

Phase 5: Apex Programming (Developer)

Objective

To extend Salesforce functionality beyond declarative tools using **Apex Triggers, Classes, Batch Apex, and Test Methods**. Apex enables complex logic, bulk data handling, and scalability that cannot be fully managed with point-and-click automation.

Key Apex Use Cases in This Project

1. Auto-Assign Event for Attendees

- **Purpose:** Automatically link an attendee to the next upcoming event if the event is not specified during registration.
- **Logic:**
 - Triggered **before insert** of Attendee__c.
 - If **Event__c** is blank → auto-populate with the **next upcoming event**.

Trigger Example – Auto-Assign Event

```
trigger AttendeeTrigger on Attendee__c (before insert) {  
  
    List<Event__c> events = [SELECT Id FROM Event__c WHERE Event_Date__c >= TODAY  
ORDER BY Event_Date__c ASC LIMIT 1];  
  
    for(Attendee__c att : Trigger.new) {  
        if(att.Event__c == null && events.size() > 0) {  
            att.Event__c = events[0].Id;  
        }  
    }  
}
```

2. Bulk Email Handler for Registrations/Feedback

- **Purpose:** Send emails to attendees in bulk without hitting Flow governor limits (>200 records).
- **Logic:**
 - **Batch Apex** processes large volumes of Attendee__c records.

- Sends registration confirmations, reminders, or feedback acknowledgments via Messaging.sendEmail().

Batch Apex Example – Registration Reminder

```
global class RegistrationReminderBatch implements Database.Batchable<sObject> {

    global Database.QueryLocator start(Database.BatchableContext bc) {

        return Database.getQueryLocator([SELECT Id, Email__c FROM Attendee__c WHERE
Registration_Confirmed__c = false]);

    }

    global void execute(Database.BatchableContext bc, List<Attendee__c> attendees) {

        List<Messaging.SingleEmailMessage> emails = new
List<Messaging.SingleEmailMessage>();

        for(Attendee__c att : attendees) {

            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();

            mail.setToAddresses(new String[] {att.Email__c});

            mail.setSubject('Event Registration Reminder');

            mail.setPlainTextBody('This is a reminder to complete your event registration.');
```

3. Prevent Duplicate Registrations

- **Purpose:** Enforce advanced validation to prevent the same attendee from registering multiple times for the same event.
- **Logic:** Apex Trigger checks for duplicate combinations of Name + Email + Event__c before inserting records.

4. Future Enhancements: Smart Event Recommendations

- **Purpose:** Recommend suitable events for attendees based on their department, past registrations, or interests.
- **Logic:** Apex Class evaluates attendee preferences and matches them with upcoming events. Results can be displayed via **Lightning Web Components (LWC)**.

Apex Components Summary

Apex Component	Type	Purpose
AttendeeTrigger	Trigger	Auto-assigns next upcoming Event if not specified during registration
RegistrationReminderBatch	Batch Apex	Sends bulk registration reminder emails without Flow limits
DuplicateRegistrationValidator	Trigger	Prevents duplicate attendee registration for same event
EventRecommendation	Apex Class	Suggests events to attendees based on department or past participation (future)
Test Classes	Apex Tests	Ensure 75%+ code coverage for deployment readiness

Benefits of Apex Programming

- Handles **bulk processing** efficiently for large attendee datasets.
- Implements **complex business logic** not possible with Flows, such as preventing duplicate registrations.
- Supports **future scalability**, e.g., AI-based event recommendations or smart notifications.
- Ensures **deployment readiness** with adequate test coverage and reliable automation.