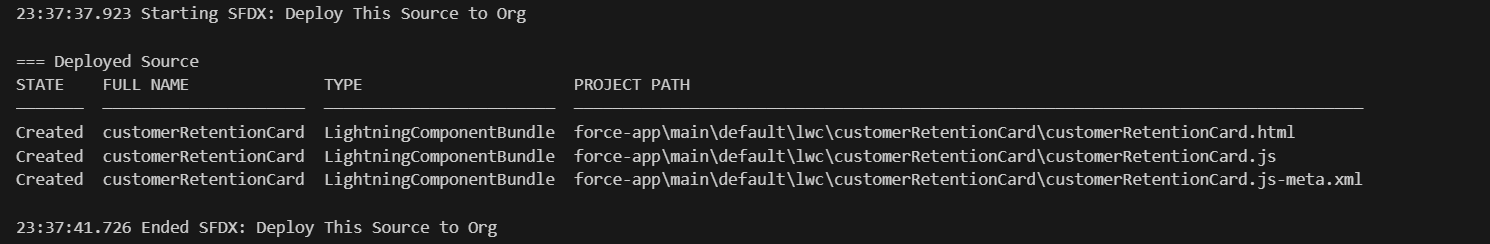
1. Now go back to your LWC folder:
2. force-app/main/default/lwc/customerRetentionCard/



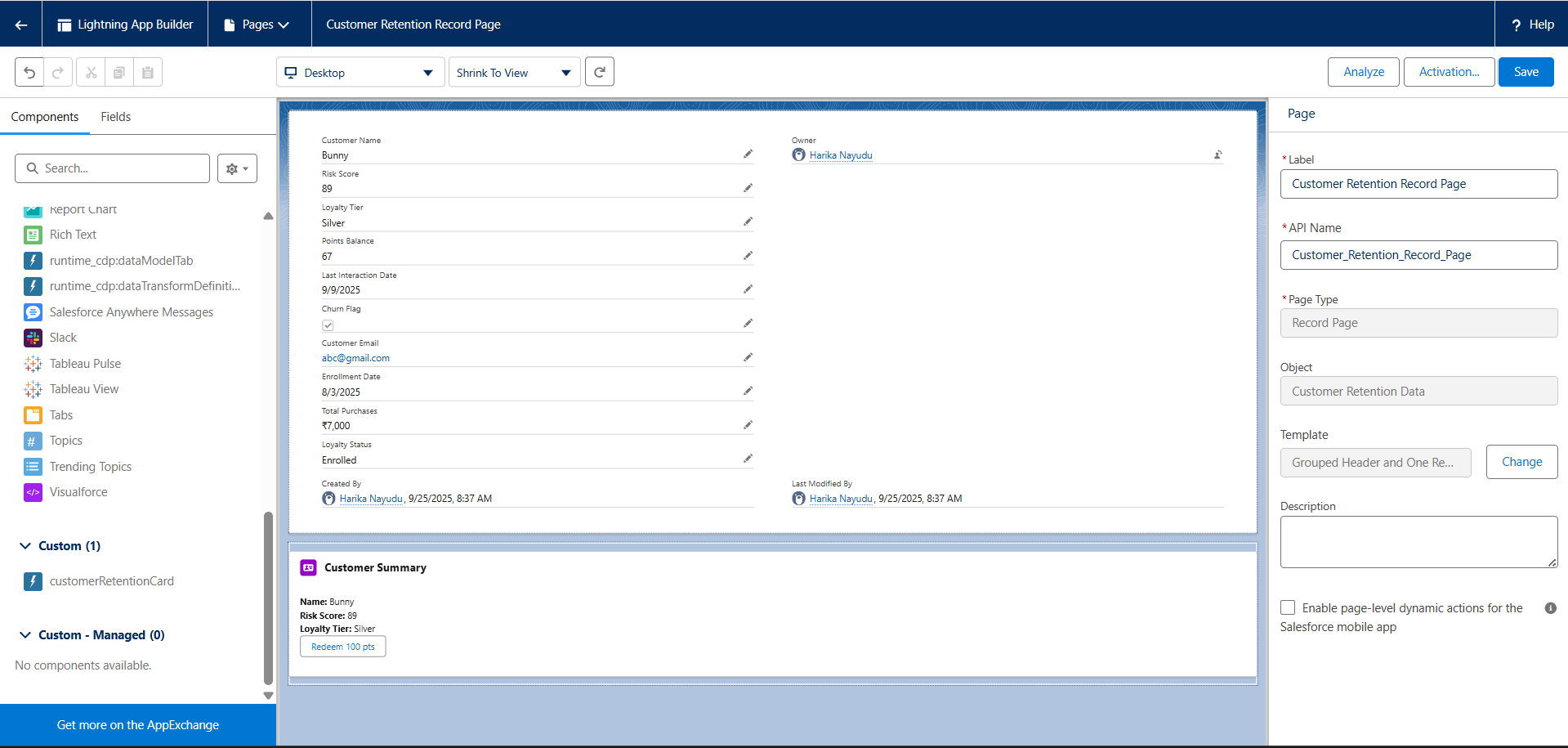




1. Right-click the folder → **SFDX: Deploy Source to Org**



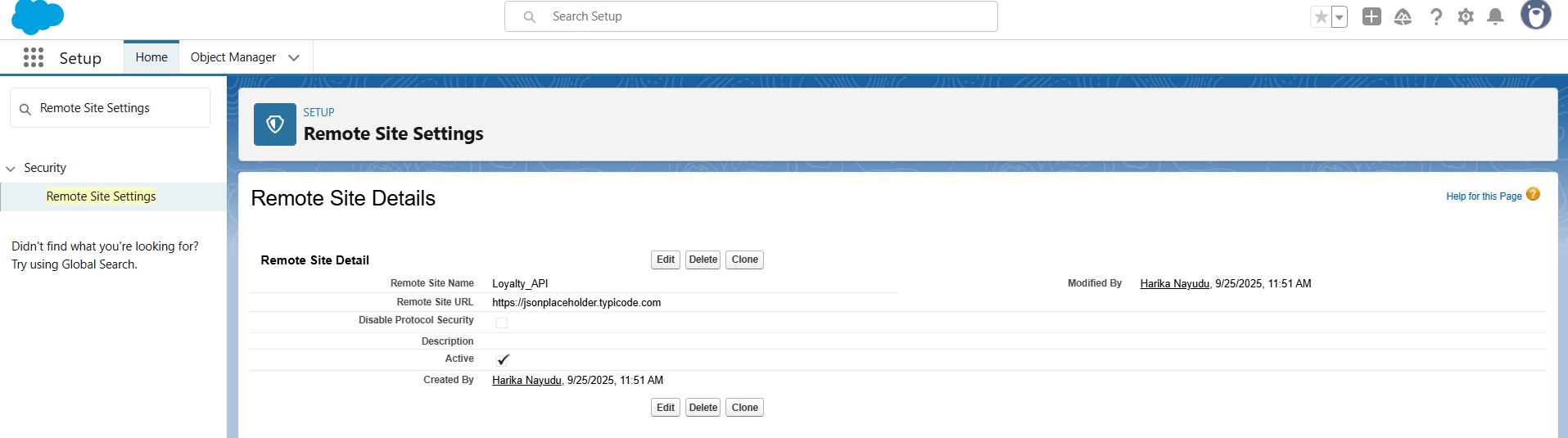
1. **Open Lightning App Builder** → Edit your Customer\_Retention\_Data\_\_c record page.
2. **Drag your LWC (customerRetentionCard)** from the **Custom** section onto the page.
3. **Save & Activate** → choose who can see the page.



**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 7: Integration & External Access**

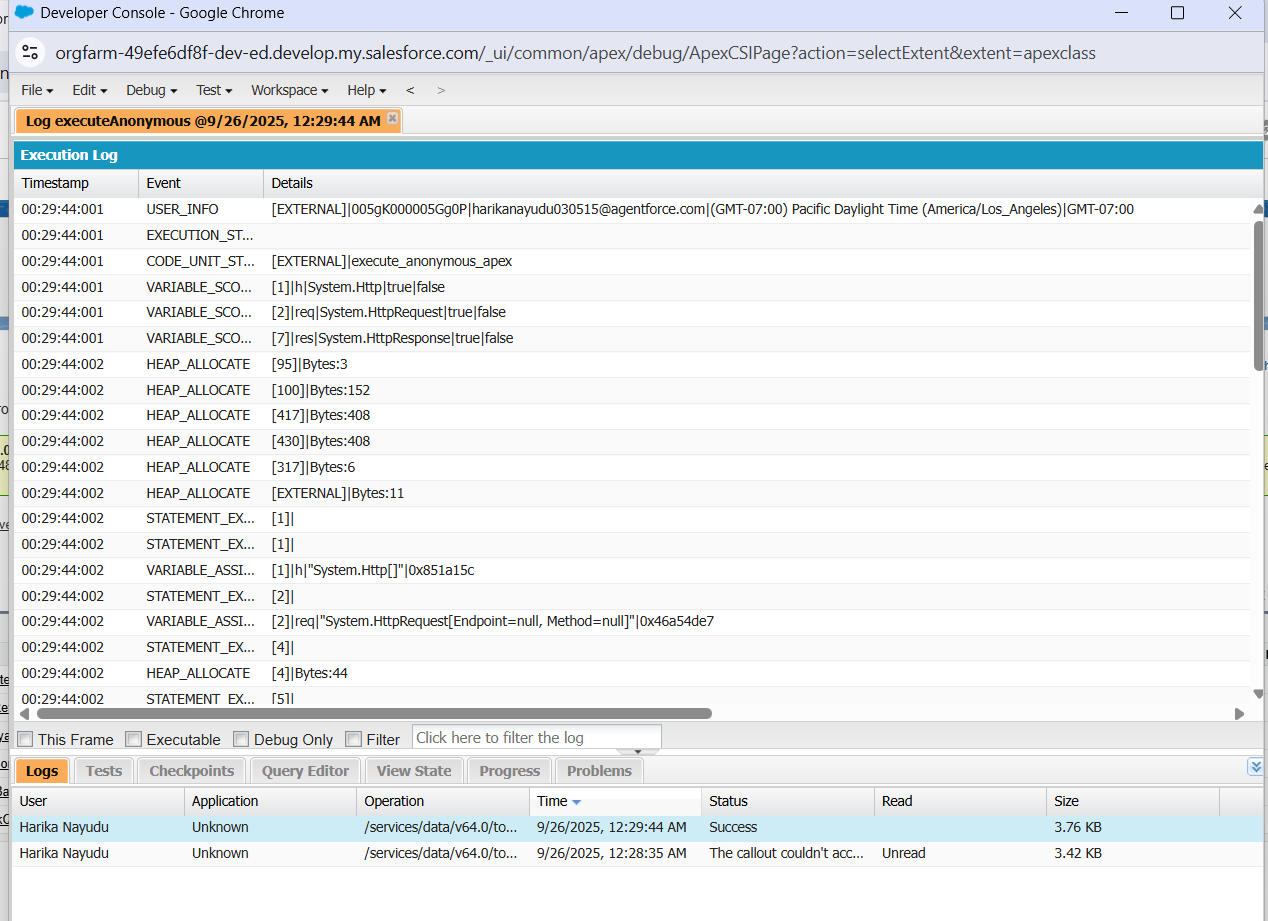
1. **Named Credential**
   * Setup → Quick Find → Named Credentials → New.
   * Name: Loyalty\_NC, URL: <https://jsonplaceholder.typicode.com>
   * Save.
2. **Remote Site Setting** (if not using Named Credential)
   * Setup → Quick Find → Remote Site Settings → New.
   * Enter API base URL → Save.



1. **REST Callout (test)**
   * Open **Developer Console** → Debug → **Open Execute Anonymous Window**.
   * Paste the code

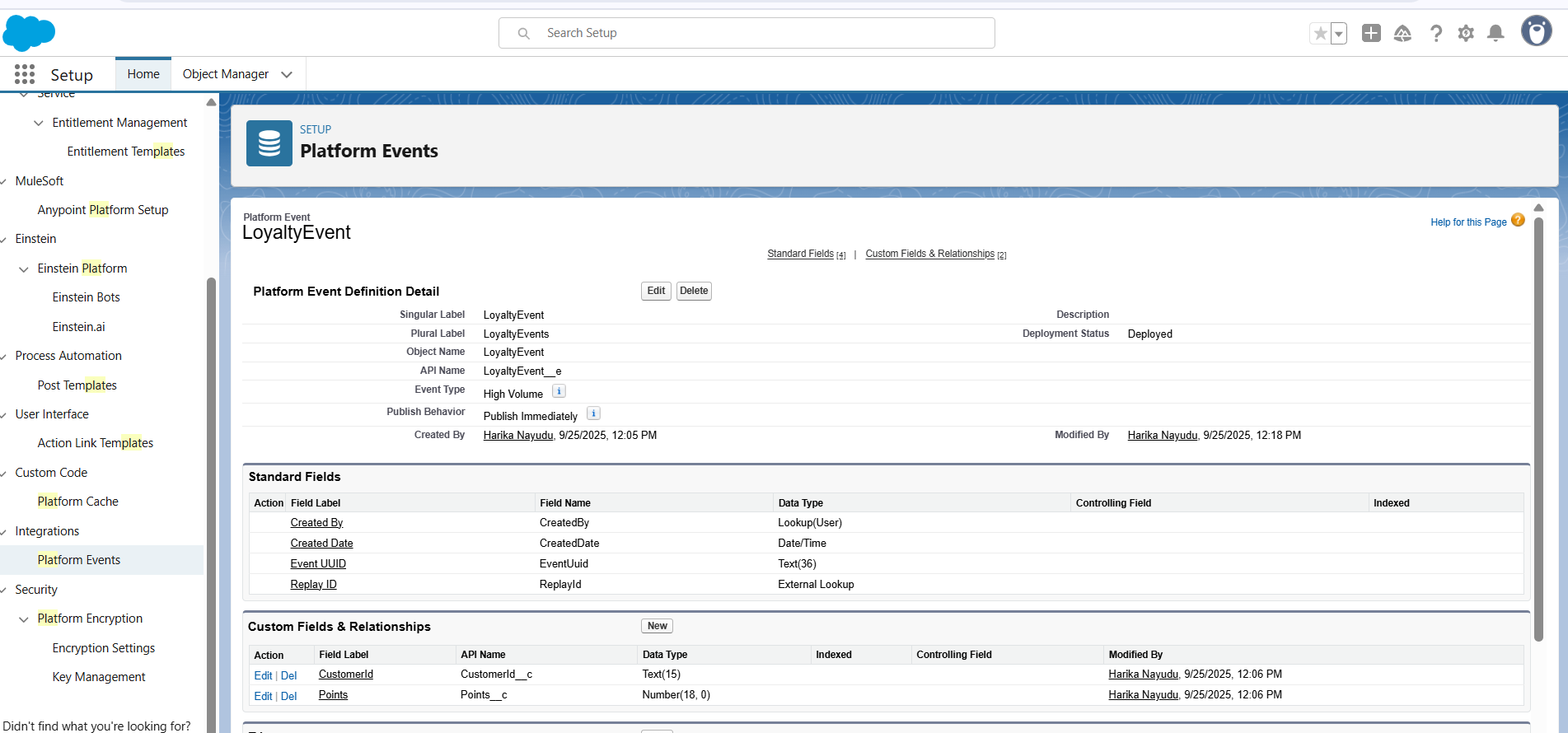


* + Check the box **Open Log** → Click **Execute**.

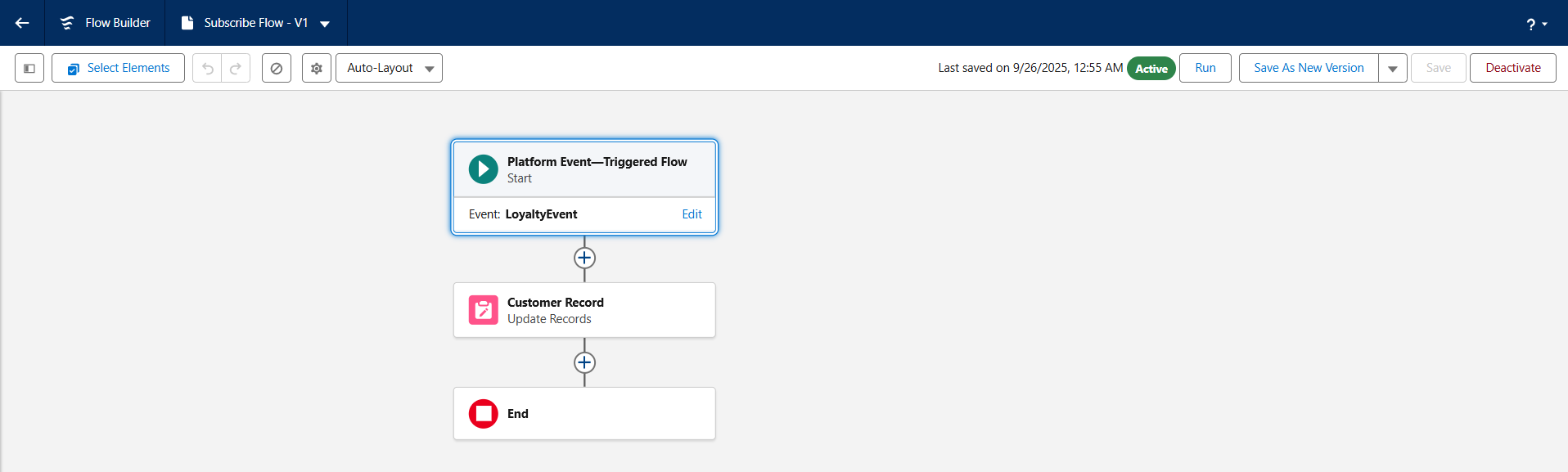


* + Run → check Debug logs for response.

1. **Platform Event**
   * Setup → Platform Events → New.
   * Name: LoyaltyEvent.
   * Add fields: CustomerId (Text), Points (Number) and Save



1. **Subscribe with Flow**
   * Setup → Flows → New → Platform Event Trigger → select LoyaltyEvent.
   * Add element: **Update Records** → update Customer\_Retention\_Data\_\_c.Points\_Balance\_\_c.
   * Save & Activate.



**Project Title: Smart Customer Retention & Loyalty Management System**

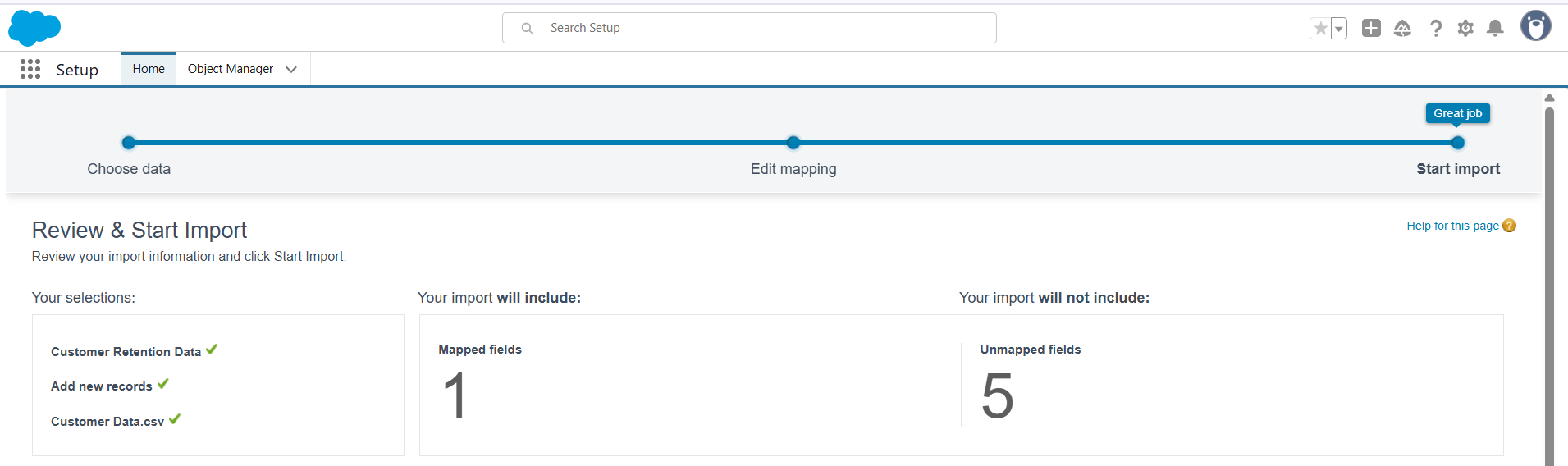
**Phase 8: Data Management & Deployment**

Got it! Let’s break **Phase 8: Data Management & Deployment** into **super clear, step-by-step instructions** so you can finish it quickly within 30 minutes. I’ll keep it easy and practical.

**Phase 8: Data Management & Deployment**

**1️⃣ Data Import Wizard**

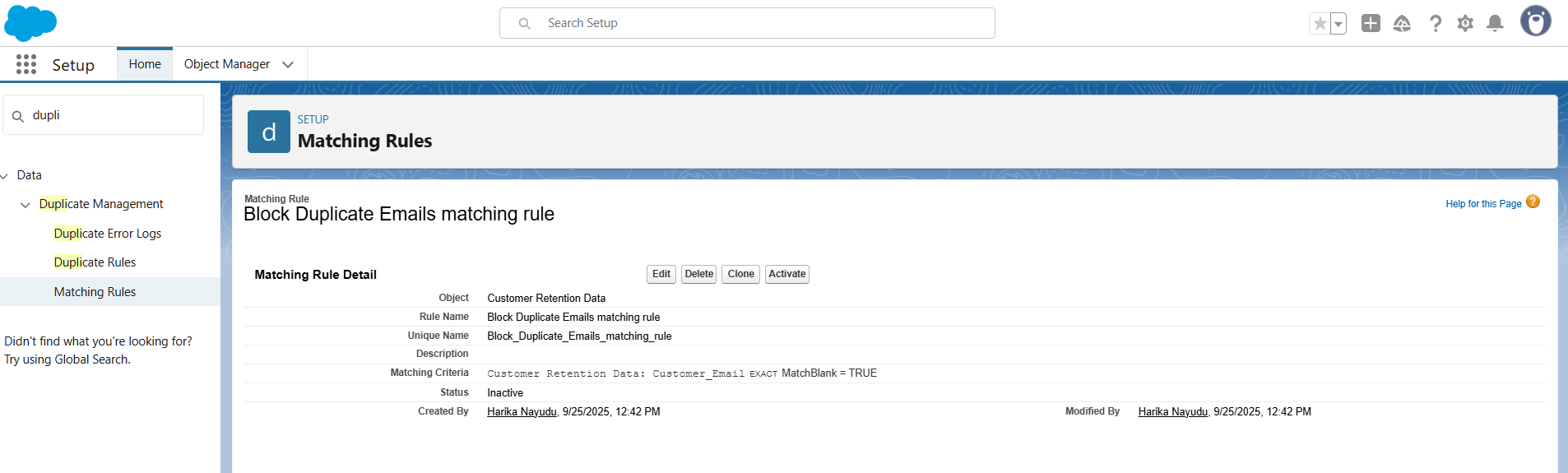
1. Go to **Setup → Quick Find → Data Import Wizard**.
2. Click **Launch Wizard**.
3. Scroll down → **Select Standard or Custom Object** → choose **Customer Retention Data**.
4. Click **Add New Records** (or choose Update if updating existing records).
5. Click **Choose CSV File → Select your file**.
6. Click **Next → Map Fields**:
   * Salesforce will try to auto-map fields
   * Double-check mapping (e.g., Name → Name, Email → Email, Total Purchases → Total\_Purchases\_\_c)
7. Click **Start Import**.
8. Wait until import completes → check **import status** and ensure no errors.

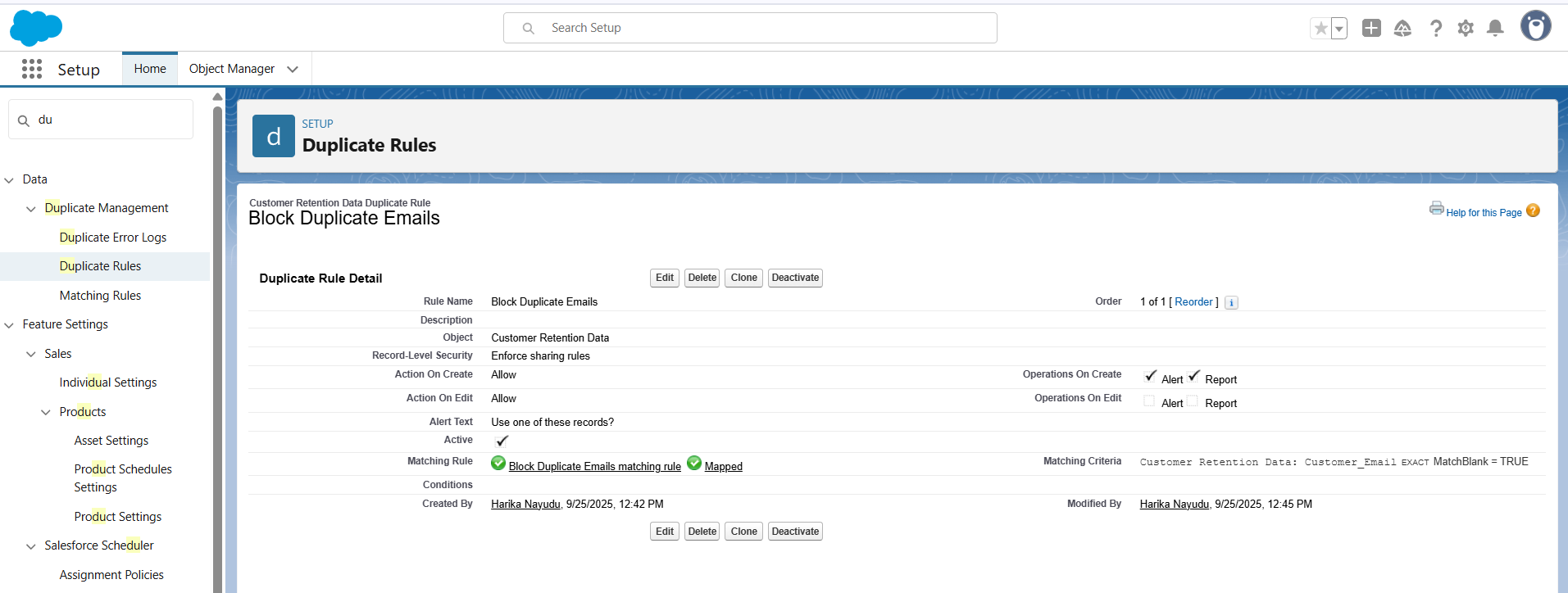




**2️⃣ Duplicate Rules**

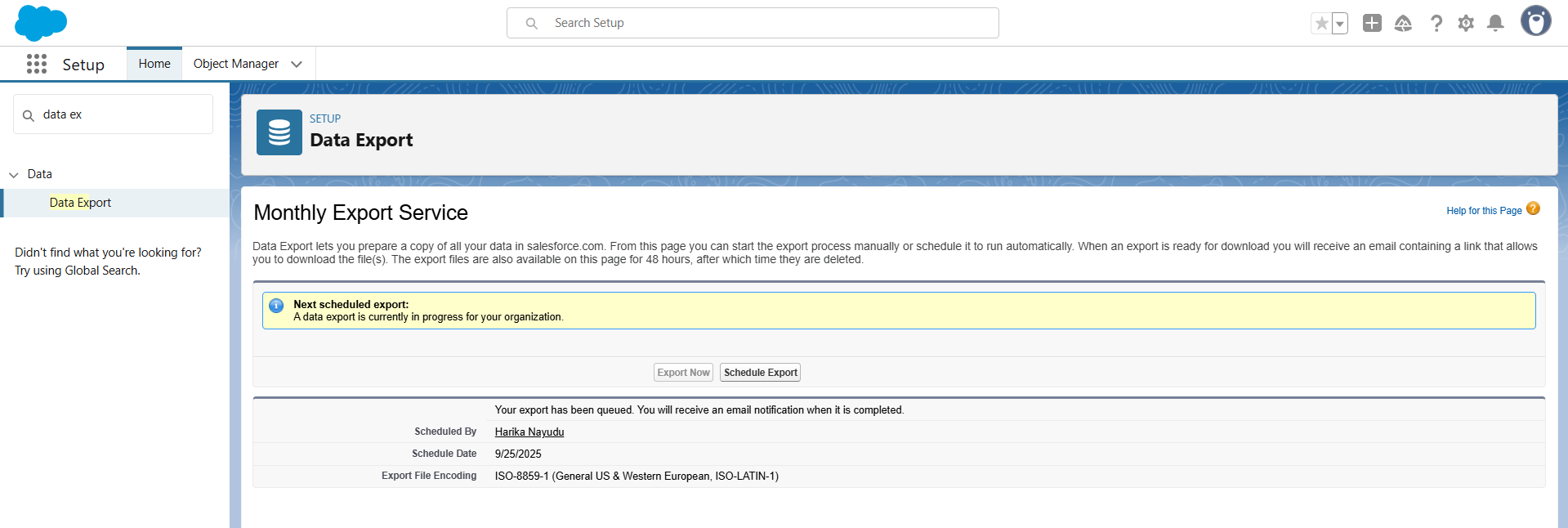
1. Go to **Setup → Quick Find → Duplicate Rules → New Rule**.
2. Select **Object:** Customer Retention Data.
3. **Rule Name:** Block Duplicate Emails (or any name you want).
4. **Matching Rule:** Match on **Email** (or any unique field) Action: **Block** → prevents saving duplicate records
5. Click **Save → Activate**.





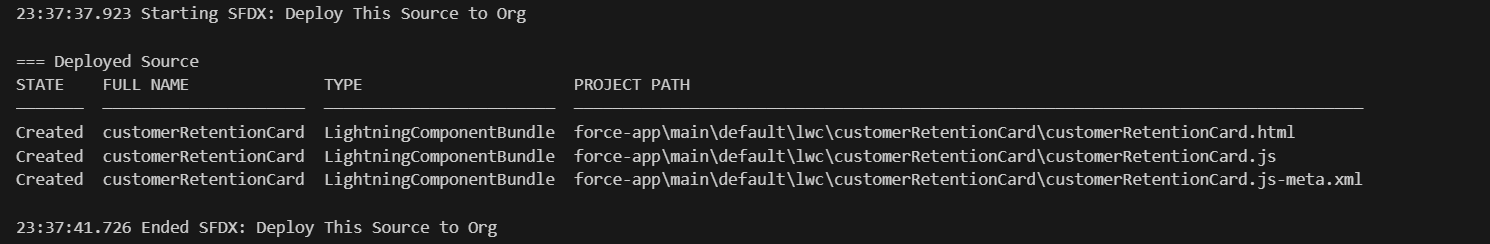
**3️⃣ Data Backup (Export)**

1. Go to **Setup → Quick Find → Data Export → Export Now**.
2. **Select Objects:**
   * Tick **Customer Retention Data** (and any other objects you want)
3. Choose **Include all data / Include attachments** if needed.
4. Click **Start Export**.
5. Wait for Salesforce to prepare the export → download ZIP file.



**4️⃣ Deployment with VS Code (SFDX)**

1. Open **VS Code → Make sure Salesforce Extension Pack is installed**.
2. Open your Salesforce project folder.
3. Right-click on the **file or folder** you want to deploy.
4. Select **SFDX: Deploy Source to Org**.
5. Wait → check the **Output panel** for success message.



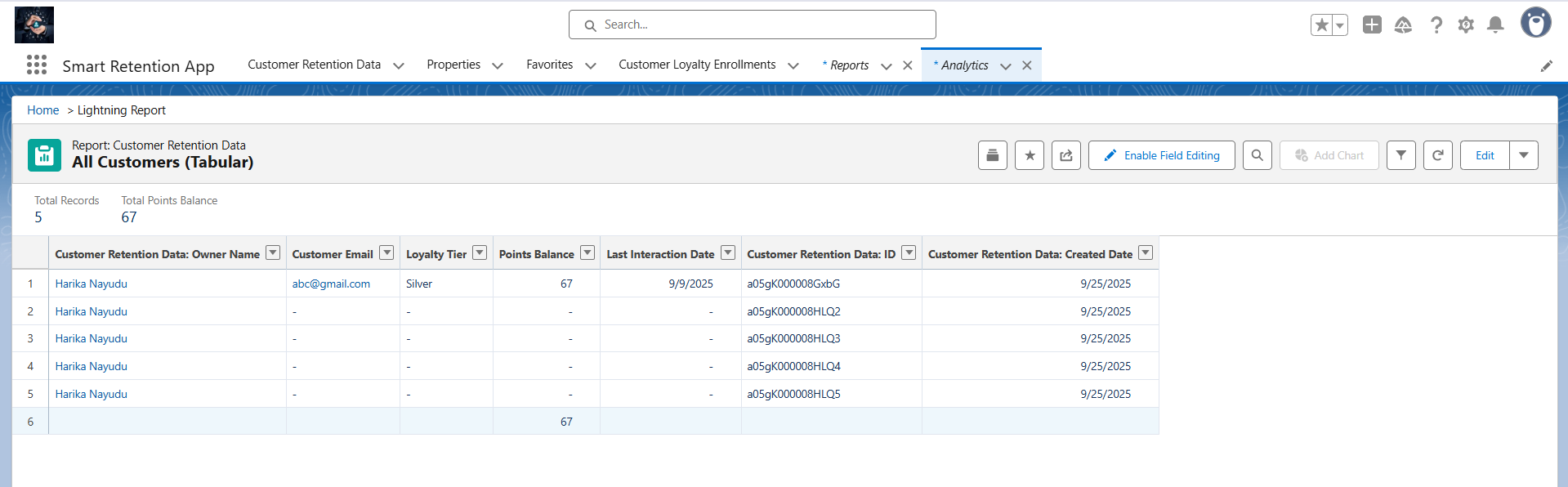
**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 9: Reporting, Dashboards & Security Review**

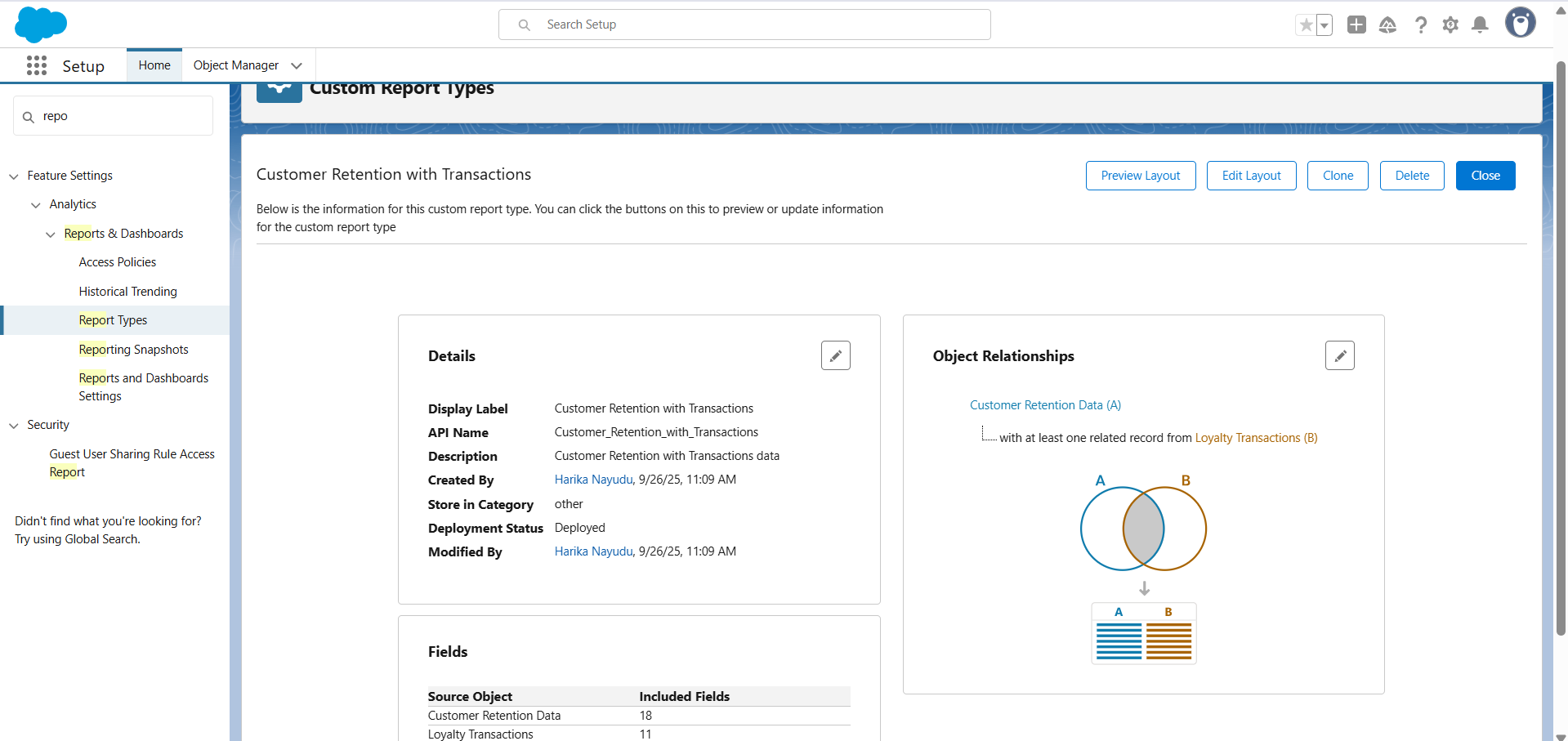
**1. Reports**

**Tabular Report (All Customers)**

1. Click **App Launcher** (waffle icon) → **Reports**.
2. Click **New Report**.
3. Search and select **Customer Retention Data** → click **Continue**.
4. In the report builder:
   * Drag columns: Customer Name, Email, Loyalty Tier, Points Balance, Last Interaction Date.
   * Remove any unwanted filters.
5. Click **Save & Run**.
   * Report Name: All Customers (Tabular) → Save.

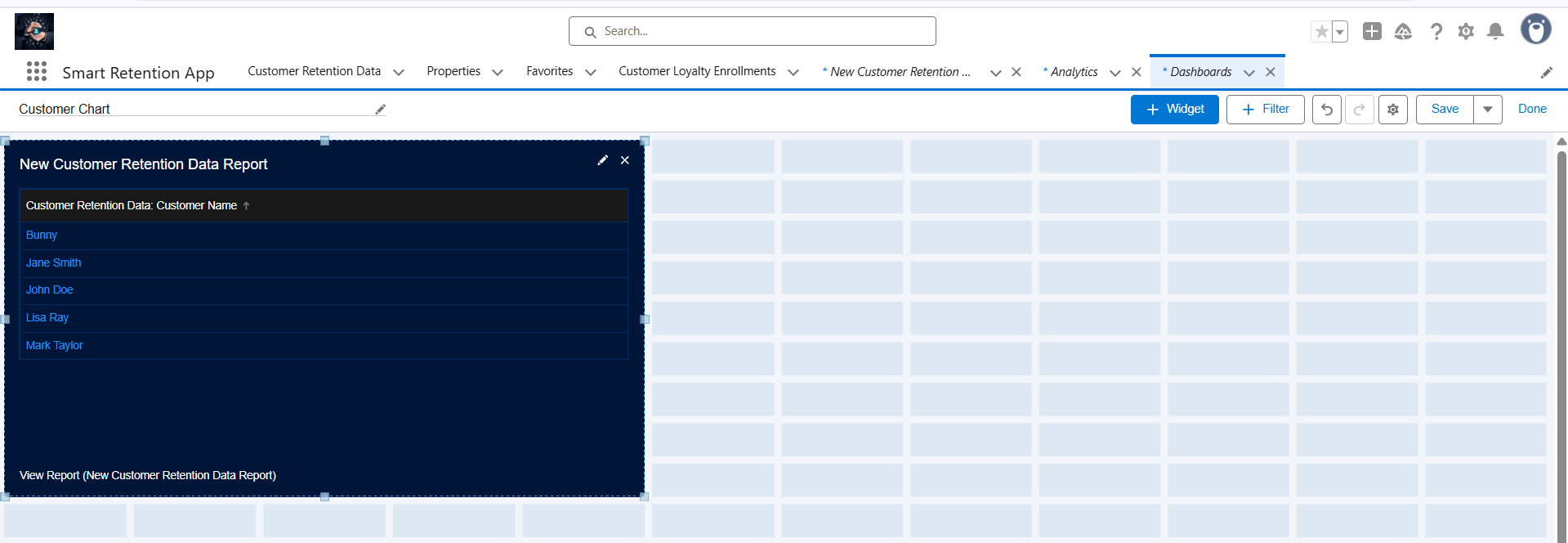
**2. Report Types**

1. Go to **Setup → Report Types** → **New Custom Report Type**.
2. Fill details:
   * **Primary Object**: Customer Retention Data.
   * **Report Type Label**: Customer Retention with Transactions.
   * Category: Other Reports.
   * Deployment Status: Deployed.
3. Click **Next**.
4. Click **Edit Layout** → Add **Loyalty Transaction** as related object.
5. Save.



**3. Dashboards**

1. App Launcher → **Dashboards** → **New Dashboard**.
2. Dashboard Name: Retention & Loyalty Dashboard.
3. Click **+ Component** → Select report Points by Loyalty Tier (Summary) → choose **Bar Chart**.
4. Click **+ Component** → Select report Churn vs Retained → choose **Pie Chart**.
5. Arrange charts → Click **Save**.



**4. Sharing Settings**

1. Setup → Search **Sharing Settings**.
2. Scroll to **Organization-Wide Defaults**.
3. Find **Customer Retention Data** → Set **Default Internal Access = Private**.
4. Save.

**Project Title: Smart Customer Retention & Loyalty Management System**

**Phase 10: Final Presentation & Demo Day**

**1. Pitch Presentation**

**Problem Statement**

Customer retention is one of the most pressing challenges faced by industries such as e-commerce, telecom, and SaaS. Businesses struggle with high churn rates due to delayed engagement, generic loyalty programs that fail to motivate customers, and manual retention processes that are both time-consuming and error-prone. In addition, organizations lack real-time visibility into customer health metrics, making it difficult for managers to identify at-risk customers and take proactive action. These issues result in revenue loss, reduced customer lifetime value, and weakened brand loyalty.

**Proposed Solution**

To address these challenges, the **Smart Customer Retention & Loyalty Management System** was developed on Salesforce. The solution leverages predictive analytics to identify at-risk customers, automated loyalty tiering to reward and retain customers effectively, and real-time dashboards to provide managers with a single view of customer health. Built using Salesforce automation tools, Apex programming, and Lightning components, the system ensures a proactive, customer-centric, and scalable approach to retention management.

**Key Features**

The system provides several important features. First, **churn detection** automatically calculates customer risk scores based on engagement activities and purchase history, while workflows and alerts notify Customer Success Managers to act quickly. Second, **loyalty tiering** enables customers to be enrolled into Bronze, Silver, Gold, or Platinum tiers automatically once they cross purchase thresholds. Third, **process automation** ensures loyalty points are updated instantly, churn alerts are triggered, and high-value redemptions follow an approval process. Finally, **dashboards and reports** offer real-time insights into churn trends, loyalty points distribution, and customer engagement metrics, empowering decision-makers with accurate, actionable data.

**Business Value**

The business impact of this solution is significant. Predictive churn scoring combined with proactive engagement can increase customer retention and improve overall ROI. Automated loyalty management enhances customer satisfaction and encourages repeat purchases, while operational efficiency improves by eliminating manual, error-prone processes. Managers and executives gain real-time visibility through dashboards, allowing them to make faster and more informed decisions. Moreover, because the solution is built on Salesforce, it is scalable and can be extended in the future with advanced features such as gamification, referral systems, and external CRM integrations.

**2. Demo Walkthrough**

**<https://drive.google.com/file/d/1wmlIsDGBuS8wf8Ir5q1vaPUwdCVu4S8Y/view?usp=sharing>**

**3. Handoff Documentation**

**1. Project Overview**

* **Project Title:** Smart Customer Retention & Loyalty Management System
* **Platform:** Salesforce (Lightning Experience)
* **Objective:** Improve customer retention and loyalty through predictive analytics, automation, and real-time dashboards.

**2. System Architecture**

* **Standard Objects:** Account, Contact
* **Custom Objects:**
  + Customer Retention Data
  + Loyalty Transaction
  + Engagement Activity
  + Customer Loyalty Enrollment (junction object)
* **Relationships:**
  + Customer Retention Data ↔ Loyalty Transaction (Lookup)
  + Customer Retention Data ↔ Engagement Activity (Master-Detail)
  + Customer Retention Data ↔ Account (Lookup)
  + Customer Loyalty Enrollment ↔ Loyalty Program (Many-to-Many)

**3. Setup & Configuration**

**Step 1: Salesforce Org Setup**

* Sign up for Salesforce Developer Edition
* Configure Lightning App: *Smart Retention App*
* Assign Permission Set: RetentionApp\_Access

**Step 2: Custom Objects & Fields**

* **Customer Retention Data** → Risk Score, Loyalty Tier, Points Balance, Churn Flag
* **Loyalty Transaction** → Transaction Type, Points Changed, Transaction Date
* **Engagement Activity** → Activity Type, Activity Date, Notes

**Step 3: Automation**

* **Validation Rule:** Prevent negative points
* **Workflow Rule:** Alert on churn flag = TRUE
* **Flows:**
  + Churn Detection Flow
  + Loyalty Points Flow
  + Auto-Enrollment Flow
* **Approval Process:** High-value redemption approvals

**4. Apex & LWC Components**

* **Triggers:** EngagementActivityTrigger
* **Handler Class:** EngagementActivityTriggerHandler
* **Queueable & Batch Apex:** Risk score updates, scheduled cleanups
* **LWC Component:** customerRetentionCard (used in Lightning Record Page)
* **Controller Class:** RetentionController

**5. Integrations**

* **Named Credential:** Loyalty\_NC → https://jsonplaceholder.typicode.com
* **REST Callouts:** Fetch loyalty-related data
* **Platform Events:** LoyaltyEvent → updates customer points via Flow

**6. Data Management**

* **Import:** Salesforce Data Import Wizard (CSV)
* **Export:** Data Export Service
* **Duplicate Rules:** Block duplicate email IDs

**7. Reporting & Dashboards**

* **Reports:**
  + All Customers (tabular)
  + Churn vs Retained Customers (summary)
  + Points by Loyalty Tier (summary)
* **Dashboard:** Retention & Loyalty Dashboard

**8. Security & Sharing**

* **Org-Wide Defaults:** Customer Retention Data = Private
* **Permission Set:** RetentionApp\_Access for CSMs
* **Role Hierarchy:** Managers > CSMs > Agents

**9. Deployment**

* Use **VS Code + Salesforce CLI (SFDX)** for deployment.
* Run **tests** before deployment (CustomerRetentionTests).
* Maintain **backup exports** before major releases.

**10. Maintenance & Next Steps**

* Monitor scheduled jobs and batch Apex logs.
* Refresh dashboards weekly for management.
* Extend loyalty program features (e.g., gamification, referral system).
* Future integrations: Payment gateways, external CRMs.