Phase 5: Apex Triggers and Classes

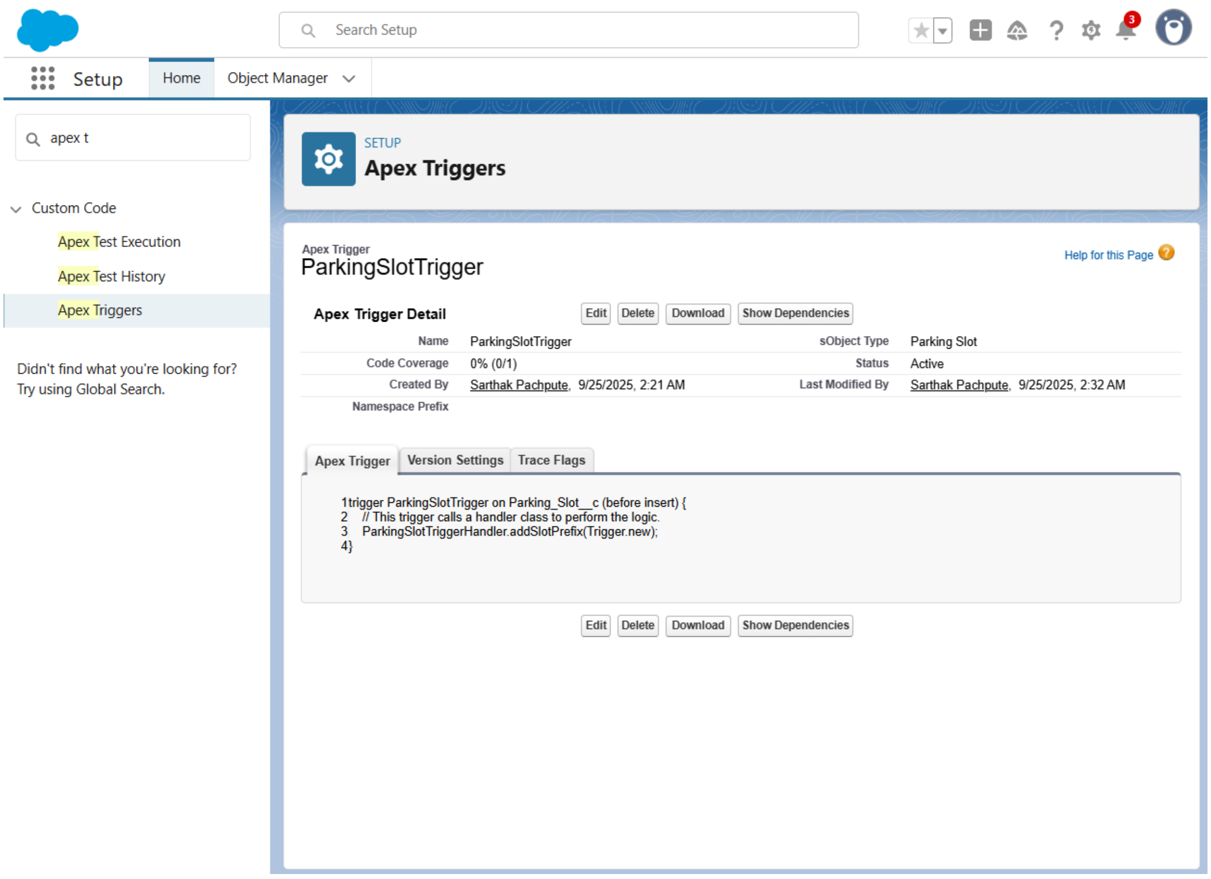
Objective

The objective of this phase was to implement complex, server-side business logic using Apex code. While declarative tools like Flow are powerful, Apex is required for more advanced automation and custom processing. This phase focuses on using an Apex Trigger and a handler class to manage the real-time status of parking slots.

Key Development Activities

1. Apex Trigger (BookingTrigger) An Apex trigger was created to execute custom logic in response to database events on the Booking object.

- **Function:** The trigger is configured to fire after a booking record is created, updated, or deleted.
- **Purpose:** It serves as the entry point for our automation. Instead of containing complex logic itself, it follows best practices by delegating the processing to a dedicated Apex handler class, ParkingSlotController.



Apex Trigger
ParkingSlotTrigger

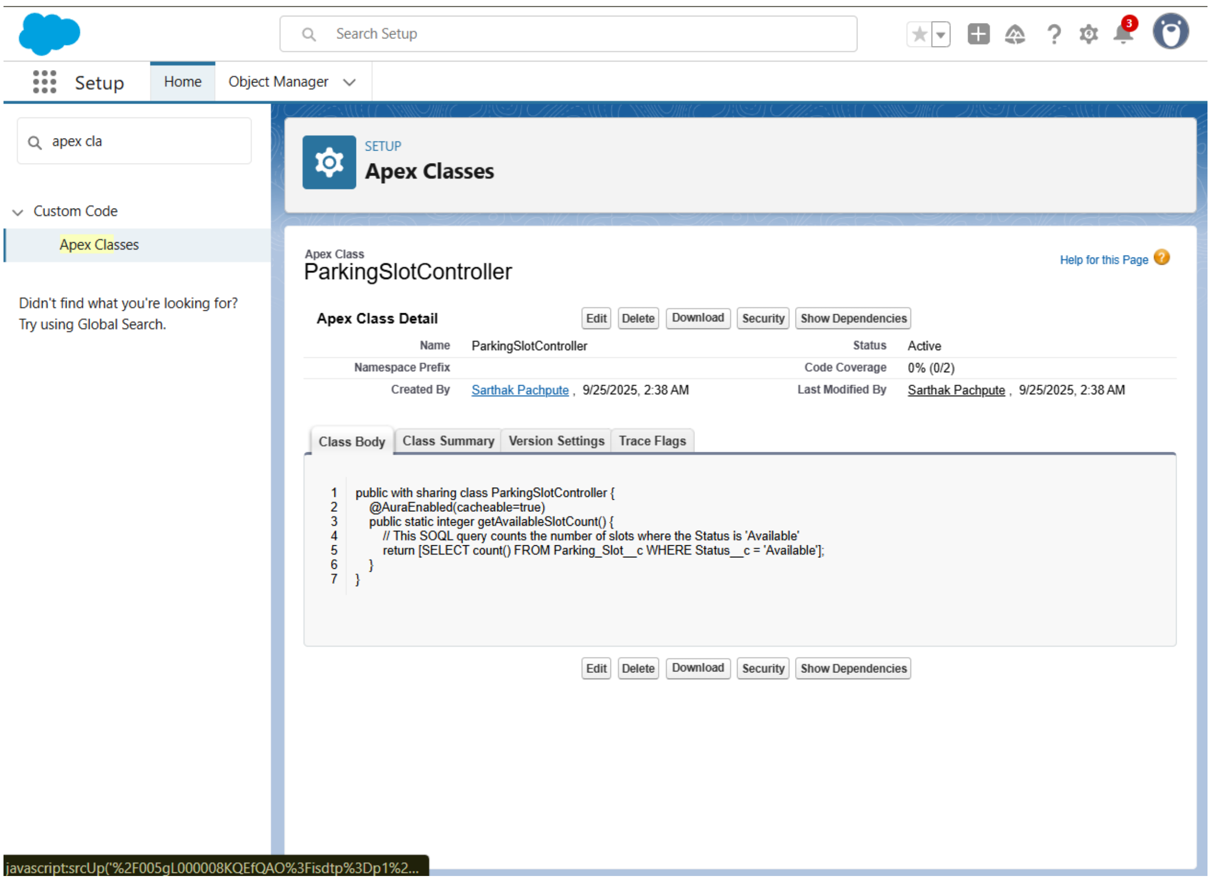
Apex Trigger Detail

Name	ParkingSlotTrigger	sObject Type	Parking Slot
Code Coverage	0% (0/1)	Status	Active
Created By	Sarthak Pachpute, 9/25/2025, 2:21 AM	Last Modified By	Sarthak Pachpute, 9/25/2025, 2:32 AM
Namespace Prefix			

```
trigger ParkingSlotTrigger on Parking_Slot__c (before insert) {
    // This trigger calls a handler class to perform the logic.
    ParkingSlotTriggerHandler.addSlotPrefix(Trigger.new);
}
```

2. Apex Class (ParkingSlotController) A separate Apex handler class, ParkingSlotController, was developed to contain the actual business logic.

- **Function:** This class includes methods that are called by the trigger. When a booking is confirmed or canceled, a method in this class is executed to find the corresponding Parking Slot and update its Status__c field to "Occupied" or "Available".
- **Purpose:** Separating the logic into a handler class makes the code cleaner, easier to test, and reusable. It is the standard for professional Apex development.



Apex Class
ParkingSlotController

Apex Class Detail

Name	ParkingSlotController	Status	Active
Namespace Prefix		Code Coverage	0% (0/2)
Created By	Sarthak Pachpute , 9/25/2025, 2:38 AM	Last Modified By	Sarthak Pachpute , 9/25/2025, 2:38 AM

```
public with sharing class ParkingSlotController {
    @AuraEnabled(cacheable=true)
    public static integer getAvailableSlotCount() {
        // This SOQL query counts the number of slots where the Status is 'Available'
        return [SELECT count() FROM Parking_Slot__c WHERE Status__c = 'Available'];
    }
}
```

Conclusion

Phase 5 marks a significant step into the programmatic capabilities of the Salesforce platform. By implementing an Apex trigger and the ParkingSlotController class, we have created a robust, real-time system for managing parking slot availability that is both powerful and scalable.