Library Management :- Phase 5: Apex Programming (Developer)

1. Classes & Objects

Created an Apex utility class **TeacherService:**

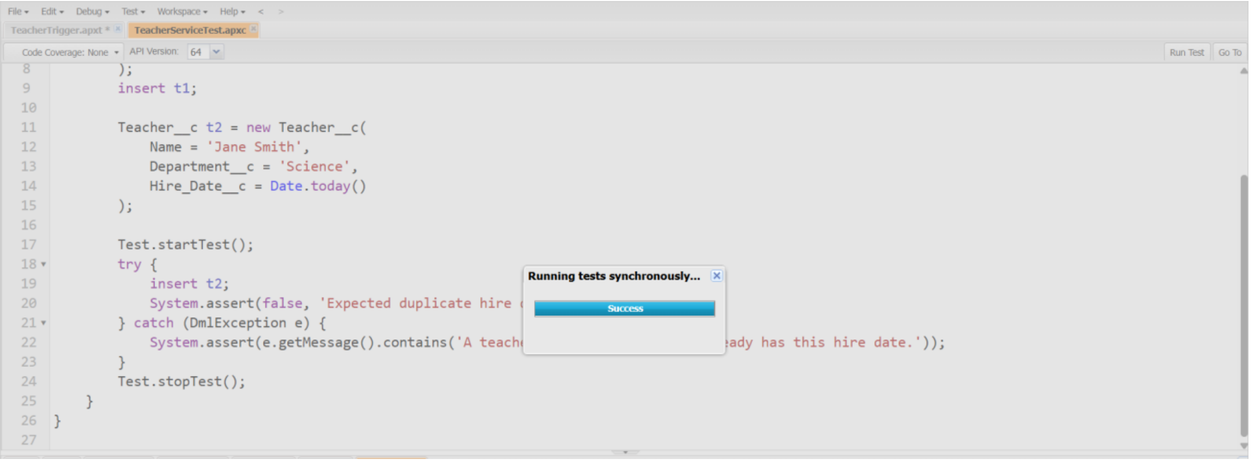
- Inner Comparator: **TeacherHireDateComparator** – sorts teachers by **Hire_Date__c.**
- Static Method:
- **preventDuplicateHireDates**(**List<Teacher_c> incoming, Map<Id, Teacher_c> oldMap**) – checks new/updated teacher records against existing teacher records to ensure no conflicting hire dates for the same department (or other criteria) and uses **addError()** to block duplicates.

File → New → Apex Class.

Name it TeacherServiceTest → click **OK**.

Paste test code:

Save.



```
        );
        insert t1;

        Teacher__c t2 = new Teacher__c(
            Name = 'Jane Smith',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        try {
            insert t2;
            System.assert(false, 'Expected duplicate hire (
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('A teach                                   eady has this hire date.'));
        }
        Test.stopTest();
    }
}
```

Running tests synchronously...

Success

2. Apex Trigger

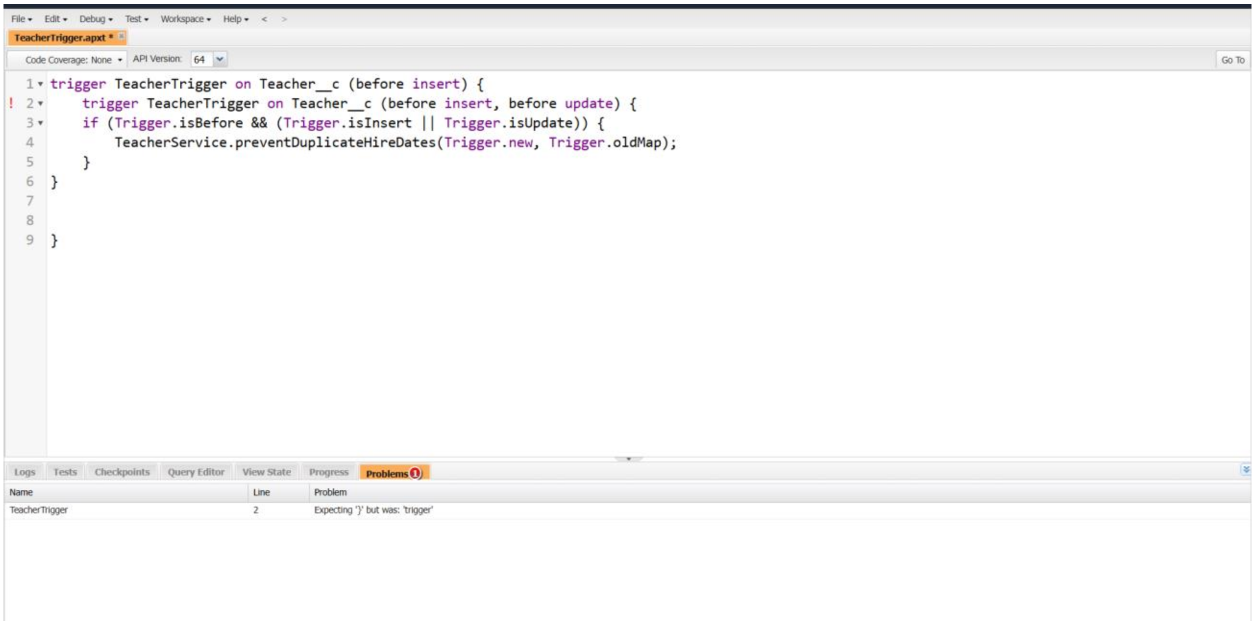
File → New → Apex Trigger.

Enter:

- Name: TeacherTrigger
- sObject: Teacher__c

Click **Submit**.

Save.

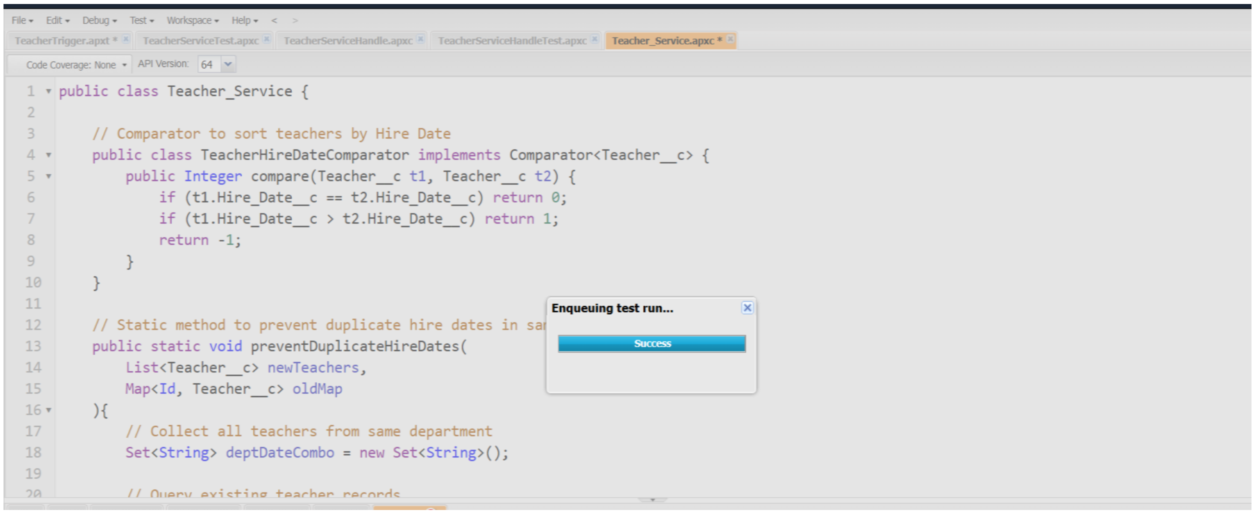


```
trigger TeacherTrigger on Teacher__c (before insert) {
    trigger TeacherTrigger on Teacher__c (before insert, before update) {
    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
        TeacherService.preventDuplicateHireDates(Trigger.new, Trigger.oldMap);
    }
}


}
```

Name	Line	Problem
TeacherTrigger	2	Expecting '}' but was: 'trigger'

- 3.Trigger Design Pattern:

Business logic is kept in the TeacherService class; the trigger on Teacher__c simply calls the class method.

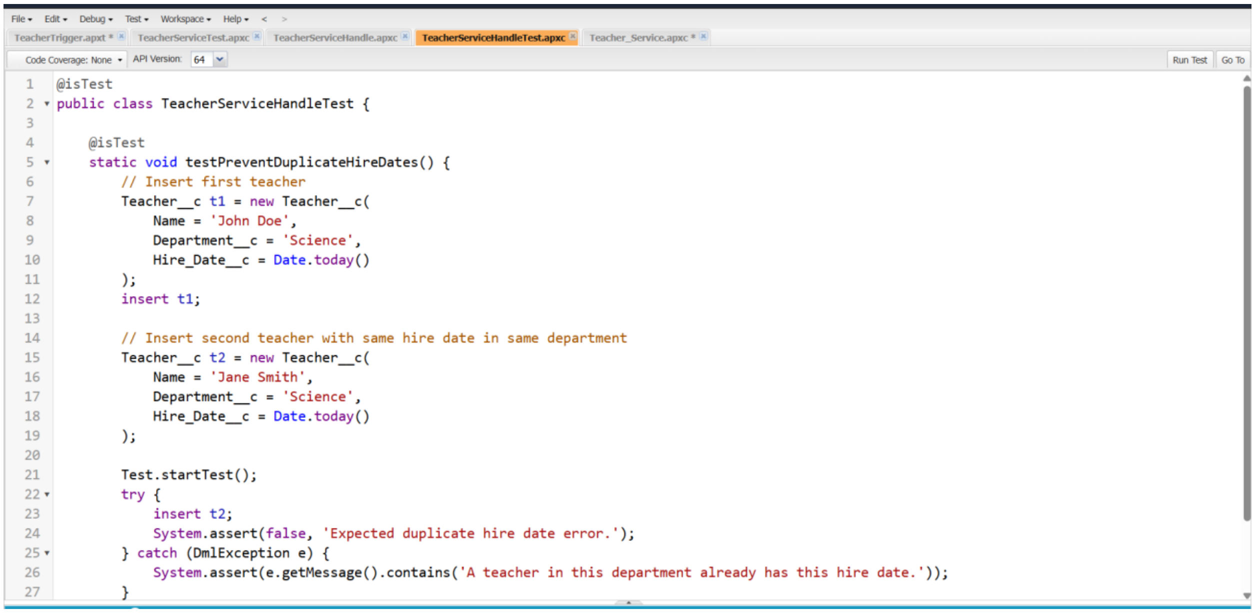


```
public class Teacher_Service {

    // Comparator to sort teachers by Hire Date
    public class TeacherHireDateComparator implements Comparator<Teacher__c> {
        public Integer compare(Teacher__c t1, Teacher__c t2) {
            if (t1.Hire_Date__c == t2.Hire_Date__c) return 0;
            if (t1.Hire_Date__c > t2.Hire_Date__c) return 1;
            return -1;
        }
    }

    // Static method to prevent duplicate hire dates in sa
    public static void preventDuplicateHireDates(
        List<Teacher__c> newTeachers,
        Map<Id, Teacher__c> oldMap
    ){
        // Collect all teachers from same department
        Set<String> deptDateCombo = new Set<String>();

        // Query existing teacher records
```

4. SOQL & Collections

Inside preventOverlap:

- **SOQL** queries existing Teacher for relevant new Teacher.



```
@isTest
public class TeacherServiceHandleTest {

    @isTest
    static void testPreventDuplicateHireDates() {
        // Insert first teacher
        Teacher__c t1 = new Teacher__c(
            Name = 'John Doe',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );
        insert t1;

        // Insert second teacher with same hire date in same department
        Teacher__c t2 = new Teacher__c(
            Name = 'Jane Smith',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        try {
            insert t2;
            System.assert(false, 'Expected duplicate hire date error.');
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('A teacher in this department already has this hire date.'));
        }
```

5. Control Statements

for loops, if conditions and **addError()** to prevent overlaps.

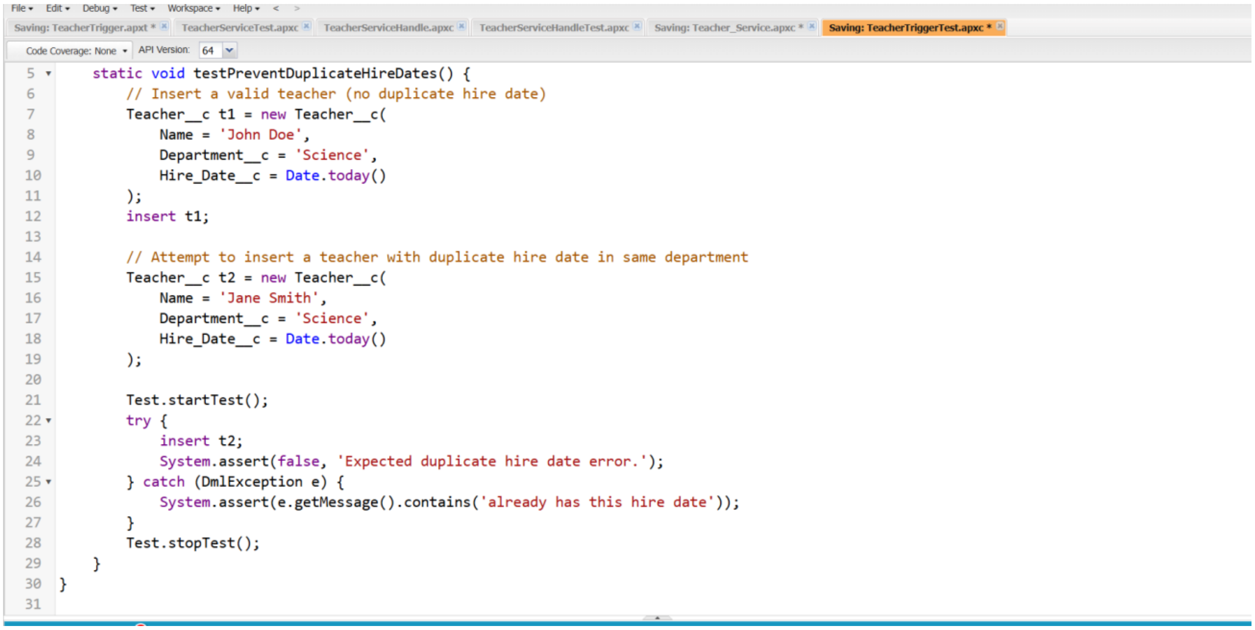
6. Test Classes (Teacher Example)

A test class TeacherTriggerTest was created to:

- Insert a valid teacher (no duplicate hire date in the same department) and verify it saves successfully.
- Attempt to insert a teacher with a duplicate hire date in the same department and verify a **DmlException** with "already has this hire date" in the message is thrown.

This test indirectly executes the logic in

TeacherServiceHandle.preventDuplicateHireDates and gives code coverage for both the trigger **(TeacherTrigger)** and the service class (**TeacherServiceHandle**).

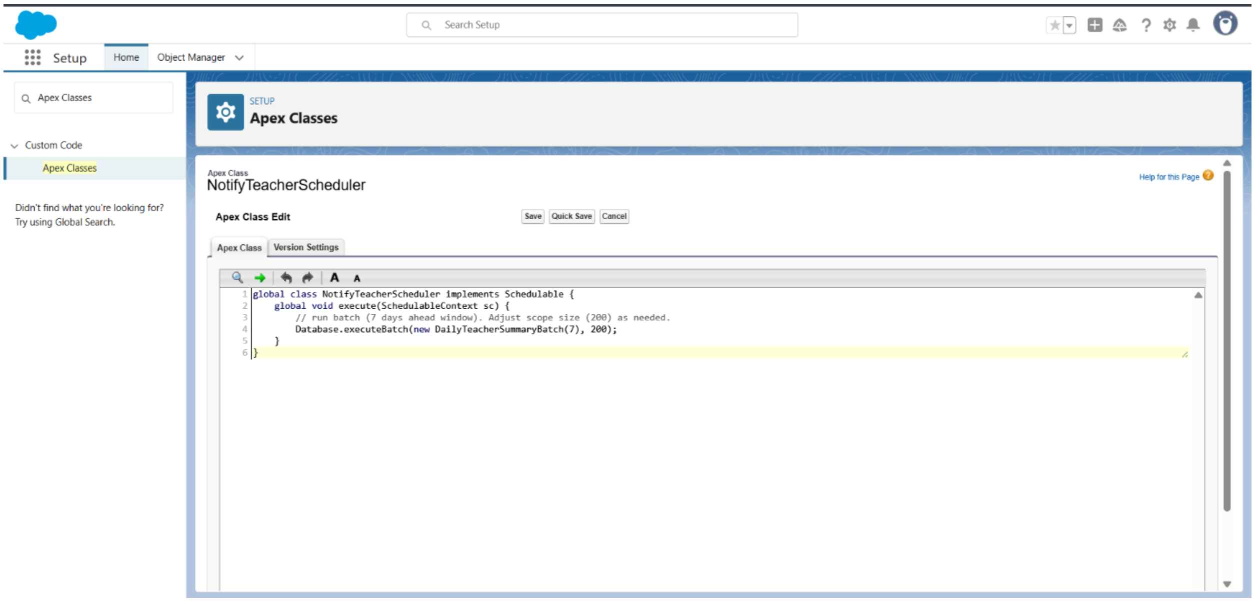

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
Saving: TeacherTrigger.apxt *
TeacherServiceTest.apxc
TeacherServiceHandle.apxc
TeacherServiceHandleTest.apxc
Saving: Teacher_Service.apxc *
Saving: TeacherTriggerTest.apxc *
Code Coverage: None ▾
API Version: 64

```
    static void testPreventDuplicateHireDates() {
        // Insert a valid teacher (no duplicate hire date)
        Teacher__c t1 = new Teacher__c(
            Name = 'John Doe',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );
        insert t1;

        // Attempt to insert a teacher with duplicate hire date in same department
        Teacher__c t2 = new Teacher__c(
            Name = 'Jane Smith',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        try {
            insert t2;
            System.assert(false, 'Expected duplicate hire date error.');
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('already has this hire date'));
        }
        Test.stopTest();
    }
}

```

7. Batch Apex and Scheduled Jobs

A Batch Apex class TeacherSummaryBatch was created to:

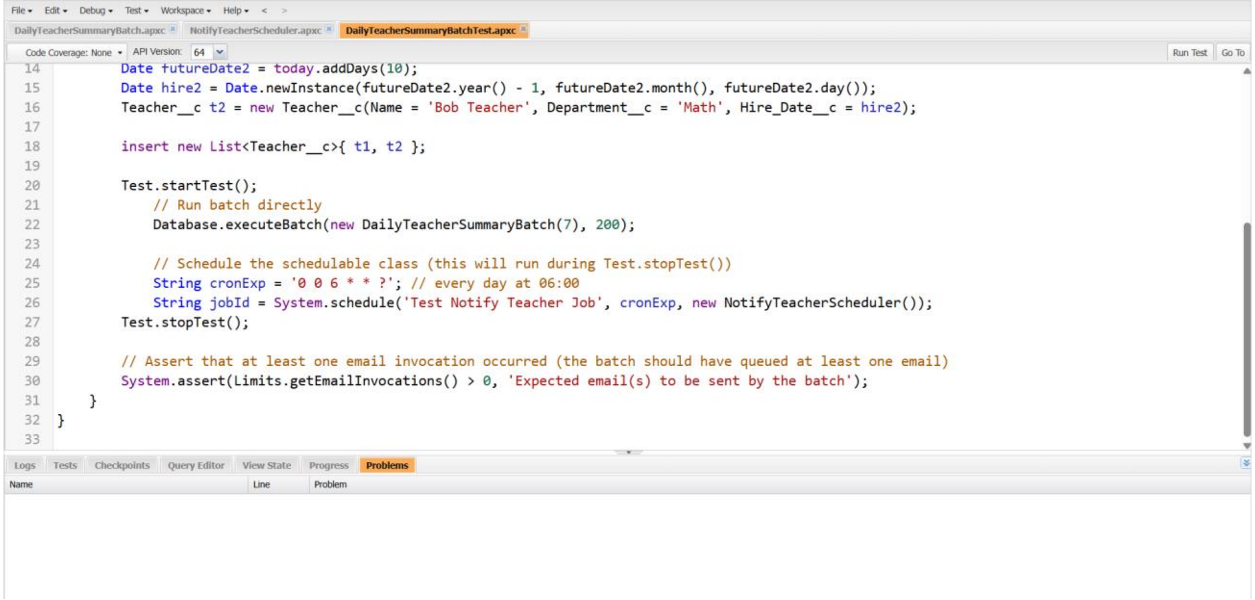
- Query recently hired teachers by department.
- Generate and log summary reports (or prepare reminders for admin review).
- Run daily at 6:00 AM using a Scheduled Apex job.



```
global class NotifyTeacherScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        // run batch (7 days ahead window). Adjust scope size (200) as needed.
        Database.executeBatch(new DailyTeacherSummaryBatch(7), 200);
    }
}
```

8. Exception Handling

Used addError() on SObjects to stop DML with a user-friendly message.



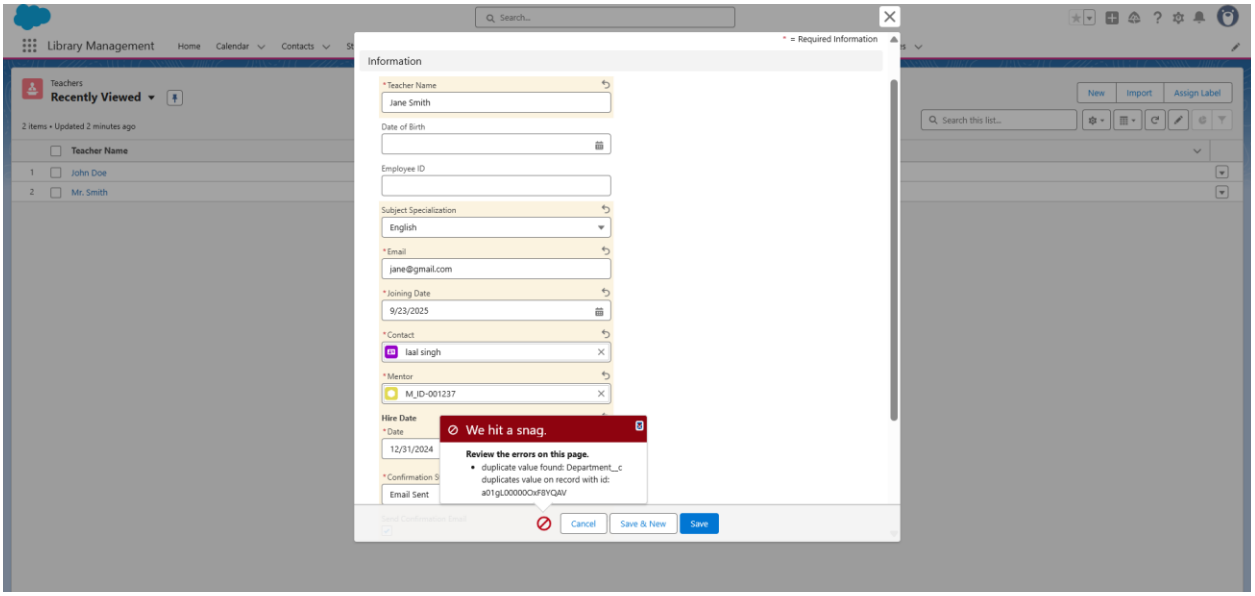
Code Coverage: None API Version: 64

```
        Date futureDate2 = today.addDays(10);
        Date hire2 = Date.newInstance(futureDate2.year() - 1, futureDate2.month(), futureDate2.day());
        Teacher__c t2 = new Teacher__c(Name = 'Bob Teacher', Department__c = 'Math', Hire_Date__c = hire2);

        insert new List<Teacher__c>{ t1, t2 };

        Test.startTest();
            // Run batch directly
            Database.executeBatch(new DailyTeacherSummaryBatch(7), 200);

            // Schedule the schedulable class (this will run during Test.stopTest())
            String cronExp = '0 0 6 * * ?'; // every day at 06:00
            String jobId = System.schedule('Test Notify Teacher Job', cronExp, new NotifyTeacherScheduler());
        Test.stopTest();

        // Assert that at least one email invocation occurred (the batch should have queued at least one email)
        System.assert(Limits.getEmailInvocations() > 0, 'Expected email(s) to be sent by the batch');
    }
}
```

9.Skipped / Not Implemented

Queueable Apex, Future Methods, Asynchronous Processing beyond the daily batch job were not implemented at this phase.

10. Testing & Results



Teachers
Recently Viewed

2 items • Updated 2 minutes ago

	Teacher Name
1	John Doe
2	Mr. Smith

* = Required Information

Information

*Teacher Name: Jane Smith

Date of Birth:

Employee ID:

Subject Specialization: English

*Email: jane@gmail.com

*Joining Date: 9/23/2025

*Contact: laal singh

*Mentor: M_ID-001237

Hire Date

*Date: 12/31/2024

*Confirmation S: Email Sent

Send Confirmation Email ☑

We hit a snag.

Review the errors on this page.

- duplicate value found: Department__c duplicates value on record with id: a01gL00000OxF8YQAV