

Phase 5: Apex Programming (Developer)

Goal:

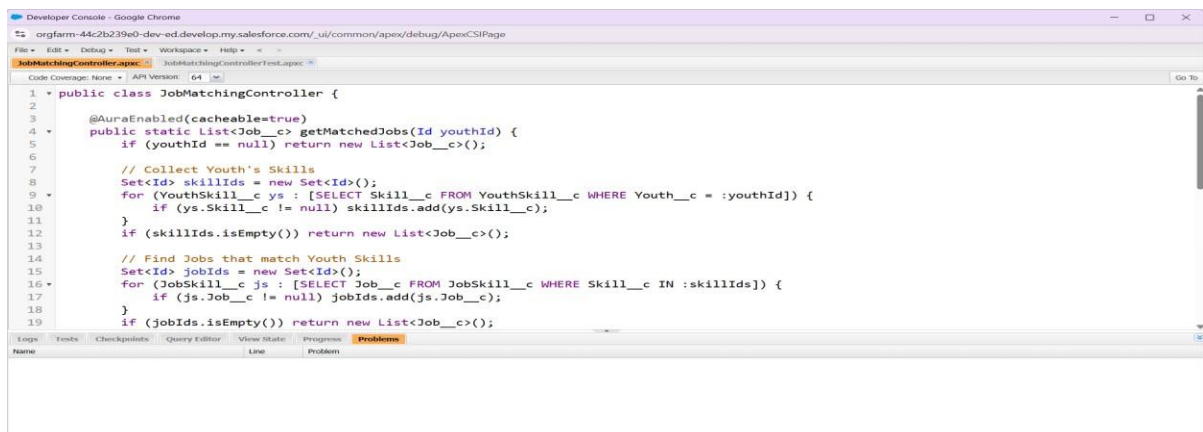
- Match youth to jobs based on skills using Apex.
- Automatically create Interview__c records.
- Handle bulk data safely and meet Salesforce deployment standards.

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**.



```
1 public class JobMatchingController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Job__c> getMatchedJobs(Id youthId) {
5         if (youthId == null) return new List<Job__c>();
6
7         // Collect Youth's Skills
8         Set<Id> skillIds = new Set<Id>();
9         for (YouthSkill__c ys : [SELECT Skill__c FROM YouthSkill__c WHERE Youth__c = :youthId]) {
10             if (ys.Skill__c != null) skillIds.add(ys.Skill__c);
11         }
12         if (skillIds.isEmpty()) return new List<Job__c>();
13
14         // Find Jobs that match Youth Skills
15         Set<Id> jobIds = new Set<Id>();
16         for (JobSkill__c js : [SELECT Job__c FROM JobSkill__c WHERE Skill__c IN :skillIds]) {
17             if (js.Job__c != null) jobIds.add(js.Job__c);
18         }
19         if (jobIds.isEmpty()) return new List<Job__c>();
20     }
21 }
```

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).



```

33
34 // Test Controller
35 Test.startTest();
36 List<Job__c> matched = JobMatchingController.getMatchedJobs(y.Id);
37 System.assertEquals(2, matched.size(), 'Youth should match 2 jobs');
38
39 String isoDate = Datetime.now().addDays(2).formatGMT('yyyy-MM-dd\'T\'HH:mm:ss\'Z\'');
40 Id interviewId = JobMatchingController.createInterview(y.Id, jobDev.Id, isoDate, UserInfo.getUserId());
41 System.assertNotEquals(null, interviewId, 'Interview Id should not be null');
42 Test.stopTest();
43
44 }

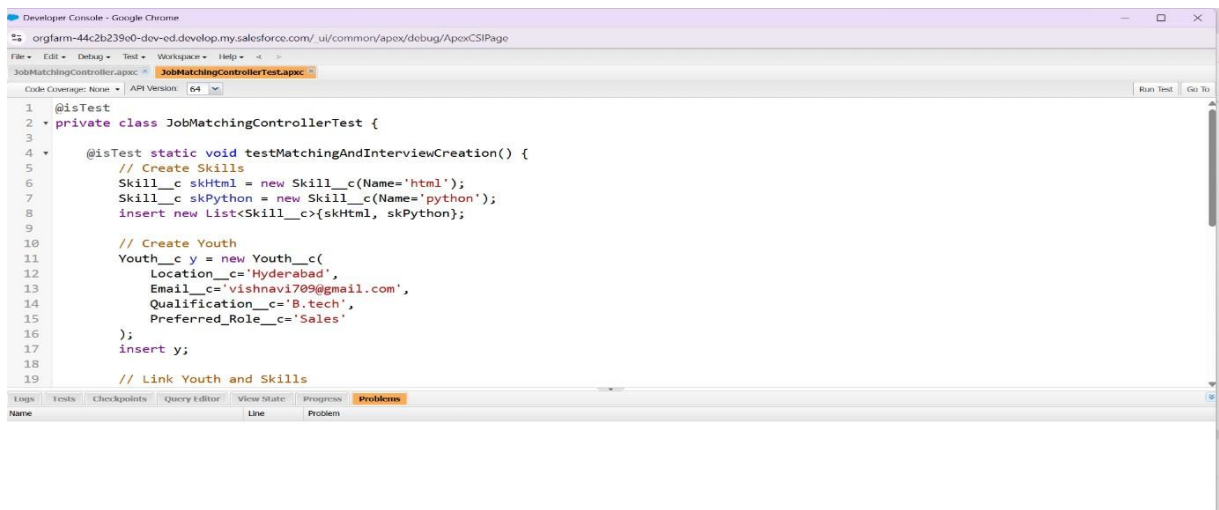
```

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**.



```

1 @isTest
2 private class JobMatchingControllerTest {
3
4     @isTest static void testMatchingAndInterviewCreation() {
5         // Create Skills
6         Skill__c skHtml = new Skill__c(Name='html');
7         Skill__c skPython = new Skill__c(Name='python');
8         insert new List<Skill__c>{skHtml, skPython};
9
10        // Create Youth
11        Youth__c y = new Youth__c(
12            Location__c='Hyderabad',
13            Email__c='vishnavi709@gmail.com',
14            Qualification__c='B.tech',
15            Preferred_Role__c='Sales'
16        );
17        insert y;
18
19        // 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.

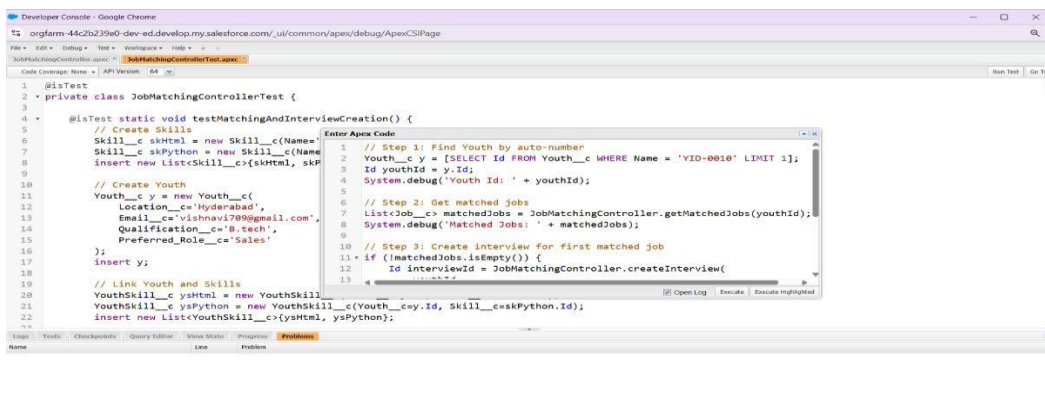
- Tests createInterview() → asserts interview Id is not null.



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.

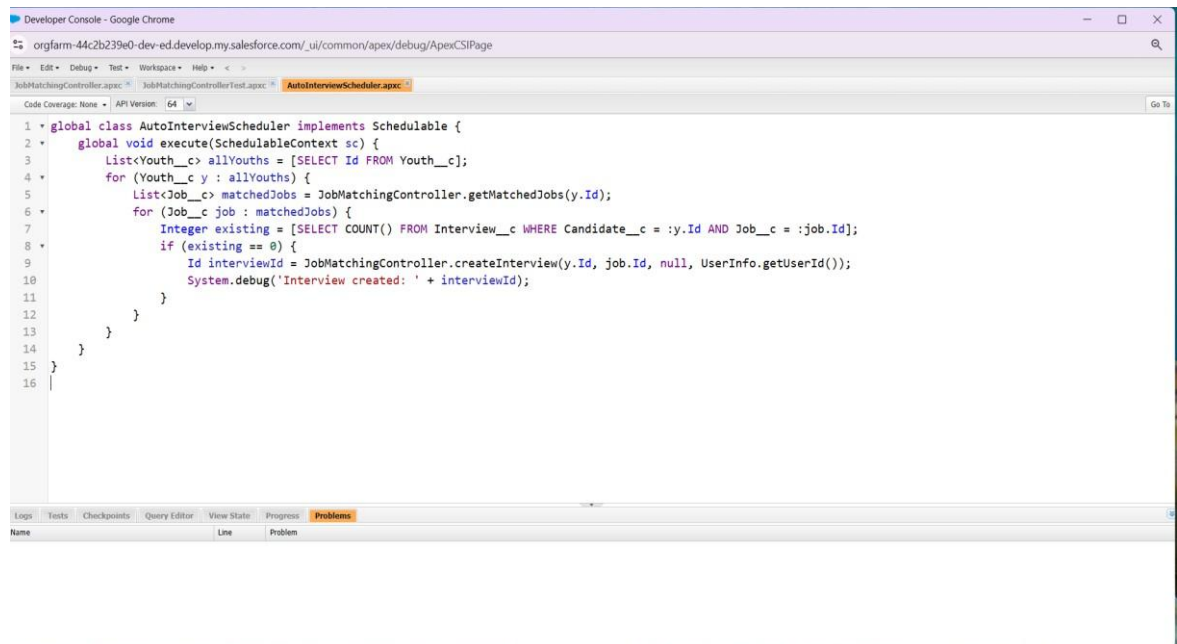
Step 4: Scheduler — AutoInterviewScheduler

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.

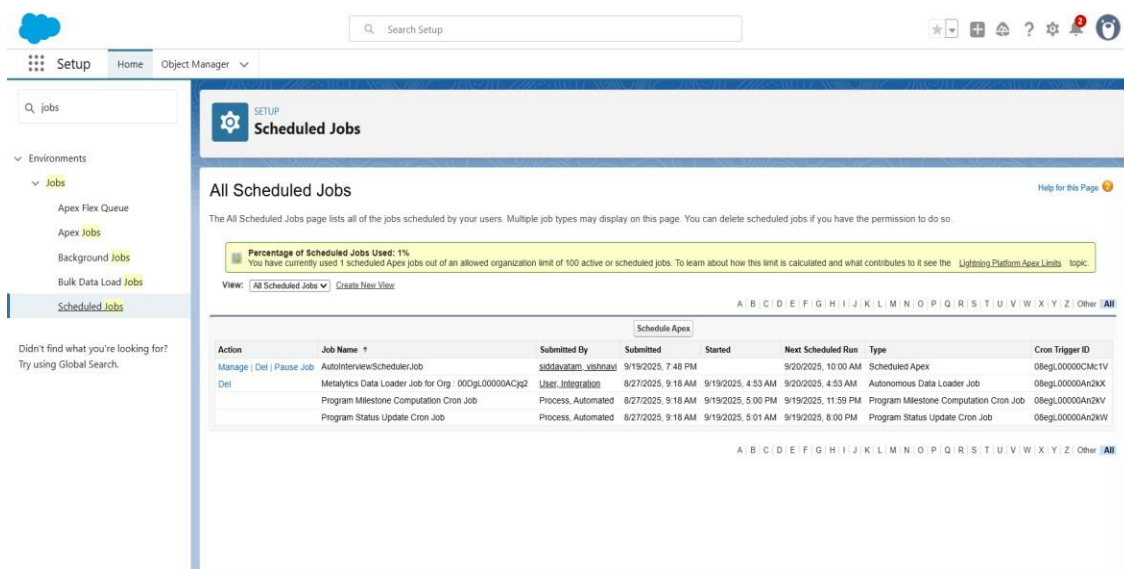
4. Click **Save** to activate scheduler.



```
1 global class AutoInterviewScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         List<Youth__c> allYouths = [SELECT Id FROM Youth__c];
4         for (Youth__c y : allYouths) {
5             List<Job__c> matchedJobs = JobMatchingController.getMatchedJobs(y.Id);
6             for (Job__c job : matchedJobs) {
7                 Integer existing = [SELECT COUNT() FROM Interview__c WHERE Candidate__c = :y.Id AND Job__c = :job.Id];
8                 if (existing == 0) {
9                     Id interviewId = JobMatchingController.createInterview(y.Id, job.Id, null, UserInfo.getUserId());
10                    System.debug('Interview created: ' + interviewId);
11                }
12            }
13        }
14    }
15 }
16 }
```

Logic:

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



The screenshot shows the Salesforce Setup page with the 'Scheduled Jobs' section selected. The page displays a table of all scheduled jobs, including their names, submission dates, and scheduled run times. A yellow banner indicates that 1% of the scheduled jobs limit has been used.

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	AutoInterviewScheduler Job	addavaram_vishnup	9/19/2025, 7:46 PM		9/20/2025, 10:00 AM	Scheduled Apex	08eg1.000000Cm1V
Del	Metastatys Data Loader Job for Org: 000G1.000000ACjg2	User_Integration	8/27/2025, 9:16 AM	9/19/2025, 4:53 AM	9/20/2025, 4:53 AM	Autonomous Data Loader Job	08eg1.000000Aa2XK
	Program Milestone Computation Cron Job	Process, Automated	8/27/2025, 9:16 AM	9/19/2025, 5:00 PM	9/19/2025, 11:59 PM	Program Milestone Computation Cron Job	08eg1.000000An2KV
	Program Status Update Cron Job	Process, Automated	8/27/2025, 9:16 AM	9/19/2025, 5:01 AM	9/19/2025, 8:00 PM	Program Status Update Cron Job	08eg1.000000An2KW

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

```
1 public class JobMatchingHelper {
2
3     public static void processYouthSkills(Id youthId, List<Messaging.SingleEmailMessage> emailsToSend, List<Interview__c> interviewsTo
4         // Get Youth Skills
5         Set<Id> youthSkillIds = new Set<Id>();
6         for(YouthSkill__c ys : [SELECT Skill__c FROM YouthSkill__c WHERE Youth__c = :youthId]){
7             if(ys.Skill__c != null) youthSkillIds.add(ys.Skill__c);
8         }
9         if(youthSkillIds.isEmpty()) return;
10
11         // Find matching jobs
12         Map<Id, Job__c> matchedJobs = new Map<Id, Job__c>();
13         for(JobSkill__c js : [SELECT Job__c, Job__r.Name FROM JobSkill__c WHERE Skill__c IN :youthSkillIds]){
14             matchedJobs.put(js.Job__c, js.Job__r);
15         }
16     }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------