PHASE 5: APEX PROGRAMMING (DEVELOPER)

Goal: Extend the Event Management System app with custom Apex logic for validations, automation, asynchronous processing, and unit testing.

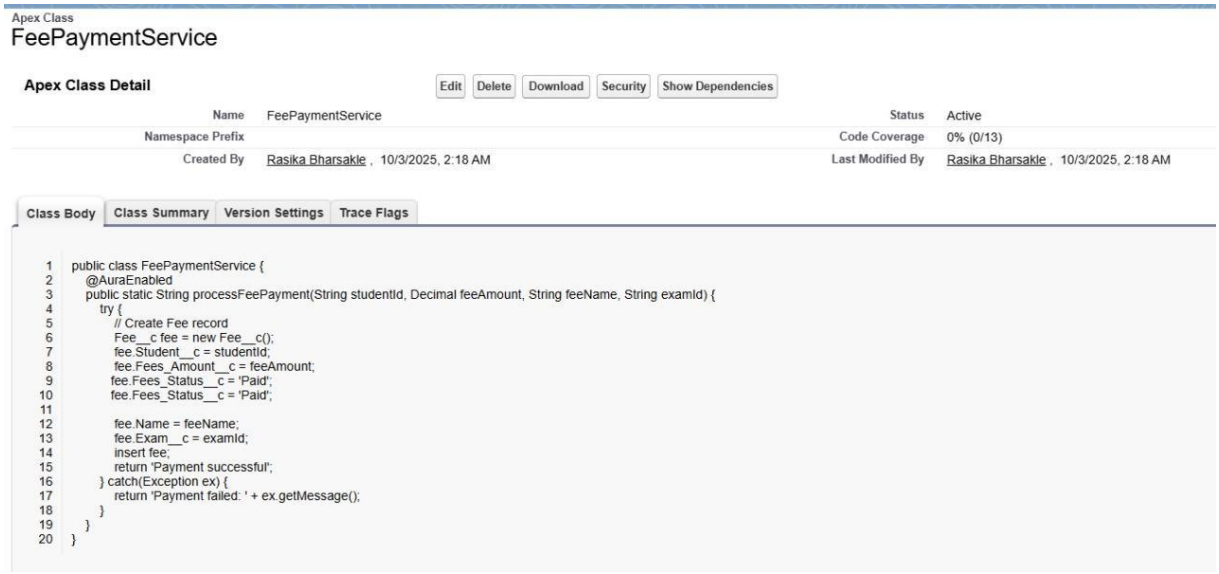
Step 1: Apex Classes & Objects

1. **Fee Payment Processing**

- Developed FeePaymentService Apex class to securely process fee payments.

- Enabled storage of payment transactions, status updates, and fee types.

- Created corresponding test classes to validate payment methods and ensure code coverage.

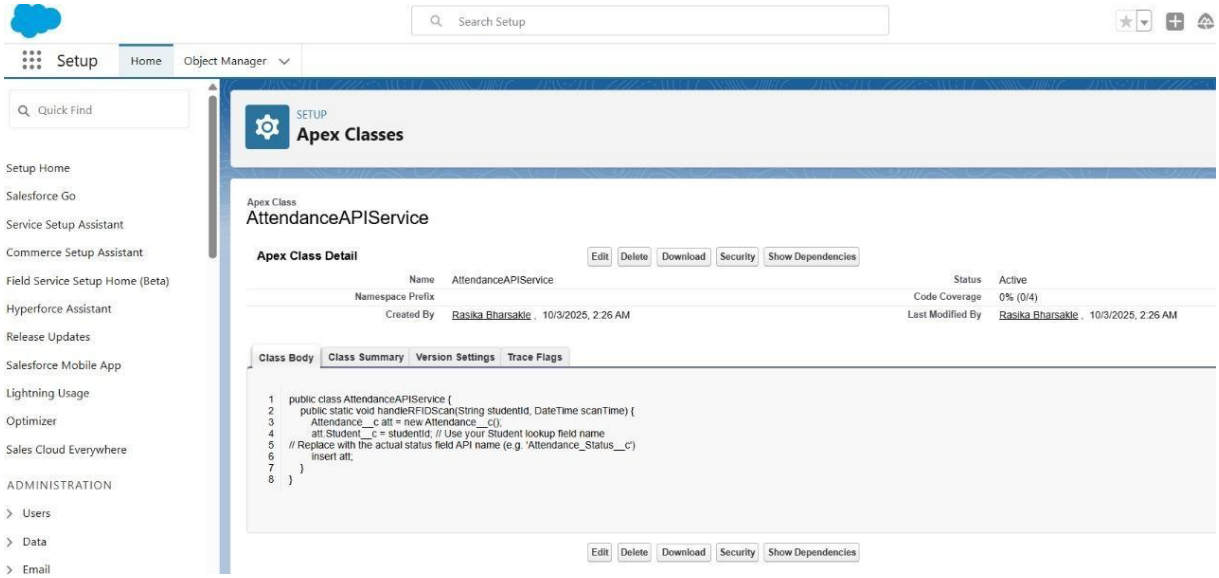


2. **Attendance Integration**

- Implemented AttendanceAPIService to process Biometric/RFID scan data.

- Automatically updated attendance records linked to student profiles.

- Developed test methods to verify attendance logging works correctly.

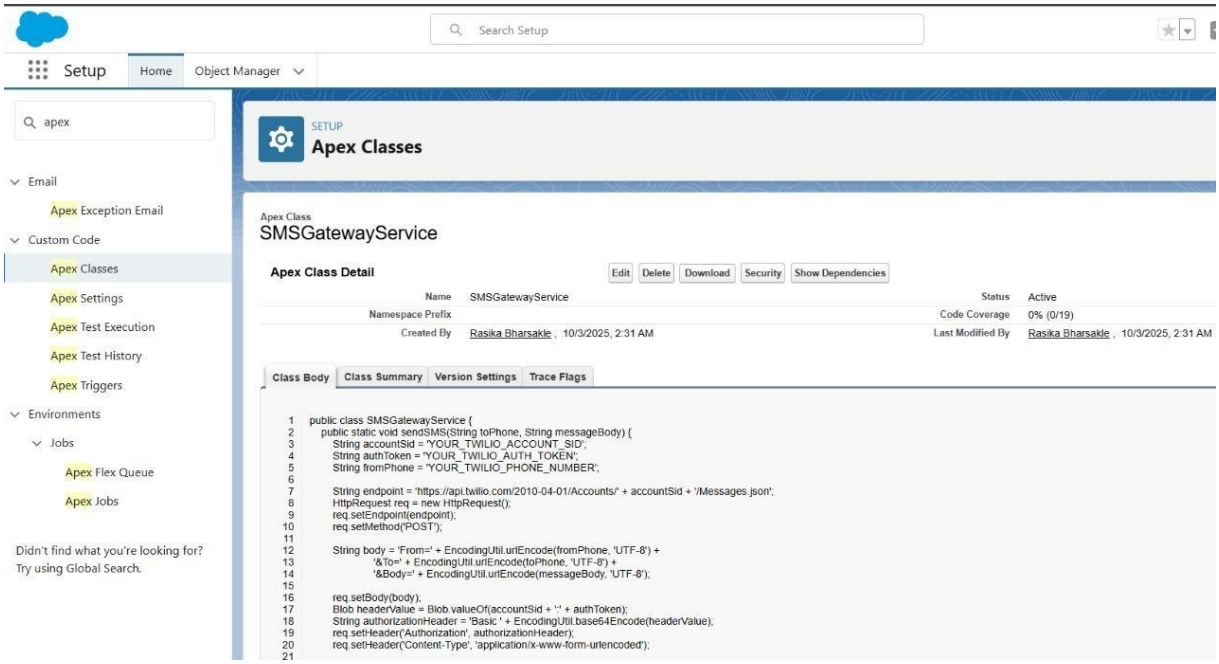


3. **SMS Gateway Integration Preparation**

- Designed SMSGatewayService Apex class framework to send SMS alerts using external SMS APIs.

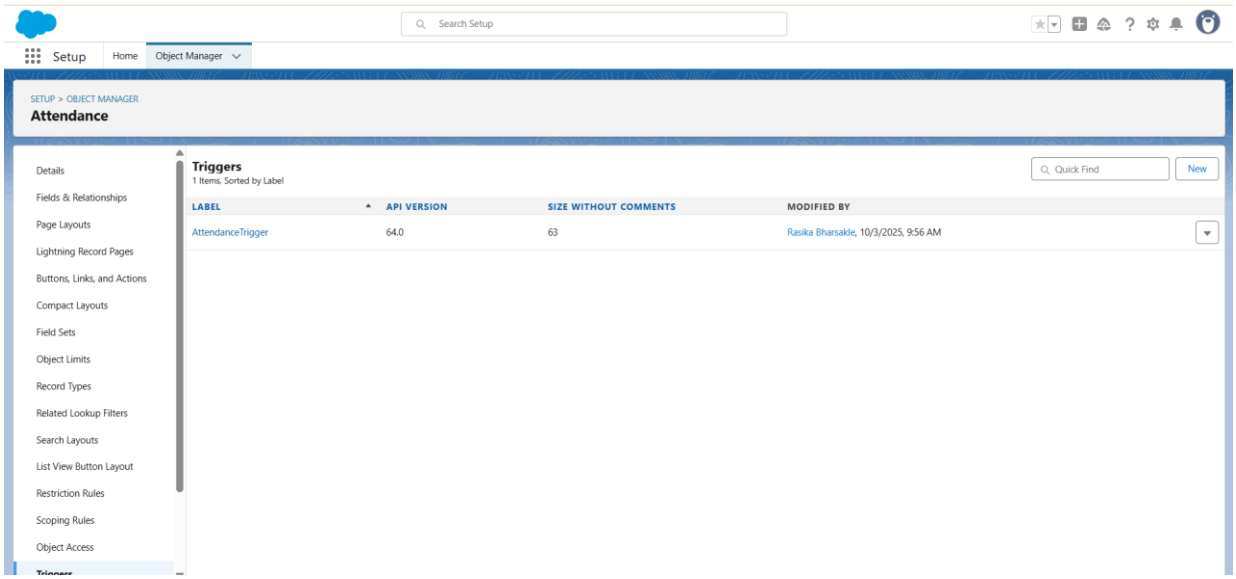
- Setup remote site settings for HTTP callouts.

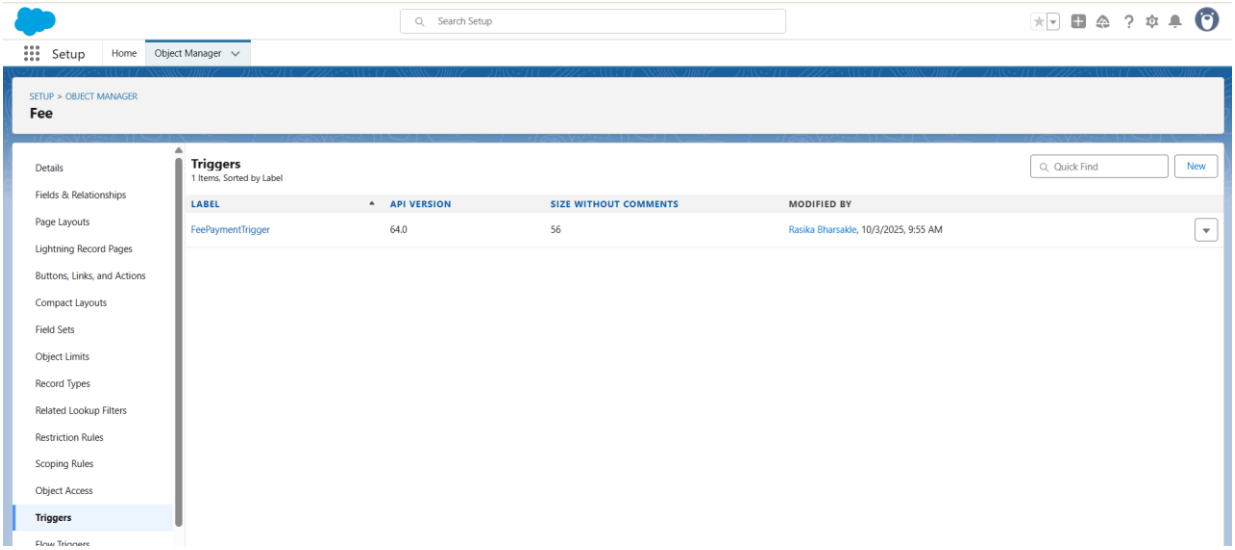
- Prepared test mocks for SMS sending validation without external dependencies.



Step 2: Apex Triggers

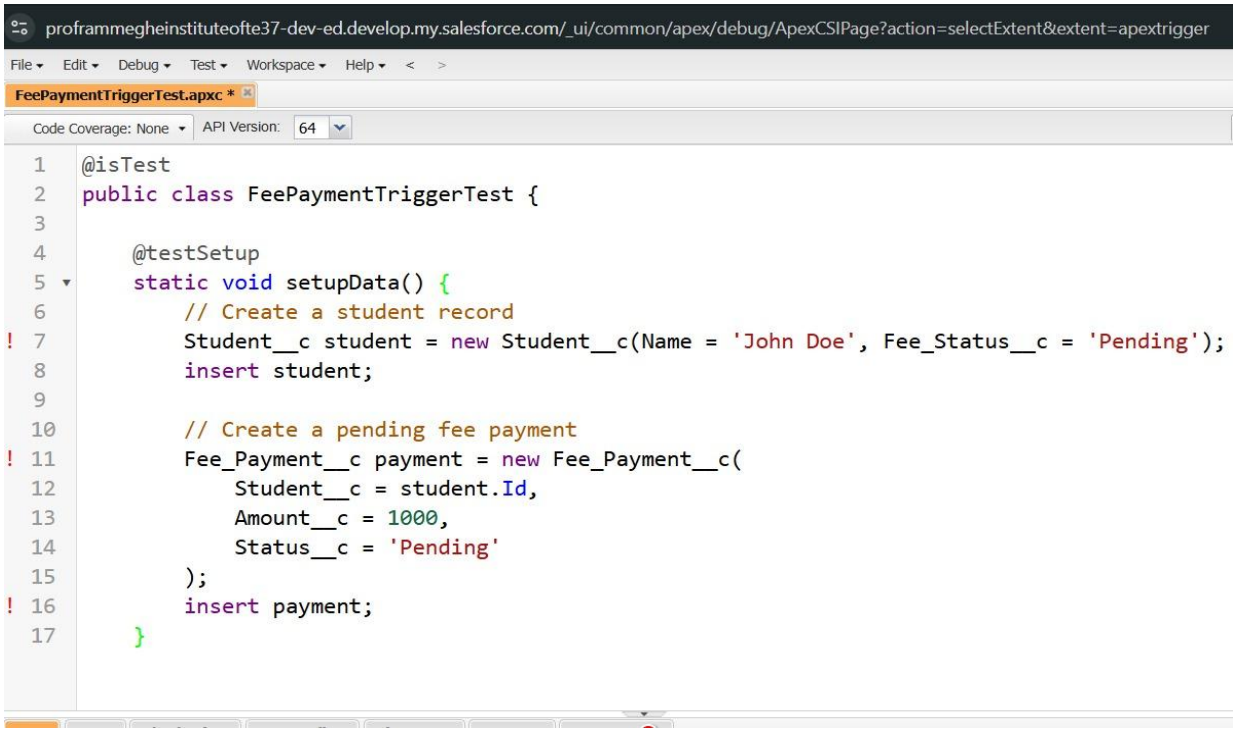
- A trigger Attendance & Fee Trigger was implemented on Attendance_c & Fee_c.
- It executes before insert and before update, ensuring invalid registrations are blocked before being committed to the database.
- The trigger itself is lean, simply delegating work to the service class.
- This follows the Trigger Handler Design Pattern, which separates trigger context from business logic.





Step 1: Test Cases

- This test class also ensures code coverage requirements are met for deployment.
- An initial failure occurred due to a conflicting Flow; this was resolved by bypassing/deactivating it during test runs.


proframmegheinstituteofte37-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apextrigger
File Edit Debug Test Workspace Help
FeePaymentTriggerTest.apxc *
Code Coverage: None
API Version: 64
@isTest
public class FeePaymentTriggerTest {
@testSetup
static void setupData() {
// Create a student record
Student__c student = new Student__c(Name = 'John Doe', Fee_Status__c = 'Pending');
insert student;
// Create a pending fee payment
Fee_Payment__c payment = new Fee_Payment__c(
Student__c = student.Id,
Amount__c = 1000,
Status__c = 'Pending'
);
insert payment;
}