Phase 5: Apex Programming (Developer) – Pet Care CRM

PetCare CRM

A Salesforce-Based Pet Shop Management System for Customer Engagement and Service Automation

Goal: Add advanced logic to handle pet service bookings efficiently and prevent conflicts.

---

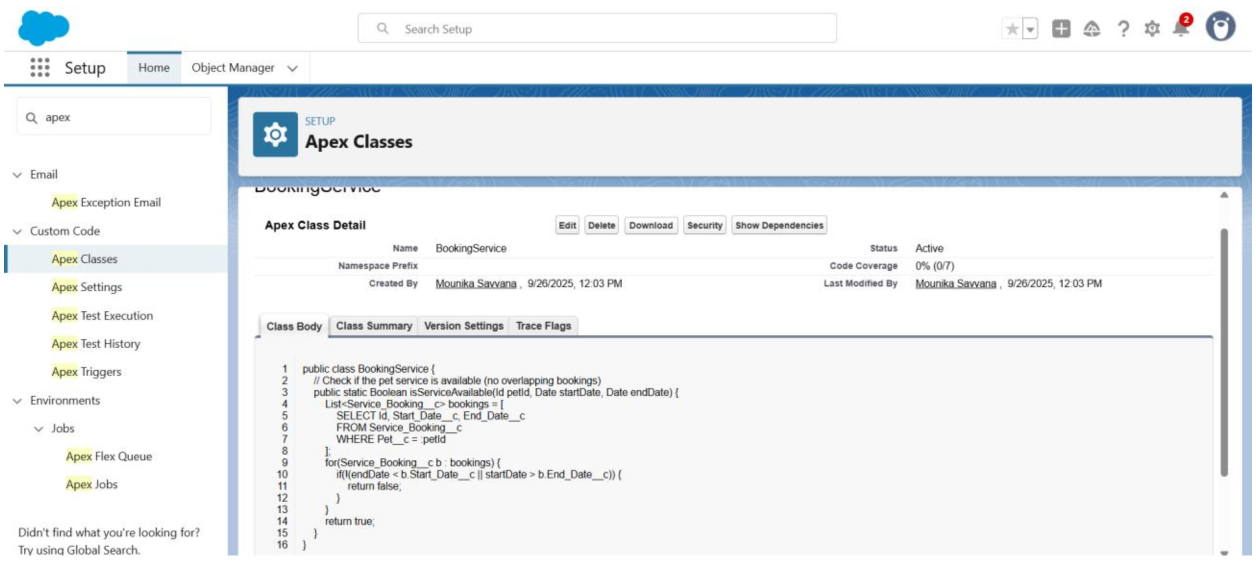
1. Classes & Objects – BookingService

Purpose:

- To create reusable logic for checking pet service availability before booking.
- Keeps code modular and easier to maintain.

Steps:

1. Go to **Setup → Apex Classes → New** in Salesforce.
2. Create a class called BookingService.
3. Add a method isServiceAvailable which:
   - Accepts a pet ID, start date, and end date.
   - Checks all existing bookings for that pet.
   - Returns false if any overlap is found; otherwise, returns true.



Apex Class Detail

Name	BookingService	Status	Active
Namespace Prefix		Code Coverage	0% (0/7)
Created By	Mounika Savvana , 9/26/2025, 12:03 PM	Last Modified By	Mounika Savvana , 9/26/2025, 12:03 PM

```
public class BookingService {
  // Check if the pet service is available (no overlapping bookings)
  public static Boolean isServiceAvailable(Id petId, Date startDate, Date endDate) {
    List<Service_Booking__c> bookings = [
      SELECT Id, Start_Date__c, End_Date__c
      FROM Service_Booking__c
      WHERE Pet__c = :petId
    ];
    for(Service_Booking__c b : bookings) {
      if(!(endDate < b.Start_Date__c || startDate > b.End_Date__c)) {
        return false;
      }
    }
    return true;
  }
}
```

2. Apex Trigger – ServiceBookingTrigger

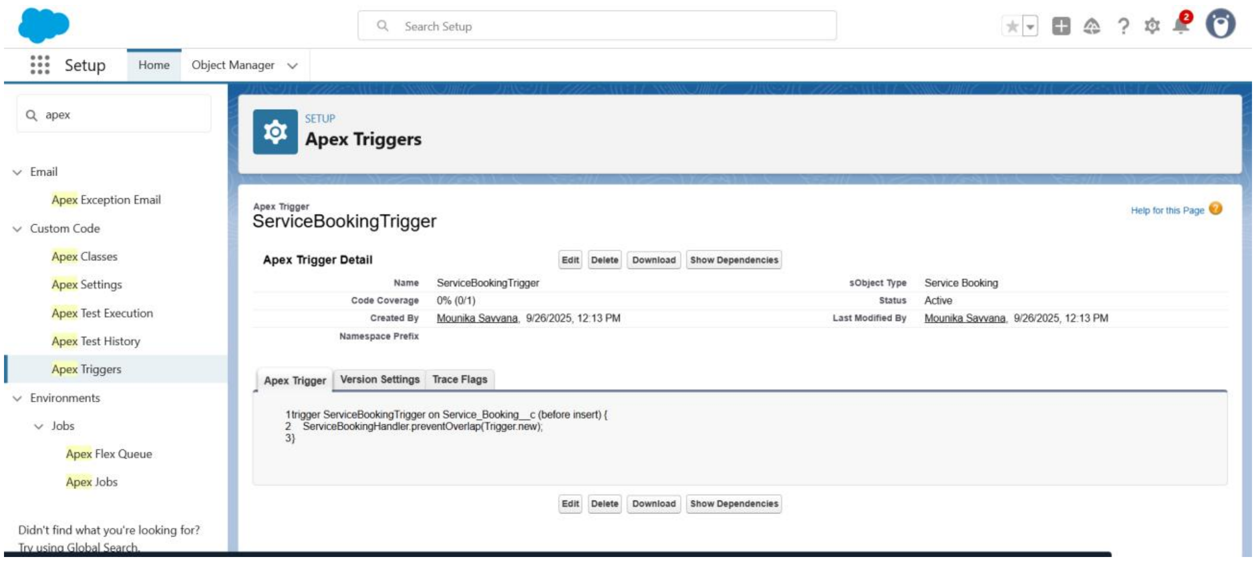
Purpose:

- To automatically check for overlapping bookings whenever a new booking is created.
- Prevents manual errors and ensures data integrity.

Steps:

1. Go to **Setup → Object Manager → Service_Booking__c → Triggers → New**.
2. Create a trigger called ServiceBookingTrigger.
3. Call the handler class from the trigger (best practice).
4. **Code:**

```
trigger ServiceBookingTrigger on Service_Booking__c (before insert) {
  ServiceBookingHandler.preventOverlap(Trigger.new);
}
```



Apex Trigger
ServiceBookingTrigger

Apex Trigger Detail

Name	ServiceBookingTrigger	sObject Type	Service Booking
Code Coverage	0% (0/1)	Status	Active
Created By	Mounika Savvana, 9/26/2025, 12:13 PM	Last Modified By	Mounika Savvana, 9/26/2025, 12:13 PM
Namespace Prefix			

```
trigger ServiceBookingTrigger on Service_Booking__c (before insert) {
  ServiceBookingHandler.preventOverlap(Trigger.new);
}
```

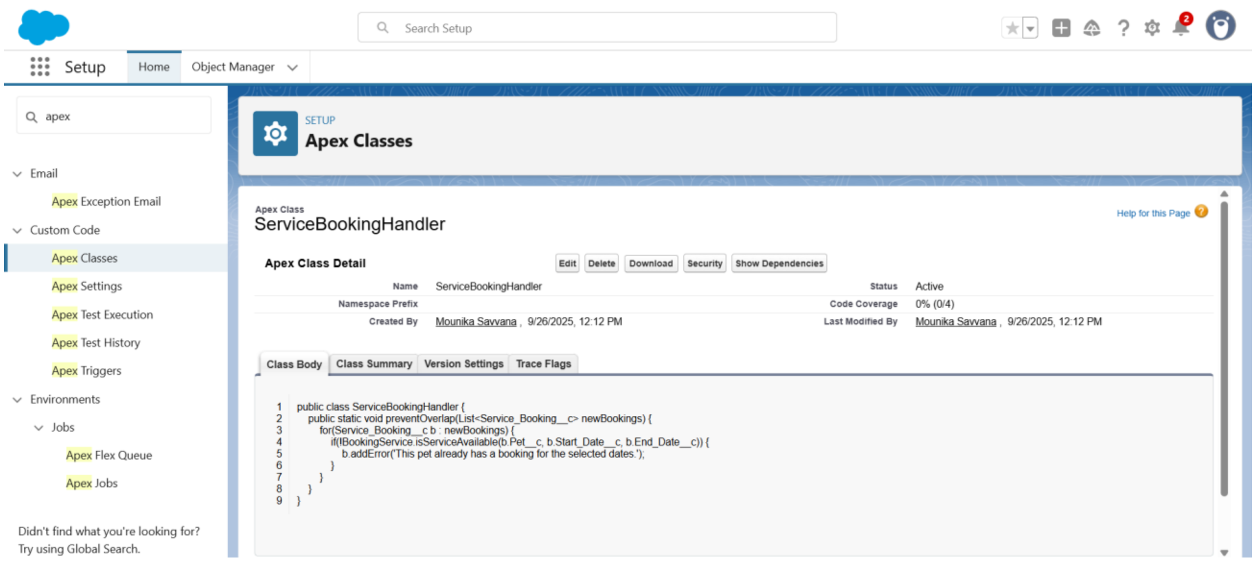
3. Trigger Handler Class – ServiceBookingHandler

Purpose:

- Separates business logic from the trigger (Trigger Design Pattern).
- Makes the system scalable and easier to maintain.

Steps:

1. Go to **Setup → Apex Classes → New**.
2. Create a class called ServiceBookingHandler.
3. Add a method preventOverlap that:
    - Loops through new bookings.
    - Checks availability using BookingService.isServiceAvailable.
    - Throws an error using addError if overlapping booking is detected.



Apex Class
ServiceBookingHandler

Apex Class Detail

Name	ServiceBookingHandler	Status	Active
Namespace Prefix		Code Coverage	0% (0/4)
Created By	Mounika Savvana , 9/26/2025, 12:12 PM	Last Modified By	Mounika Savvana , 9/26/2025, 12:12 PM

```
public class ServiceBookingHandler {
    public static void preventOverlap(List<Service_Booking__c> newBookings) {
        for(Service_Booking__c b : newBookings) {
            if(!BookingService.isServiceAvailable(b.Pet__c, b.Start_Date__c, b.End_Date__c)) {
                b.addError('This pet already has a booking for the selected dates.');
            }
        }
    }
}
```

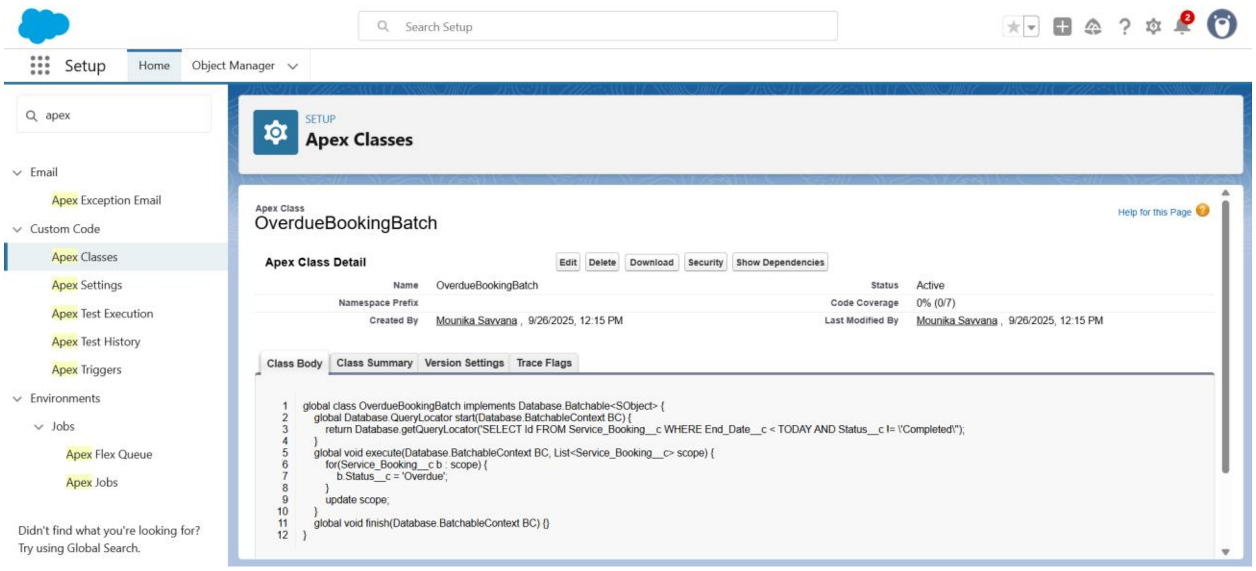
4. Batch Apex – Overdue Bookings

Purpose:

- Automatically mark bookings as overdue if the end date has passed.
- Runs as a background job nightly.

Steps:

1. Create a new Apex class implementing Database.Batchable<SObject>.
2. Query all bookings that have ended but are not marked as “Completed”.
3. Update their status to “Overdue”.



Apex Class
OverdueBookingBatch

Apex Class Detail

Name	OverdueBookingBatch	Status	Active
Namespace Prefix		Code Coverage	0% (0/7)
Created By	Mounika Savvana , 9/26/2025, 12:15 PM	Last Modified By	Mounika Savvana , 9/26/2025, 12:15 PM

```
global class OverdueBookingBatch implements Database.Batchable<SObject> {
    global Database.QueryLocator start(Database.BatchableContext BC) {
        return Database.getQueryLocator('SELECT Id FROM Service_Booking__c WHERE End_Date__c < TODAY AND Status__c != \'Completed\'');
    }
    global void execute(Database.BatchableContext BC, List<Service_Booking__c> scope) {
        for(Service_Booking__c b : scope) {
            b.Status__c = 'Overdue';
        }
        update scope;
    }
    global void finish(Database.BatchableContext BC) {}
}
```

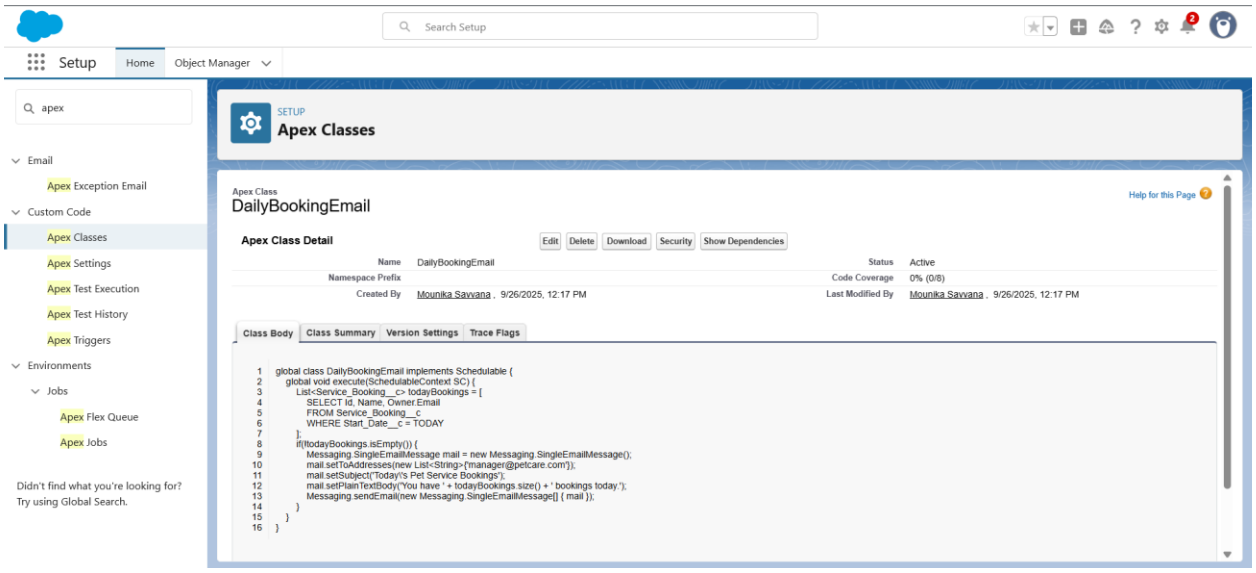
5. Scheduled Apex – Daily Booking Email

Purpose:

- Send a daily email to the manager with today's bookings.

Steps:

1. Create a new Apex class implementing Schedulable.
2. Query all bookings for today.
3. Use Messaging.SingleEmailMessage to send a summary email.
4. Schedule this class to run every morning.



Apex Class
DailyBookingEmail

Apex Class Detail

Name	DailyBookingEmail	Status	Active
Namespace Prefix		Code Coverage	0% (0/8)
Created By	Mounika Savvana , 9/26/2025, 12:17 PM	Last Modified By	Mounika Savvana , 9/26/2025, 12:17 PM

```
global class DailyBookingEmail implements Schedulable {
    global void execute(SchedulableContext SC) {
        List<Service_Booking__c> todayBookings = [
            SELECT Id, Name, Owner.Email
            FROM Service_Booking__c
            WHERE Start_Date__c = TODAY
        ];
        if(!todayBookings.isEmpty()) {
            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
            mail.setToAddresses(new List<String>{'manager@petcare.com'});
            mail.setSubject('Today\'s Pet Service Bookings');
            mail.setPlainTextBody('You have ' + todayBookings.size() + ' bookings today.');
            Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
        }
    }
}
```

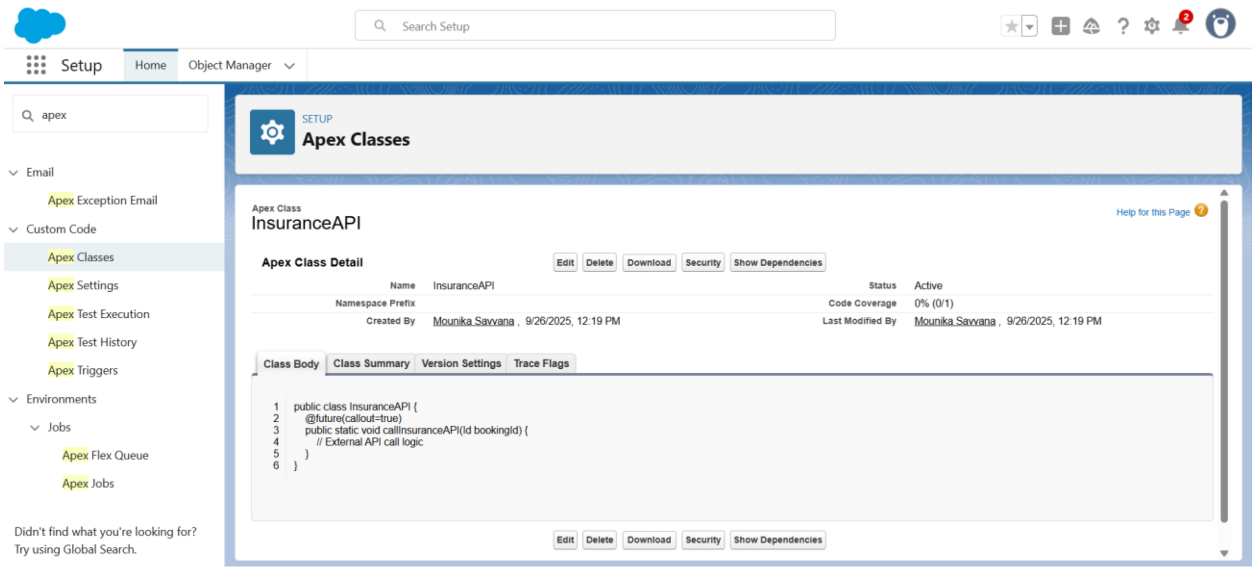
6. Future Method – Async API Call for Insurance

Purpose:

- Call external pet insurance API asynchronously.
- Ensures that the main booking process is not delayed by API calls.

Steps:

1. Create a new Apex class.
2. Add a @future(callout=true) method to make the external call.



Apex Class
InsuranceAPI

Apex Class Detail

Name	InsuranceAPI	Status	Active
Namespace Prefix		Code Coverage	0% (0/1)
Created By	Mounika Savvana , 9/26/2025, 12:19 PM	Last Modified By	Mounika Savvana , 9/26/2025, 12:19 PM

```
public class InsuranceAPI {
    @future(callout=true)
    public static void callInsuranceAPI(Id bookingId) {
        // External API call logic
    }
}
```

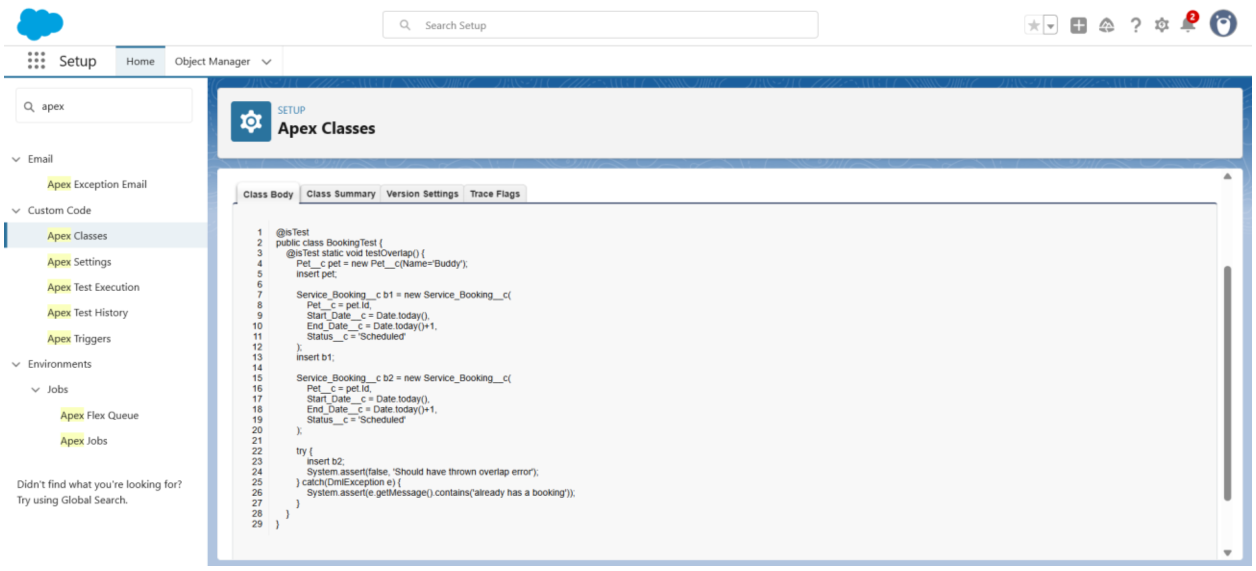
7. Test Class – BookingTest

Purpose:

- Ensure triggers and booking logic work correctly.
- Verify that overlapping bookings are blocked.

Steps:

1. Create a new Apex class with @isTest.
2. Insert a pet record.
3. Insert a first booking successfully.
4. Try inserting an overlapping booking and assert that an error occurs.



```
@isTest
public class BookingTest {
    @isTest static void testOverlap() {
        Pet__c pet = new Pet__c(Name='Buddy');
        insert pet;

        Service_Booking__c b1 = new Service_Booking__c(
            Pet__c = pet.Id,
            Start_Date__c = Date.today(),
            End_Date__c = Date.today()+1,
            Status__c = 'Scheduled'
        );
        insert b1;

        Service_Booking__c b2 = new Service_Booking__c(
            Pet__c = pet.Id,
            Start_Date__c = Date.today(),
            End_Date__c = Date.today()+1,
            Status__c = 'Scheduled'
        );

        try {
            insert b2;
            System.assert(false, 'Should have thrown overlap error');
        } catch(DmlException e) {
            System.assert(e.getMessage().contains('already has a booking'));
        }
    }
}
```