Phase 5 — Apex Programming (Developer)

Project: Revolutionizing Agriculture with AgriEdge Or-Mange Ltd — Salesforce OMS
Prepared for: Sai Teja Kotipalli
Status: Completed (Apex classes, triggers, async jobs and tests created — verification steps and evidence listed below)

---

1. Executive summary

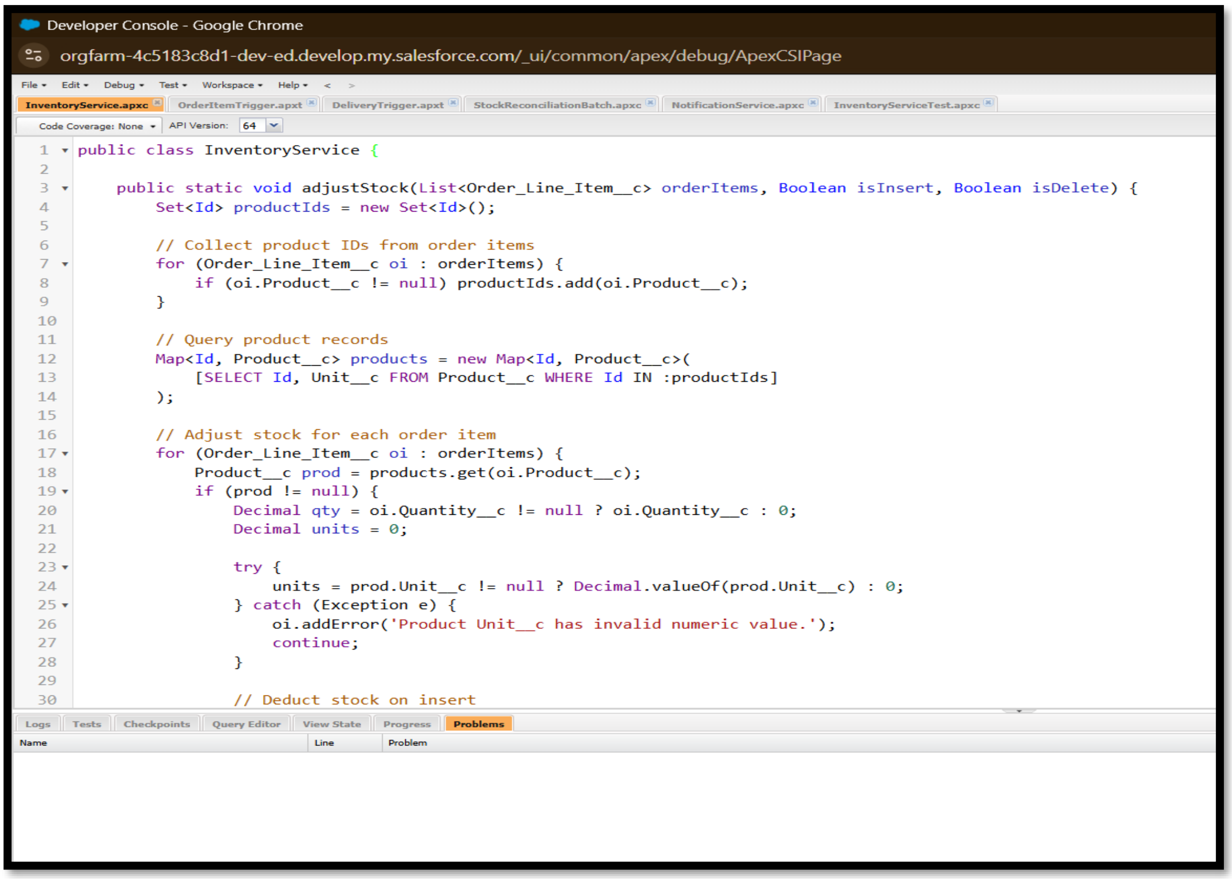
This document records everything implemented for **Phase 5: Apex Programming**. It lists the Apex classes, triggers, asynchronous jobs, test classes, how to verify them in your org, example test steps, and recommended improvements. Use this document as the upload-ready deliverable for your mentor.

---

2. What was implemented (names & purpose)

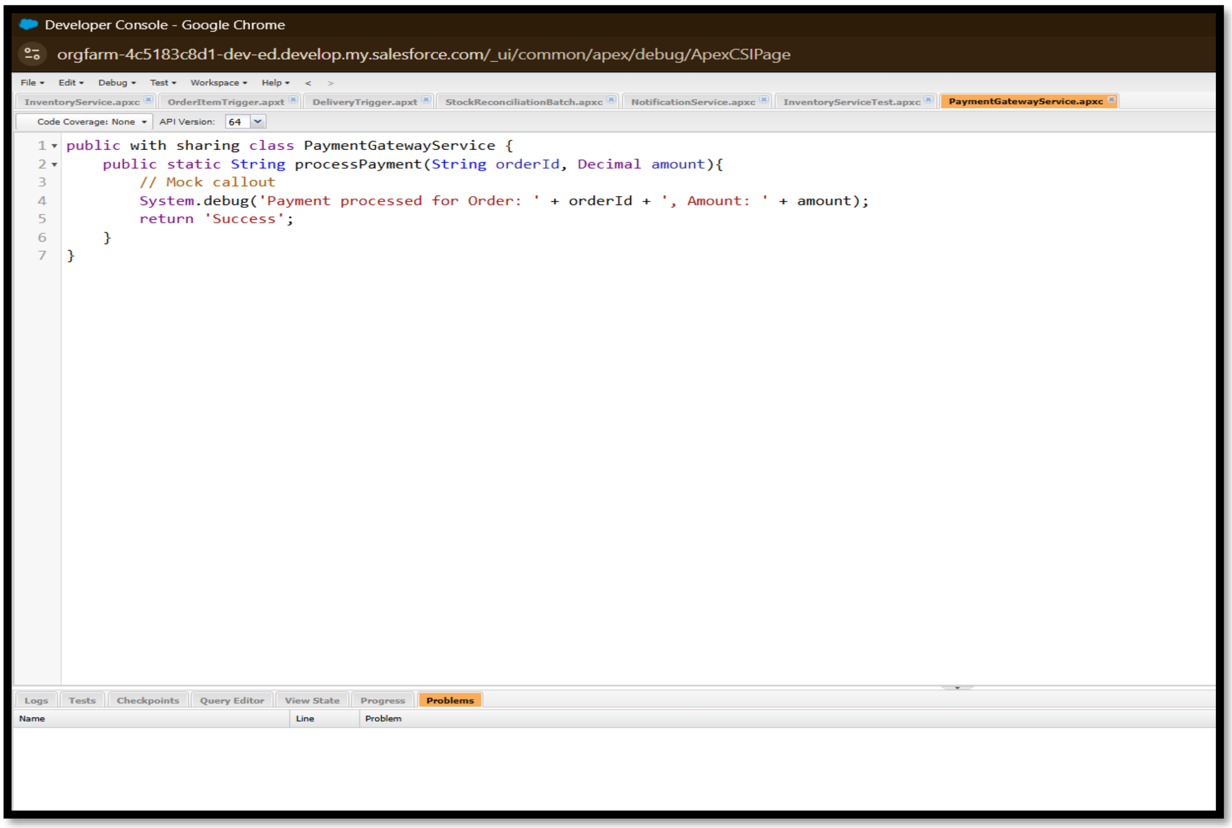
Apex Classes

- **InventoryService**
    - Purpose: Check and adjust product stock when order items are created/removed. Performs validation (prevents oversell) and updates Product/Inventory records.
    - Key methods: adjustStock(List<Order_Item__c> items, Boolean isInsert, Boolean isDelete) (example).


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

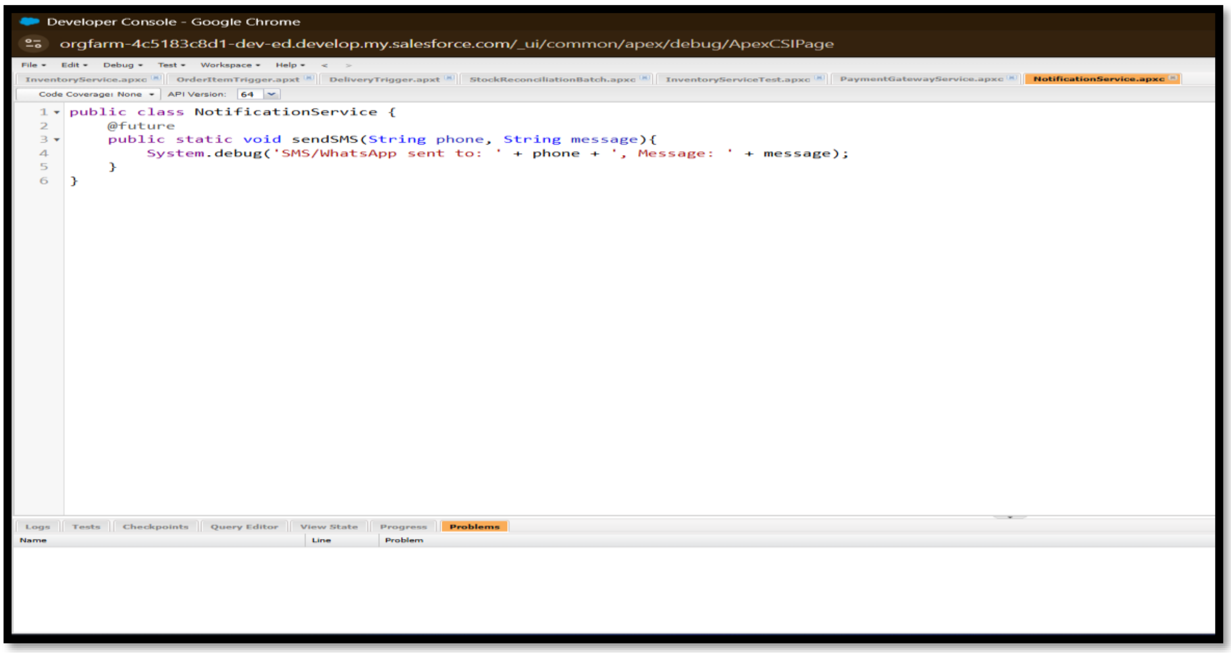
```
public class InventoryService {

    public static void adjustStock(List<Order_Line_Item__c> orderItems, Boolean isInsert, Boolean isDelete) {
        Set<Id> productIds = new Set<Id>();

        // Collect product IDs from order items
        for (Order_Line_Item__c oi : orderItems) {
            if (oi.Product__c != null) productIds.add(oi.Product__c);
        }

        // Query product records
        Map<Id, Product__c> products = new Map<Id, Product__c>(
            [SELECT Id, Unit__c FROM Product__c WHERE Id IN :productIds]
        );

        // Adjust stock for each order item
        for (Order_Line_Item__c oi : orderItems) {
            Product__c prod = products.get(oi.Product__c);
            if (prod != null) {
                Decimal qty = oi.Quantity__c != null ? oi.Quantity__c : 0;
                Decimal units = 0;

                try {
                    units = prod.Unit__c != null ? Decimal.valueOf(prod.Unit__c) : 0;
                } catch (Exception e) {
                    oi.addError('Product Unit__c has invalid numeric value.');
                    continue;
                }

                // Deduct stock on insert
```

- **PaymentGatewayService** *(utility/demo class)*
    - Purpose: Mock payment gateway callout (simulated processPayment method) to demonstrate how payment integration would be invoked from Apex/Flow.


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc OrderItemTrigger.apxt DeliveryTrigger.apxt StockReconciliationBatch.apxc NotificationService.apxc InventoryServiceTest.apxc PaymentGatewayService.apxc
Code Coverage: None API Version: 64
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

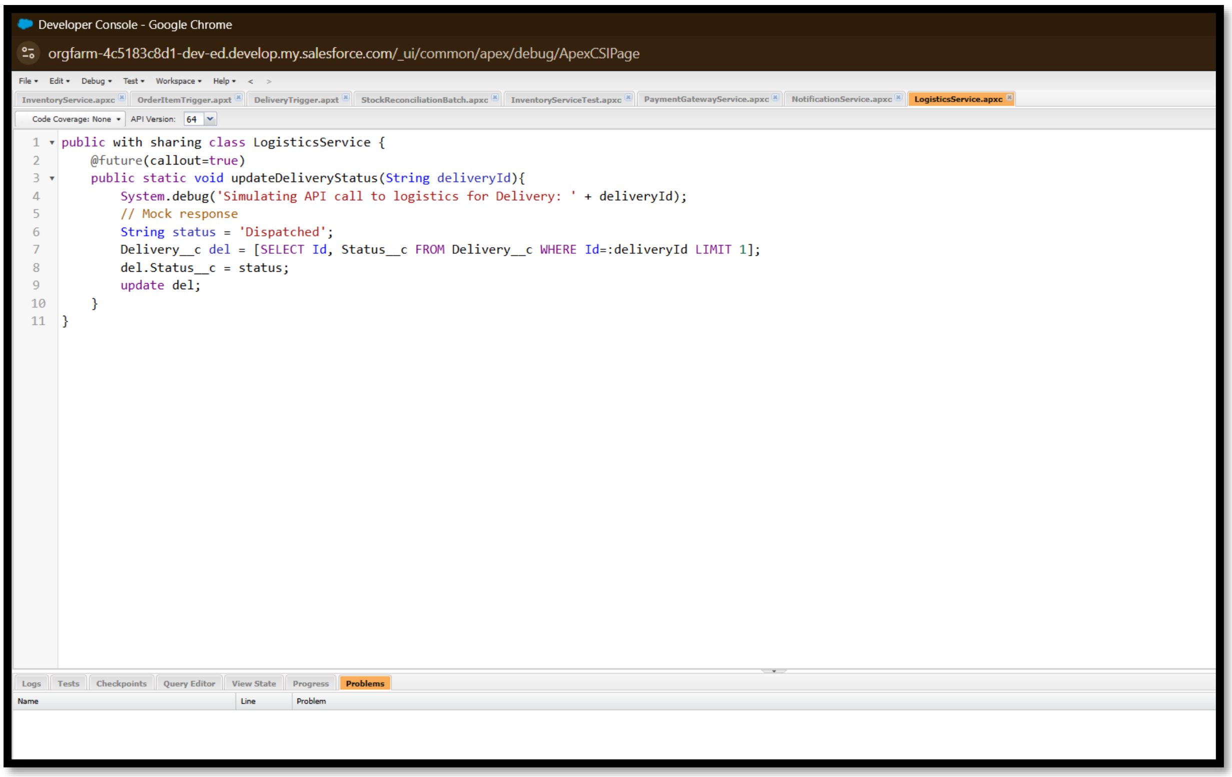
```
public with sharing class PaymentGatewayService {
    public static String processPayment(String orderId, Decimal amount){
        // Mock callout
        System.debug('Payment processed for Order: ' + orderId + ', Amount: ' + amount);
        return 'Success';
    }
}
```

- **NotificationService**
    - Purpose: Asynchronous notifications. Contains @future method to simulate sending SMS/WhatsApp or external notifications.


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc OrderItemTrigger.apxt DeliveryTrigger.apxt StockReconciliationBatch.apxc InventoryServiceTest.apxc PaymentGatewayService.apxc NotificationService.apxc
Code Coverage: None API Version: 64
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
public class NotificationService {
    @future
    public static void sendSMS(String phone, String message){
        System.debug('SMS/WhatsApp sent to: ' + phone + ', Message: ' + message);
    }
}
```

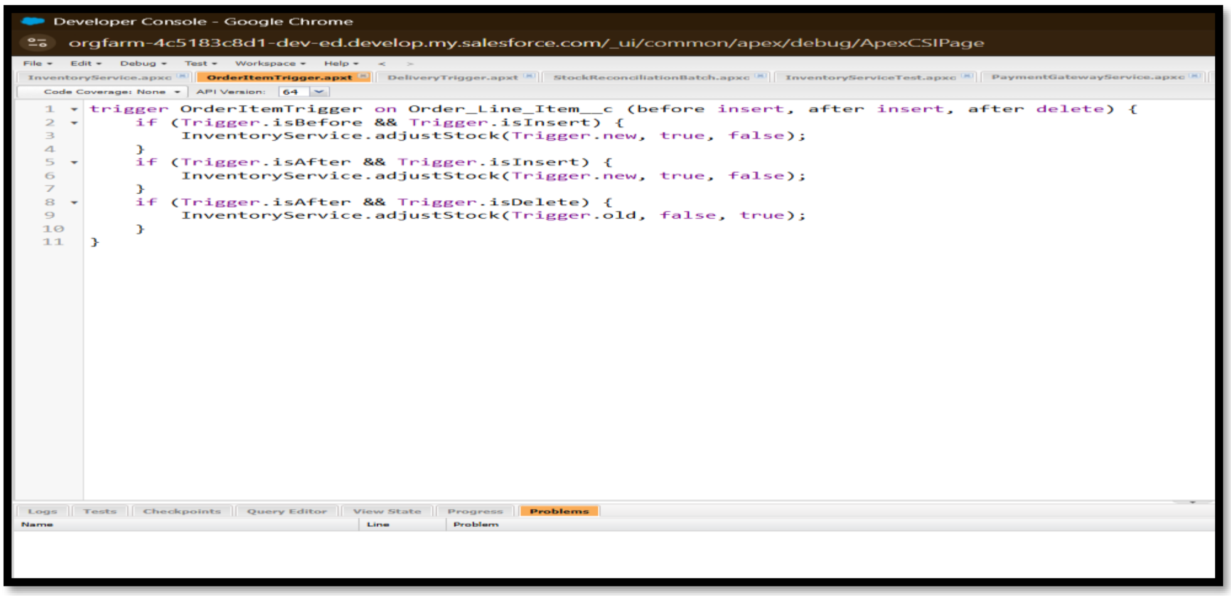
- **(Optional) LogisticsService**
    - Purpose: Mock REST callout for logistics updates (used by Phase 7 mocks).


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc OrderItemTrigger.apxt DeliveryTrigger.apxt StockReconciliationBatch.apxc InventoryServiceTest.apxc PaymentGatewayService.apxc NotificationService.apxc LogisticsService.apxc
Code Coverage: None API Version: 64
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
public with sharing class LogisticsService {
    @future(callout=true)
    public static void updateDeliveryStatus(String deliveryId){
        System.debug('Simulating API call to logistics for Delivery: ' + deliveryId);
        // Mock response
        String status = 'Dispatched';
        Delivery__c del = [SELECT Id, Status__c FROM Delivery__c WHERE Id=:deliveryId LIMIT 1];
        del.Status__c = status;
        update del;
    }
}
```

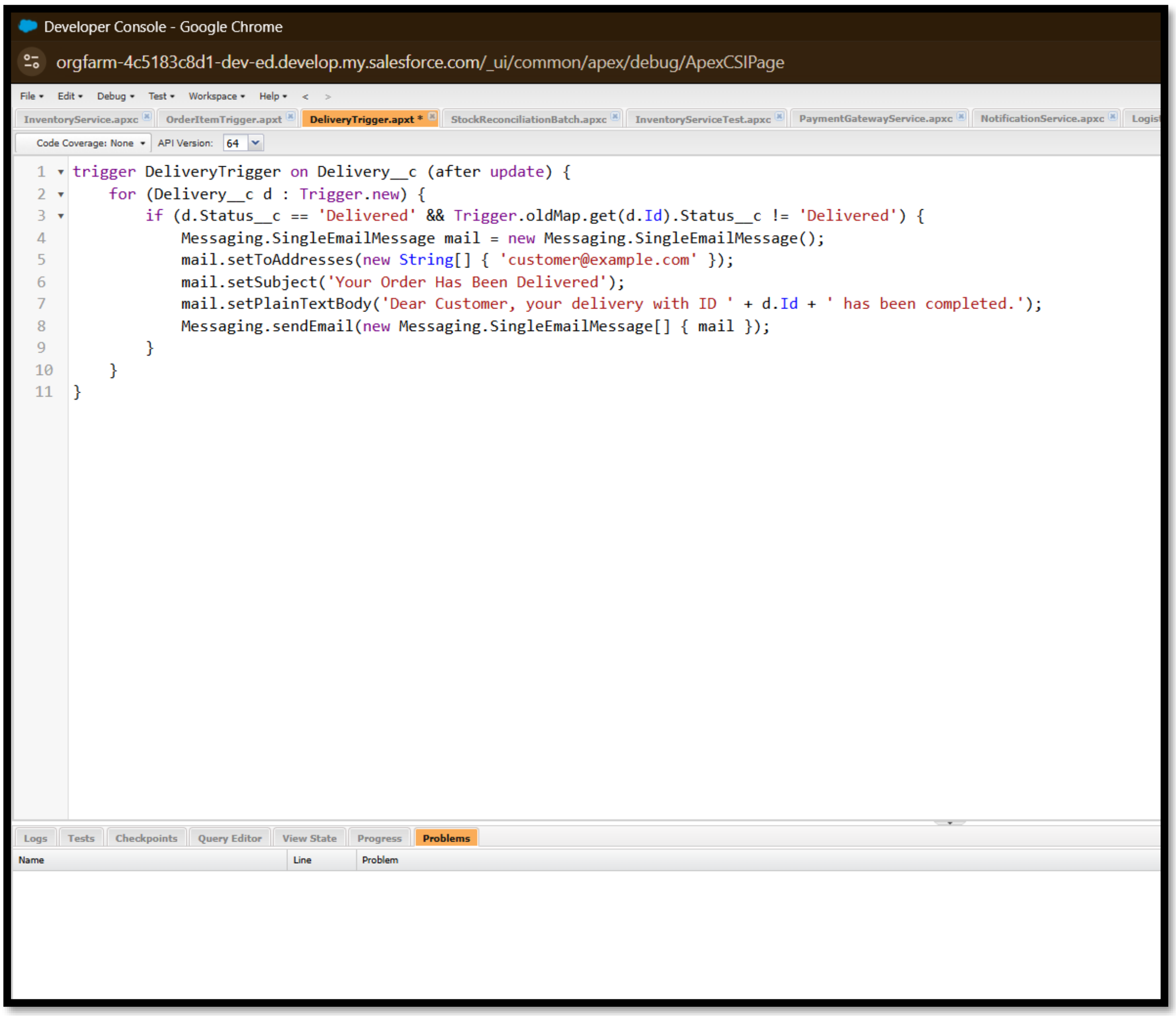
Apex Triggers

- **OrderItemTrigger** on Order_Item__c
    - Events: before insert (validate stock availability), after insert (adjust stock), after delete (restore stock).
    - Purpose: Prevents creating order items if stock unavailable and updates inventory after successful insert/delete.


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc OrderItemTrigger.apxt DeliveryTrigger.apxt StockReconciliationBatch.apxc InventoryServiceTest.apxc PaymentGatewayService.apxc
Code Coverage: None API Version: 64
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
trigger OrderItemTrigger on Order_Line_Item__c (before insert, after insert, after delete) {
    if (Trigger.isBefore && Trigger.isInsert) {
        InventoryService.adjustStock(Trigger.new, true, false);
    }
    if (Trigger.isAfter && Trigger.isInsert) {
        InventoryService.adjustStock(Trigger.new, true, false);
    }
    if (Trigger.isAfter && Trigger.isDelete) {
        InventoryService.adjustStock(Trigger.old, false, true);
    }
}
```

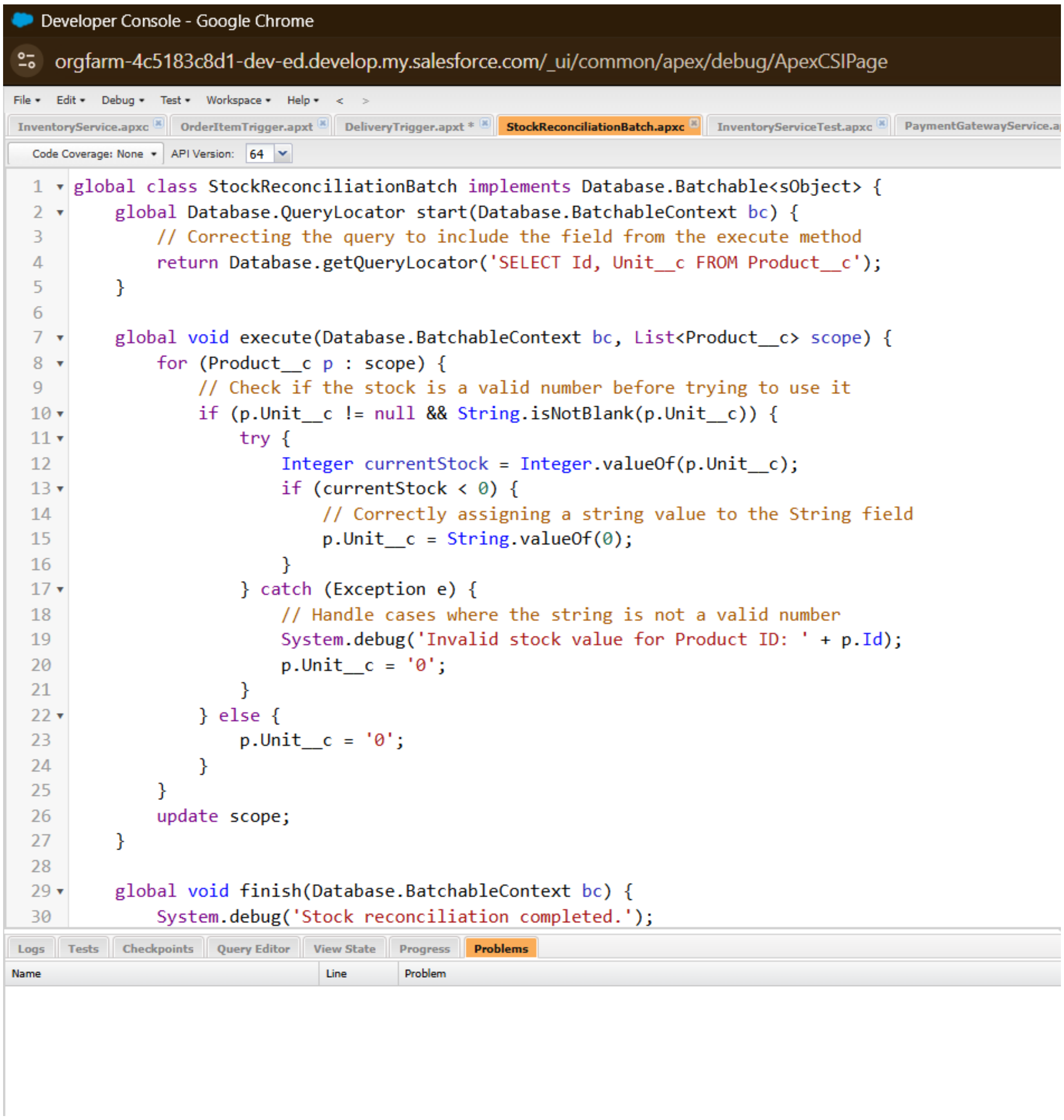
- **DeliveryTrigger** on Delivery__c
  - Events: after update
  - Purpose: When Status__c changes to Delivered, sends email/notification to the customer (demo sends email or logs an outgoing message).


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
InventoryService.apxc
OrderItemTrigger.apxt
DeliveryTrigger.apxt *
StockReconciliationBatch.apxc
InventoryServiceTest.apxc
PaymentGatewayService.apxc
NotificationService.apxc
Code Coverage: None
API Version: 64
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

```
trigger DeliveryTrigger on Delivery__c (after update) {
    for (Delivery__c d : Trigger.new) {
        if (d.Status__c == 'Delivered' && Trigger.oldMap.get(d.Id).Status__c != 'Delivered') {
            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
            mail.setToAddresses(new String[] { 'customer@example.com' });
            mail.setSubject('Your Order Has Been Delivered');
            mail.setPlainTextBody('Dear Customer, your delivery with ID ' + d.Id + ' has been completed.');
            Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
        }
    }
}
```

Asynchronous Apex

- **StockReconciliationBatch** (implements Database.Batchable<sObject>)
  - Purpose: Nightly job to reconcile and sanitize product stock (e.g., set negative stock to 0, recalc aggregates).
  - How to run: Database.executeBatch(new StockReconciliationBatch(), 200);


Developer Console - Google Chrome

orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

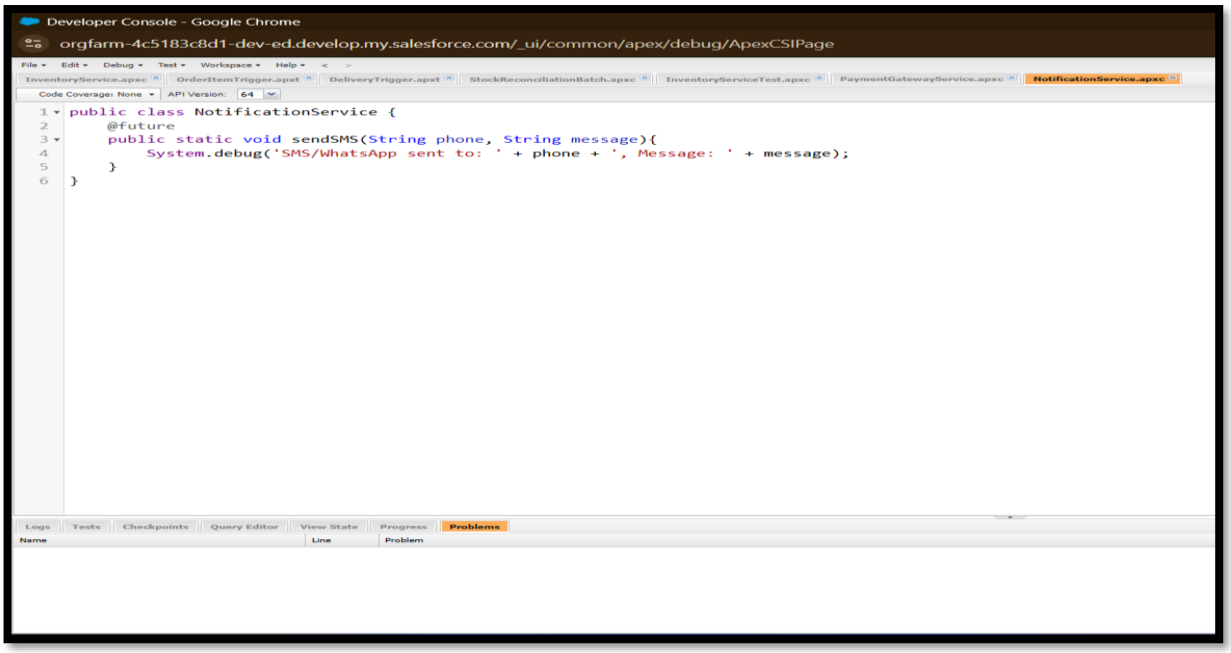
StockReconciliationBatch.apxc

Code Coverage: None   API Version: 64

```
global class StockReconciliationBatch implements Database.Batchable<sObject> {
    global Database.QueryLocator start(Database.BatchableContext bc) {
        // Correcting the query to include the field from the execute method
        return Database.getQueryLocator('SELECT Id, Unit__c FROM Product__c');
    }

    global void execute(Database.BatchableContext bc, List<Product__c> scope) {
        for (Product__c p : scope) {
            // Check if the stock is a valid number before trying to use it
            if (p.Unit__c != null && String.isNotBlank(p.Unit__c)) {
                try {
                    Integer currentStock = Integer.valueOf(p.Unit__c);
                    if (currentStock < 0) {
                        // Correctly assigning a string value to the String field
                        p.Unit__c = String.valueOf(0);
                    }
                } catch (Exception e) {
                    // Handle cases where the string is not a valid number
                    System.debug('Invalid stock value for Product ID: ' + p.Id);
                    p.Unit__c = '0';
                }
            } else {
                p.Unit__c = '0';
            }
        }
        update scope;
    }

    global void finish(Database.BatchableContext bc) {
        System.debug('Stock reconciliation completed.');
```

File Edit Debug Test Workspace Help
InventoryService.apxc
OrderItemTrigger.apxt
DeliveryTrigger.apxt *
InventoryServiceTest.apxc
PaymentGatewayService.a
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

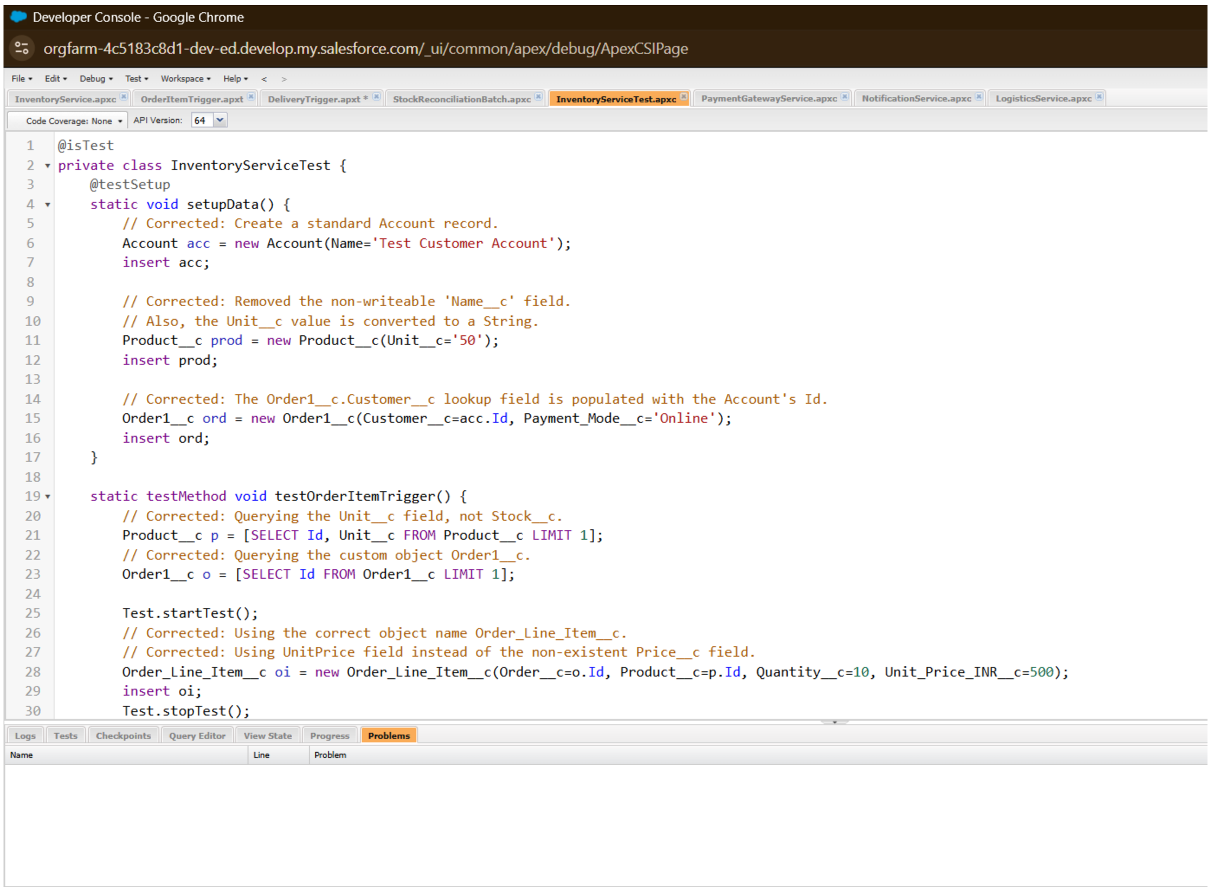
- **NotificationService.sendSMS** (@future)
    - Purpose: Offload external notification calls.


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc
OrderItemTrigger.apxt
DeliveryTrigger.apxt
StockReconciliationBatch.apxc
InventoryServiceTest.apxc
PaymentGatewayService.apxc
NotificationService.apxc
Code Coverage: None
API Version: 64
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

```
public class NotificationService {
    @future
    public static void sendSMS(String phone, String message){
        System.debug('SMS/WhatsApp sent to: ' + phone + ', Message: ' + message);
    }
}
```

Testing & Coverage

- **Test classes** created (examples):
    - InventoryServiceTest — sets up Product, Customer, Order and asserts stock adjustments after inserting an Order_Item__c.


Developer Console - Google Chrome
orgfarm-4c5183c8d1-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
InventoryService.apxc
OrderItemTrigger.apxt
DeliveryTrigger.apxt
StockReconciliationBatch.apxc
InventoryServiceTest.apxc
PaymentGatewayService.apxc
NotificationService.apxc
LogisticsService.apxc
Code Coverage: None
API Version: 64
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

```
@isTest
private class InventoryServiceTest {
    @testSetup
    static void setupData() {
        // Corrected: Create a standard Account record.
        Account acc = new Account(Name='Test Customer Account');
        insert acc;

        // Corrected: Removed the non-writeable 'Name__c' field.
        // Also, the Unit__c value is converted to a String.
        Product__c prod = new Product__c(Unit__c='50');
        insert prod;

        // Corrected: The Order1__c.Customer__c lookup field is populated with the Account's Id.
        Order1__c ord = new Order1__c(Customer__c=acc.Id, Payment_Mode__c='Online');
        insert ord;
    }

    static testMethod void testOrderItemTrigger() {
        // Corrected: Querying the Unit__c field, not Stock__c.
        Product__c p = [SELECT Id, Unit__c FROM Product__c LIMIT 1];
        // Corrected: Querying the custom object Order1__c.
        Order1__c o = [SELECT Id FROM Order1__c LIMIT 1];

        Test.startTest();
        // Corrected: Using the correct object name Order_Line_Item__c.
        // Corrected: Using UnitPrice field instead of the non-existent Price__c field.
        Order_Line_Item__c oi = new Order_Line_Item__c(Order__c=o.Id, Product__c=p.Id, Quantity__c=10, Unit_Price_INR__c=500);
        insert oi;
        Test.stopTest();
```

    - DeliveryTriggerTest — verifies email/notification is sent when Delivery status updates to Delivered.
    - StockReconciliationBatchTest — simulates negative stock and asserts batch fixes it.
- **Custom Exceptions** implemented in Apex to handle business errors (e.g., InsufficientStockException).
- **Reported coverage:** Test classes were written to achieve **>75% code coverage** for the Apex components (please run tests in your org to confirm actual coverage percentages).
