

Guest360 CRM Project – Phase 5: Apex Programming (Developer)

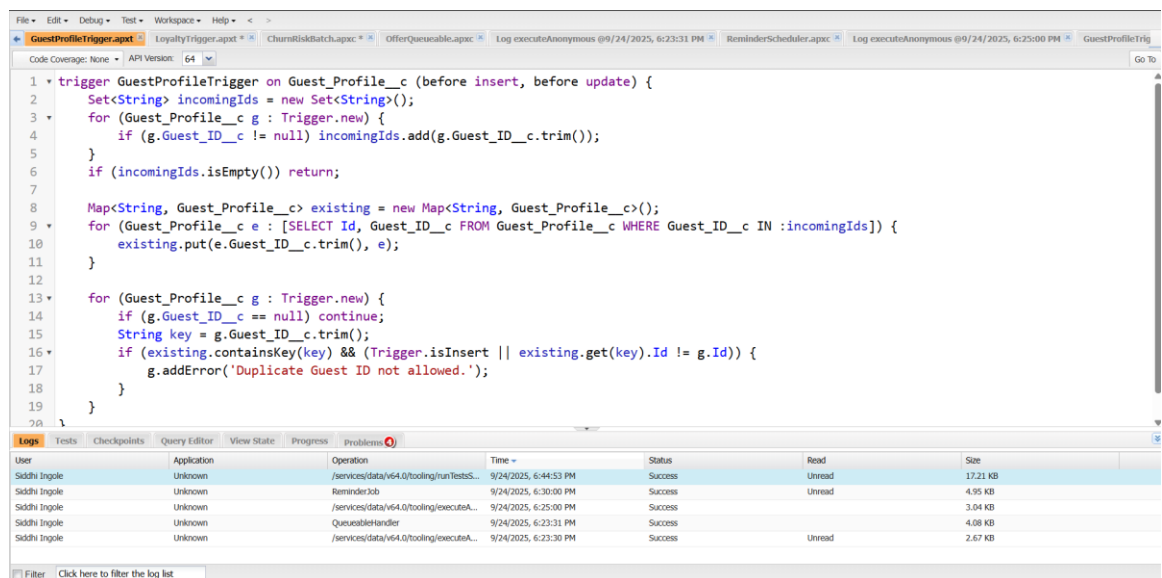
This document explains Phase 5 (Apex Programming) of the Guest360 CRM Project. It provides step-by-step click instructions with code and screenshot placeholders.

Step 1: Prevent Duplicate Guest IDs (Trigger)

Path: **Setup → Object Manager → Guest Profile → Triggers → New**

Steps:

1. Click New → Name: GuestProfileTrigger → sObject: Guest_Profile__c



Step 2: Personalize Loyalty (Trigger)

Path: **Setup → Object Manager → Reservation → Triggers → New**

Steps:

1. Name: LoyaltyTrigger → sObject: Reservation__c

```

1 trigger LoyaltyTrigger on Reservation__c (after insert, after update) {
2     Set<Id> guestIds = new Set<Id>();
3     for (Reservation__c r : Trigger.new) if (r.Guest__c != null) guestIds.add(r.Guest__c);
4     if (guestIds.isEmpty()) return;
5
6     Map<Id, Integer> counts = new Map<Id, Integer>();
7     for (AggregateResult ar : [
8         SELECT Guest__c g, COUNT(Id) c
9         FROM Reservation__c
10        WHERE Guest__c IN :guestIds
11        GROUP BY Guest__c
12    ]) {
13        counts.put((Id)ar.get('g'), ((Long)ar.get('c')).intValue());
14    }
15
16    List<Guest_Profile__c> toUpdate = new List<Guest_Profile__c>();
17    for (Id gid : counts.keySet()) {
18        if (counts.get(gid) > 5) toUpdate.add(new Guest_Profile__c(Id = gid, Loyalty_Member__c = true));
19    }
20    if (!toUpdate.isEmpty()) update toUpdate;
21}

```

User	Application	Operation	Time	Status	Read	Size
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/runTests...	9/24/2025, 6:44:53 PM	Success	Unread	17.21 KB
Siddhi Ingole	Unknown	Reminder Job	9/24/2025, 6:30:00 PM	Success	Unread	4.95 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:25:00 PM	Success	Unread	3.04 KB
Siddhi Ingole	Unknown	QueueableHandler	9/24/2025, 6:23:31 PM	Success	Unread	4.08 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:23:30 PM	Success	Unread	2.67 KB

Step 3: Batch Apex – Churn Risk Analysis

Path: Setup → Apex Classes → New

Steps:

1. Name: ChurnRiskBatch

```

1 global class ChurnRiskBatch implements Database.Batchable<SObject> {
2
3     global Database.QueryLocator start(Database.BatchableContext bc) {
4         return Database.getQueryLocator(
5             'SELECT Id, Last_Reservation_Date__c FROM Guest_Profile__c'
6         );
7     }
8
9     global void execute(Database.BatchableContext bc, List<Guest_Profile__c> scope) {
10        Date cutoff = Date.today().addMonths(-6);
11        List<Guest_Profile__c> updateList = new List<Guest_Profile__c>();
12
13        for (Guest_Profile__c g : scope) {
14            if (g.Last_Reservation_Date__c != null && g.Last_Reservation_Date__c < cutoff) {
15                g.Churn_Risk__c = true;
16                updateList.add(g);
17            }
18        }
19
20        if (!updateList.isEmpty()) {
21            update updateList;
22        }
23    }
24}

```

User	Application	Operation	Time	Status	Read	Size
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/runTests...	9/24/2025, 6:44:53 PM	Success	Unread	17.21 KB
Siddhi Ingole	Unknown	Reminder Job	9/24/2025, 6:30:00 PM	Success	Unread	4.95 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:25:00 PM	Success	Unread	3.04 KB
Siddhi Ingole	Unknown	QueueableHandler	9/24/2025, 6:23:31 PM	Success	Unread	4.08 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:23:30 PM	Success	Unread	2.67 KB

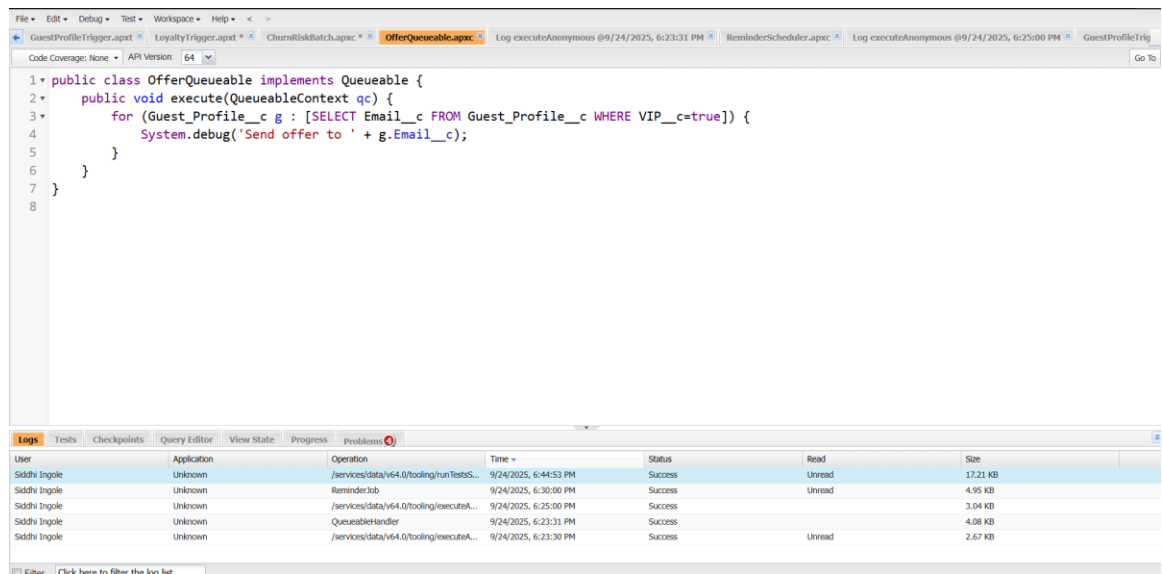
Save and run: Debug → Open Execute Anonymous → Database.executeBatch(new ChurnRiskBatch(), 200)

Step 4: Queueable Apex – Send Offers

Path: **Setup** → **Apex Classes** → **New**

Steps:

1. Name: OfferQueueable
- 2.

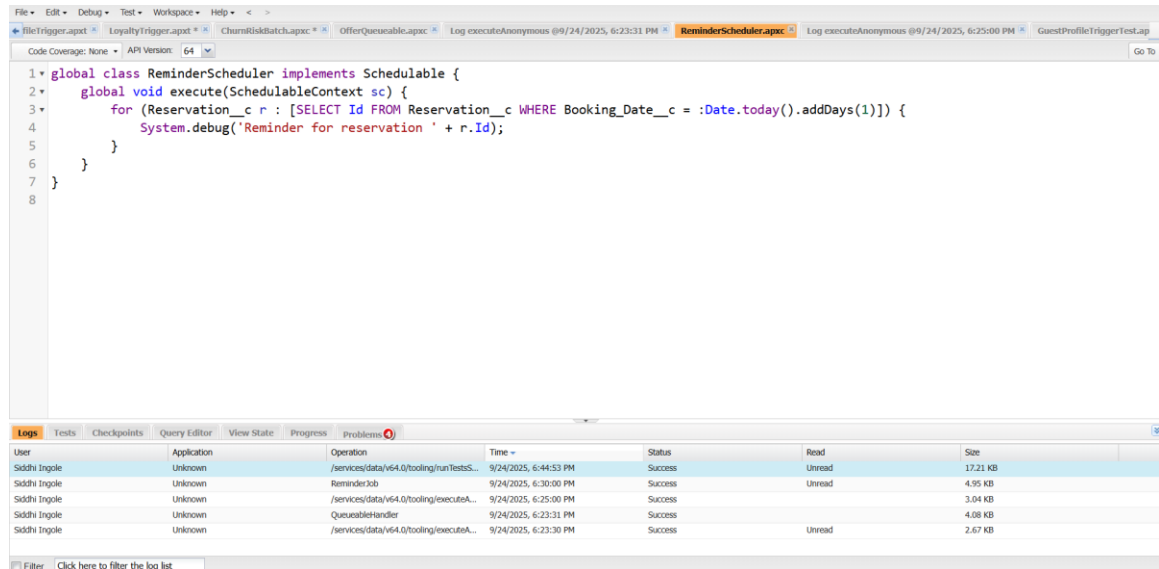


Save and run: Execute Anonymous → `System.enqueueJob(new OfferQueueable())`

Step 5: Scheduled Apex – Reminders

Path: Setup → Apex Classes → New

Steps:



The screenshot displays the Salesforce IDE interface. The top pane shows the code for the `ReminderScheduler` class, which implements the `Schedulable` interface. The `execute` method queries for reservations due within the next day and logs their IDs. The bottom pane shows the 'Logs' tab with a table of execution results.

```
1 global class ReminderScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         for (Reservation__c r : [SELECT Id FROM Reservation__c WHERE Booking_Date__c = :Date.today().addDays(1)]) {
4             System.debug('Reminder for reservation ' + r.Id);
5         }
6     }
7 }
8
```

User	Application	Operation	Time	Status	Read	Size
Siddhi Ingole	Unknown	/services/data/v4.0/tooling/runTests...	9/24/2025, 6:44:53 PM	Success	Unread	17.21 KB
Siddhi Ingole	Unknown	ReminderJob	9/24/2025, 6:30:00 PM	Success	Unread	4.95 KB
Siddhi Ingole	Unknown	/services/data/v4.0/tooling/executeA...	9/24/2025, 6:25:00 PM	Success	Unread	3.04 KB
Siddhi Ingole	Unknown	QueueableHandler	9/24/2025, 6:23:31 PM	Success	Unread	4.08 KB
Siddhi Ingole	Unknown	/services/data/v4.0/tooling/executeA...	9/24/2025, 6:23:30 PM	Success	Unread	2.67 KB

Save and schedule: Setup → Apex Classes → Schedule Apex → Select ReminderScheduler → Daily.

Step 6: Test Class

Path: Setup → Apex Classes → New

Steps:

1. Name: GuestProfileTriggerTest

@isTest

```
private class GuestProfileTriggerTest {
    @isTest static void testDuplicate() {
        Guest_Profile__c g1 = new Guest_Profile__c(Name='Test1', Guest_ID__c='X001');
        insert g1;
        Guest_Profile__c g2 = new Guest_Profile__c(Name='Test2', Guest_ID__c='X001');
        Test.startTest();
        try { insert g2; System.assert(false); } catch (DmlException e) {}
        Test.stopTest();
    }
}
```

}

2. Save → Test → New Run → Select GuestProfileTriggerTest → Run.

The screenshot shows a development environment with a code editor and a log viewer. The code editor displays a C# test class named `GuestProfileTriggerTest` with the following code:

```
1 @isTest
2 * private class GuestProfileTriggerTest {
3 *     @isTest static void testDuplicate() {
4 *         Guest_Profile__c g1 = new Guest_Profile__c(Name='Test1', Guest_ID__c='X001');
5 *         insert g1;
6 *         Guest_Profile__c g2 = new Guest_Profile__c(Name='Test2', Guest_ID__c='X001');
7 *         Test.startTest();
8 *         try { insert g2; System.assert(false); } catch (DmlException e) { }
9 *         Test.stopTest();
10 *     }
11 * }
12 }
```

The log viewer at the bottom shows a table with the following data:

User	Application	Operation	Time	Status	Read	Size
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/runTests...	9/24/2025, 6:44:53 PM	Success	Unread	17.21 KB
Siddhi Ingole	Unknown	ReminderJob	9/24/2025, 6:30:00 PM	Success	Unread	4.95 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:25:00 PM	Success	Unread	3.04 KB
Siddhi Ingole	Unknown	QueueableHandler	9/24/2025, 6:23:31 PM	Success	Unread	4.08 KB
Siddhi Ingole	Unknown	/services/data/v64.0/tooling/executeA...	9/24/2025, 6:23:30 PM	Success	Unread	2.67 KB