Phase 5: Apex Programming (Developer)

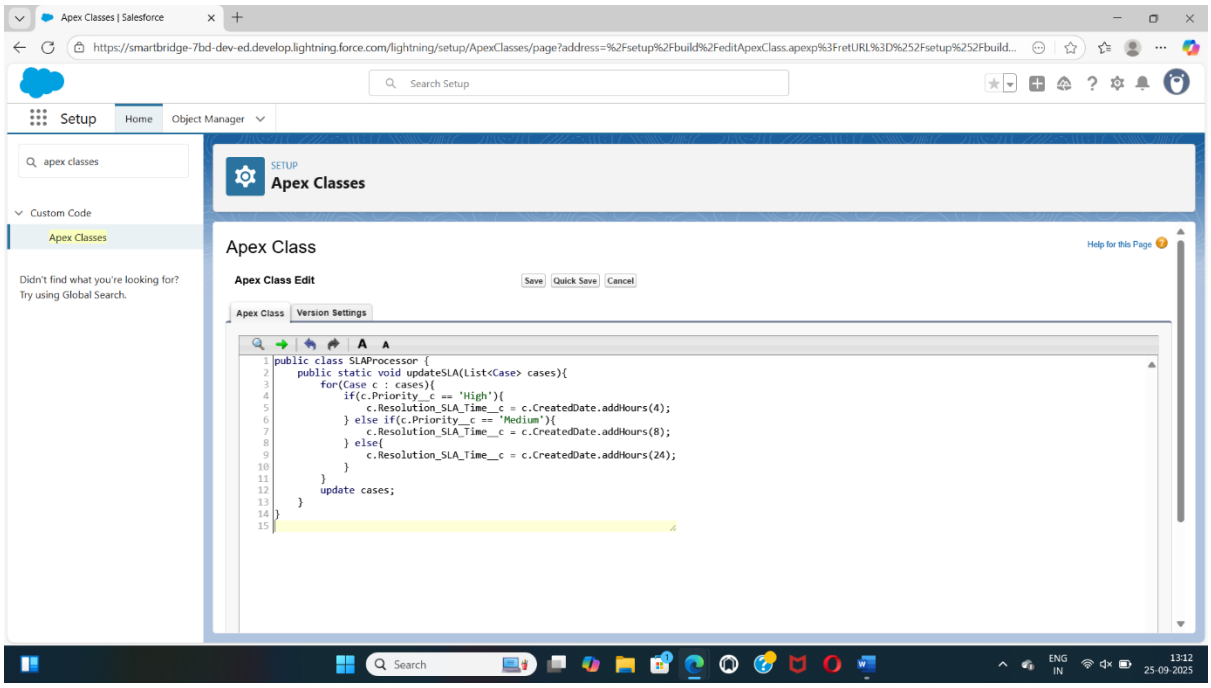
Objective:
Implement custom logic to automate case priority assignment, SLA calculation, and multi-channel notifications using Apex. This ensures that incoming customer queries from Email, WhatsApp, Chat, and Social Media are processed efficiently with minimal manual intervention.

1. Apex Trigger for Case Priority

Purpose:
Automatically set the **Priority** of a case based on the **Sentiment** of the customer message. This helps agents focus on urgent or negative cases first.

Implementation Steps:

1. Go to **Setup → Object Manager → Case → Triggers → New Trigger**
2. Trigger Name: CasePriorityTrigger
3. Select Event: before insert

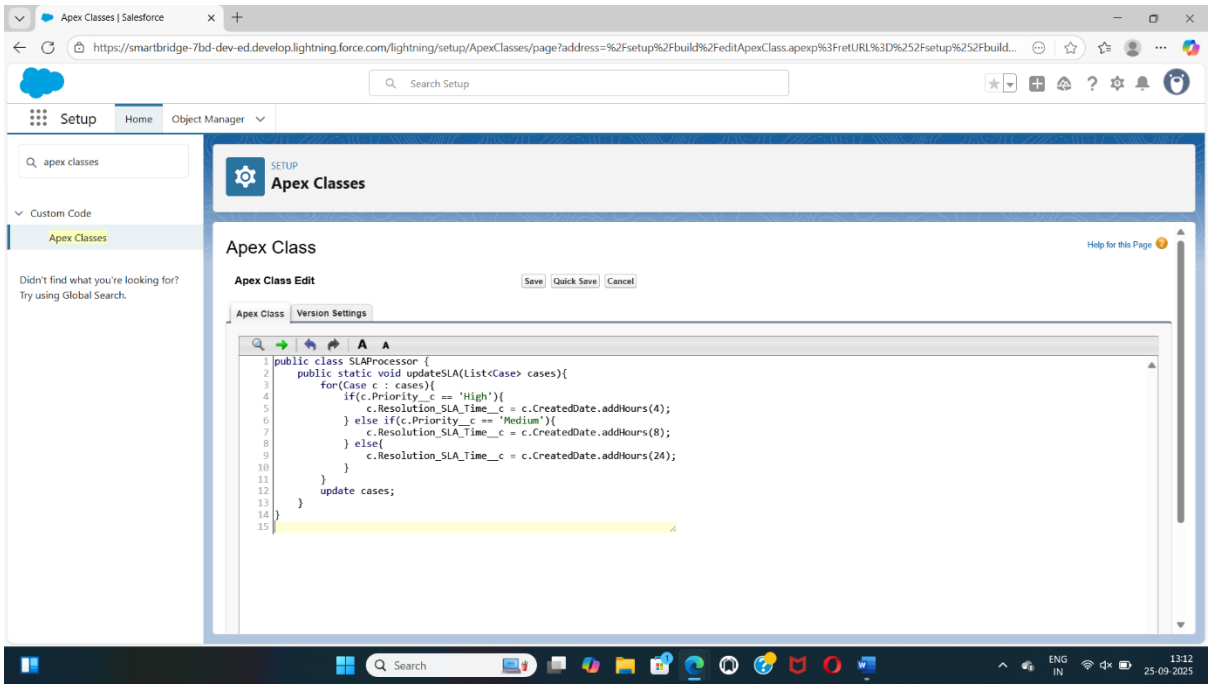


2. Apex Class for SLA Calculation

Purpose:
Automatically calculate **Resolution SLA Time** based on the priority of a case to ensure timely response.

Implementation Steps:

1. Go to **Setup → Apex Classes → New**
2. Class Name: SLAProcessor



Important Notes:

- Resolution_SLA_Time_c is a **custom Date/Time field** in the Case object.
- This class can be called from **triggers** or **flows** to automatically update SLA when a case is created.

3. Apex Integration for WhatsApp Notifications

Purpose:
Send automated messages via WhatsApp when a new case is created. Ensures customers receive timely confirmation of their query.

Implementation Steps:

1. Setup → **Named Credentials → Add Twilio/WhatsApp API credentials**
2. Create Apex class WhatsAppNotifier using HTTP callouts:

```
public class WhatsAppNotifier {
    public static void sendMessage(String toNumber, String message) {
        HttpRequest req = new HttpRequest();
        req.setEndpoint('callout:TwilioAPI/Messages');
        req.setMethod('POST');
        req.setBody('To=' + toNumber + '&Body=' + message);
        Http http = new Http();
        HttpResponse res = http.send(req);
    }
}
```

3.Call this class from **Case trigger or Flow** after Case creation.

4. Test Classes

Purpose:
Salesforce requires **minimum 75% code coverage** for Apex deployment. Test classes ensure triggers and classes work correctly.

Implementation Steps:

1. Create a test class TestCasePriorityTrigger

```
@isTest
public class TestCasePriorityTrigger {
    @isTest static void testCasePriority() {
        Case c = new Case(
            Subject = 'Test WhatsApp Case',
            Sentiment__c = 'Negative'
        );
        insert c;
        System.assertEquals('High', c.Priority);
    }
}
```

2. **p → Apex Test Execution**
3. Confirm code coverage ≥75% and that logic works correctly.

Expected Output / Result:

- New Cases automatically get **Priority** based on sentiment.
- **SLA resolution time** is calculated and updated automatically.
- Customers receive **WhatsApp notifications** immediately.
- Developers can safely deploy Apex logic with test coverage confirmed.