

# EduConnect : Smart Learning and Enrollment CRM Portal(Phase 5)

## Goal:

The goal of this project is to build a Salesforce-based Student Enrollment and Learning Portal that automates the complete admission process. It will streamline student inquiries, course enrollments, and fee tracking while reducing manual effort and delays. Automated notifications and reminders will improve communication with students and faculty. Overall, the system will enhance efficiency, accuracy, and reporting, ensuring a smoother enrollment experience for educational institutes.

## Problem Statement:

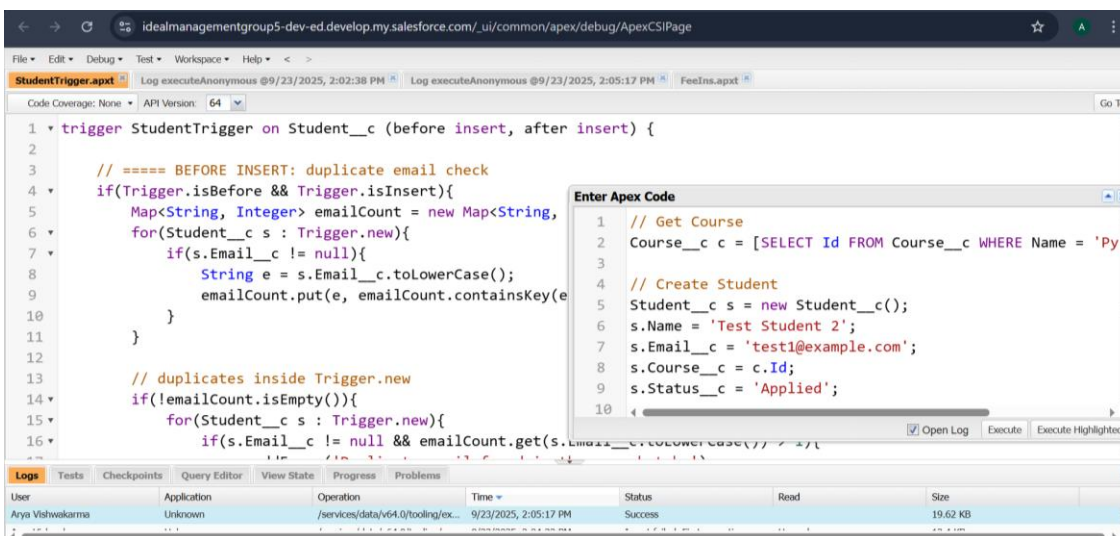
Educational institutes face challenges in managing student inquiries, admissions, and course enrollments due to manual processes. This causes delays, missed follow-ups, and inefficiency. A Salesforce-based portal will automate lead capture, admissions, fee tracking, and notifications, ensuring faster enrollment and better reporting.

## Phase 5: Apex Programming (Developer)

Phase 5 focuses on **Apex Programming** to enhance system functionality through custom logic. In this phase, custom **Apex Triggers, Classes, and SOQL queries** were developed to automate business processes and provide advanced functionalities for the Smart Student Enrollment and Learning Portal.

## Apex Trigger

- Developed an Apex Trigger to automatically generate an **Enrollment record** whenever a new **Student** is added to a **Course**.
- This ensures enrollments are consistently created without requiring manual intervention.

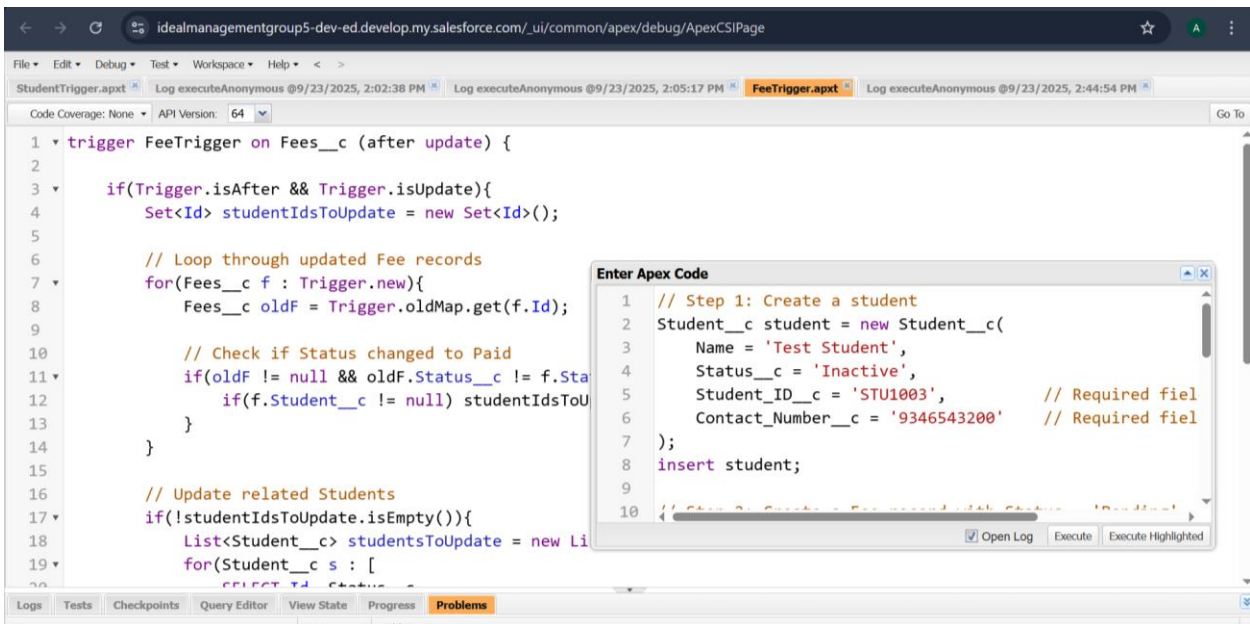


The screenshot displays the Salesforce IDE interface. The main editor shows the code for a trigger named `StudentTrigger` on the `Student__c` object. The trigger is configured to execute before insert and after insert. The code includes a comment indicating a 'BEFORE INSERT: duplicate email check' and logic to create a new enrollment record if the email is not already in the system. A second comment indicates 'duplicates inside Trigger.new' and logic to handle duplicates. The bottom of the IDE shows a table with columns: User, Application, Operation, Time, Status, Read, and Size. The table contains one row of data for a user named 'Arya Vishwakarma'.

```
1 trigger StudentTrigger on Student__c (before insert, after insert) {
2
3     // ===== BEFORE INSERT: duplicate email check
4     if (Trigger.isBefore && Trigger.isInsert) {
5         Map<String, Integer> emailCount = new Map<String, Integer>();
6         for (Student__c s : Trigger.new) {
7             if (s.Email__c != null) {
8                 String e = s.Email__c.toLowerCase();
9                 emailCount.put(e, emailCount.containsKey(e) ? emailCount.get(e) + 1 : 1);
10            }
11        }
12
13        // duplicates inside Trigger.new
14        if (!emailCount.isEmpty()) {
15            for (Student__c s : Trigger.new) {
16                if (s.Email__c != null && emailCount.get(s.Email__c.toLowerCase()) > 1) {
17                    // Duplicate email found
18                }
19            }
20        }
21    }
22}
```

```
1 // Get Course
2 Course__c c = [SELECT Id FROM Course__c WHERE Name = 'Python'];
3
4 // Create Student
5 Student__c s = new Student__c();
6 s.Name = 'Test Student 2';
7 s.Email__c = 'test1@example.com';
8 s.Course__c = c.Id;
9 s.Status__c = 'Applied';
10
```

User	Application	Operation	Time	Status	Read	Size
Arya Vishwakarma	Unknown	/services/data/v64.0/tooling/execute/apex/StudentTrigger.apex	9/23/2025, 2:05:17 PM	Success		19.62 KB



## SOQL Queries

- Designed SOQL queries to fetch **student-course enrollment reports**.
- Generated **fee collection summaries** per batch and course.

```

21
22 // duplicates against existing records
23 Set<String> emails = emailCount.keySet();
24 if(!emails.isEmpty()){
25     for(Student__c existing : [
26         SELECT Email__c FROM Student__c WHERE Email__c IN :emails
27     ]){
28         String em = existing.Email__c == null ? '' : existing.Email__c.toLowerCase();
29         for(Student__c s : Trigger.new){
30             if(s.Email__c != null && s.Email__c.toLowerCase() == em){
31                 s.addError('A student with this Email already exists.');

```

```

45
46 Map<Id, Course__c> courseMap = new Map<Id, Course__c>();
47 if(!courseIds.isEmpty()){
48     courseMap = new Map<Id, Course__c>(
49         [SELECT Id, Course_Fees__c FROM Course__c WHERE Id IN :courseIds]
50     );
51 }
52

```

```

16 // Update related Students
17 if(!studentIdsToUpdate.isEmpty()){
18     List<Student__c> studentsToUpdate = new List<Student__c>();
19     for(Student__c s : [
20         SELECT Id, Status__c
21         FROM Student__c
22         WHERE Id IN :studentIdsToUpdate
23     ]){
24         s.Status__c = 'Active';
25         studentsToUpdate.add(s);
26     }

```

## Test Classes

- Added Test Classes to validate Apex logic and ensure **code coverage**.
- Improved reliability and enabled successful deployments.
- Maintained compliance with Salesforce **best practices**.

The screenshot shows the Salesforce 'Students' list view. The table contains four rows: 'Test Student', 'Test Student 2', 'Test Student', and 'Arya'. A red bracket highlights the first three rows, with the text 'Made Using Apex Triggers Automatically (Test Student)' next to it. The interface includes a search bar, filters, and action buttons like 'New', 'Import', 'Change Owner', and 'Assign Label'.

The screenshot shows the details of a 'Test Student 2' record. A red arrow points to the 'Status' field, which is labeled 'APPLIED'. The 'Details' tab is active, showing fields like 'Status', 'Created By', 'Last Modified By', 'Student Name', 'Student\_id', 'Email', 'Date of Birth', and 'Age'. The 'Activity' panel on the right shows 'Upcoming & Overdue' activities, with a message 'No activities to show. Get started by sending an email, scheduling a task, and more.'

idealmanagementgroup5-dev-ed.develop.lightning.force.com/lightning/r/Student\_\_c/a02Qy00000NBAJIIAP/view

Search...

Sales Home Opportunities Leads Tasks Files Accounts Contacts Campaigns Dashboards Reports Chatter Groups Calendar Students More

Student  
Test Student

New Contact Edit New Opportunity

ACTIVE

Related Details

Status  
Active

Created By  
Arya Vishwakarma, 23/09/2025, 2:44 pm

Owner  
Arya Vishwakarma

Last Modified By  
Arya Vishwakarma, 23/09/2025, 2:44 pm

Student Information

Student Name  
Test Student

Student\_id  
STU1003

Email

Date of Birth

Age

Contact Number  
9346543200

Address

Activity

Filters: All time • All activities • All types

Refresh Expand All View All

Upcoming & Overdue

No activities to show.  
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

To Do List

## Conclusion

Phase 5 successfully introduced **Apex Programming** into the project. By implementing triggers, classes, and SOQL queries, along with robust test coverage, the system achieved advanced automation and custom logic. This ensured **accurate fee calculations, streamlined enrollments, and comprehensive reporting**, strengthening the portal's functionality.