Phase 7 — Integration & External Access(Online Appointment Booking for legal services)
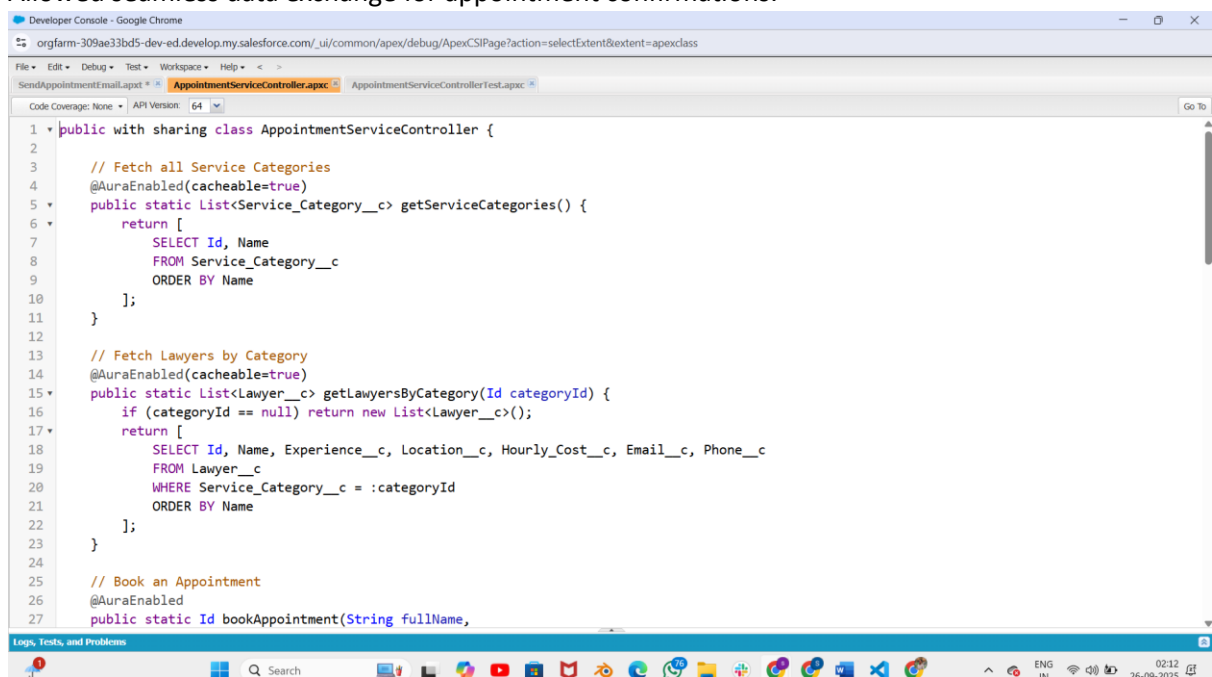
Topics Implemented

1. ### Named Credentials

   - Configured to securely store authentication credentials for external systems.

   - Allowed Apex callouts without exposing sensitive information.

2. ### External Services

   - Registered external APIs to fetch or send data for appointment verification or notifications.

   - Enabled declarative invocation via Flows and Apex.
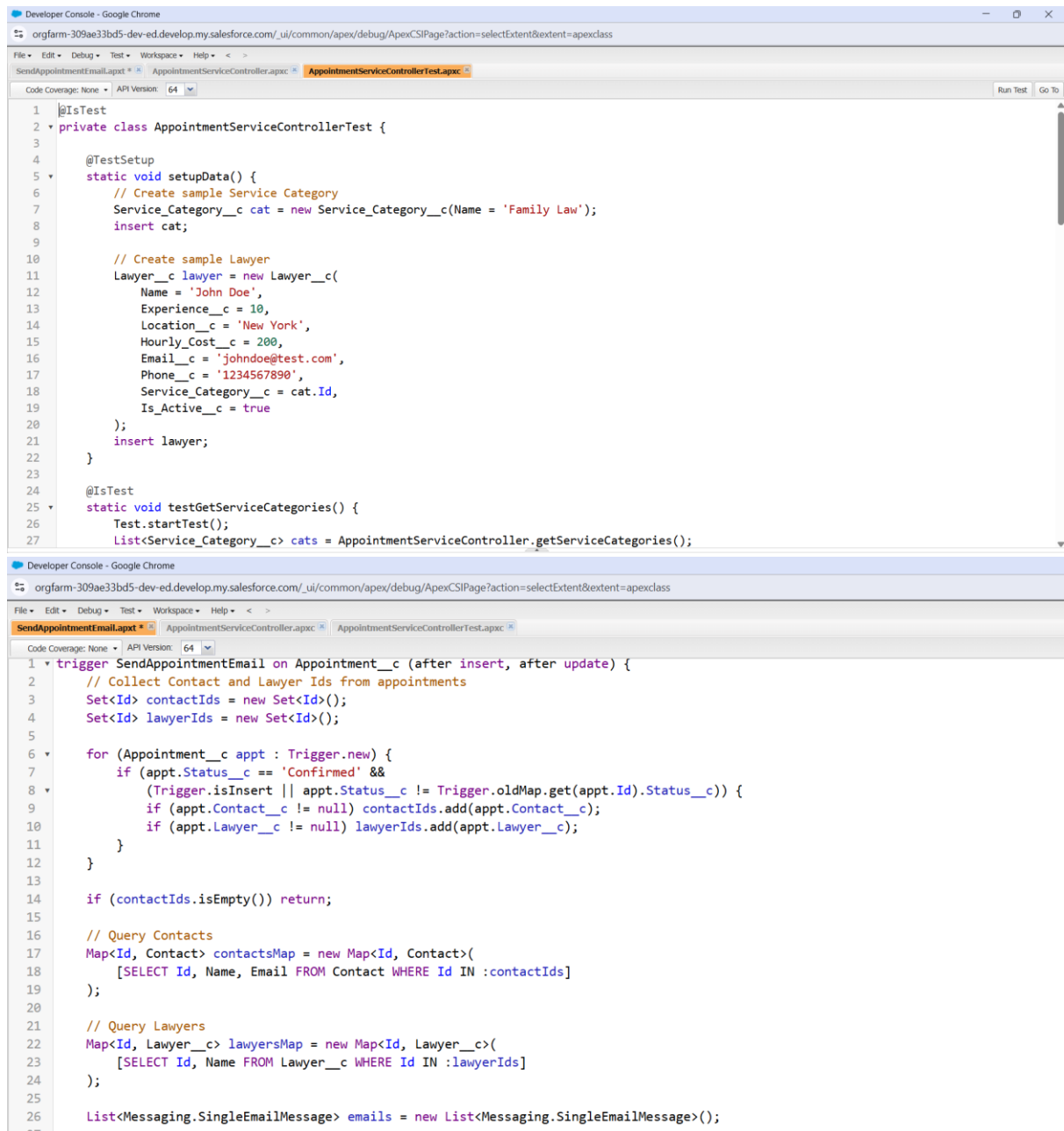
3. ### Web Services (REST/SOAP)

   - Implemented REST callouts to integrate with external systems, such as legal document services or payment gateways.

   - Allowed seamless data exchange for appointment confirmations.

```
1    @IsTest
2  ▾ private class AppointmentServiceControllerTest {
3
4        @TestSetup
5  ▾     static void setupData() {
6            // Create sample Service Category
7            Service_Category__c cat = new Service_Category__c(Name = 'Family Law');
8            insert cat;
9
10           // Create sample Lawyer
11           Lawyer__c lawyer = new Lawyer__c(
12               Name = 'John Doe',
13               Experience__c = 10,
14               Location__c = 'New York',
15               Hourly_Cost__c = 200,
16               Email__c = 'johndoe@test.com',
17               Phone__c = '1234567890',
18               Service_Category__c = cat.Id,
19               Is_Active__c = true
20           );
21           insert lawyer;
22       }
23
24       @IsTest
25 ▾     static void testGetServiceCategories() {
26           Test.startTest();
27           List<Service_Category__c> cats = AppointmentServiceController.getServiceCategories();
```



```
1  ▾ trigger SendAppointmentEmail on Appointment__c (after insert, after update) {
2        // Collect Contact and Lawyer Ids from appointments
3        Set<Id> contactIds = new Set<Id>();
4        Set<Id> lawyerIds = new Set<Id>();
5
6  ▾     for (Appointment__c appt : Trigger.new) {
7            if (appt.Status__c == 'Confirmed' &&
8  ▾             (Trigger.isInsert || appt.Status__c != Trigger.oldMap.get(appt.Id).Status__c)) {
9                if (appt.Contact__c != null) contactIds.add(appt.Contact__c);
10               if (appt.Lawyer__c != null) lawyerIds.add(appt.Lawyer__c);
11           }
12       }
13
14       if (contactIds.isEmpty()) return;
15
16       // Query Contacts
17       Map<Id, Contact> contactsMap = new Map<Id, Contact>(
18           [SELECT Id, Name, Email FROM Contact WHERE Id IN :contactIds]
19       );
20
21       // Query Lawyers
22       Map<Id, Lawyer__c> lawyersMap = new Map<Id, Lawyer__c>(
23           [SELECT Id, Name FROM Lawyer__c WHERE Id IN :lawyerIds]
24       );
25
26       List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
27
```

4. **Callouts**

- o Performed HTTP callouts to external APIs to fetch lawyer availability or send notifications.

- o Implemented error handling in Apex to ensure reliability.

5. **Platform Events**

- o Used for real-time notifications when an appointment is booked, updated, or canceled.

- o Enabled asynchronous communication with external systems or internal processes.

6. **Change Data Capture (CDC)**

    o   Subscribed to CDC events on Appointment__c and Lawyer__c.

    o   Ensured external systems stayed in sync with Salesforce data changes.

7. **Remote Site Settings**

    o   Configured approved endpoints for external API callouts.

    o   Ensured secure and authorized communication with third-party services.

Outcome:

- Appointments and lawyer data could be shared securely with external systems.

- Real-time updates and notifications ensured both clients and lawyers had accurate information.

- External integrations enhanced the functionality and reliability of the online legal services booking system.