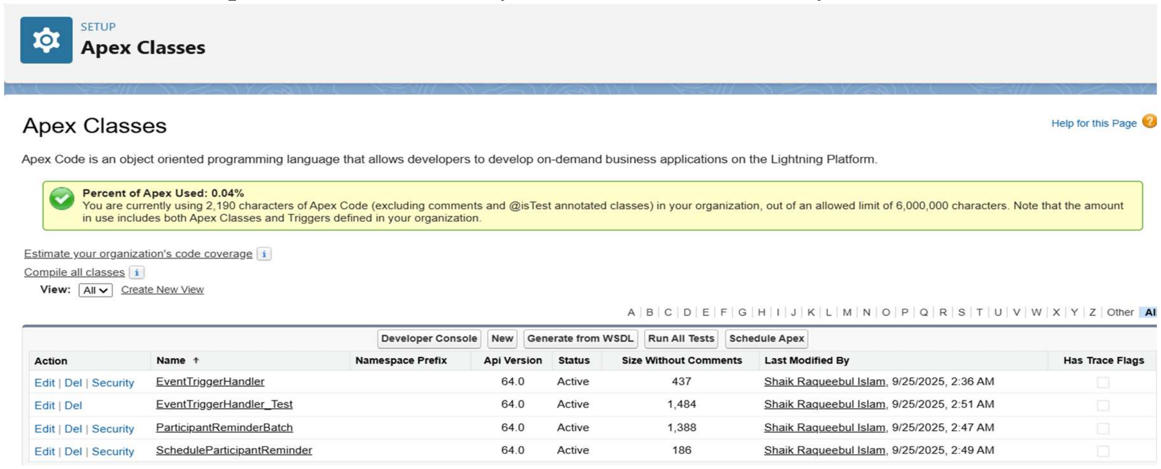
CAMPUS EVENT MANAGEMENT SYSTEM

NAME: SHAIK RAQUEEBUL ISLAM

Phase 5: Apex Programming (Developer)

Stage 1: Classes & Objects

- In this phase, we created Apex classes to encapsulate business logic for automating event-related processes. For instance, the EventTriggerHandler class includes static methods to handle validation rules such as preventing duplicate events. Using static methods ensures modularity and reusability across different classes, which improves maintainability and reduces redundancy in the code.


SETUP
Apex Classes
Help for this Page
Developer Console
New
Generate from WSDL
Run All Tests
Schedule Apex
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

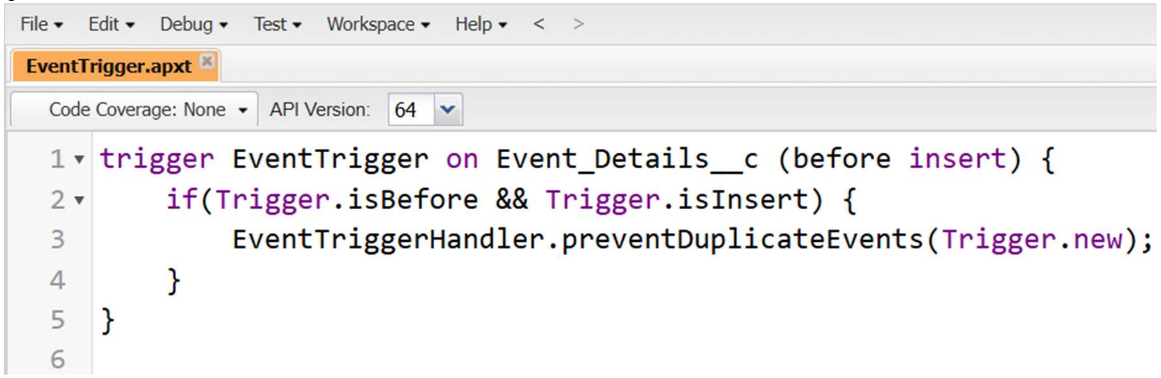
Percent of Apex Used: 0.04%
You are currently using 2,190 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage
Compile all classes
View: All Create New View

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	EventTriggerHandler		64.0	Active	437	Shaik Raqueebul Islam, 9/25/2025, 2:36 AM	☐
Edit | Del	EventTriggerHandler_Test		64.0	Active	1,484	Shaik Raqueebul Islam, 9/25/2025, 2:51 AM	☐
Edit | Del | Security	ParticipantReminderBatch		64.0	Active	1,388	Shaik Raqueebul Islam, 9/25/2025, 2:47 AM	☐
Edit | Del | Security	ScheduleParticipantReminder		64.0	Active	186	Shaik Raqueebul Islam, 9/25/2025, 2:49 AM	☐

Stage 2: Apex Triggers (Before/After Insert/Update/Delete)

- We implemented triggers on the Event_Details__c object to enforce business rules automatically. The trigger ensures that no duplicate event names are created by calling the EventTriggerHandler class before record insertion. This approach guarantees that all event records follow consistent rules without manual checks.



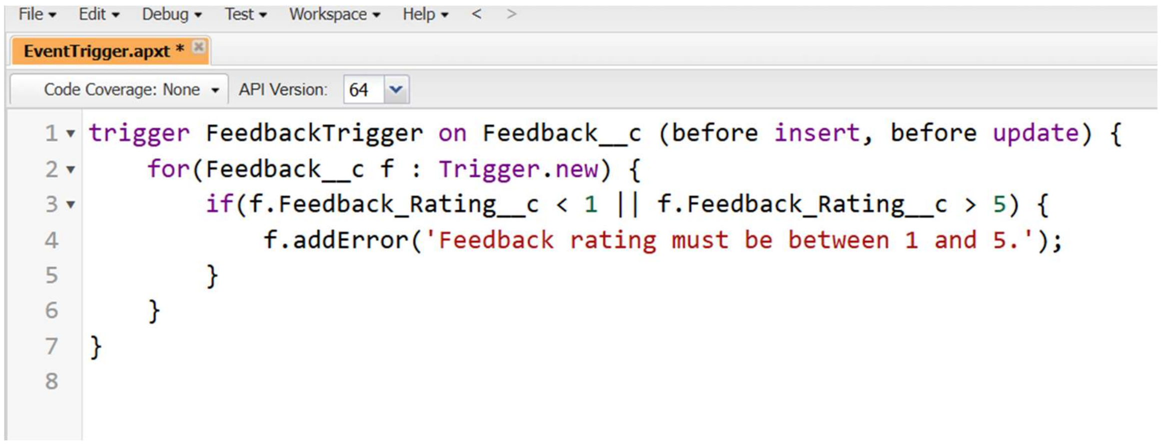
```
trigger EventTrigger on Event_Details__c (before insert) {
    if(Trigger.isBefore && Trigger.isInsert) {
        EventTriggerHandler.preventDuplicateEvents(Trigger.new);
    }
}
```

Stage 3: SOQL & SOSL

- SOQL was utilized extensively to query Participants and Feedback for reports, validations, and automated emails. For example, we fetch all participants registered for an event to send reminders or validate feedback. SOSL was not required since queries were object-specific.

Stage 4: Control Statements

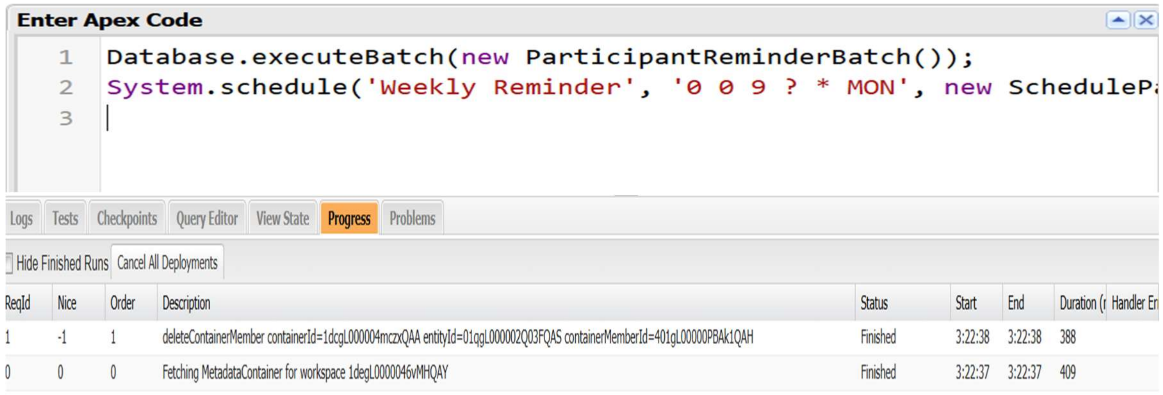
- We used control statements such as loops and conditional checks to enforce data integrity. For example, feedback ratings were validated to ensure they fall within a range of 1–5. This prevents bad data from entering the system and maintains clean, reliable records.



```
trigger FeedbackTrigger on Feedback__c (before insert, before update) {
    for(Feedback__c f : Trigger.new) {
        if(f.Feedback_Rating__c < 1 || f.Feedback_Rating__c > 5) {
            f.addError('Feedback rating must be between 1 and 5.');
        }
    }
}
```

Stage 5: Batch Apex

- To automate participant reminders, we created a batch Apex class. The batch queries participants whose events are approaching and sends them email reminders. Batch Apex enables processing of large volumes of records efficiently without hitting governor limits.



```
Database.executeBatch(new ParticipantReminderBatch());
System.schedule('Weekly Reminder', '0 0 9 ? * MON', new ScheduleP
```

ReqId	Nice	Order	Description	Status	Start	End	Duration (r	Handler Er
1	-1	1	deleteContainerMember containerId=1dcgL000004mczxQAA entityId=01qgL000002Q03FQAS containerMemberId=401gL00000PBAk1QAH	Finished	3:22:38	3:22:38	388	
0	0	0	Fetching MetadataContainer for workspace 1degL0000046vMHQAY	Finished	3:22:37	3:22:37	409	