

Phase 5 — Apex Programming (Developer)

Objective: Implement server-side logic in Salesforce using Apex to automate tasks like updating booking status, sending notifications, bulk processing, and scheduling reports.

A. Trigger: Update Campaign Donation Totals After Donation

Steps (Click by Click):

1. Go to **Setup** → **Quick Find** → **Object Manager** → Select **Donation**.
2. Click **Triggers** → **New**.
3. Enter Trigger Name: **DonationTrigger**.
4. Copy-paste the code:

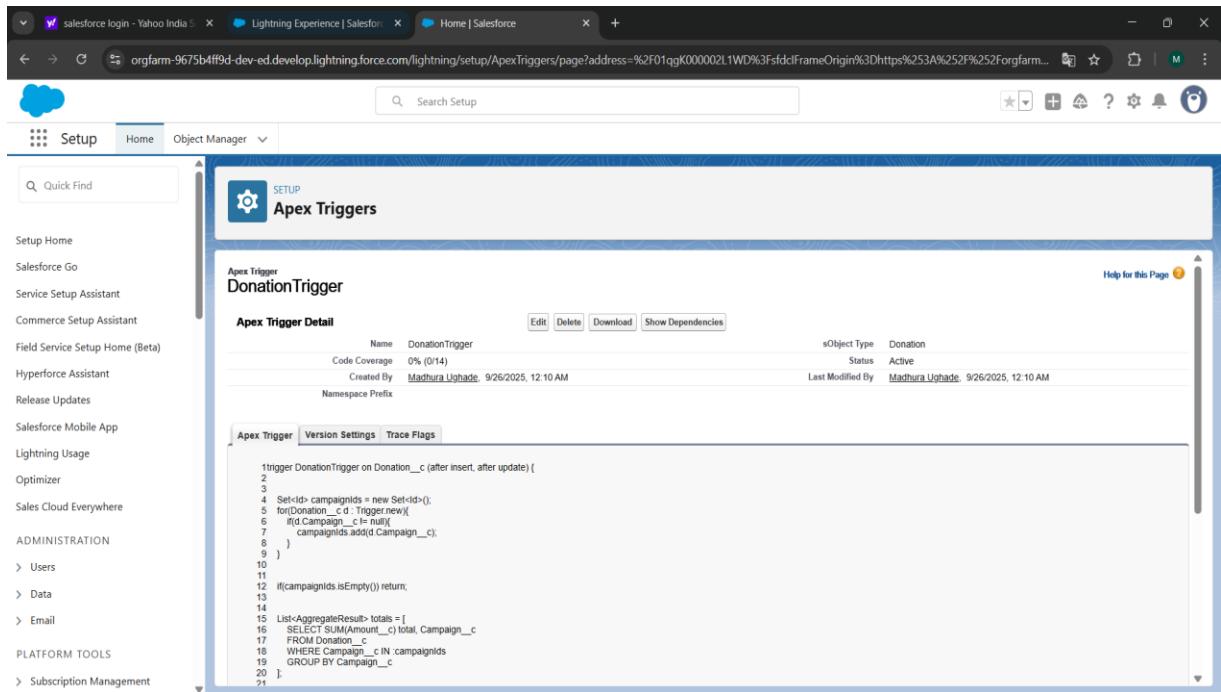
```
trigger DonationTrigger on Donation__c (before insert, after update) {  
    // Before Insert: Assign Donor ID  
    if(Trigger.isBefore && Trigger.isInsert){  
        for(Donation__c d : Trigger.new){  
            if(d.Donor_ID__c == null){  
                d.Donor_ID__c = 'DNR-' +  
                    String.valueOf(System.currentTimeMillis());  
            }  
        }  
  
        // After Update: Update Campaign Totals  
        if(Trigger.isAfter && Trigger.isUpdate){  
            Set<Id> campaignIds = new Set<Id>();  
            for(Donation__c d : Trigger.new){  
                if(d.Campaign__c != null){  
                    campaignIds.add(d.Campaign__c);  
                }  
            }  
        }  
    }  
}
```

```
}
```

```
}
```

```
List<Campaign_c> campaignsToUpdate = new  
List<Campaign_c>();  
for(Campaign_c c : [  
    SELECT Id, Total_Donations__c,  
        (SELECT Amount_c FROM Donations_r)  
    FROM Campaign_c  
    WHERE Id IN :campaignIds  
]) {  
    Decimal total = 0;  
    for(Donation_c d : c.Donations_r) {  
        total += d.Amount__c;  
    }  
    c.Total_Donations__c = total;  
    campaignsToUpdate.add(c);  
}  
  
if(!campaignsToUpdate.isEmpty()) {  
    update campaignsToUpdate;  
}  
}
```

6. Test: Create a **Donation** record with Campaign → Check if campaign Total_Donation__c update automatically.



B. SOQL Queries: Fetch Top Destinations

Steps (Click by Click):

1. Click **Gear Icon → Developer Console**.
2. In Developer Console, click **Query Editor** tab.
3. Paste the query:

```

SELECT Donor__r.Name, SUM(Amount__c) total
FROM Donation__c
WHERE CreatedDate = LAST_N_MONTHS:6
GROUP BY Donor__r.Name
ORDER BY total DESC
LIMIT 5

```

4. Click **Execute** → Results show top 5 donation in last 6 months.

C. Queueable Apex: Bulk Donation Imports

Steps (Click by Click):

1. Go to **Setup → Apex Classes → New**.
2. Name class: **DonationQueueable**.

3. Paste code: public class DonationQueueable implements Queueable

{

```
public List<Donation__c> newDonations;  
public DonationQueueable(List<Donation__c> donations){  
    this.newDonations = donations;  
}  
  
public void execute(QueueableContext context){  
    insert newDonations;  
}
```

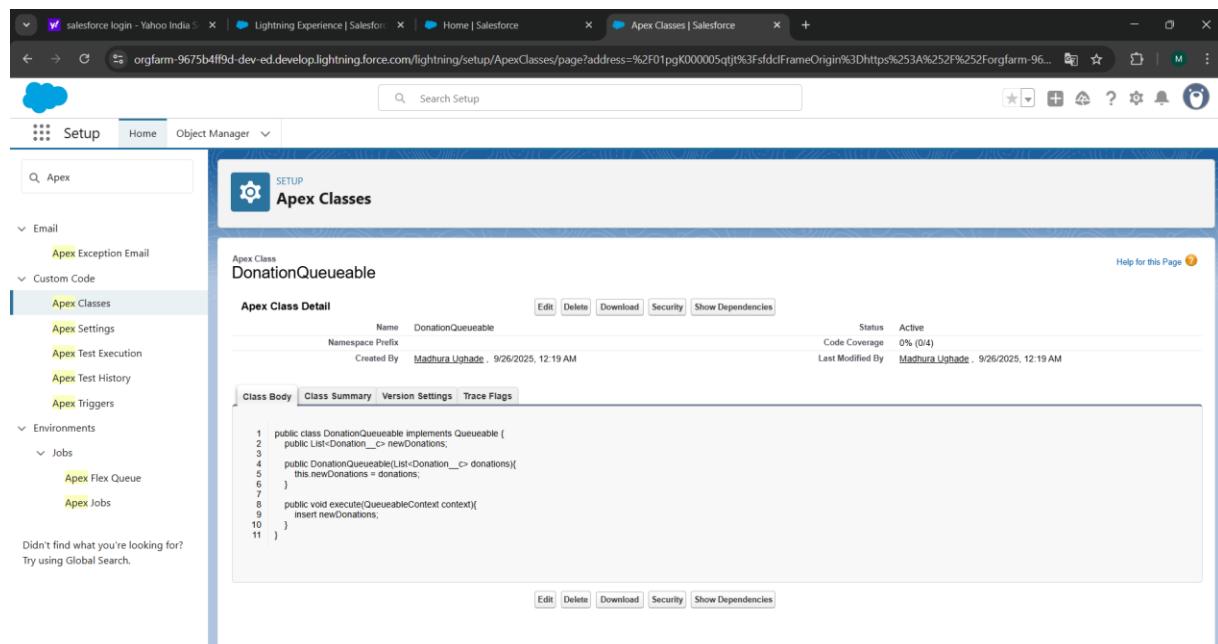
}

4. Click **Save**.

5. To execute, open **Developer Console → Execute Anonymous Window** and run:

```
System.enqueueJob(new DonationQueueable);
```

6. Bulk leads are inserted asynchronously.



D. Test Classes (Click by Click)

Purpose: Ensure triggers, batch jobs are working and deployable.

Steps:

1. **Setup** → **Apex Classes** → **New** → Name: TestDonationTrigger.

2. Paste code: @IsTest

```
public class TestDonationTrigger {
```

```
    @IsTest static void testCampaignUpdate() {
```

```
        // Step 1: Create test donor
```

```
        Donor_c donor = new Donorc(Name='Test Donor',  
        Email_c='test@example.com');  
        insert donor;
```

```
        // Step 2: Create test campaign
```

```
        Campaign_c camp = new Campaignc(Name='Test Campaign',  
        Total_Donations_c=0);  
        insert camp;
```

```
        // Step 3: Insert donation
```

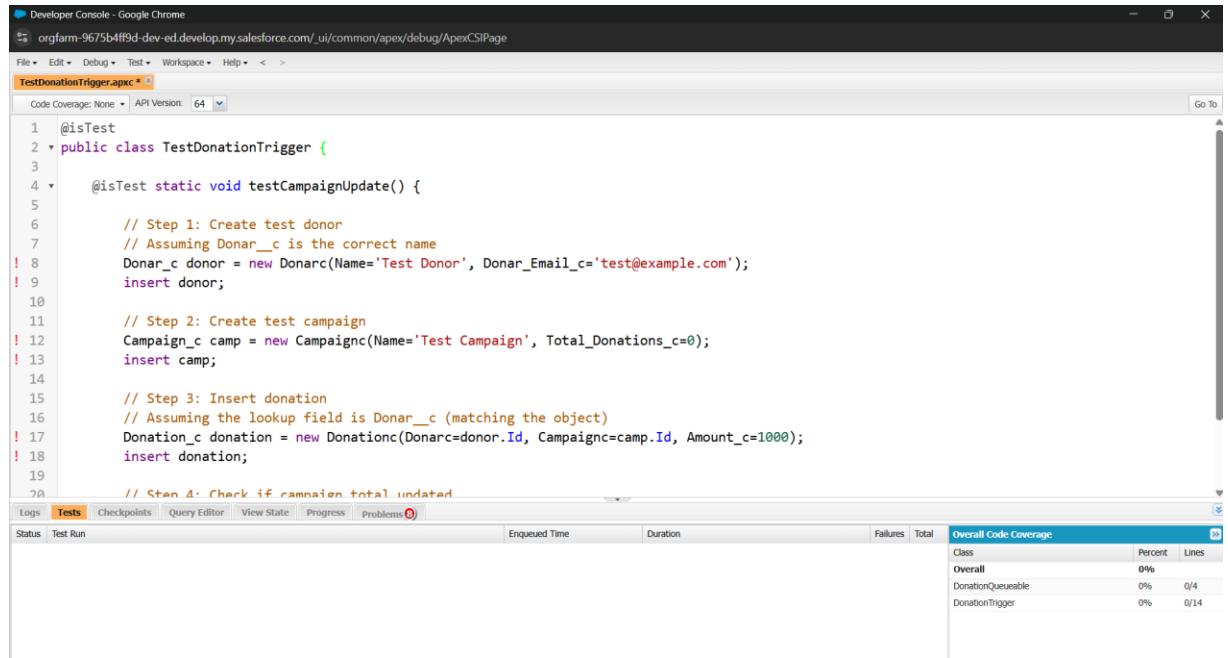
```
        Donation_c donation = new Donationc(Donorc=donor.Id,  
        Campaignc=camp.Id, Amount_c=1000);  
        insert donation;
```

```
        // Step 4: Check if campaign total updated
```

```
        Campaign_c updatedCamp = [SELECT Total_Donationsc FROM  
        Campaign_c WHERE Id=:camp.Id];  
        System.assertEquals(1000, updatedCamp.Total_Donations_c);  
    }
```

}Click Save.

3. Run All Tests → Ensure code coverage ≥ 75%.



```
1  @isTest
2  public class TestDonationTrigger {
3
4  @isTest static void testCampaignUpdate() {
5
6      // Step 1: Create test donor
7      // Assuming Donar__c is the correct name
8      Donar__c donor = new Donar__c(Name='Test Donor', Donar_Email__c='test@example.com');
9      insert donor;
10
11     // Step 2: Create test campaign
12     Campaign__c camp = new Campaign__c(Name='Test Campaign', Total_Donations__c=0);
13     insert camp;
14
15     // Step 3: Insert donation
16     // Assuming the lookup field is Donar__c (matching the object)
17     Donation__c donation = new Donation__c(Donar__c=donor.Id, Campaign__c=camp.Id, Amount__c=1000);
18     insert donation;
19
20     // Step 4: Check if campaign total updated

```

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage												
						<table border="1"><thead><tr><th>Class</th><th>Percent</th><th>Lines</th></tr></thead><tbody><tr><td>Overall</td><td>0%</td><td></td></tr><tr><td>DonationQueueable</td><td>0%</td><td>0/4</td></tr><tr><td>DonationTrigger</td><td>0%</td><td>0/14</td></tr></tbody></table>	Class	Percent	Lines	Overall	0%		DonationQueueable	0%	0/4	DonationTrigger	0%	0/14
Class	Percent	Lines																
Overall	0%																	
DonationQueueable	0%	0/4																
DonationTrigger	0%	0/14																

All Apex automation is now tested and ready for deployment.