

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 Booking Status when Payment is Received

Steps (Click by Click):

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

```
5. trigger PaymentTrigger on Payment__c (after insert, after update) {  
    Set<Id> bookingIds = new Set<Id>();  
    for(Payment__c p : Trigger.new){  
        if(p.Booking__c != null){  
            bookingIds.add(p.Booking__c);  
        }  
    }  
  
    List<Booking__c> bookingsToUpdate = new List<Booking__c>();  
    for(Booking__c b : [  
        SELECT Id, Total_Amount__c, Total_Paid__c, Booking_Status__c  
        FROM Booking__c  
        WHERE Id IN :bookingIds  
    ]){  
        if(b.Total_Paid__c >= b.Total_Amount__c && b.Booking_Status__c != 'Confirmed'){  
            b.Booking_Status__c = 'Confirmed';  
            bookingsToUpdate.add(b);  
        }  
    }  
  
    if(bookingsToUpdate.size() > 0){  
        update bookingsToUpdate;
```

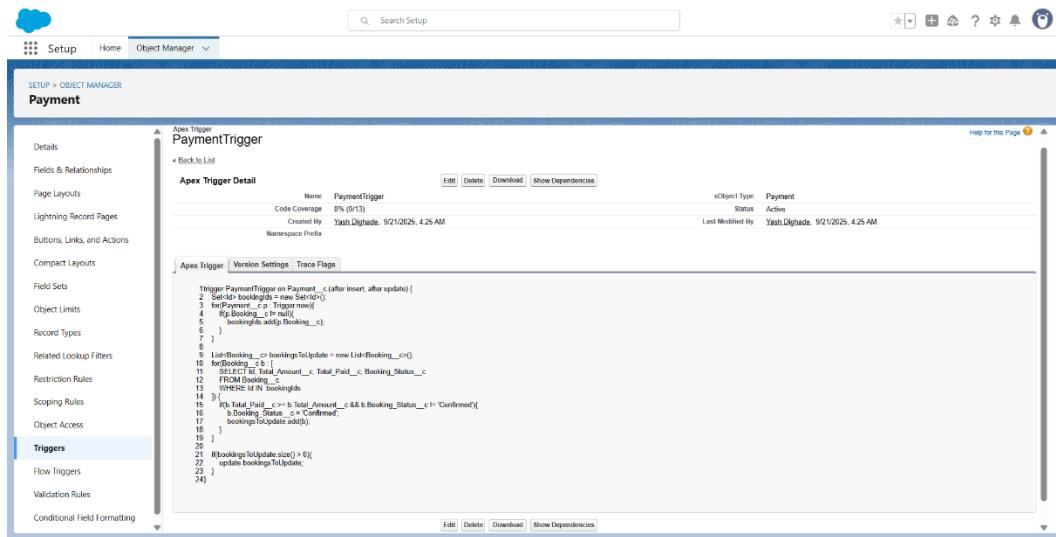
```

    }
}

```

6. Click Save.

7. Test: Create a **Payment** record with Status = Paid → Booking status should automatically update.



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 Travel_Package__r.Destination__c, COUNT(Id) total
FROM Booking__c
WHERE CreatedDate = LAST_N_MONTHS:6
GROUP BY Travel_Package__r.Destination__c
ORDER BY total DESC
LIMIT 5

```

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

C. Queueable Apex: Bulk Lead Imports

Steps (Click by Click):

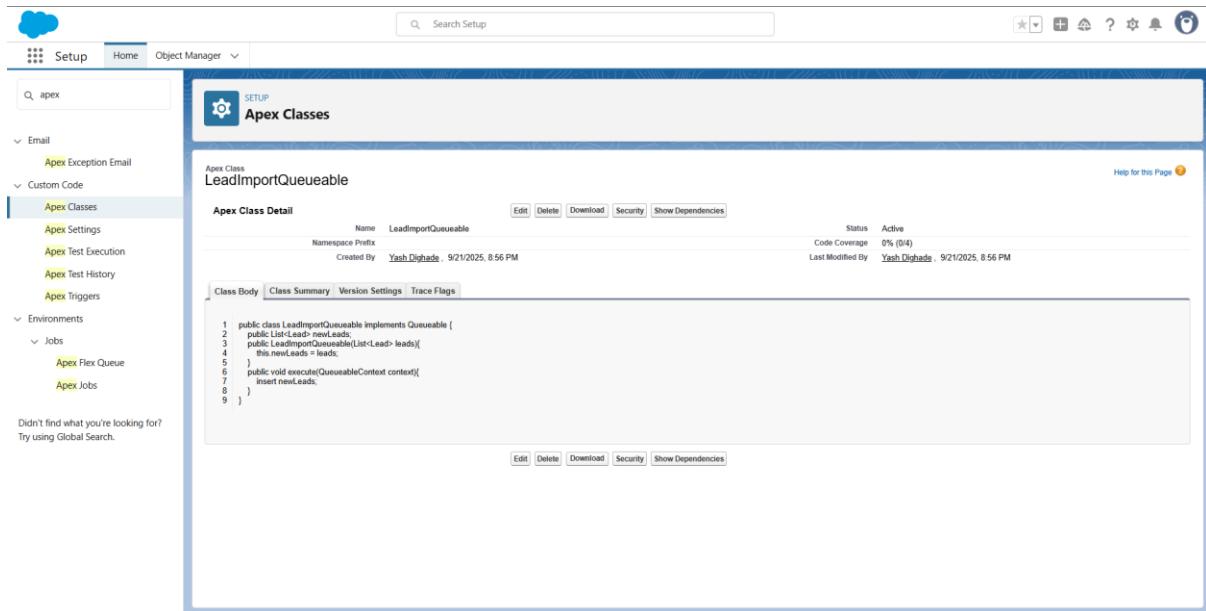
1. Go to **Setup** → **Apex Classes** → **New**.
2. Name class: `LeadImportQueueable`.
3. Paste code:

```
public class LeadImportQueueable implements Queueable {  
    public List<Lead> newLeads;  
    public LeadImportQueueable(List<Lead> leads){  
        this.newLeads = leads;  
    }  
    public void execute(QueueableContext context){  
        insert newLeads;  
    }  
}
```

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

```
System.enqueueJob(new LeadImportQueueable(campaignLeads));
```

6. Bulk leads are inserted asynchronously.



D. Test Classes (Click by Click)

Purpose: Ensure triggers, batch jobs, and future methods are covered for deployment.

Steps:

1. **Setup → Apex Classes → New** → Name: TestBookingTrigger.
2. Paste code:

@IsTest

```

public class TestBookingTrigger {
    @IsTest static void testPaymentTrigger(){
        Customer__c cust = new Customer__c(First_Name__c='Test',
                                         Last_Name__c='User', Email__c='test@example.com');
        insert cust;
    }
}

```

```

Booking__c book = new Booking__c(Customer__c=customer.Id,
                                  Travel_Start_Date__c=Date.today(), Booking_Status__c='Pending');

insert book;

```

```

    Payment__c pay = new Payment__c(Booking__c=book.Id,
Amount__c=1000, Status__c='Paid');

insert pay;

```

```

Booking__c updatedBooking = [SELECT Booking_Status__c FROM
Booking__c WHERE Id=:book.Id];

System.assertEquals('Confirmed', updatedBooking.Booking_Status__c);

}
}
```

3. Click Save.

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

Status	Test Run	Enqueued Time	Duration	Failures	Total
x	707gk00000DnH72	Mon Sep 22 2025 09:31:00 GM...		1	1

Overall Code Coverage		
Class	Percent	Lines
Overall	0%	
LeadImportQueueable	0%	0/4
PaymentTrigger	0%	0/13

- ✓ All Apex automation is now tested and ready for deployment.