

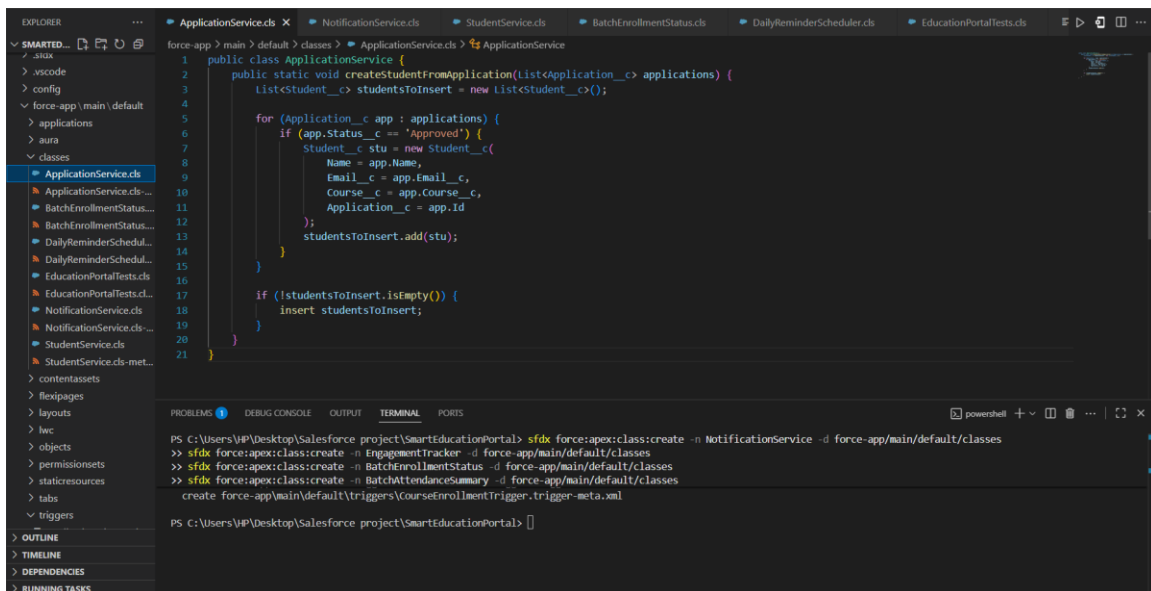
Smart Education Portal

Phase 5: Apex Programming (Developer)

In this phase, Apex programming concepts are implemented to enhance the Smart Education Portal. This includes Apex classes, triggers, batch processes, scheduled jobs, SOQL/SOSL queries, and test classes.

1. Apex Classes

- ApplicationService.cls → Handles student admission workflows (auto-create Student__c record once an Application__c is approved)

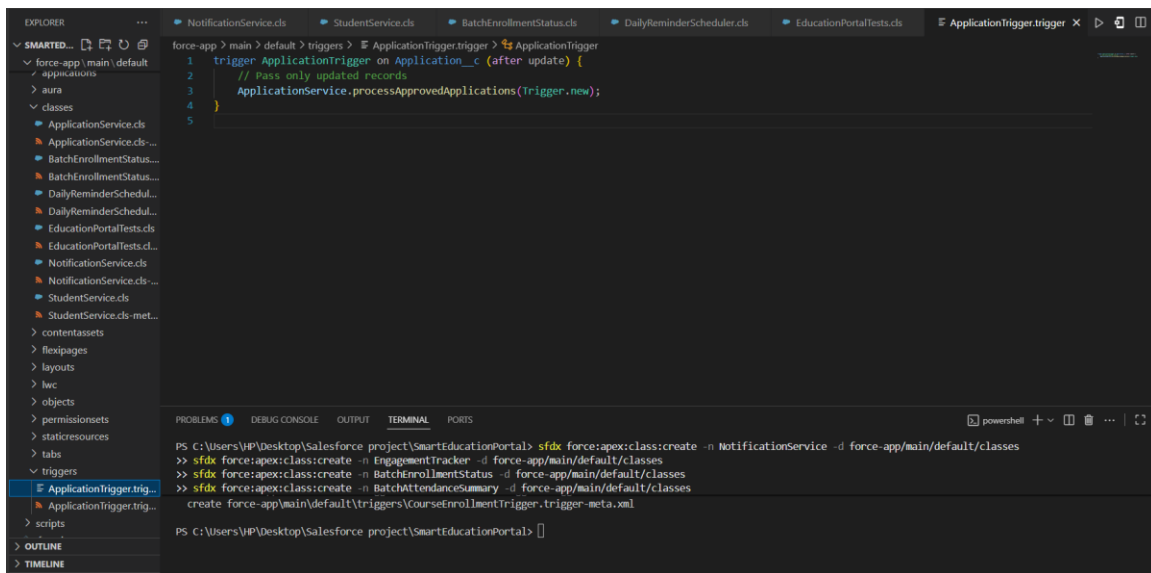


```
1 public class ApplicationService {
2     public static void createStudentFromApplication(List<Application__c> applications) {
3         List<Student__c> studentsToInsert = new List<Student__c>();
4
5         for (Application__c app : applications) {
6             if (app.Status__c == 'Approved') {
7                 Student__c stu = new Student__c(
8                     Name = app.Name,
9                     Email_c = app.Email_c,
10                    Course_c = app.Course_c,
11                    Application__c = app.Id
12                );
13                studentsToInsert.add(stu);
14            }
15        }
16
17        if (!studentsToInsert.isEmpty()) {
18            insert studentsToInsert;
19        }
20    }
21 }
```

```
PS C:\Users\HP\Desktop\Salesforce project\SmartEducationPortal> sfdx force:apex:class:create -n NotificationService -d force-app/main/default/classes
>> sfdx force:apex:class:create -n EngagementTracker -d force-app/main/default/classes
>> sfdx force:apex:class:create -n BatchEnrollmentStatus -d force-app/main/default/classes
>> sfdx force:apex:class:create -n BatchAttendanceSummary -d force-app/main/default/classes
create force-app/main/default/triggers/courseenrollmenttrigger.trigger-meta.xml
PS C:\Users\HP\Desktop\Salesforce project\SmartEducationPortal>
```

2. Apex Triggers

- ApplicationTrigger → Prevents duplicate applications for the same student and program.



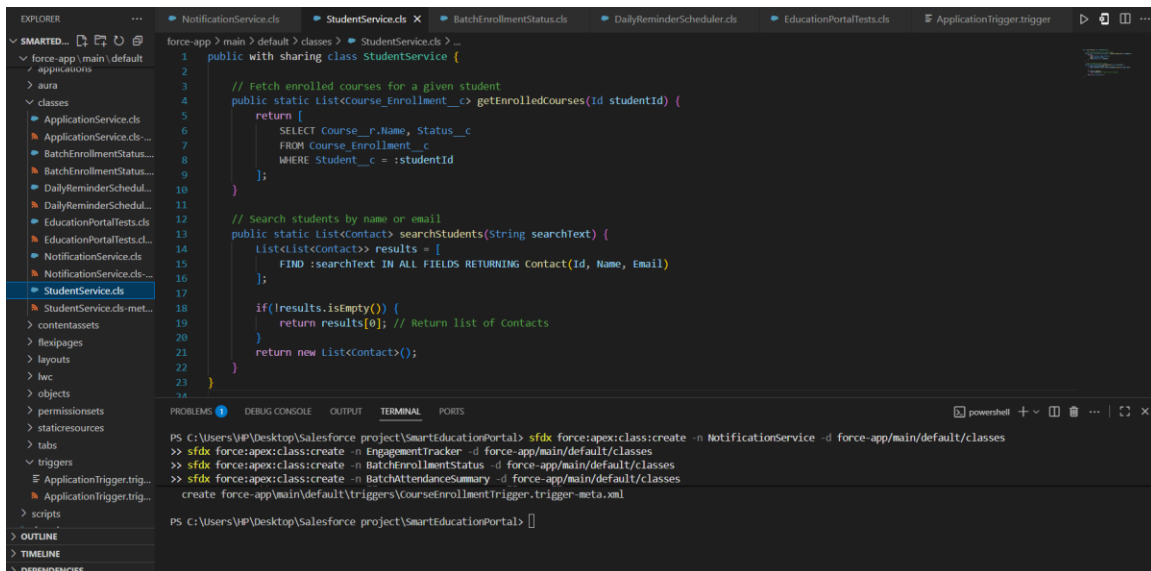
3. SOQL & SOSL Queries

SOQL Example: Fetch enrolled courses for a student:

```
List<Course_Enrollment__c> enrollments = [SELECT Course__r.Name, Status__c FROM
Course_Enrollment__c WHERE Student__c = :studentId];
```

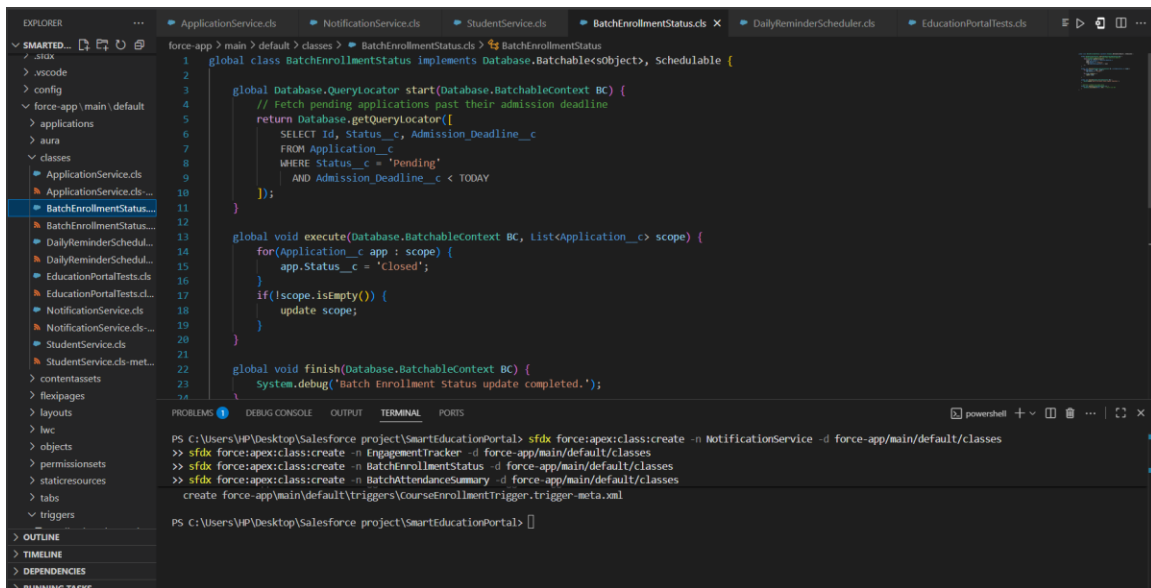
SOSL Example: Search students by name/email across records:

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



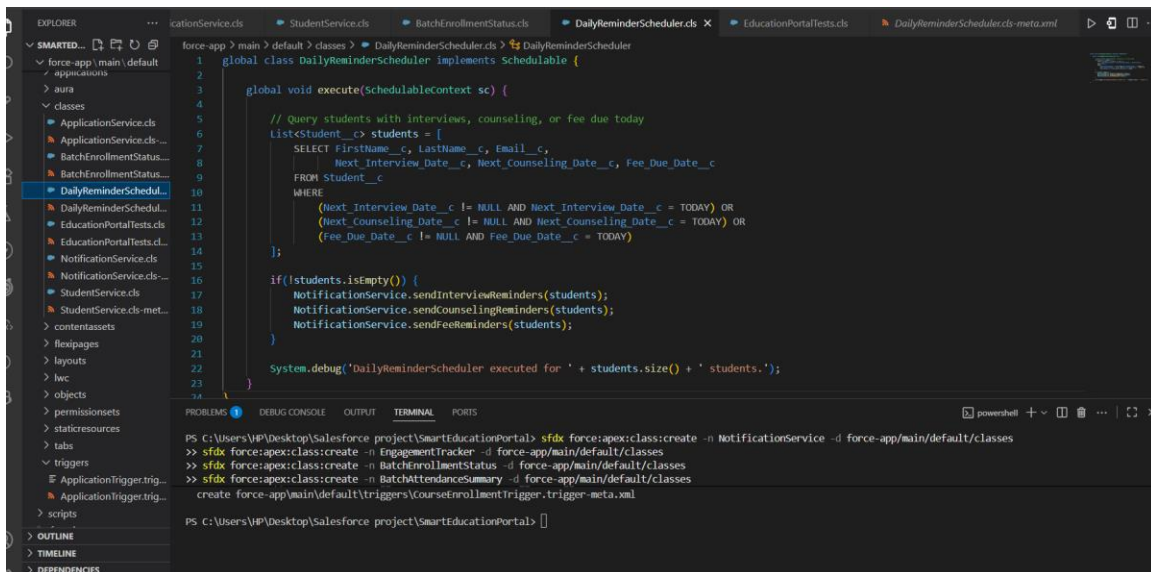
4. Batch Apex

- BatchEnrollmentStatus.cls → Updates all pending applications after the admission deadline to 'Closed'.



5. Scheduled Apex

- DailyReminderScheduler.cls → Sends reminders every morning for:
 - Today's interviews
 - Upcoming counseling sessions
 - Fee due notifications



6. Test Classes

- EducationPortalTests.cls → Covers triggers, batch jobs, and service classes. Creates sample students, applications, courses, and enrollments to validate:
 - Duplicate prevention
 - Seat availability updates

- Automatic reminders
- Batch jobs updating statuses

The screenshot shows the Visual Studio Code interface for a Salesforce project named 'SmartEducationPortal'. The Explorer pane on the left shows the project structure, including the 'force-app/main/default/classes' directory. The main editor displays the 'EducationPortalTests.cls' file, which contains a test class for the 'EducationPortalTests' namespace. The test class includes a static method 'testApplicationTriggerAndDuplicatePrevention()' that creates a test application, approves it, and verifies the student record.

```

1  @isTest
2  private class EducationPortalTests {
3
4
5      @isTest static void testApplicationTriggerAndDuplicatePrevention() {
6          // Create test Application
7          Application__c app = new Application__c{
8              FirstName__c = 'Test',
9              LastName__c = 'Student',
10             Email__c = 'test@student.com',
11             Status__c = 'pending',
12             Program__c = 'computer Science'
13         };
14         insert app;
15
16         // Approve the application - triggers ApplicationService
17         app.Status__c = 'Approved';
18         update app;
19
20         // Verify Student__c created
21         Student__c student = [SELECT Id, FirstName__c, LastName__c, Email__c, Program__c
22                               FROM Student__c WHERE Application__c = :app.Id LIMIT 1];
23         System.assertEquals('test', student.FirstName__c);
24
25         // Test Application same application again - should not create duplicate
26     }
27 }

```

The terminal window at the bottom shows the command prompt for the project, with the following commands and output:

```

PS C:\Users\JP\Desktop\Salesforce project\SmartEducationPortal> sfdx force:apex:class:create -n NotificationService -d force-app/main/default/classes
>> sfdx force:apex:class:create -n EngagementTracker -d force-app/main/default/classes
>> sfdx force:apex:class:create -n BatchEnrollmentStatus -d force-app/main/default/classes
>> sfdx force:apex:class:create -n BatchAttendanceSummary -d force-app/main/default/classes
create force-app/main/default/triggers/CourseEnrollmentTrigger.trigger-meta.xml
PS C:\Users\JP\Desktop\Salesforce project\SmartEducationPortal>

```