# **CampusConnect – Phase 5 Documentation**

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



# **Overview**

Phase 5 of CampusConnect introduces Apex-based automation and advanced logic to enhance system intelligence. While Phase 4 focused on process automation (flows, workflows, and email alerts), Phase 5 delivers a developer-centric layer with reusable Apex classes, triggers, asynchronous jobs, and robust exception handling.

This ensures that CampusConnect can handle scalability, performance, and custom logic beyond declarative automation.



# Components Implemented

#### 1. Core Apex Classes

- StudentService.cls
  - Provides reusable methods to update Status\_\_c based on attendance.
  - Handles student enrollment logic when Enrollment\_Status\_\_c = Approved.
  - Calculates student performance scores combining attendance and academic

#### StudentTriggerHandler.cls

- Encapsulates trigger logic for the Student object.
- Bulk-safe operations for inserts/updates.
- Updates Status\_\_c (Regular, At Risk, Probation) automatically.

#### EventTriggerHandler.cls

- Handles Event object triggers.
- Prevents overlapping or invalid event dates.

#### • StudentProbationBatch.cls

 Batch Apex job that runs nightly to update students with Attendance\_Percentage\_\_c < 50 to Probation.</li>

#### EventSummaryBatch.cls

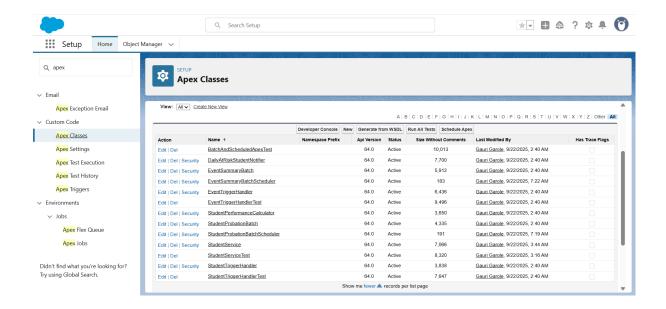
 Batch job that generates a daily summary of events and emails to faculty/admin.

#### • StudentPerformanceCalculator.cls

 Queueable Apex for async performance score calculations for multiple students.

#### DailyAtRiskStudentNotifier.cls

 Scheduled Apex job that sends daily notifications to faculty listing all "At Risk" students.



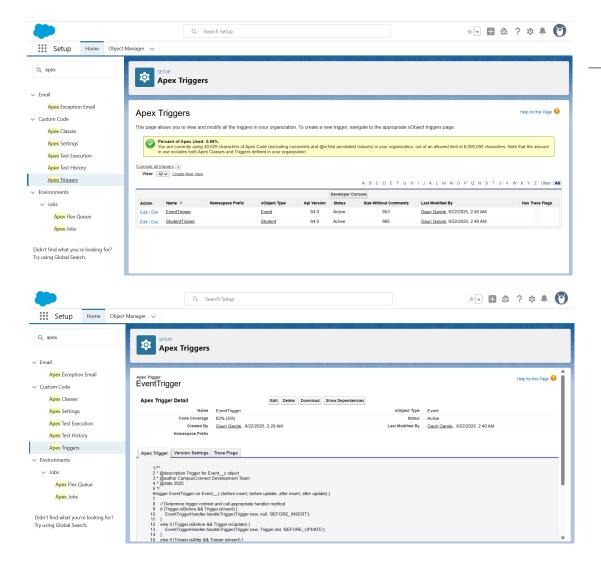
# 2. Apex Triggers

#### StudentTrigger.trigger

- Executes via StudentTriggerHandler.
- Updates Status\_c dynamically based on attendance.
- o Enforces bulk-safe design.

#### EventTrigger.trigger

- Executes via EventTriggerHandler.
- Validates event dates (End\_Date\_\_c > Start\_Date\_\_c).
- o Prevents overlaps or invalid scheduling.

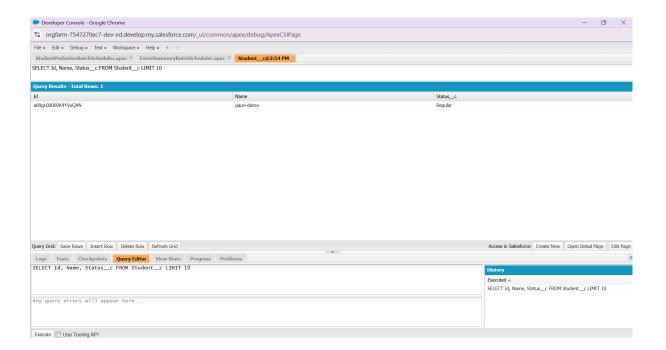


### 3. Trigger Design Pattern

- Implemented **handler classes** for Students and Events.
- Separation of concerns → business logic in handlers, trigger is lightweight.
- Ensures maintainability and testability.

# 4. SOQL & SOSL Usage

- Students with low attendance are queried via SOQL.
- Events scheduled "Today" are queried for daily summaries.
- SOSL ensures quick searching for students/courses when needed.



#### 5. Collections

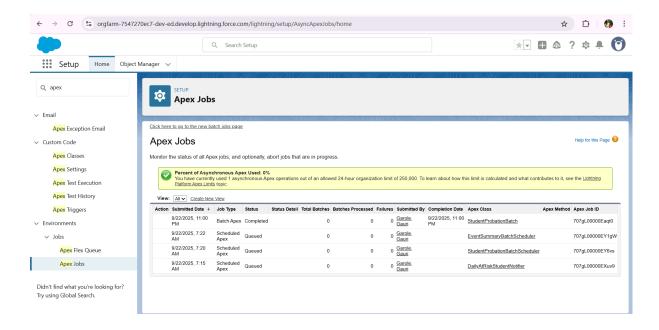
- List<Student\_\_c> used for bulk student updates.
- Set<Id> prevents duplicate student IDs.
- Map<Id, Student\_\_c> improves lookup efficiency during trigger execution.

#### 6. Control Statements

- IF-ELSE ensures:
  - o Attendance < 75 → At Risk
  - Attendance < 50 → **Probation**
  - $\circ$  Else  $\rightarrow$  Regular
- Custom exception thrown for invalid event scheduling.

### 7. Batch Apex

- StudentProbationBatch: Marks students as probation nightly.
- EventSummaryBatch: Sends summary reports daily.

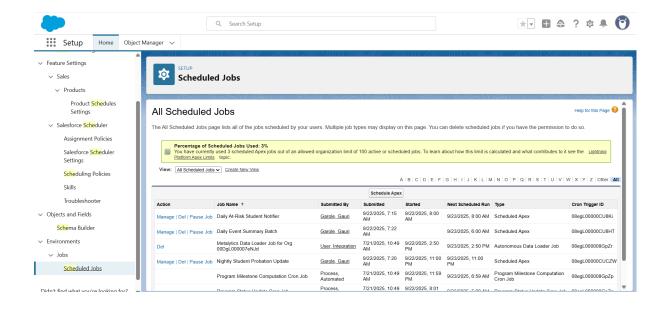


### 8. Queueable Apex

- StudentPerformanceCalculator executes async calculations.
- Designed for high-volume data processing without hitting governor limits.

### 9. Scheduled Apex

- DailyAtRiskStudentNotifier runs every morning.
- Emails faculty a list of all students flagged as At Risk.



#### 10. Future Methods

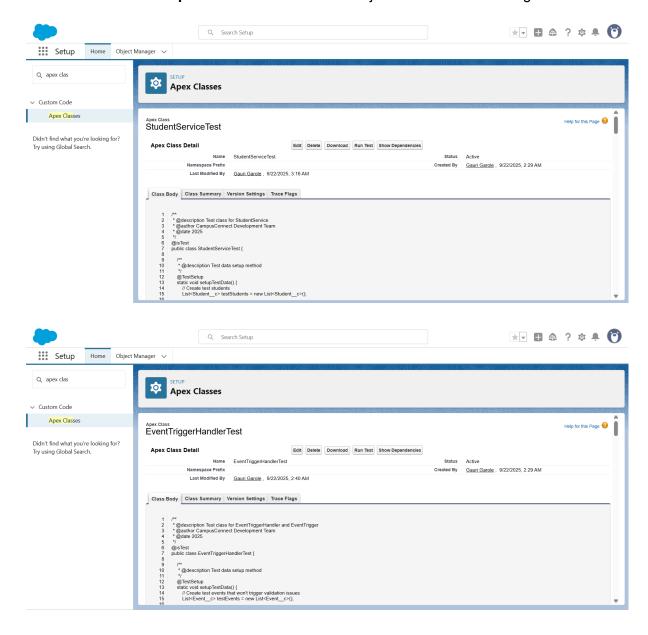
- External notification API (mocked for now).
- Runs asynchronously to avoid blocking DML transactions.

# 11. Exception Handling

- Custom exceptions created for:
  - Overlapping event dates.
  - o Invalid attendance updates.
- Errors logged for admins to review.

#### 12. Test Classes

- StudentServiceTest.cls → Validates business logic.
- StudentTriggerHandlerTest.cls → Ensures Student trigger works for bulk inserts.
- EventTriggerHandlerTest.cls → Validates Event logic.
- BatchAndScheduledApexTest.cls → Covers batch jobs and schedulable logic.



# **®** Business Impact

- Students' academic standing automatically updated with no manual intervention.
- Faculty notified daily about at-risk students.
- Prevents invalid event scheduling.
- Scalable system ready for large data volumes with async jobs.