

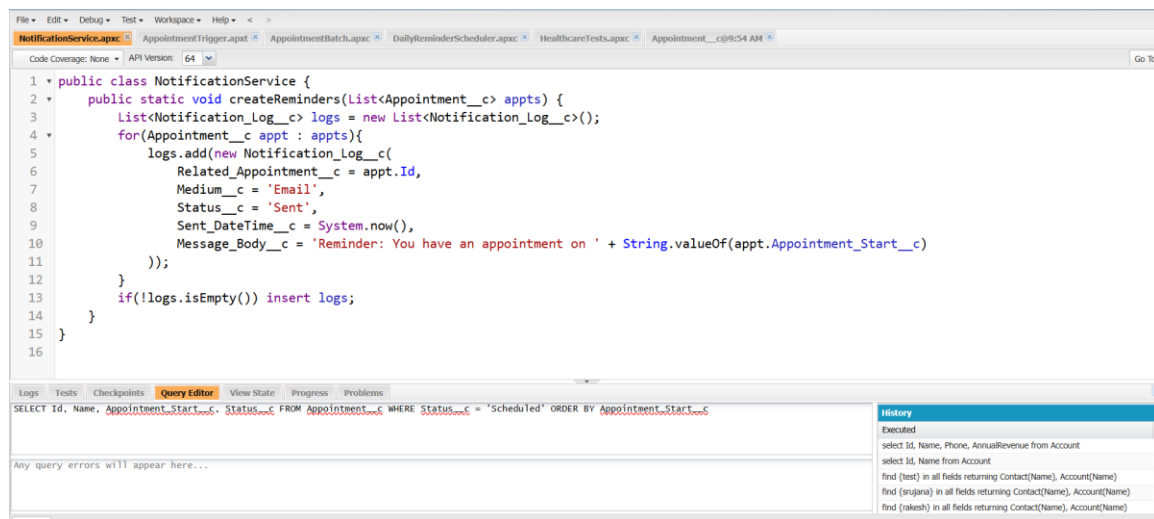
Smart HealthCare Portal

Phase 5: Apex Programming (Developer)

In this phase, Apex programming concepts were implemented to enhance the Smart Healthcare CRM project. This included writing Apex classes, triggers, batch processes, scheduled jobs, SOQL and SOSL queries, and test classes to ensure coverage and reliability.

1. Apex Classes

A utility class NotificationService was created to handle automatic reminder creation by inserting Notification_Log__c records when an appointment requires reminders.



```
1 public class NotificationService {
2     public static void createReminders(List<Appointment__c> appts) {
3         List<Notification_Log__c> logs = new List<Notification_Log__c>();
4         for(Appointment__c appt : appts){
5             logs.add(new Notification_Log__c(
6                 Related_Appointment__c = appt.Id,
7                 Medium__c = 'Email',
8                 Status__c = 'Sent',
9                 Sent_DateTime__c = System.now(),
10                Message_Body__c = 'Reminder: You have an appointment on ' + String.valueOf(appt.Appointment_Start__c)
11            ));
12        }
13        if(!logs.isEmpty()) insert logs;
14    }
15 }
16
```

SELECT Id, Name, Appointment_Start__c, Status__c FROM Appointment__c WHERE Status__c = 'Scheduled' ORDER BY Appointment_Start__c

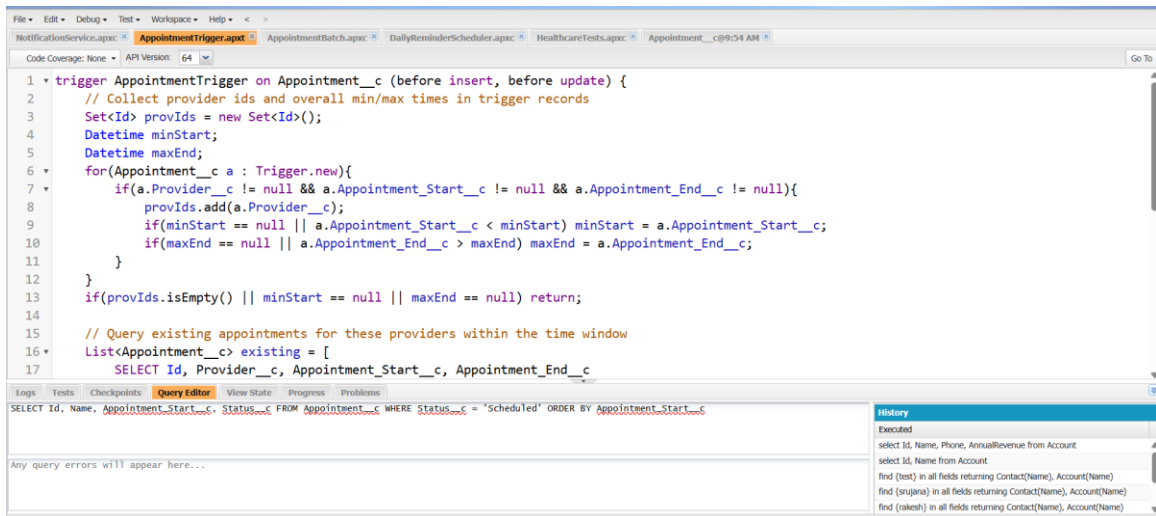
Any query errors will appear here...

History

- Executed
- select Id, Name, Phone, AnnualRevenue from Account
- select Id, Name from Account
- find (test) in all fields returning Contact(Name), Account(Name)
- find (srujana) in all fields returning Contact(Name), Account(Name)
- find (rakesh) in all fields returning Contact(Name), Account(Name)

2. Apex Triggers

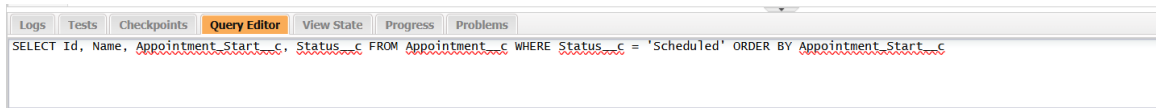
An Apex Trigger on Appointment__c was developed to prevent double booking of providers. It ensures that no two appointments overlap for the same provider.



3. SOQL and SOSL Queries

SOQL Example: Fetch scheduled appointments for a provider.

SOSL Example: Search for a patient by name.



List<List<Contact>> results = [FIND 'Anita*' IN ALL FIELDS RETURNING Contact(Id, Name, Phone, Email)];

System.debug(results);

4. Batch Apex

A batch class `AppointmentBatch` was created to mark past appointments as Completed automatically.

The screenshot shows the IDE with the `AppointmentBatch.apex` file open. The code defines a class `AppointmentBatch` that implements `Database.Batchable<SObject>`. It includes methods for `start`, `execute`, and `finish`. The `execute` method iterates over a scope of `Appointment__c` records, updating their status to 'Completed'.

Below the code editor, the 'Logs' tab is active, displaying a table of execution logs for the 'AppointmentBatch' application.

User	Application	Operation	Time	Status	Read	Size
Srujana Balam	Unknown	/services/data/v44.0/tooling/executeA...	9/25/2025, 9:54:35 AM	The Apex job named "Daily Appointment..."	Unread	4.75 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:27 AM	Success	Unread	3.08 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:26 AM	Success	Unread	3.75 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Success	Unread	2.42 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Update failed. First exception on row 0...	Unread	9.72 KB
Srujana Balam	Unknown	/services/data/v44.0/tooling/executeA...	9/25/2025, 9:42:24 AM	Success	Unread	4.06 KB

5. Scheduled Apex

A scheduled class `DailyReminderScheduler` was created to send reminders every morning at 8 AM for appointments scheduled within the next day.

The screenshot shows the IDE with the `DailyReminderScheduler.apex` file open. The code defines a class `DailyReminderScheduler` that implements `Schedulable`. It includes a `execute` method that queries for upcoming appointments and calls `NotificationService.createReminders`.

Below the code editor, the 'Logs' tab is active, displaying a table of execution logs for the 'DailyReminderScheduler' application.

User	Application	Operation	Time	Status	Read	Size
Srujana Balam	Unknown	/services/data/v44.0/tooling/executeA...	9/25/2025, 9:54:35 AM	The Apex job named "Daily Appointment..."	Unread	4.75 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:27 AM	Success	Unread	3.08 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:26 AM	Success	Unread	3.75 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Success	Unread	2.42 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Update failed. First exception on row 0...	Unread	9.72 KB
Srujana Balam	Unknown	/services/data/v44.0/tooling/executeA...	9/25/2025, 9:42:24 AM	Success	Unread	4.06 KB

6. Test Classes

A test class `HealthcareTests` was developed to cover triggers, batch apex, and the notification service. It creates test data including providers and patients (Contacts), verifies that double booking is prevented, ensures batch jobs mark past appointments completed, and confirms notification logs are created.

The screenshot shows an IDE with the following components:

- Code Editor:** Contains the `HealthcareTests` class with two utility methods: `createProvider` and `createPatient`.
- Log Table:** A table with columns: User, Application, Operation, Time, Status, Read, and Size. It lists several log entries for user 'Srujana Balam'.

```
@isTest
private class HealthcareTests {

    // Utility method to create a provider
    private static Provider__c createProvider(String name) {
        Provider__c prov = new Provider__c(Name = name, Specialty__c = 'Cardiology');
        insert prov;
        return prov;
    }

    // Utility method to create a patient (Contact)
    private static Contact createPatient(String fname, String lname) {
        Contact pat = new Contact(FirstName = fname, LastName = lname, Email = fname+'.'+lname+'@test.com');
        insert pat;
        return pat;
    }
}
```

User	Application	Operation	Time	Status	Read	Size
Srujana Balam	Unknown	/services/data/v64.0/tooling/executeA...	9/25/2025, 9:54:35 AM	The Apex job named "Daily Appointment...	Unread	4.75 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:27 AM	Success	Unread	3.08 KB
Srujana Balam	Unknown	Batch Apex	9/25/2025, 9:42:26 AM	Success	Unread	3.75 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Success	Unread	2.42 KB
Srujana Balam	Unknown	SerialBatchApexRangeChunkHandler	9/25/2025, 9:42:26 AM	Update failed. First exception on row 0...	Unread	9.72 KB
Srujana Balam	Unknown	/services/data/v64.0/tooling/executeA...	9/25/2025, 9:42:24 AM	Success		4.06 KB

7. Summary

In Phase 5, Apex programming was applied to automate and enforce complex business logic that could not be handled by declarative tools alone. This included preventing provider double-booking, automatically marking past appointments as completed, sending daily reminders, and ensuring robust testing with Apex test classes. Together, these features strengthened the Smart Healthcare CRM and made it more reliable for end users.