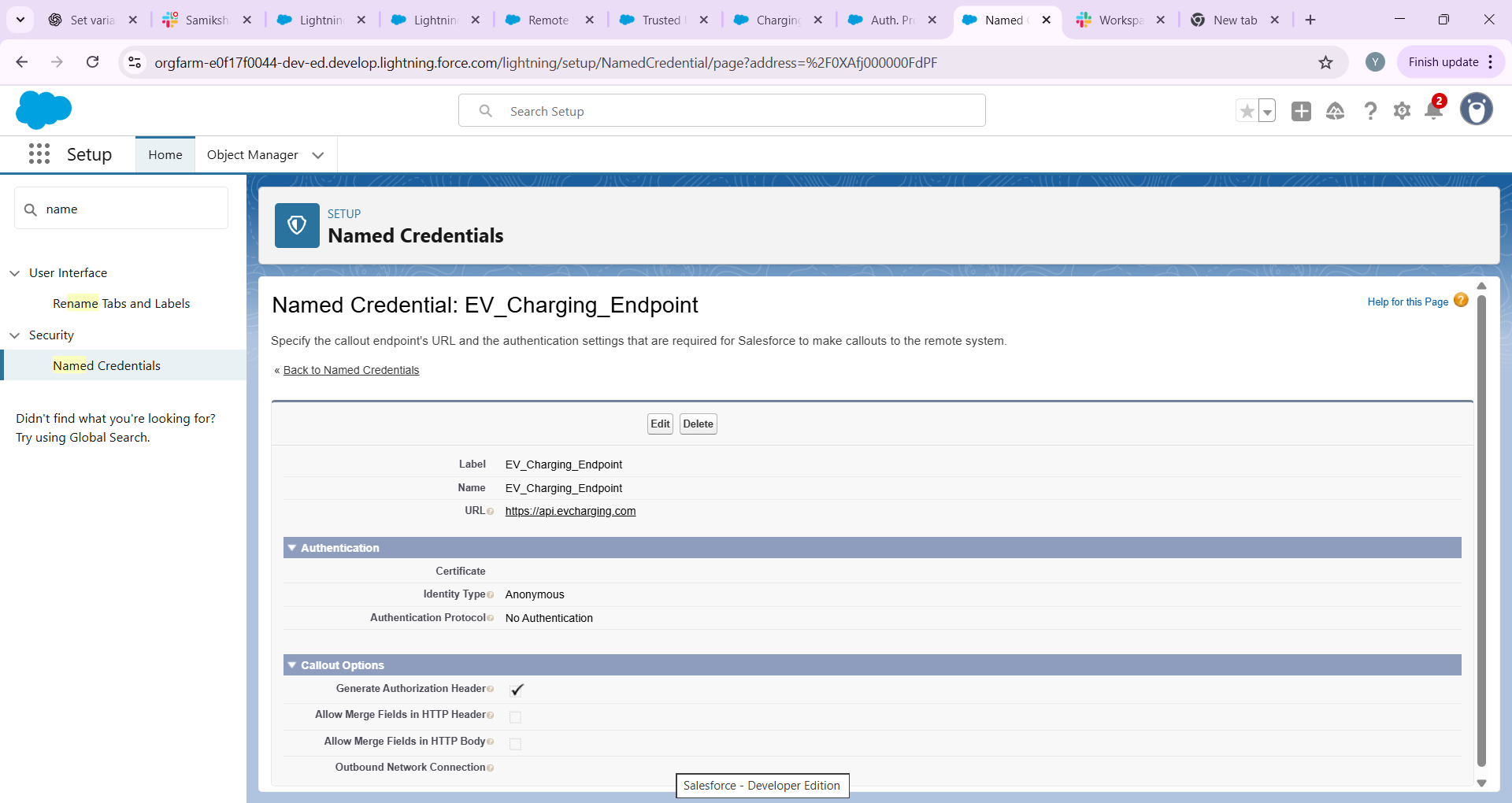
**EV Charging CRM Project – Phase 7: Integration & External Access**  
  
  
 Objective  
Implement external system integration for EV Charging CRM to connect with APIs, publish events, and track changes without relying on UI or Apex triggers (can be added in later phases).  
  
  
  
**Step 1: Named Credentials**1. Go to Setup → Security → Named Credentials.  
2. Click New Named Credential.  
3. Fill in details:  
 - Label: `Charging\_API\_Credential`  
 - Name: `Charging\_API\_Credential`  
 - URL: `https://api.evcharging.com`  
 - Identity Type: Named Principal  
 - Authentication: Password or OAuth (based on API)  
4. Click Save.  
  
  
  


**2. External Services in Salesforce**

**External Services** is a declarative tool in Salesforce that allows you to connect to external APIs without writing custom Apex code.

**Key components:**

1. **Named Credential / External Credential**
   * Stores authentication details (OAuth 2.0, API key) securely.
   * Ensures that Salesforce can call the external API safely.
2. **Open API / Swagger specification**
   * Defines the structure of the API: endpoints, methods (GET, POST), parameters, and responses.
   * Salesforce reads this schema to generate actions that can be used in Flows or Apex.
3. **Flow / Apex Integration**
   * Once registered, the API operations appear as **invocable actions** in Flow Builder or as generated Apex classes.
   * Allows users to trigger external API calls directly from Salesforce processes.

**3. Implementation Steps (Conceptual)**

1. **Create Named Credential / External Credential**
   * Normally, this stores authentication information (OAuth client ID/secret or API key).
2. **Register External Service**
   * Upload a sample OpenAPI schema that represents the external API endpoints.
   * Link it to the Named Credential for authentication.
3. **Use in Flow / Apex**
   * Drag and drop generated actions into Flows.
   * Map input/output variables to Salesforce records for automation.

*Note:* In this project, we do not have a real EV Charging API. Therefore, a **sample OpenAPI schema** is used to demonstrate the process conceptually. In a real-world scenario, the API provider would supply the actual OpenAPI specification and credentials.

**4. Sample Use Case**

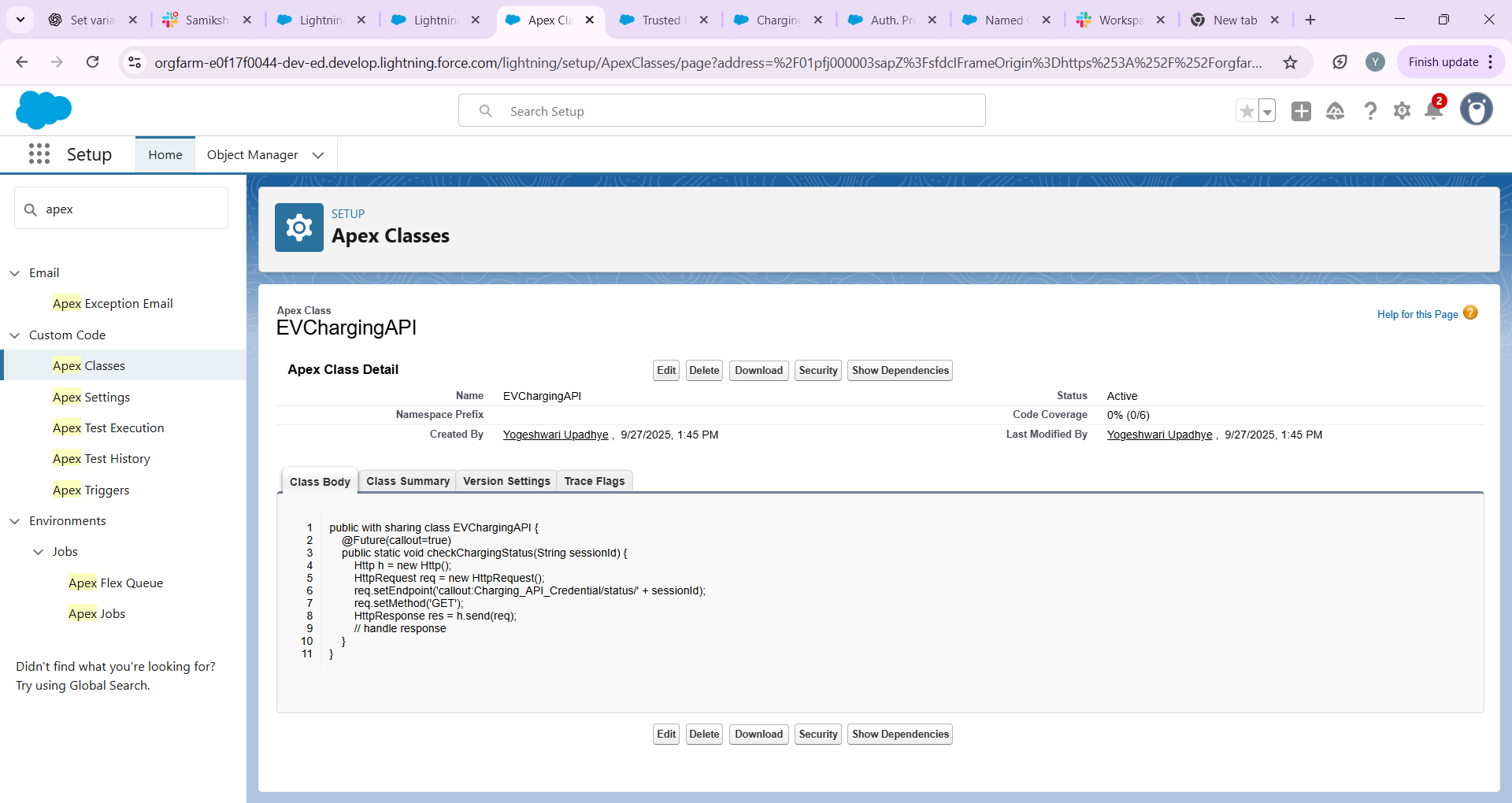
**Use Case:** Automatically calculate the total amount for a charging session using an external EV Charging API.

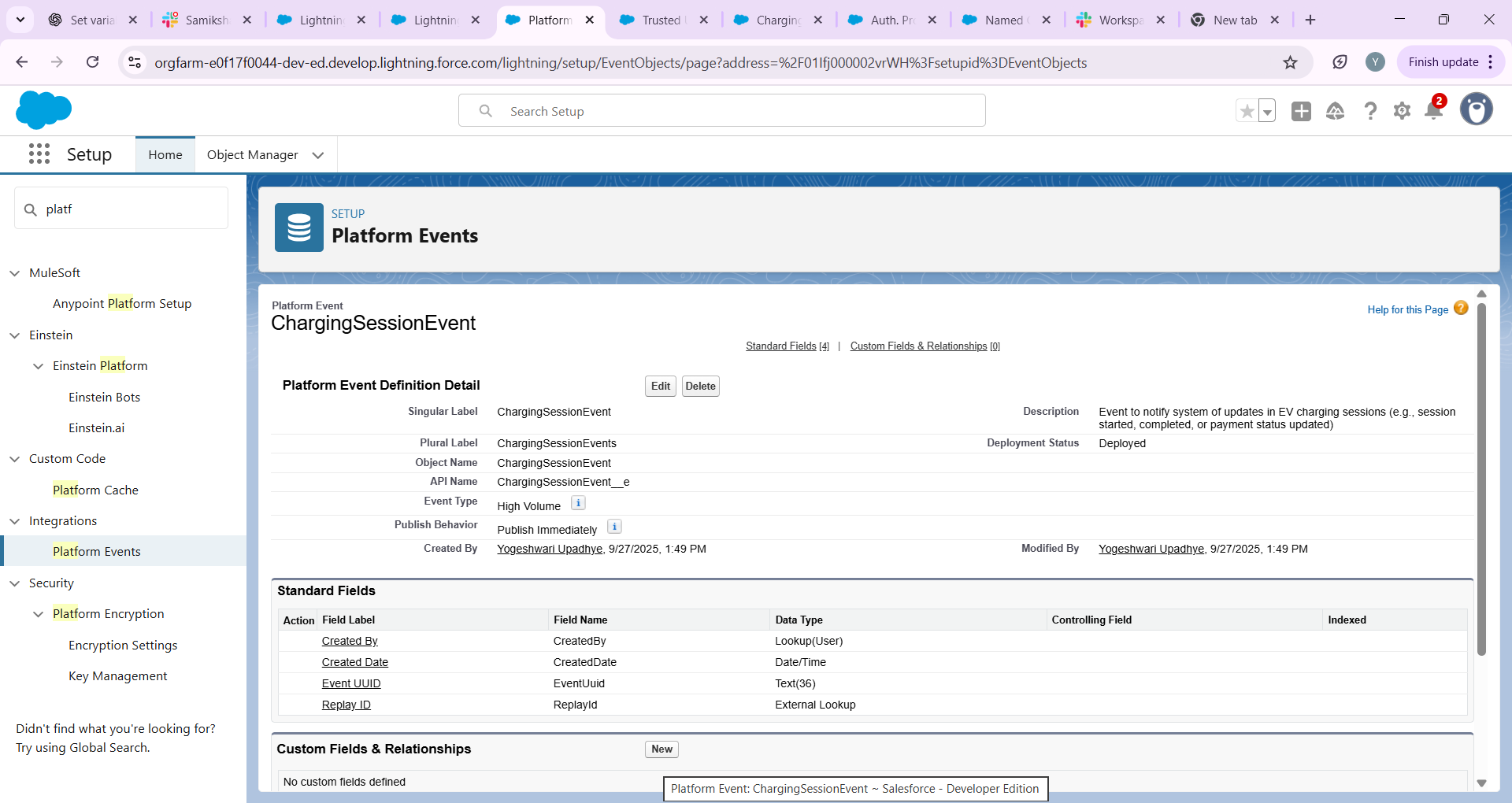
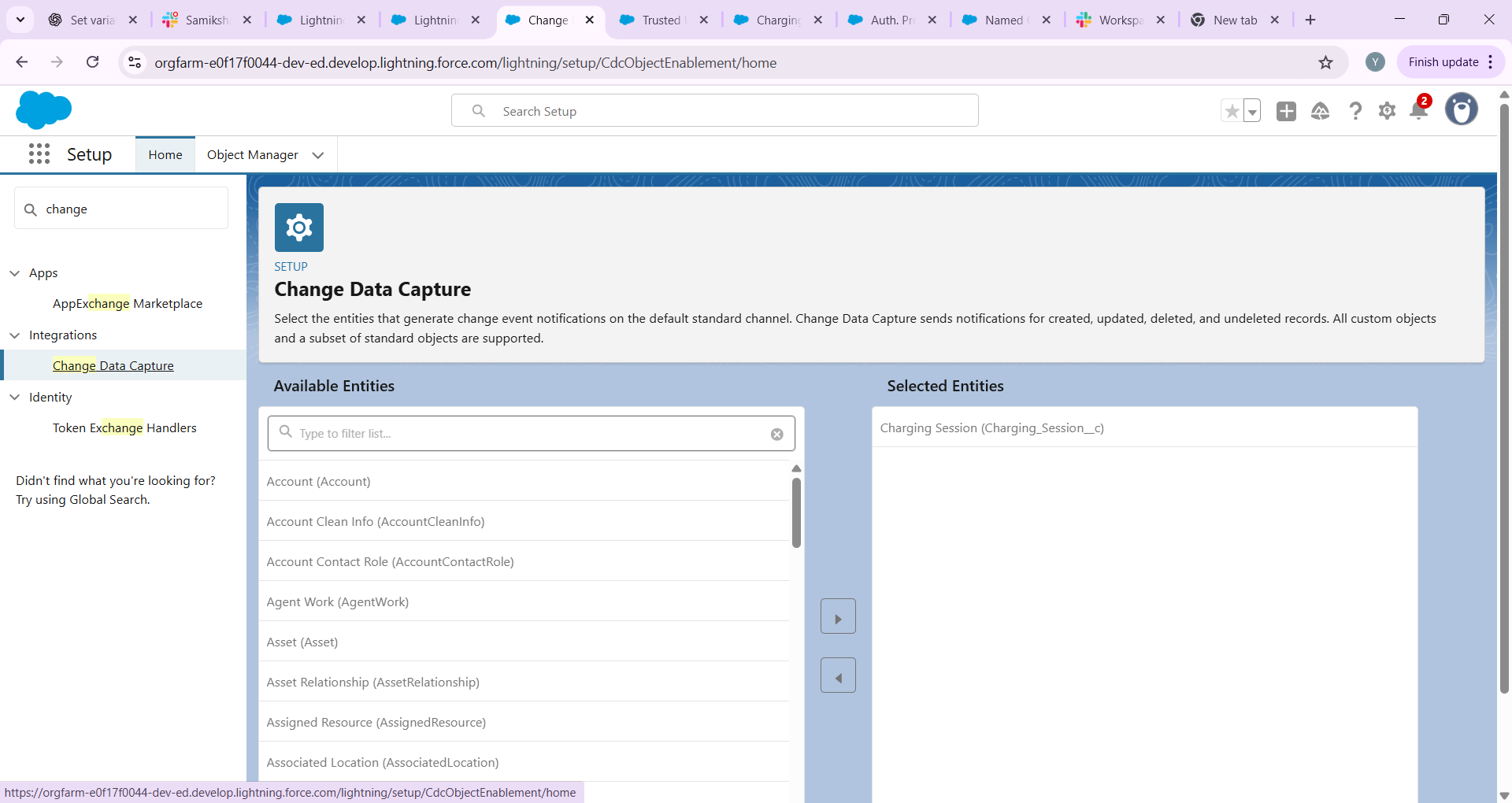
1. **Scenario:** A customer starts a charging session.
2. **Flow:** Salesforce Flow triggers an API call (via External Service) to fetch session status and calculate total energy consumed.
3. **Result:** The total amount is automatically updated in the Charging\_Session\_\_c record without manual input.

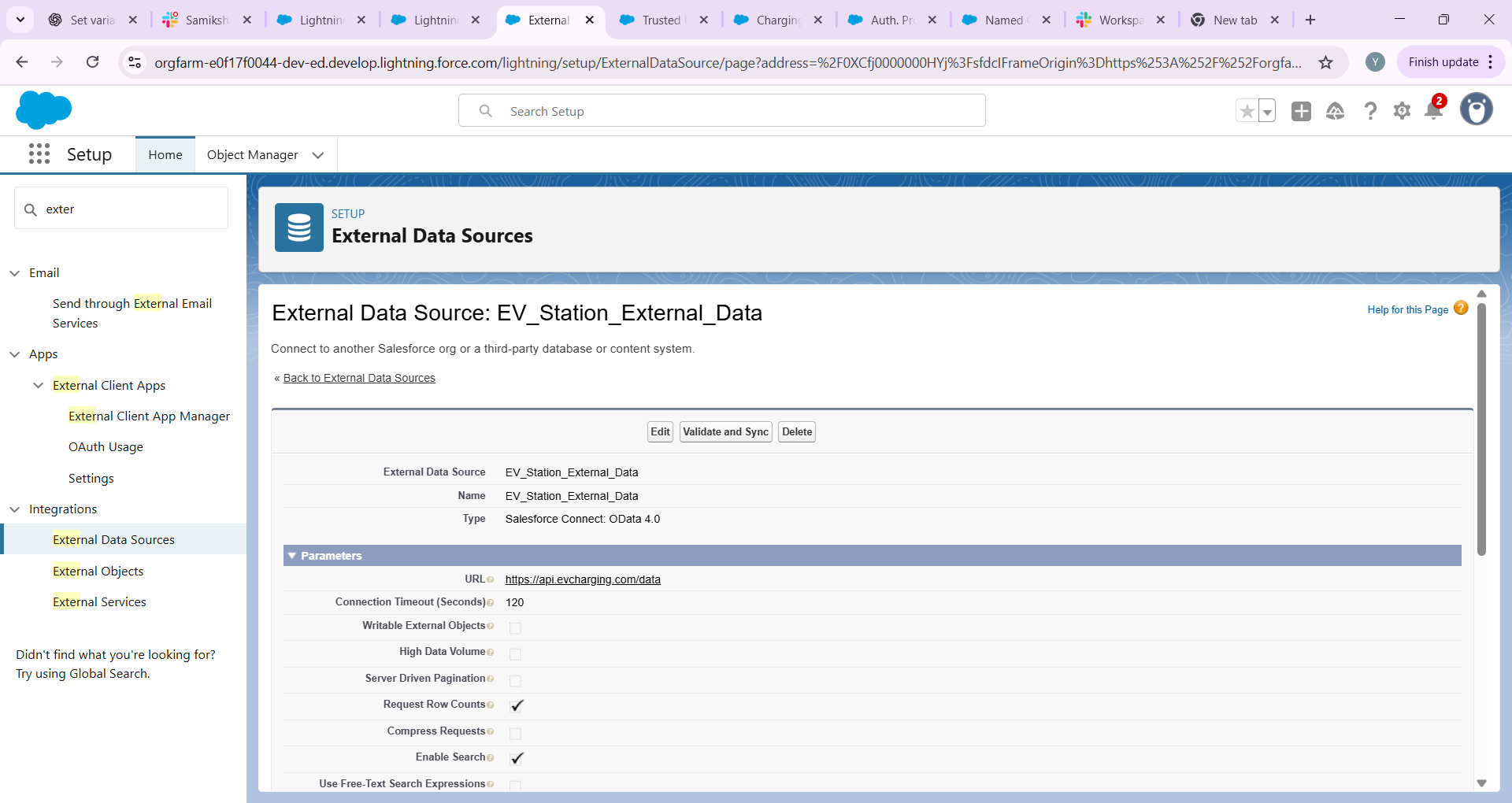
**Note:** In this demo project, the API call is simulated using a sample OpenAPI schema. The Flow is configured to show how the integration would work in a real environment.

Top of Form

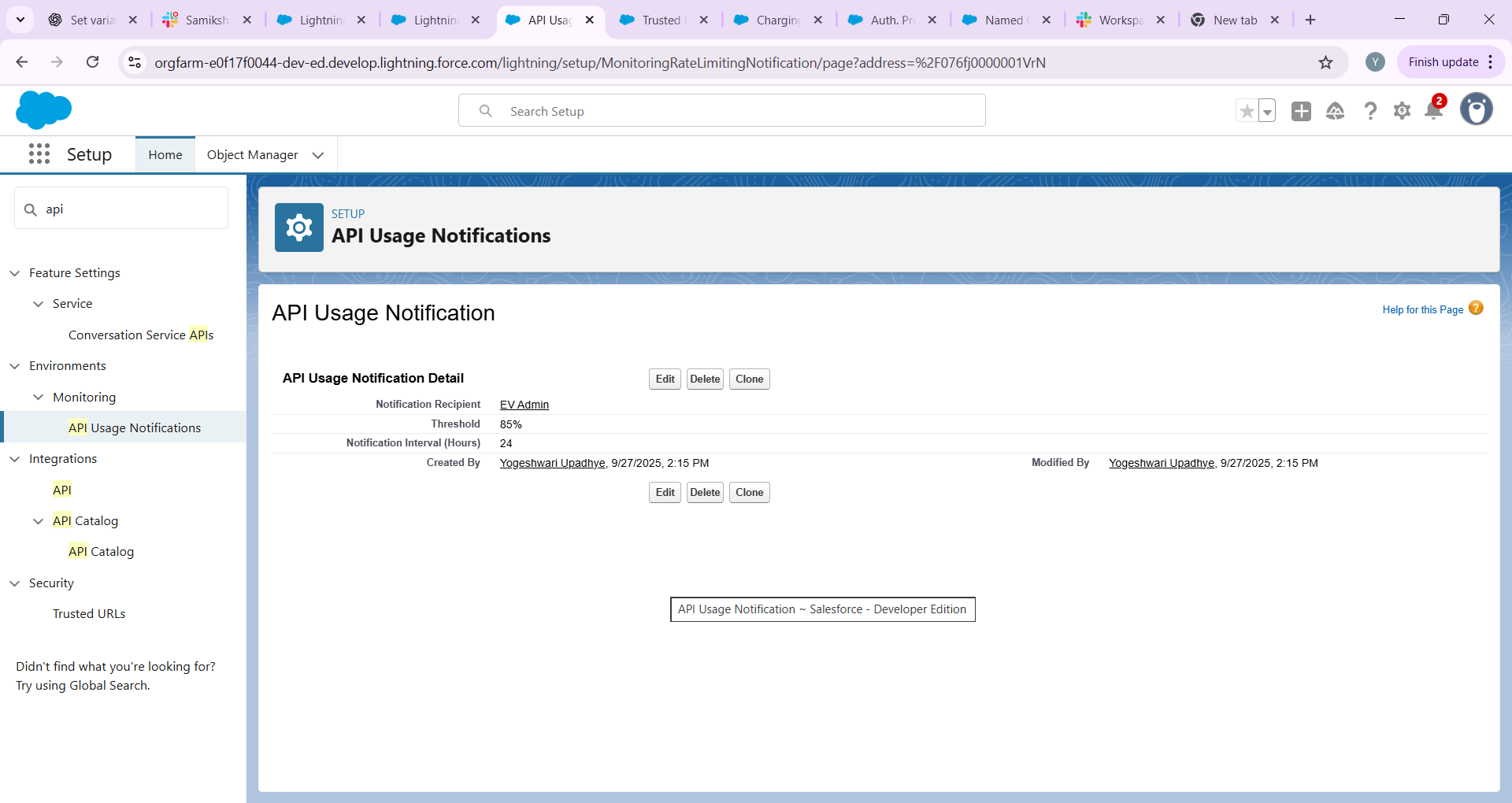
Bottom of Form

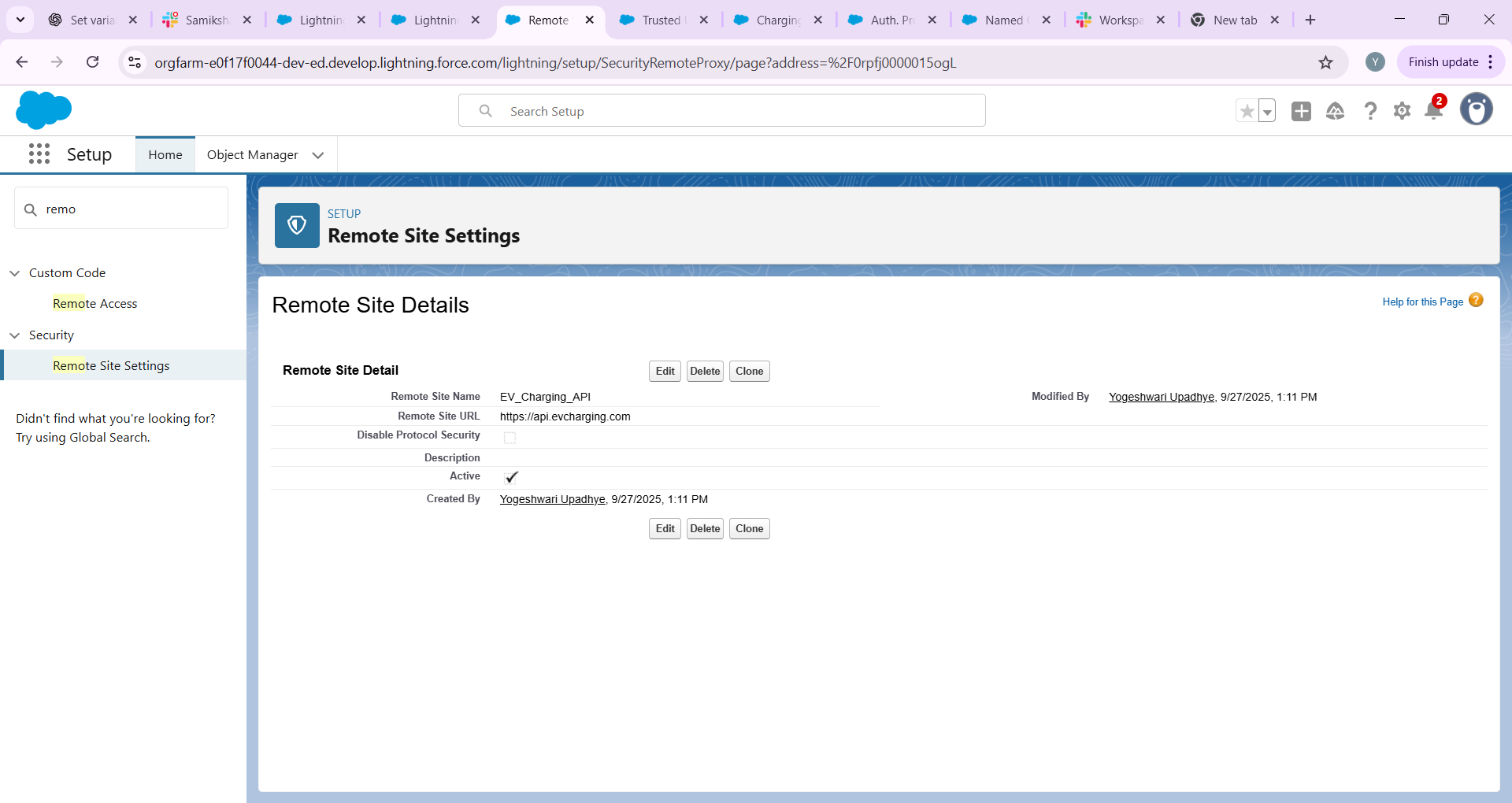
**Step 3: REST Callouts**  
1. Create Apex class for callout:  
  
```apex  
public with sharing class EVChargingAPI {  
 @Future(callout=true)  
 public static void checkChargingStatus(String sessionId) {  
 Http h = new Http();  
 HttpRequest req = new HttpRequest();  
 req.setEndpoint('callout:Charging\_API\_Credential/status/' + sessionId);  
 req.setMethod('GET');  
 HttpResponse res = h.send(req);  
 // handle response  
 }  
}  
```  
  


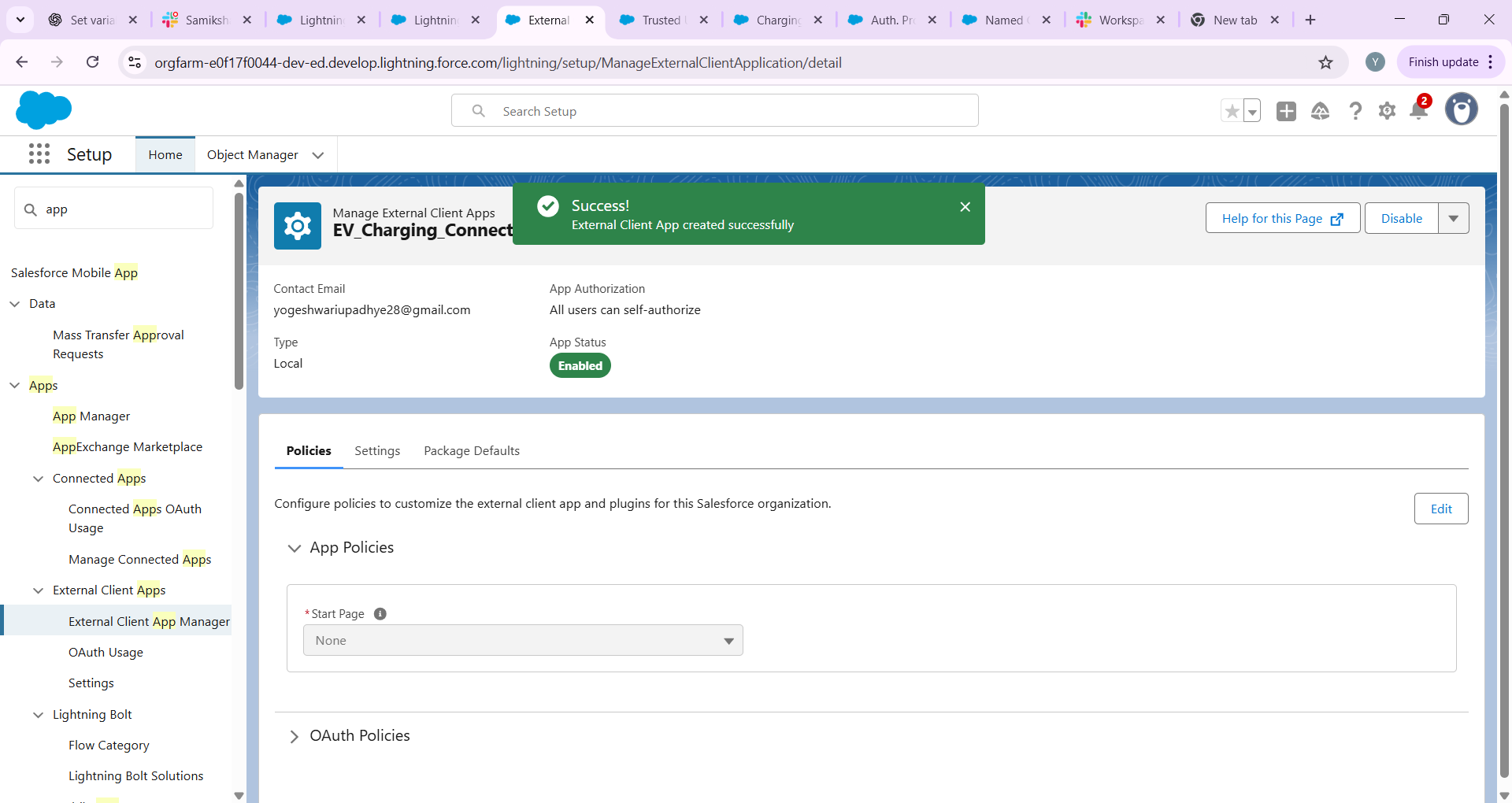
**Step 4: Platform Events**1. Setup → Platform Events → New Platform Event  
 - Name: `Charging\_Slot\_Updated\_\_e`  
 - Fields: `Slot\_ID\_\_c`, `Status\_\_c`  
2. Publish event in Flow whenever slot status changes.  
  
  
  
  
  
 **Step 5: Change Data Capture**  
1. Setup → Change Data Capture → Select Object: `Charging\_Session\_\_c`  
2. Subscribe to updates → notify external system via Flow or Apex.  
  


**Step 6: Salesforce Connect (Optional)**- Use Salesforce Connect to access external objects if EV stations data is outside Salesforce.  
  


**Step 7: API Limits, OAuth, Remote Site Settings**1. Setup → API Usage Notifications → Monitor API calls.  
2. Setup → Remote Site Settings → Add external API endpoints if not using Named Credential.  
3. Setup → Connected Apps / OAuth → For external system login.







**Future Enhancements**  
- Link UI components (Screen Flow / LWC) to trigger external API calls.  
- Add Apex triggers for advanced validation once Phase 5 is completed.  
- Use async processing for batch updates.