Phase 7: Integration & External Access

Objective

The objective of this phase is to **enable communication between Salesforce and external systems** (like websites, ERP, or customer portals) using secure and scalable integration methods.
This ensures that complaint and customer data can be shared, updated, and synchronized across platforms seamlessly.

1. Named Credentials

Description:

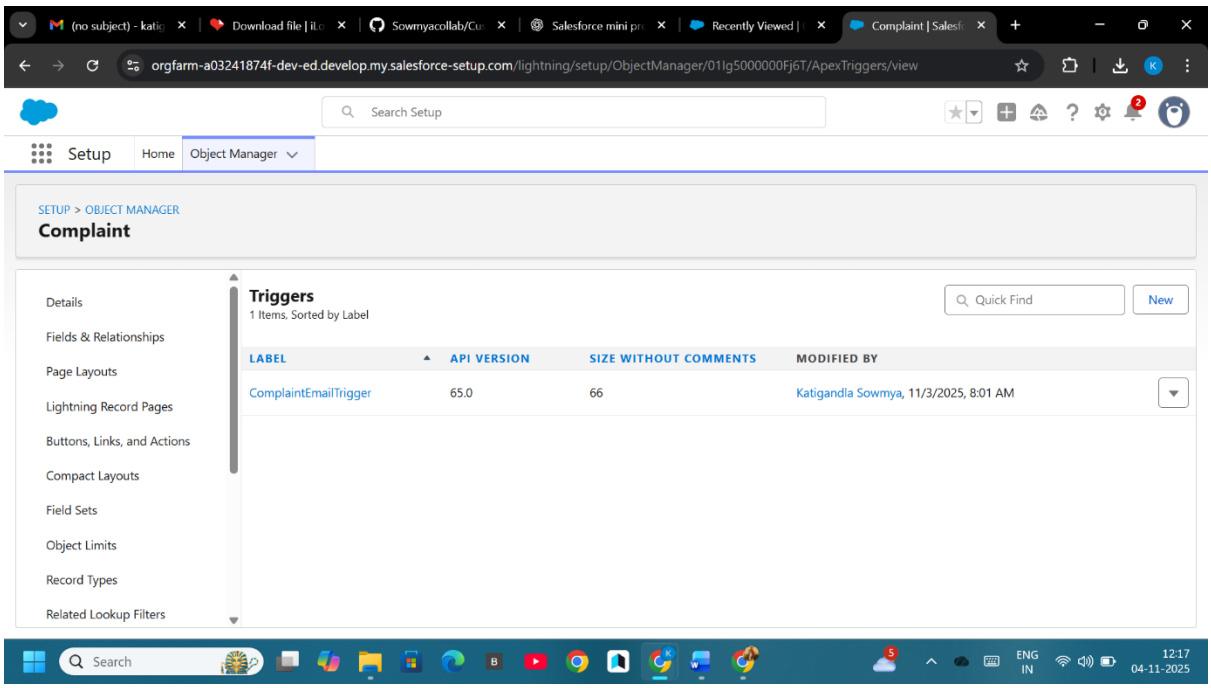
Named Credentials in Salesforce are used to **store external system credentials (like API URLs, usernames, passwords, or OAuth tokens)** in a secure and managed way.

Purpose:

- Simplifies API callouts by avoiding hardcoding authentication details.
- Centralizes management of credentials for integration endpoints.

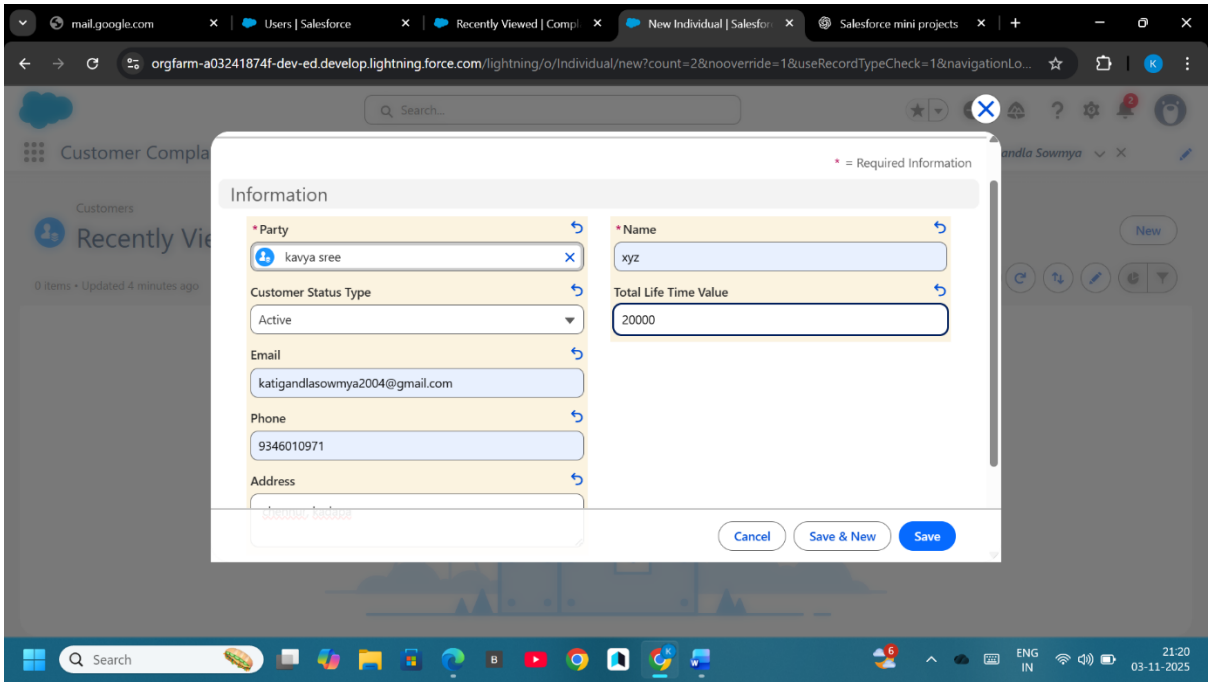
Example:

1. Go to **Setup → Named Credentials → New Named Credential**
2. Enter:
    - **Label:** External Complaint API
    - **URL:** https://externalcomplaintsapi.com
    - **Authentication:** Password or OAuth 2.0
3. Save.



Usage in Apex:

```
HttpRequest req = new HttpRequest();
req.setEndpoint('callout:External_Complaint_API/complaints');
req.setMethod('GET');
HttpResponse res = new Http().send(req);
System.debug(res.getBody());
```



🌐 2. External Services

Description:

External Services allow you to connect **Salesforce Flows** with **external REST APIs** automatically using **Schema or OpenAPI specifications**.

Purpose:

- No-code integration
- Used to send complaint data to external tracking systems.

Example:

- Import an **OpenAPI schema** of an external complaint service.
- Define actions like **"Create External Ticket"** in Flow when a new complaint is created.

💻 3. Web Services (REST / SOAP)

Description:

Salesforce can **consume (use)** or **expose (provide)** web services to integrate with other applications.

A. REST API Integration:

Used for lightweight, modern integrations with JSON.

Example (Apex REST Callout):

HttpRequest req = new HttpRequest();

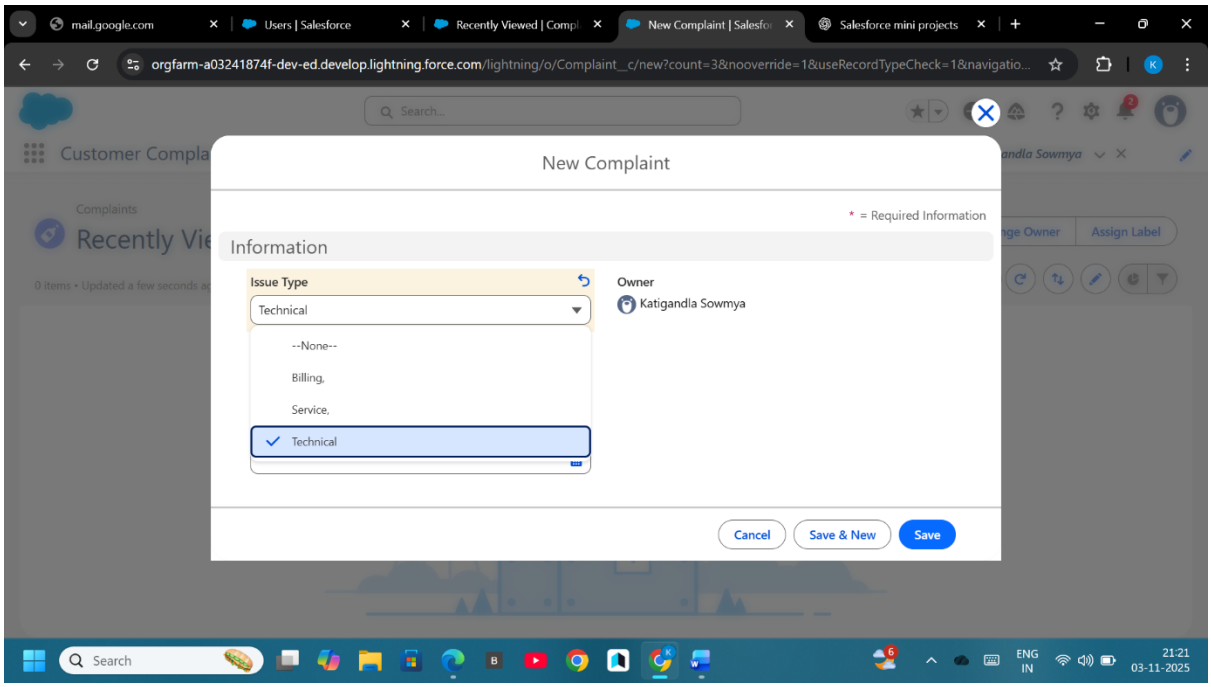
req.setEndpoint('https://externalapi.com/support');

req.setMethod('POST');

req.setHeader('Content-Type', 'application/json');

req.setBody('{"ComplaintId":"C001","Status":"New"}');

HttpResponse res = new Http().send(req);



B. SOAP Web Services:

Used for enterprise systems (ERP/legacy apps).

Example:
Salesforce exposes its **WSDL** so external systems can call its services to create or update complaints.

🔄 4. Callouts

Description:

Apex **Callouts** are used to send requests to external systems (e.g., notify a third-party CRM when a complaint is resolved).

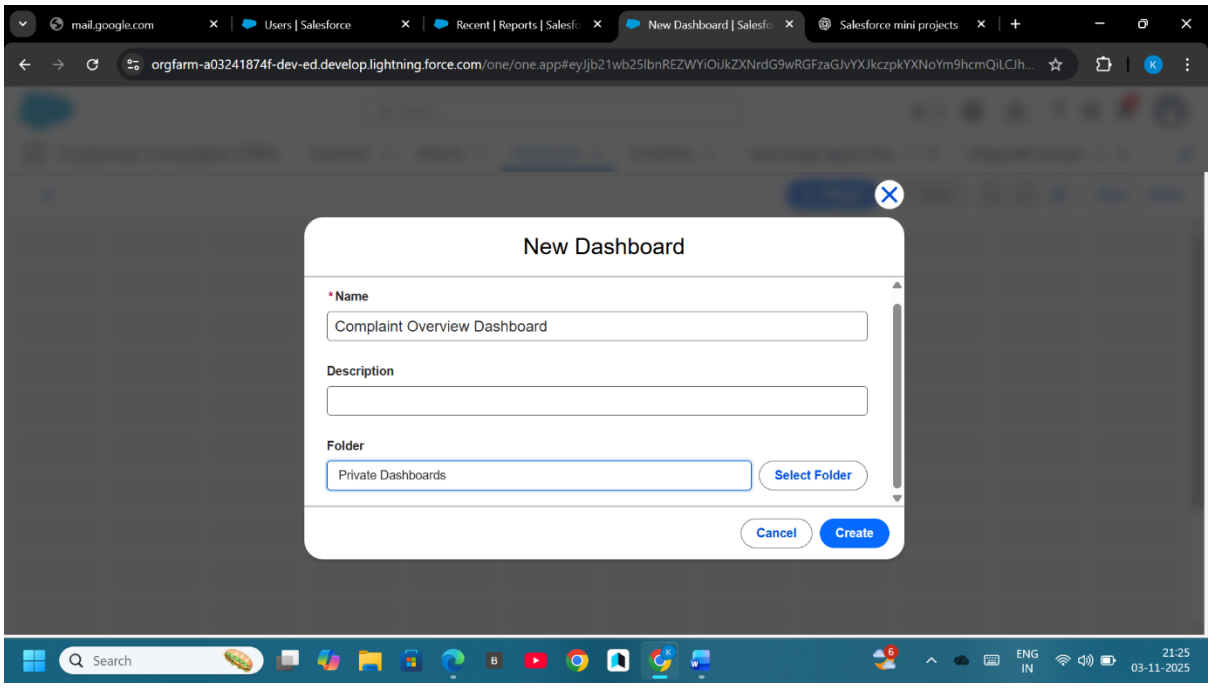
Example Use Case:

When Complaint Status = "Resolved" → Send a callout to external service to update complaint record.

```
public class ComplaintCallout {

    @future(callout=true)
```

```
    public static void notifyExternal(String complaintId) {
        Complaint__c comp = [SELECT Name, Status__c FROM Complaint__c WHERE Id =
:complaintId];
        HttpRequest req = new HttpRequest();
        req.setEndpoint('callout:External_Complaint_API/notify');
        req.setMethod('POST');
        req.setBody(JSON.serialize(comp));
        new Http().send(req);
    }
}
```



⚙ 5. Platform Events

Description:

Platform Events are Salesforce's **event-driven architecture** to communicate changes between Salesforce and other systems **in real-time**.

Use Case:

When a new complaint is created → Publish an event "ComplaintCreatedEvent" → External systems subscribe to receive the data.

Example:

```
Complaint_Event__e event = new Complaint_Event__e(
    ComplaintId__c = 'C001',
    Status__c = 'New'
```

);

Database.SaveResult result = EventBus.publish(event);

🔁 6. Change Data Capture (CDC)

Description:

Change Data Capture automatically tracks **create, update, delete** operations and shares these changes to external systems in real-time.

Use Case:

Whenever a **Complaint__c** record is updated, the change is sent to an external service or data warehouse.

Steps:

1. Go to **Setup → Change Data Capture**
2. Select **Complaint__c** object.
3. Save and subscribe using Salesforce API or middleware (like MuleSoft).

🌍 7. Salesforce Connect

Description:

Salesforce Connect allows you to **view and work with data stored in external systems** as if it were native Salesforce data — without importing it.

Use Case:

- View external **Customer** or **Complaint** records stored in an ERP or Service Database.

Steps:

1. Setup → **External Data Source → New**
2. Type: **OData 4.0**
3. Enter URL: (e.g., https://externalcrm.com/odata)
4. Validate and **Sync** tables → External Objects appear in Salesforce.

📊 8. API Limits

Description:

Salesforce imposes daily API usage limits depending on the **edition and license**.

Use Case:

Monitor API calls when external integrations are active to avoid exceeding limits.

Check Limits:

- Go to **Setup → System Overview**
- Or query:

SELECT Name, DailyApiRequests, RemainingApiRequests FROM Organization

🔒 9. OAuth & Authentication

Description:

OAuth 2.0 provides a secure way to connect Salesforce with external systems without exposing passwords.

Use Case:

Allow external apps (like a chatbot or website) to access Salesforce data securely.

Steps:

1. Setup → **App Manager → New Connected App**
2. Enable **OAuth Settings**
3. Add Callback URL & Scopes
4. Use Consumer Key/Secret in external application

🌐 10. Remote Site Settings

Description:

Salesforce blocks external callouts by default for security.
You must whitelist external URLs using **Remote Site Settings**.

Steps:

1. Setup → **Remote Site Settings → New Remote Site**
2. Remote Site Name: Complaint_API
3. URL: https://externalcomplaintsapi.com
4. Save.

Now Apex callouts to that URL are allowed.