**Phase 5: Apex Programming (Developer)**

**Objectives**

Build advanced automation and integrations beyond what Admin tools can achieve using **Apex Classes, Triggers, Batch Processing, and Asynchronous Operations**.

**Activities**

1. **Apex Classes**
   * **RefundHandler.cls**: Handles refund processing logic.
   * **ReplacementManager.cls**: Automates replacement order creation.
   * **CourierAPIService.cls**: Manages courier tracking integration.
2. **Apex Triggers**
   * **Before Insert**: Validate return reason.
   * **After Insert**: Auto-generate Return ID and link to Order.
   * **After Update**: Update courier tracking in Return\_\_c when courier API updates status.
3. **Trigger Design Pattern**
   * Centralized handler framework (one trigger per object, delegating logic to handler classes).
4. **SOQL & SOSL Queries**
   * SOQL: Fetch all pending refunds for a given customer.
   * SOSL: Search across multiple objects for order-related complaints.
5. **Collections (List, Set, Map)**
   * Map<OrderId, Refund\_\_c> used for batch processing refunds.
6. **Batch Apex**
   * Nightly batch job for reconciling refunds with payment gateway.
7. **Queueable Apex**
   * Queue courier label generation for large volumes.
8. **Scheduled Apex**
   * Weekly SLA breach summary email to management.
9. **Future Methods**
   * Async callouts to courier tracking APIs.
10. **Exception Handling**

* Custom exception object to log failed refund attempts.

1. **Test Classes**

* Achieved 85% coverage.
* Used mock classes for external courier API.

1. **Asynchronous Processing**

* High-volume refund reconciliation jobs processed outside synchronous limits.

**Deliverables**

* **Apex Class Repository in GitHub**
* **Trigger Handler Framework**
* **Test Coverage Report (>75%)**

Name- Shubhangi Vishwakarma

College- Gyan Ganga Institute of Technology and Sciences

Email- shubhangi.vishwakarma.cs22@ggits.net

TCS Phase 2 : Batch 5