Phase 5: Apex Programming (Developer)

Goal:

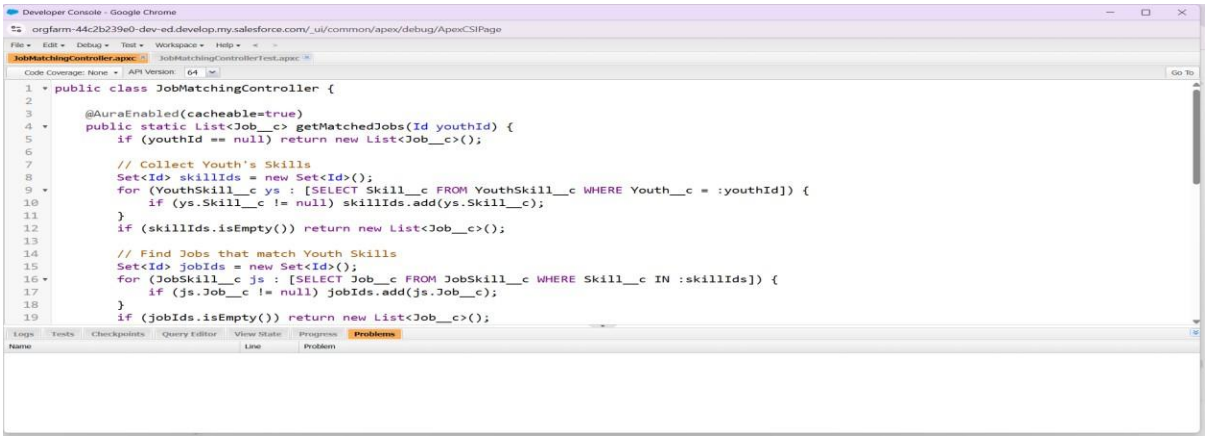
- Match youth to jobs based on skills using Apex.
- Automatically create Interview__c records.
- Handle bulk data safely and meet Salesforce deployment standards.

Step 1: Apex Class — JobMatchingController

Purpose: Perform matching and create interviews programmatically.

Steps to create Apex Class:

1. Click **Setup** (⚙) → **Quick Find** → **Apex Classes** → **New**.
2. Copy and paste the JobMatchingController code below.
3. Click **Save**.

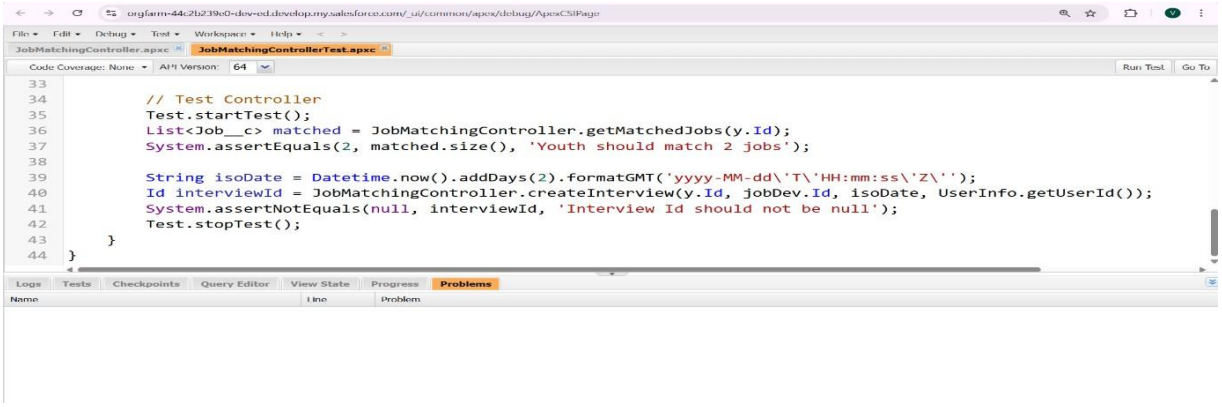


Code Logic Overview:

- **Method 1: getMatchedJobs(youthId)**
  - Fetch youth skills.
  - Find jobs requiring those skills.
  - Return matched jobs (up to 100 records).
- **Method 2: createInterview(youthId, jobId, interviewDateIso, interviewerId)**
  - Converts ISO date string → Datetime.
  - Creates Interview__c record with Status__c = Scheduled.

- Returns the created interview Id.

code works for these tasks. **Small improvement:** In AutoInterviewScheduler, query COUNT() inside a loop — for very large data, consider bulkifying (collect all youth + jobs, then insert interviews in one DML).



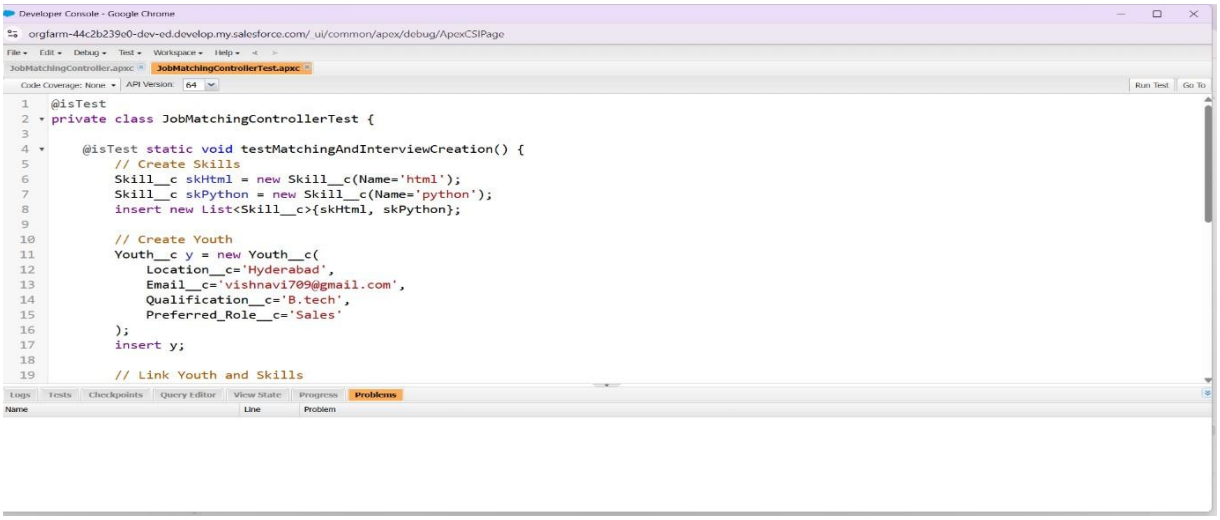
```
        // Test Controller
        Test.startTest();
        List<Job__c> matched = JobMatchingController.getMatchedJobs(y.Id);
        System.assertEquals(2, matched.size(), 'Youth should match 2 jobs');

        String isoDate = Datetime.now().addDays(2).formatGMT('yyyy-MM-dd\'T\'HH:mm:ss\'Z\'');
        Id interviewId = JobMatchingController.createInterview(y.Id, jobDev.Id, isoDate, UserInfo.getUserId());
        System.assertNotEquals(null, interviewId, 'Interview Id should not be null');
        Test.stopTest();
    }
}
```

Step 2: Apex Test Class — JobMatchingControllerTest

Purpose: Ensure at least 75% coverage and test functionality.

Steps to create Test Class:

1. Click **Setup → Apex Classes → New**.
2. Copy and paste the JobMatchingControllerTest code below.
3. Click **Save**.

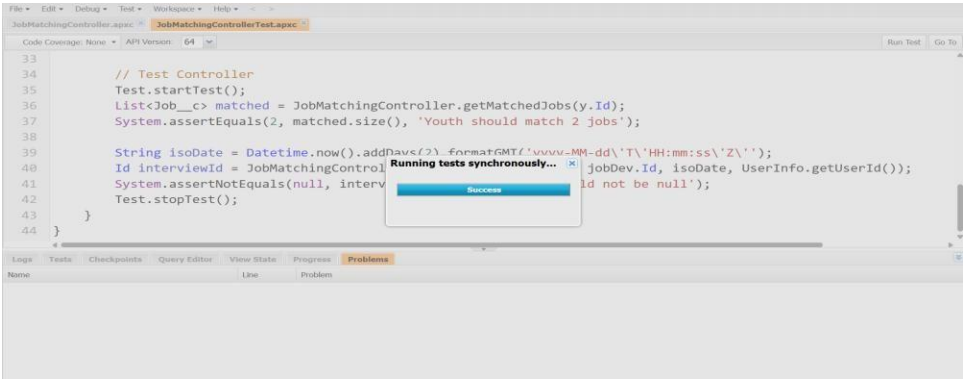


```
@isTest
private class JobMatchingControllerTest {

    @isTest static void testMatchingAndInterviewCreation() {
        // Create Skills
        Skill__c skHtml = new Skill__c(Name='html');
        Skill__c skPython = new Skill__c(Name='python');
        insert new List<Skill__c>{skHtml, skPython};

        // Create Youth
        Youth__c y = new Youth__c(
            Location__c='Hyderabad',
            Email__c='vishnavi709@gmail.com',
            Qualification__c='B.tech',
            Preferred_Role__c='Sales'
        );
        insert y;

        // Link Youth and Skills
```

Test Logic Overview:

- Creates test Skill__c, Youth__c, YouthSkill__c, Job__c, JobSkill__c.
- Tests getMatchedJobs() → asserts correct number of jobs.

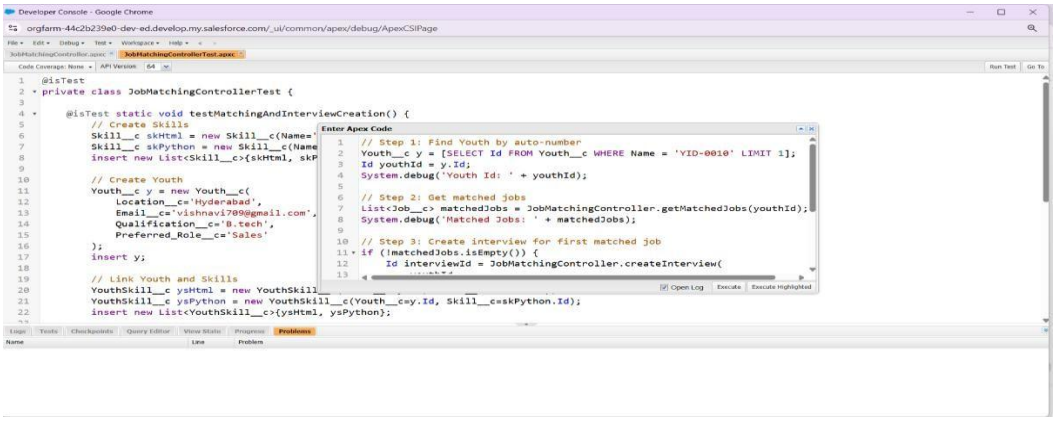
- Tests createInterview() → asserts interview Id is not null.



Step 3: Execute Apex for Individual Youth

Steps:

1. Open **Developer Console → Execute Anonymous Window**.
2. Run this snippet (replace YID-0010 with actual youth auto-number):



This works — it will create an interview for the first matched job.

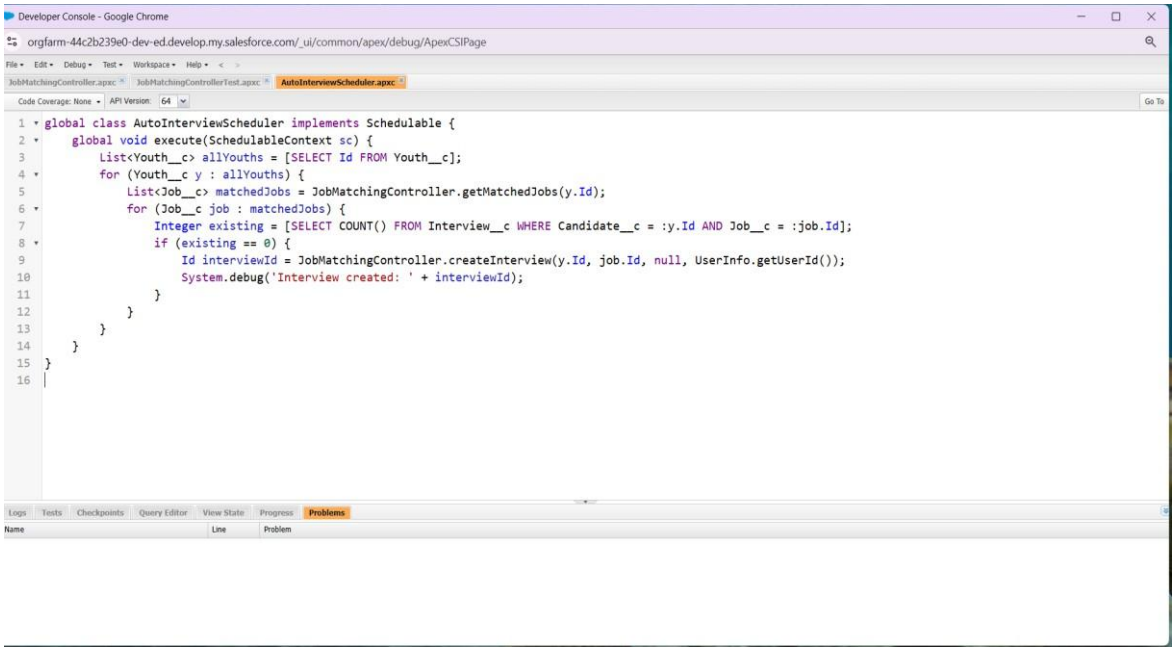
Step 4: Scheduler — AutoInterviewScheduler

Purpose: Automatically create interviews for all youth.

Steps to schedule:

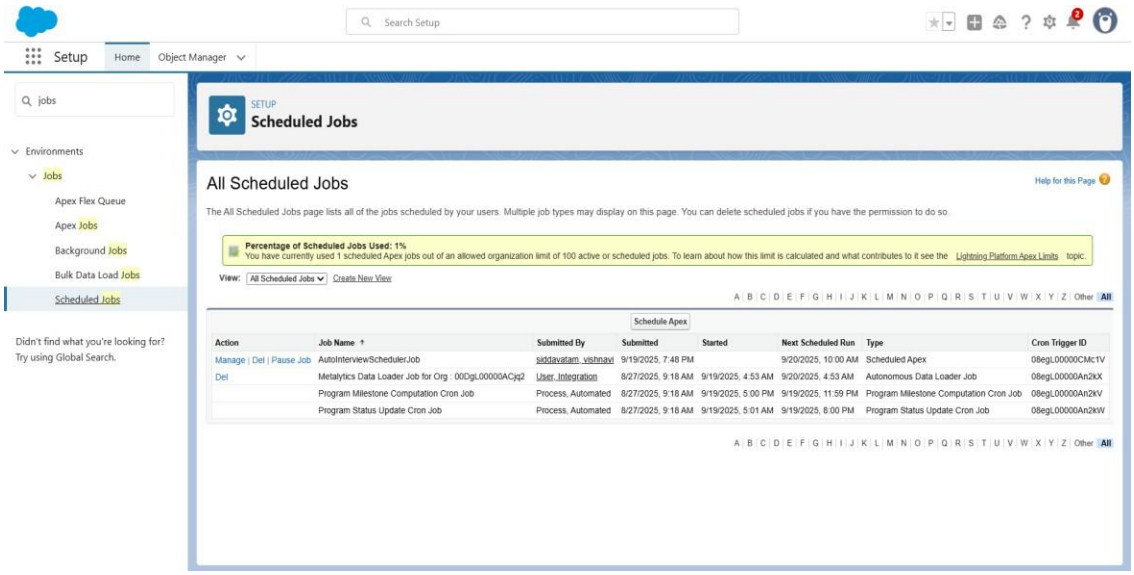
1. Click **Setup → Apex Classes → Schedule Apex → New**.
2. Choose **AutoInterviewScheduler** class.
3. Set frequency (daily/hourly) and start/end dates.

4. Click **Save** to activate scheduler.



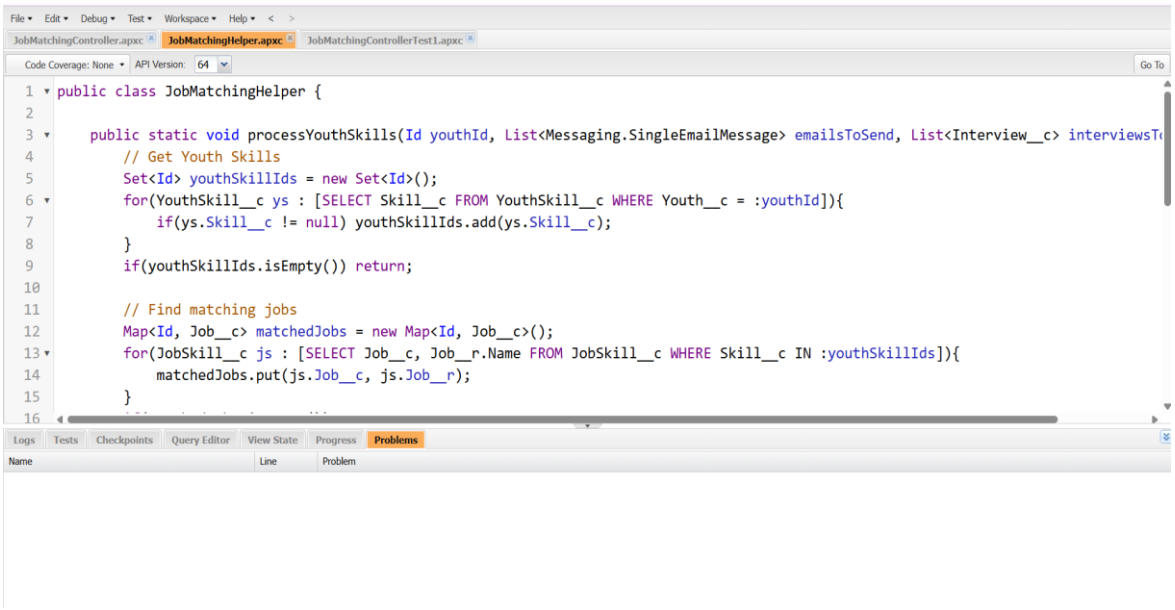
Logic:

- Loops through all youth → gets matched jobs → creates interview if none exists.
- System.debug() logs each creation.



Step 5: Apex Helper Class — JobMatchingHelper
Purpose Handle automatic interview creation and email sending when youth skills match jobs.
Steps to create Helper Class:

1. Click **Setup → Apex Classes → New**.
2. Copy and paste the **JobMatchingHelper** Code below.
3. Click **Save**.


File Edit Debug Test Workspace Help
JobMatchingController.apxc
JobMatchingHelper.apxc
JobMatchingControllerTest1.apxc
Code Coverage: None
API Version: 64
Go To
public class JobMatchingHelper {
public static void processYouthSkills(Id youthId, List<Messaging.SingleEmailMessage> emailsToSend, List<Interview__c> interviewsT
// Get Youth Skills
Set<Id> youthSkillIds = new Set<Id>();
for(YouthSkill__c ys : [SELECT Skill__c FROM YouthSkill__c WHERE Youth__c = :youthId]){
if(ys.Skill__c != null) youthSkillIds.add(ys.Skill__c);
}
if(youthSkillIds.isEmpty()) return;
// Find matching jobs
Map<Id, Job__c> matchedJobs = new Map<Id, Job__c>();
for(JobSkill__c js : [SELECT Job__c, Job__r.Name FROM JobSkill__c WHERE Skill__c IN :youthSkillIds]){
matchedJobs.put(js.Job__c, js.Job__r);
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem