

Phase 5: Apex Programming (Developer)

Goal:

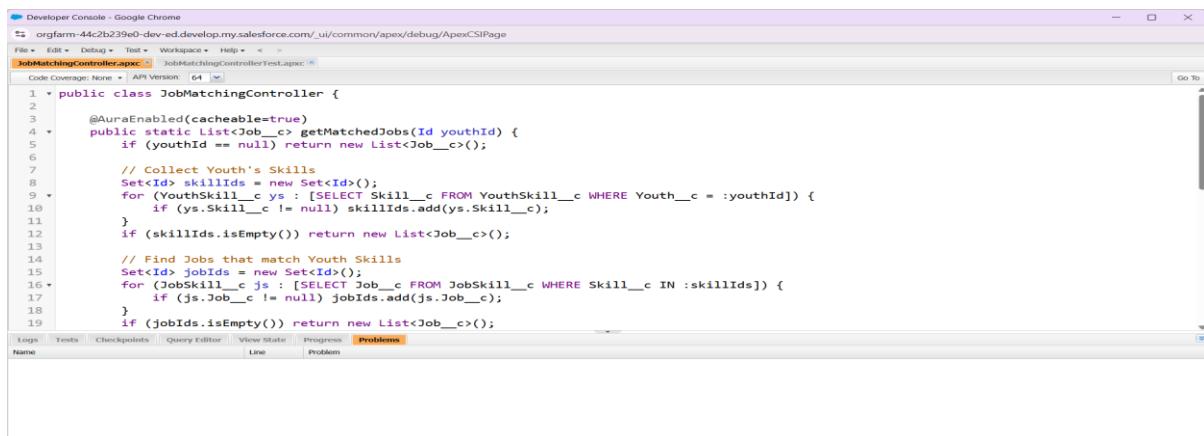
- Match youth to jobs based on skills using Apex.
- Automatically create Interview__c records.
- Handle bulk data safely and meet Salesforce deployment standards.

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**.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-44c2b239e0-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is JobMatchingController.apex. The code editor contains the following Apex class:

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

The bottom navigation bar shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected.

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.assertEquals(null, interviewId, 'Interview Id should not be null');
42
43     Test.stopTest();
44 }

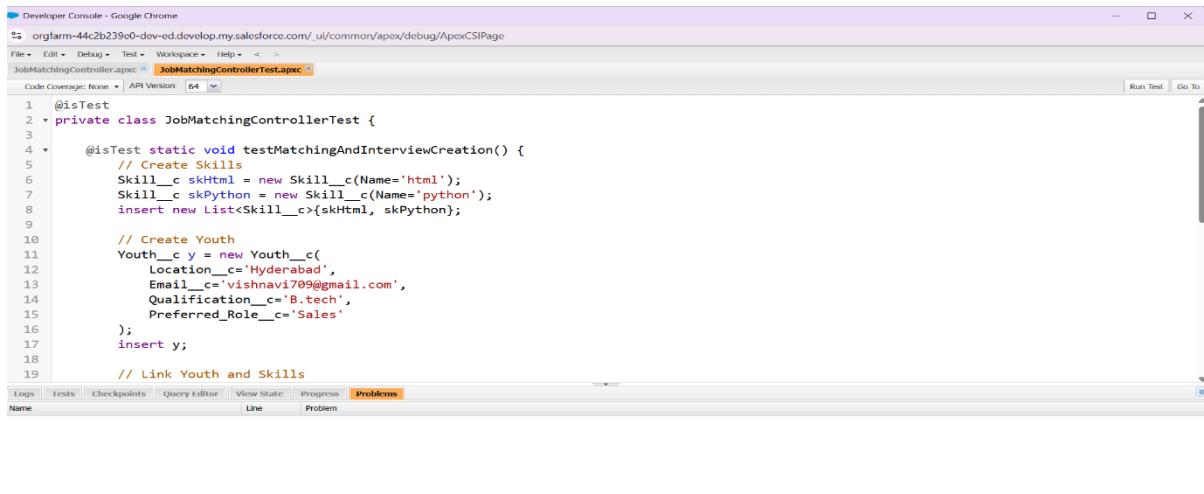
```

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.

The screenshot shows the Salesforce IDE interface with the file `JobMatchingControllerTest.apc` open. The code is a test class for the `JobMatchingController`. It includes a test method that creates a youth, adds skills, finds matched jobs, and creates an interview. A tooltip from the IDE indicates that the test is running synchronously. The status bar at the bottom shows "Success".

```

File • Edit • Debug • Test • Workspace • Help • 
JobMatchingControllerTest.apc • JobMatchingControllerTest.apc 
Code Coverage: None • API Version: 64 
33 // Test Controller
34 Test.startTest();
35 List<Job__c> matched = JobMatchingController.getMatchedJobs(y.Id);
36 System.assertEquals(2, matched.size(), 'Youth should match 2 jobs');
37
38 String isoDate = Datetime.now().addHours(2).formatGMT('yyyy-MM-dd\T\HH:mm:ss\Z');
39 Id interviewId = JobMatchingController.createInterview(jobDev.Id, isoDate, UserInfo.getUserId());
40 System.assertEquals(null, interviewId, 'Id not be null');
41 Test.stopTest();
42
43 }
44 
```

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):

The screenshot shows the Salesforce Developer Console with an anonymous window open. The code in the window is a test method for the `JobMatchingController`. It performs three steps: finding a youth by auto-number, getting matched jobs, and creating an interview for the first matched job. A tooltip highlights the line `System.debug('Matched Jobs: ' + matchedJobs);` with the message "Execute highlighted".

```

Developer Console - Google Chrome
File • Edit • Debug • Test • Workspace • Help • 
JobMatchingController.apc • JobMatchingControllerTest.apc 
Code Coverage: None • API Version: 64 
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='vishnav1709@gmail.com',
14            Qualification__c='B.tech',
15            Preferred_Role__c='Sales');
16        insert y;
17
18        // Link Youth and Skills
19        YouthSkill__c ysHTML = new YouthSkill__c({y.Id, Skill__c=skHTML});
20        YouthSkill__c ysPython = new YouthSkill__c({y.Id, Skill__c=skPython.Id});
21        insert new List<YouthSkill__c>({ysHTML, ysPython});
22    }
23 
```

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

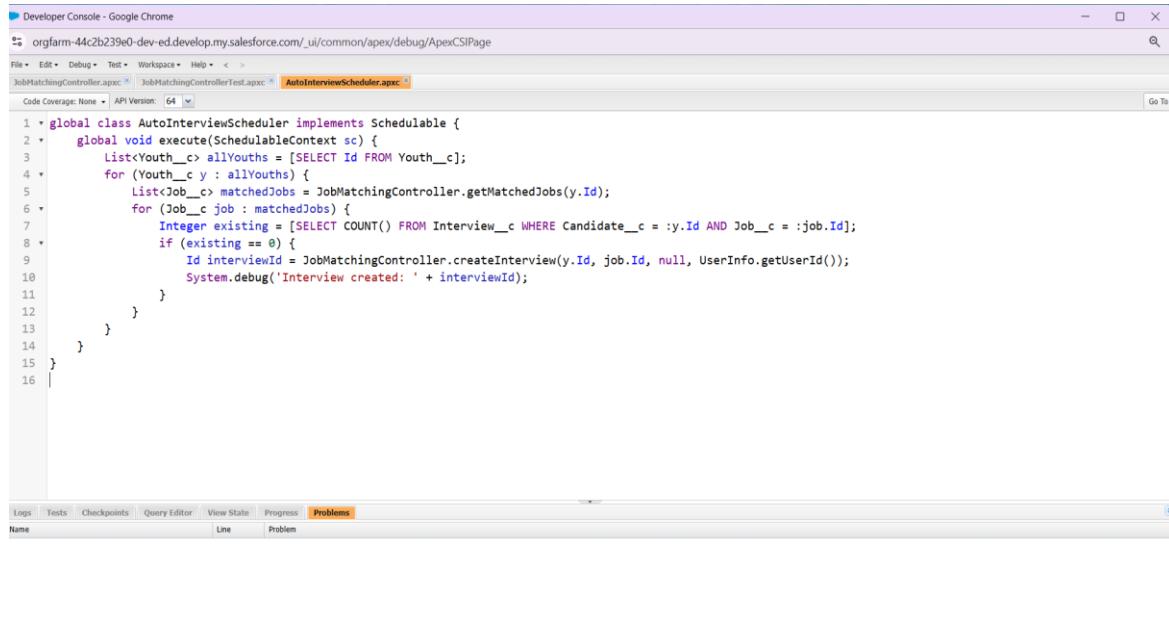
Step 4: Scheduler — AutoInterviewScheduler

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.



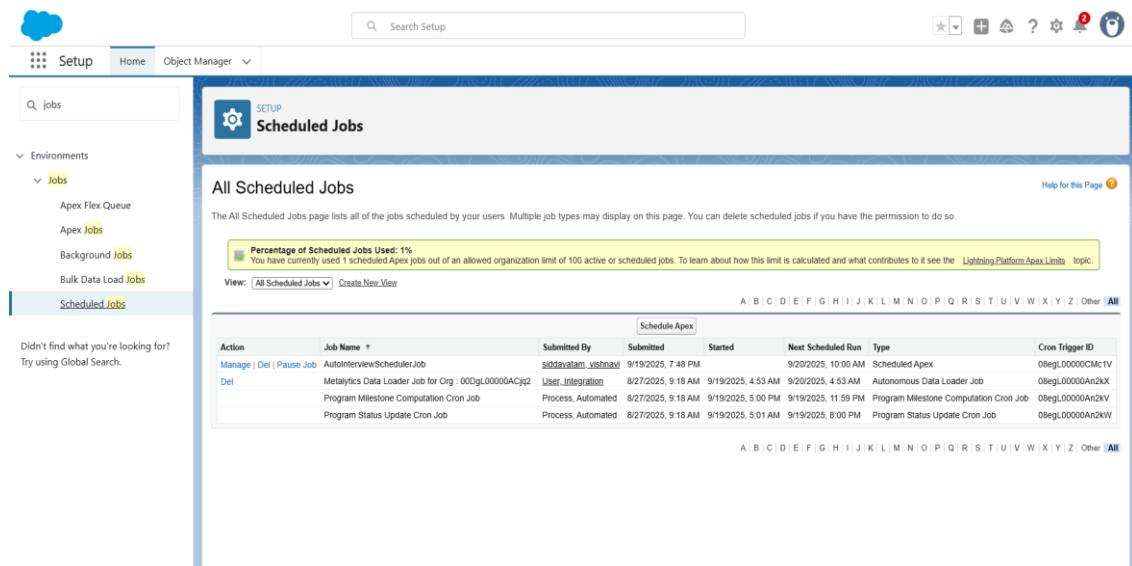
```

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.



Scheduled Jobs

All Scheduled Jobs

Percentage of Scheduled Jobs Used: 1%

Help for this Page

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	AutoInterviewScheduler	saddavatam_vishnau	9/19/2025, 7:48 PM		9/20/2025, 10:00 AM	Scheduled Apex	0begI,00000CMc1V
Del	Metalytics Data Loader Job for Org : 00Dg1.00000A0Cp2	User, Integration	9/27/2025, 9:18 AM	9/19/2025, 4:53 AM	9/20/2025, 4:53 AM	Autonomous Data Loader Job	0begI,00000An2kX
	Program Milestone Computation Cron Job	Process, Automated	8/27/2025, 9:18 AM	9/19/2025, 5:00 PM	9/19/2025, 11:59 PM	Program Milestone Computation Cron Job	0begI,00000An2kV
	Program Status Update Cron Job	Process, Automated	8/27/2025, 9:18 AM	9/19/2025, 5:01 AM	9/19/2025, 8:00 PM	Program Status Update Cron Job	0begI,00000An2kW