

Project Title: SkillMatch Connect-Smart Skill-Based Student Job Recommendation System

Project Overview:

SkillMatch Connect is a Salesforce-based intelligent job recommendation system that connects students, educational institutions, and employers through skill-focused matchmaking. The platform enables students to register their profiles, showcase their skills, and receive personalized job or internship recommendations based on real-time analysis. Employers can post openings with specific skill requirements, while the system automatically maps qualified candidates using Salesforce automation and AI-powered logic. It manages the full recruitment lifecycle—from application to selection—while offering transparency, detailed analytics, and scalable integration for institutions and organizations.

Objectives

- Enable students to create profiles and register their skills, qualifications, and career interests.
- Allow companies and institutions to post job and internship listings with skill criteria.
- Automate skill-based job matching using Salesforce CRM data models and automation tools.
- Provide real-time notifications and application status updates to students via email.
- Generate analytical dashboards and reports for administrators and educational partners.
- Ensure a secure, scalable, and easily maintainable CRM environment to support continuous student-employer engagement.

Phase 1: Problem Understanding & Industry Analysis

Requirement Gathering

- Conducted sessions with NGOs, Youth Coordinators, and Employers.
- Core requirements:
 1. Youth registration with basic details and skills.
 2. Employer job postings.
 3. Auto-matching of youth ↔ jobs based on skills.
 4. Automatic interview creation and tracking.

5. Email notifications at each step.
6. Reports and dashboards for monitoring.

Scope: Digital bridge between youth, Admins and employers.

Stakeholder Analysis

- **Primary:** Youth, Employers, NGOs/Placement Coordinators.
- **Secondary:** System Administrators, Salesforce Platform.

Business Process Mapping

Current Process (Manual): Manual resume submission → NGOs connect youth → Communication gaps → No centralized reporting.

Proposed Process (Automated via Salesforce): Youth registration → Employer job postings → Auto-matching → Interviews scheduled → Emails sent → Dashboards.

Industry-Specific Use Case Analysis

- Existing job portals are generic and not NGO-specific.
- CareerBridge centralizes youth/job data, provides skill-based matching, and automated interview lifecycle.

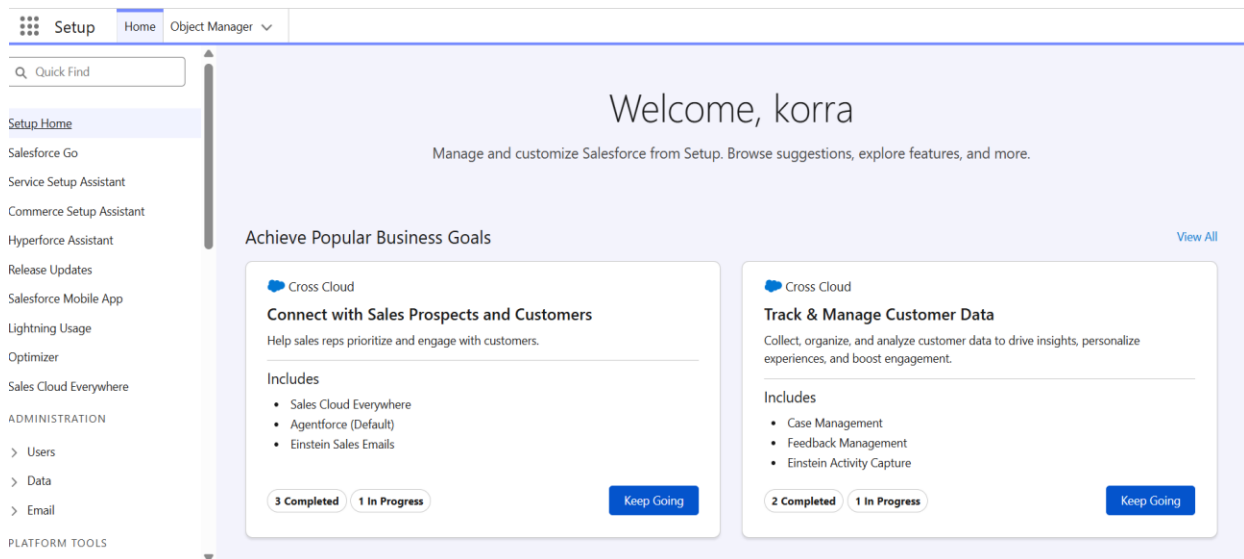
AppExchange Exploration

- Explored recruitment apps; none provide NGO-focused youth employability features.
- Decision: Build custom solution using Salesforce objects (Youth, Job, Skill, Interview), flows, and custom emails.

Phase 2: Org Setup & Configuration

Step 1: Open Setup

1. Login to **Salesforce Lightning**.
2. Click the **Gear (⚙️) icon** in the top-right → select **Setup**.




Step 2: Update Company Information

1. In Setup, use **Quick Find** → type **Company Information** → open it.
2. Click **Edit**.
3. Update:
 - **Organization Name:** KSRM College of Engineering
 - **Default Time Zone:** (09:00 – 18:00)
4. Click **Save**

Step 4: Add Holidays

1. Quick Find → **Holidays** → click **New**.
2. Add important dates, for example:
 - Independence Day
 - Republic Day
3. Click **Save**.

Why: Salesforce respects holidays for time-dependent automation (tasks, emails).

| <div><div> <small>SETUP</small></div><div>Object Manager</div><div><small>2 Items. Sorted by Label</small></div></div> <div><input type="text" value="Q job"/> Schema Builder Create ▼</div> | | | | | | |
|---|---|------------|---------------|-------------|---------------|----------|
| LABEL | ▲ | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED |
| Job | | Job_c | Custom Object | | 11/3/2025 | ✓ |
| JobSkill | | JobSkill_c | Custom Object | | 11/3/2025 | ✓ |


Step 5: Create Test Users

You need at least **3 users**: Youth Coordinator, Employer, NGO Admin.

1. Quick Find → **Users** → click **New User**.
2. Fill example details

| Name | Username | Profile | Role |
|-------------------|---|----------------------|-------------------|
| Youth Coordinator | your.email+yc@sandbox.my.salesforce.com | System Administrator | Youth Coordinator |
| Employer | employer@sandbox.my.salesforce.com | Standard User | Employer |
| NGO Admin | ngo@sandbox.my.salesforce.com | Standard User | NGO Manager |

3. Click **Save** after each user.

 **SETUP**
Users

All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. [Let's Go](#)

View: All Users Edit Create New User

New User Reset Password(s) Add Multiple Users

| <input type="checkbox"/> Action | Full Name ↑ | Alias | Username | Role | Active | Profile |
|---|-------------------|-------|---|---------------------|--------|----------------------------------|
| <input type="checkbox"/> Edit | admin_Placement | padmi | placement@admin.com | SVP_Human Resources | ✓ | System Administrator |
| <input type="checkbox"/> Edit | admin_Youth | yadmi | youth.admin@my.com | | ✓ | System Administrator |
| <input type="checkbox"/> Edit | Chatter_Expert | | chatty.00dg50000012imkeaa.dvus5joe1joo@chatter.salesforce.com | | ✓ | Chatter Free User |
| <input type="checkbox"/> Edit | Coordinator_Youth | ycoor | youth@coordinator.com | CEO | ✓ | Standard Platform User |
| <input type="checkbox"/> Edit | EPIC_OrgFarm | OEPIC | epic.ftb3b92bd750@orgfarm.salesforce.com | | ✓ | System Administrator |
| <input type="checkbox"/> Edit | sunil.kumar.korra | kor | korrasunilkumar04509@agentforce.com | | ✓ | System Administrator |
| <input type="checkbox"/> Edit | User_Integration | integ | integration@00dg50000012imkeaa.com | | ✓ | Analytics Cloud Integration User |
| <input type="checkbox"/> Edit | User_Security | sec | insightssecurity@00dg50000012imkeaa.com | | ✓ | Analytics Cloud Security User |

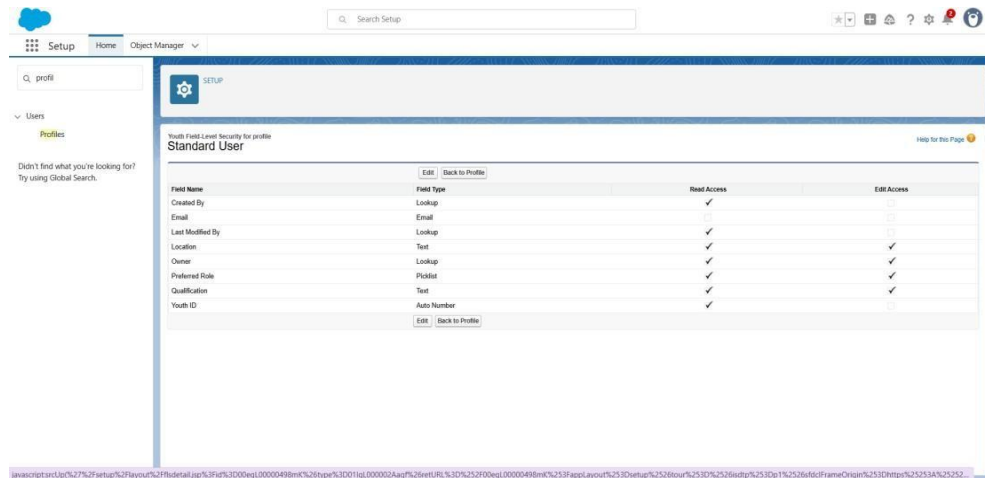
New User Reset Password(s) Add Multiple Users

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Other [All](#)

Step 6: Configure Profiles (Object Permissions)

1. Quick Find → **Profiles** → open a profile (e.g., *Standard User*).
2. Scroll to **Object Settings** → select object Youth__c → click **Edit**.
3. Grant permissions:
 - **Read, Create, Edit** (as required)

4. Click **Save**



Step 7: Create Role Hierarchy

1. Quick Find → **Roles** → click **Set Up Roles** → **Add Role**.

2. Example hierarchy:

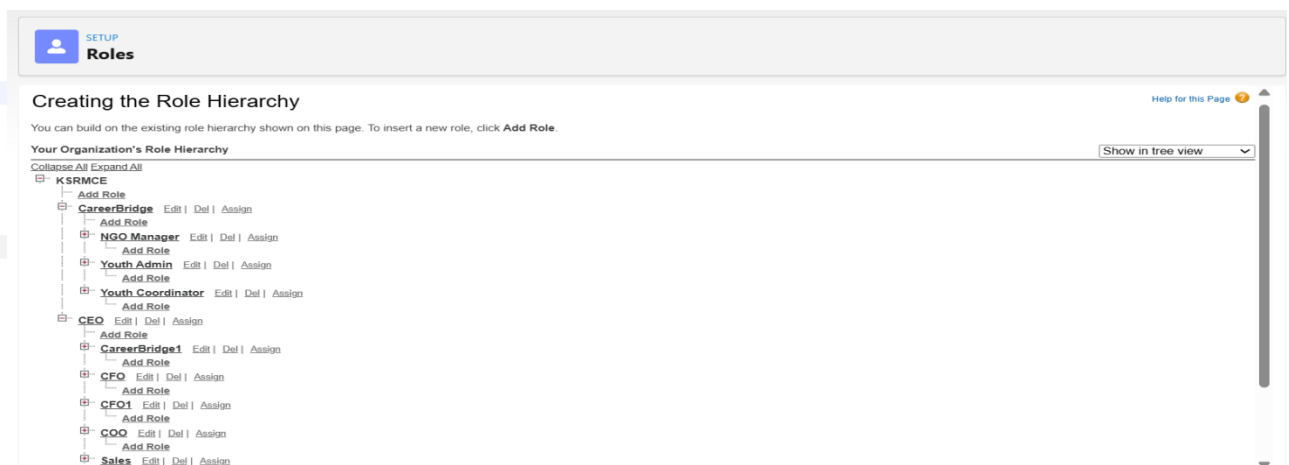
CareerBridge

├── NGO Manager

├── Youth Coordinator

└── Recruiter

3. Click **Save**.



Step 8: Permission Sets

1. Quick Find → **Permission Sets** → click **New**.
2. Label: Interview Scheduler PS → **Save**.
3. Click **Manage Assignments** → assign to users who need extra permissions.

Why: Grants additional permissions without changing profiles.

Step 9: Set Org-Wide Defaults (OWD) & Sharing

1. Quick Find → **Sharing Settings** → click **Edit**.
2. Set defaults:

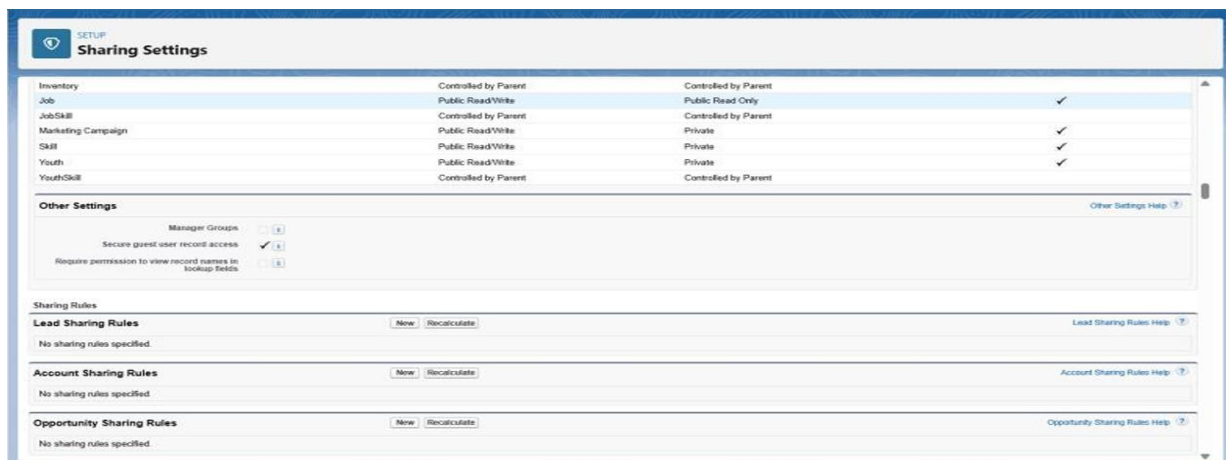
| Object | Default Access |
|--------|----------------|
|--------|----------------|

| | |
|----------|---------|
| Youth__c | Private |
|----------|---------|

| | |
|--------|------------------|
| Job__c | Public Read Only |
|--------|------------------|

| | |
|--------------|---------|
| Interview__c | Private |
|--------------|---------|

1. Create a **Sharing Rule**:
 - Scroll to **Youth__c** → click **New**.
 - Name: Share Youth to Recruiters
 - Rule Type: Owner-based
 - Shared To: Role → Recruiter
 - Click **Save**.



Step 10: Enable Admin Login Access

1. Quick Find → **Users** → select a user → click the down-arrow → **Login**.
2. If option missing: Setup → **Login Access Policies** → enable.

Why: Admins can log in as other users for testing.

Phase 3: Data Modeling & Relationships

This phase covers creating custom objects, fields, relationships, layouts, and adding sample data.

Step 1: Create Custom Objects

Navigation:

Setup → Object Manager → Create → Custom Object

Objects to create:

| Object Name | API Name | Purpose |
|-------------|----------|--------------------------------|
| Youth | Youth_c | Tracks youth registration info |
| Job | Job_c | Tracks job postings |

Skill Skill_c Stores skill names like Java, Python

Interview Interview_c Stores interview details

YouthSkill YouthSkill_c Links Youth → Skill (Master-Detail on Youth, Lookup to Skill)

JobSkill JobSkill_c Links Job → Skill (Master-Detail on Job, Lookup to Skill)

Setup Home Object Manager ▾

Object Manager
2 Items. Sorted by Label

Q you Schema Builder Create ▾

| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED | |
|------------|--------------|---------------|-------------|---------------|----------|---|
| Youth | Youth_c | Custom Object | | 11/3/2025 | ✓ | ▾ |
| YouthSkill | YouthSkill_c | Custom Object | | 11/3/2025 | ✓ | ▾ |

Setup Home Object Manager ▾

Object Manager
2 Items. Sorted by Label

Q job Schema Builder Create ▾

| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED | |
|----------|------------|---------------|-------------|---------------|----------|---|
| Job | Job_c | Custom Object | | 11/3/2025 | ✓ | ▾ |
| JobSkill | JobSkill_c | Custom Object | | 11/3/2025 | ✓ | ▾ |

Setup Home Object Manager ▾

Object Manager
2 Items. Sorted by Label

Q inter Schema Builder Create ▾

| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED | |
|-----------------|-------------------|---------------|-------------|---------------|----------|---|
| Interview | Interview_c | Custom Object | | 11/3/2025 | ✓ | ▾ |
| Interview Skill | Interview_Skill_c | Custom Object | | 11/3/2025 | ✓ | ▾ |

Step 2: Create Fields

Each object needs fields to capture record details.

Youth_c Fields

| Field Name | Type | Notes |
|------------------|-----------|------------------------|
| Email_c | Email | Email address of youth |
| Location_c | Text(255) | City/region |
| Qualification__c | Text | Education level |

Preferred_Role__c Picklist Options: Developer, Designer, Sales

Job__c Fields

| Field Name | Type | Notes |
|------------------------|-------------|-----------------------|
| Location__c | Text(255) | City/region of job |
| Description__c | Long Text | Job description |
| Required_Experience__c | Number/Text | Experience in years |
| Status__c | Picklist | Options: Open, Closed |

Interview__c Fields

| Field Name | Type | Notes |
|-------------------|-------------------|---|
| Interview_Date__c | DateTime | Scheduled date/time |
| Status__c | Picklist | Scheduled, Completed, Cancelled, Feedback_Given |
| Candidate__c | Lookup → Youth__c | Links to the youth being interviewed |
| Job__c | Lookup → Job__c | Links to the job |
| Interviewer__c | Lookup → User | Recruiter/interviewer |
| Feedback__c | Long Text Area | Interview feedback |

Setup Home Object Manager

SETUP > OBJECT MANAGER

Youth

Details

Fields & Relationships

8 Items, Sorted by Field Label

Q, Quick Find New Deleted Fields Field Dependencies Set History Tracking

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|-------------------|--------------------|-------------------|---------|
| Created By | CreatedById | Lookup(User) | | |
| Email | Email__c | Email | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Location | Location__c | Text(255) | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Preferred Role | Preferred_Role__c | Picklist | | |
| Qualification | Qualification__c | Text(100) | | |
| Youth ID | Name | Auto Number | | ✓ |

Setup Home Object Manager

SETUP > OBJECT MANAGER

Job

Details

Fields & Relationships
8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|---------------------|------------------------|-----------------------|-------------------|---------|
| Created By | CreatedById | Lookup(User) | | |
| Description | Description__c | Long Text Area(32768) | | |
| Job Name | Name | Text(80) | | ✓ |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Location | Location__c | Text(255) | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Required Experience | Required_Experience__c | Number(18, 0) | | |
| Status | Status__c | Picklist | | |

Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access

SETUP > OBJECT MANAGER

Interview

Details

Fields & Relationships
13 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|-------------------|-----------------------|-------------------|---------|
| Candidate | Candidate__c | Lookup(Youth) | | ✓ |
| Candidate Name | Candidate_Name__c | Formula (Text) | | |
| Created By | CreatedById | Lookup(User) | | |
| Feedback | Feedback__c | Long Text Area(32768) | | |
| Interview Date | Interview_Date__c | Date/Time | | |
| Interview ID | Name | Auto Number | | ✓ |
| Interviewer | Interviewer__c | Lookup(User) | | ✓ |
| Job | Job__c | Lookup(Job) | | ✓ |
| Job Title | Job_Title__c | Formula (Text) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User,Group) | | ✓ |
| Result | Result__c | Picklist | | |
| Status | Status__c | Picklist | | |

Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting

Step 3: Create Relationships

YouthSkill__c

- Master-Detail → Youth__c
- Lookup → Skill__c

JobSkill__c

- Master-Detail → Job__c

- Lookup → Skill__c

Step 4 : Configure Page

Layouts Youth__c Layout

- Fields: Email, Location, Qualification, Preferred Role
- Related Lists: YouthSkill__c, Interview__c

Job__c Layout

- Fields: Location, Description, Required Experience, Status
- Related Lists: JobSkill__c, Interview__c

Interview__c Layout

- Fields: Candidate, Job, Interview Date, Interviewer, Status, Feedback

Step 5 : Add Sample Records

- **Skills:** Java, Python, HTML
- **Youth:** Test Youth → Email: test@y.com, Location: Hyderabad
- **Job:** Java Developer → Location: Hyderabad, Status: Open

Link YouthSkill & JobSkill

- Test Youth → Java (YouthSkill)
- Java Developer Job → Java (JobSkill)

CareerBridge

Jobs

JobSkills

Youths

YouthSkills

Skills

Interviews

Dashboards

Reports

Accounts

Contacts

YouthSkills

Recently Viewed

New

Import

Assign Label

3 items • Updated a few seconds ago

Search this list...

| | <div><input type="checkbox"/> YouthSkill (junction) Name</div> | |
|---|--|-------------|
| 1 | <div><input type="checkbox"/> Sales Engineer</div> | <div></div> |
| 2 | <div><input type="checkbox"/> Developer-python</div> | <div></div> |
| 3 | <div><input type="checkbox"/> Developer-html</div> | <div></div> |

CareerBridge

Jobs

JobSkills

Youths

YouthSkills

Skills

Interviews

Dashboards

Reports

Accounts

Contacts

JobSkills

Recently Viewed

New

Import

Assign Label

2 items • Updated a few seconds ago

Search this list...

| | <div><input type="checkbox"/> JobSkill Name</div> | |
|---|---|-------------|
| 1 | <div><input type="checkbox"/> Designer-python</div> | <div></div> |
| 2 | <div><input type="checkbox"/> Developer-html</div> | <div></div> |

Phase 4: Process Automation (Admin)

Goal: Automate youth–job matching, interview scheduling, and email notifications, while ensuring data integrity.

Step 1: Create Email Templates

1. Click **Setup** (⚙️) → In Quick Find, type **Email Templates** → Select **Email Templates**.
 2. Click **New Email Template** → Select **Lightning Email Template**.
 3. Fill details like **Name** and **Subject**, then design the email body.
 4. Save the template.
- **Welcome Email** → Sent after youth registration.

The screenshot shows the Salesforce Classic Email Templates interface. At the top, there's a 'SETUP' button and the title 'Classic Email Templates'. Below this, the 'Text Email Template' section shows the name 'welcome youth' and a preview of the email body. The 'Email Template Detail' section contains fields for 'Email Template Name', 'Template Unique Name', 'Encoding', 'Author', 'Description', 'Created By', and 'Modified By'. The 'Email Template' section at the bottom shows the 'Subject' and a 'Plain Text Preview' of the email body.

Text Email Template
welcome youth
Preview your email template below.

Email Template Detail [Edit] [Delete] [Clone]

| | | | |
|---------------------------------|--|-------------------|---------------------------------------|
| Email Templates from Salesforce | Unified Public Classic Email Templates | Available For Use | <input type="checkbox"/> |
| Email Template Name | welcome youth | Last Used Date | |
| Template Unique Name | welcome_youth | Times Used | |
| Encoding | Unicode (UTF-8) | | |
| Author | korra sunil kumar [Change] | | |
| Description | | | |
| Created By | korra sunil kumar: 11/3/2025, 8:42 PM | Modified By | korra sunil kumar: 11/3/2025, 8:42 PM |

[Edit] [Delete] [Clone]

Email Template [Send Text and Verify Merge Fields]

Subject | Welcome to SkillMatch-Connect


Plain Text Preview

Hello YID-0043,

Thank you for registering with SkillMatch-Connect. Your Skills(Developer) are now recorded.
We Will notify you when Your Skills Matching jobs are posted.

Regards,
SkillMatch-Connect Team

- **Interview Scheduled Email** → Sent after youth is matched with a job and interview is created.

 **Classic Email Templates**

Text Email Template

Interview

Preview your email template below.

Email Template Detail

Email Templates from Salesforce

Unified Public Classic Email Templates

Email Template Name

Interview

Available For Use

☐

Template Unique Name

Interview_scheduled

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

korra_sunil kumar [\[Change\]](#)

Description

Created By

korra_sunil kumar, 11/3/2025, 8:55 PM

Modified By

korra_sunil kumar, 11/3/2025, 8:55 PM

Edit

Delete

Clone

Email Template

Send Test and Verify Merge Fields


Subject

Interview Scheduled

Plain Text Preview

Hello {{Interview__c.Candidate_Name__c}},
Congratulations! You have been shortlisted for the {{Interview__c.Job_Title__c}} position.
Your interview is scheduled on {{Interview__c.Interview_Date__c}} {{Interview__c.Interview_Time_Only__c}}.
Please be prepared.
Best regards,
SkillMatch-Connect

- **Interview Completed Email** → Sent after interview status is marked as Completed.

 **Classic Email Templates**

Text Email Template

Interview Complete

Preview your email template below.

Email Template Detail

Email Templates from Salesforce

Unified Public Classic Email Templates

Email Template Name

Interview Complete

Available For Use

☐

Template Unique Name

Interview_Complete

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

korra_sunil kumar [\[Change\]](#)

Description

Created By

korra_sunil kumar, 11/3/2025, 8:58 PM

Modified By

korra_sunil kumar, 11/3/2025, 8:58 PM

Edit

Delete

Clone

Email Template

Send Test and Verify Merge Fields

Subject


Interview Completed Scheduled

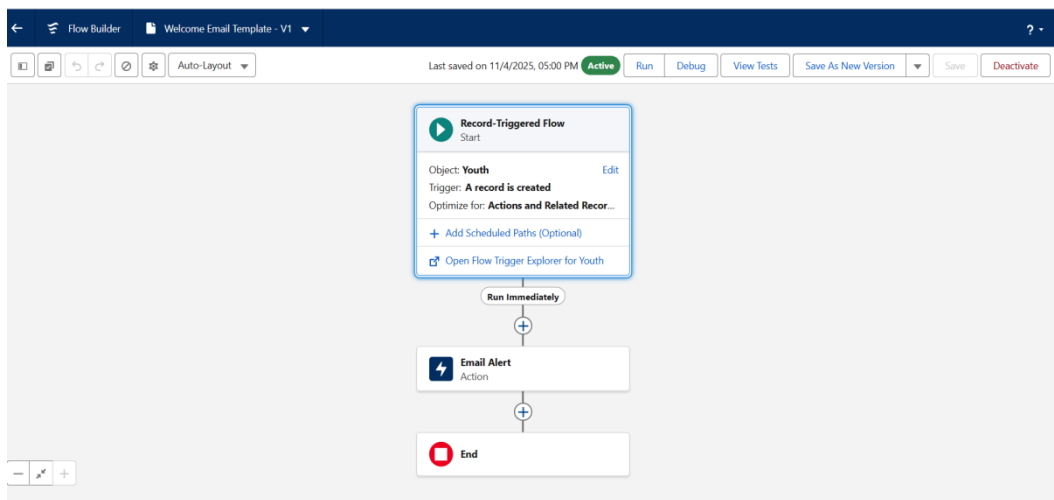
Plain Text Preview

Hello {{Interview__c.Candidate_Name__c}},
Your interview for {{Interview__c.Job__r.Name}} has been successfully marked as Completed.
Result: {{Interview__c.Result__c}}
Thank you for attending.
Best regards,
SkillMatch-Connect

Step 2: Build Flows

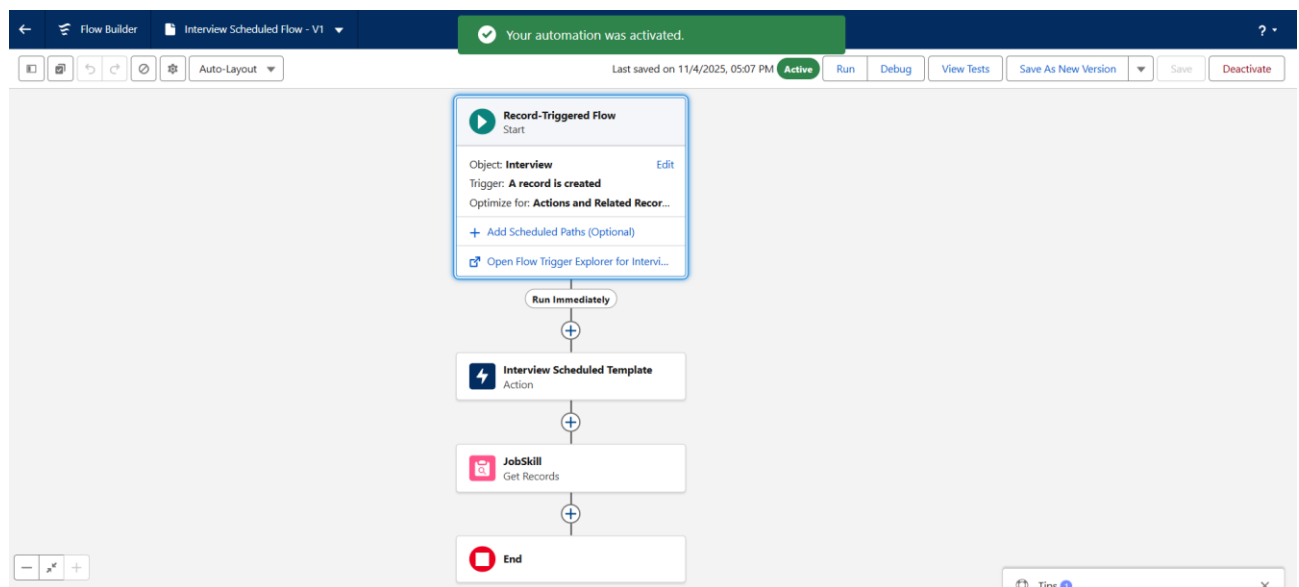
1. Welcome Email Flow (After Registration)

1. Click **Setup**  → Quick Find → **Flows** → **New Flow**.
2. Select **Record-Triggered Flow**.
3. Choose **Object: Youth__c** → Trigger: **When record is created** → **After Save**.
4. Add **Action** → **Send Email** → Select **Welcome Email Template**.
5. Save and Activate.



2. Interview Scheduled Flow (After Match & Scheduling)

1. Click **Setup** → **Flows** → **New Flow**.
 2. Select **Record-Triggered Flow** → **Object: Job_c** → Trigger: **Created or Updated** → **After Save**.
 3. Add **Get Records** → **JobSkill__c** (fetch required skills).
 4. Add **Get Records** → **YouthSkill__c** (find youth with matching skills).
 5. Add **Loop** → For each Youth, **Create Interview__c record** (Candidate, Job, Status = Scheduled, Date).
 6. Add **Action** → **Send Email** → Select **Interview Scheduled Template**.
- Save and Activate

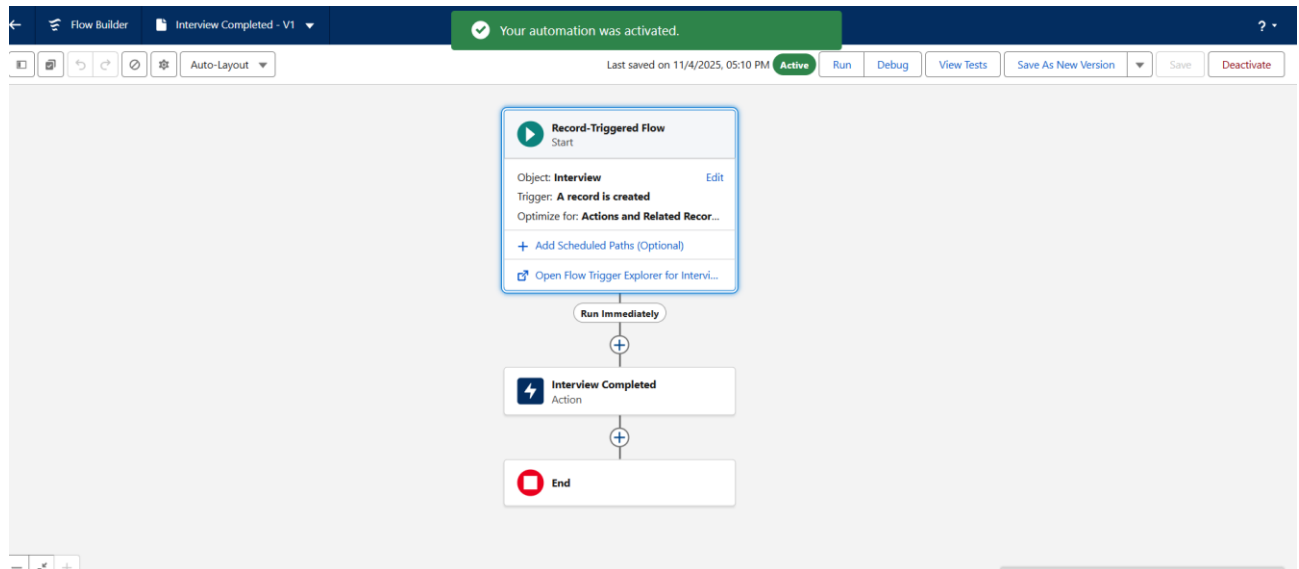


3. Interview Completed Flow (After Interview Completion)

6. Click **Setup** → **Flows** → **New Flow**.
7. Select **Record-Triggered Flow** → **Object: Interview__c**.
8. Trigger: **When record is updated** → **After Save**.
9. Add **Decision Element** → Condition: Status__c = Completed.

10. Add Action → **Send Email** → Select **Interview Completed Template**.

11. Save and Activate.



Step 3: Add Validation Rules

1. Click **Setup** → **Object Manager** → **Youth__c** → **Validation Rules** → **New**.

2. Create Rule: **Email Required**

- Formula: ISBLANK(Email__c)
- Error: "Email is required."

The screenshot shows the Object Manager page for the 'Youth Validation Rule'. The page includes a 'Validation Rule Detail' section with the following information:

| Validation Rule Detail | | Active |
|-------------------------|---------------------------------------|--|
| Rule Name | Email_Required | ✓ |
| Error Condition Formula | ISBLANK(\$!User Email) | |
| Error Message | Email is required. | Error Location: Top of Page |
| Description | | |
| Created By | korra sunil kumar: 11/4/2025, 3:49 AM | Modified By: korra sunil kumar: 11/4/2025, 3:49 AM |

3. Create Rule: **Location Required**

- Formula: ISBLANK(Location__c)
- Error: “Location is required.”

Youth Validation Rule

[Back to Youth](#)

[Help for this Page](#)

Validation Rule Detail

[Edit](#) [Clone](#)

| | | | |
|-------------------------|---|----------------|--|
| Rule Name | Location_Required | Active | ✓ |
| Error Condition Formula | ISBLANK(Location__c) | | |
| Error Message | Location is required. | Error Location | Top of Page |
| Description | Youth Validation Rule ~ Salesforce - Developer Edition Location | | |
| Created By | Youth Coordinator 9/13/2025, 9:15 AM | Modified By | Youth Coordinator 9/13/2025, 9:15 AM |

[Edit](#) [Clone](#)

Phase 5: Apex Programming (Developer)

Goal:

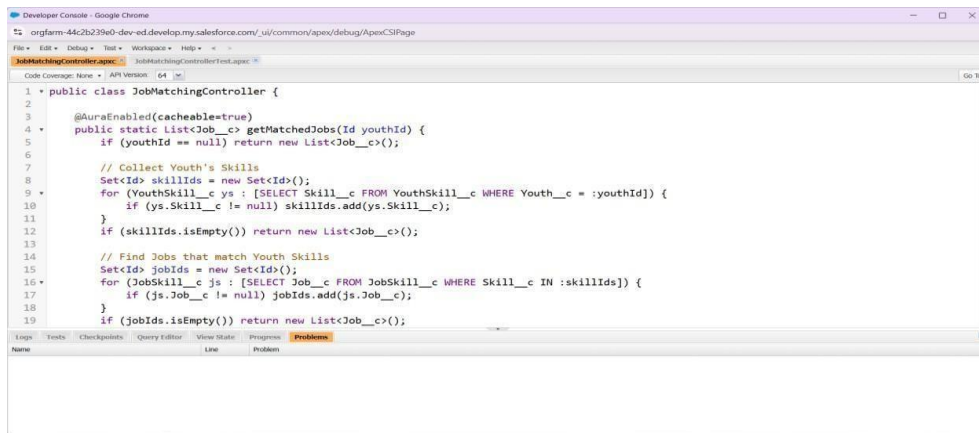
- Match youth to jobs based on skills using Apex.
- Automatically create Interview__c records.
- Handle bulk data safely and meet Salesforce deployment standards.

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).

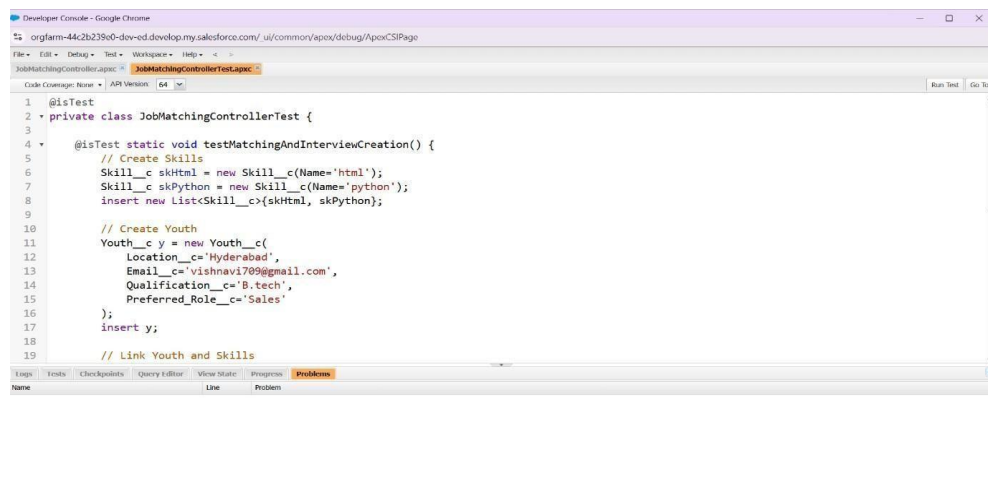


Step 2: Apex Test Class — JobMatchingControllerTest

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:

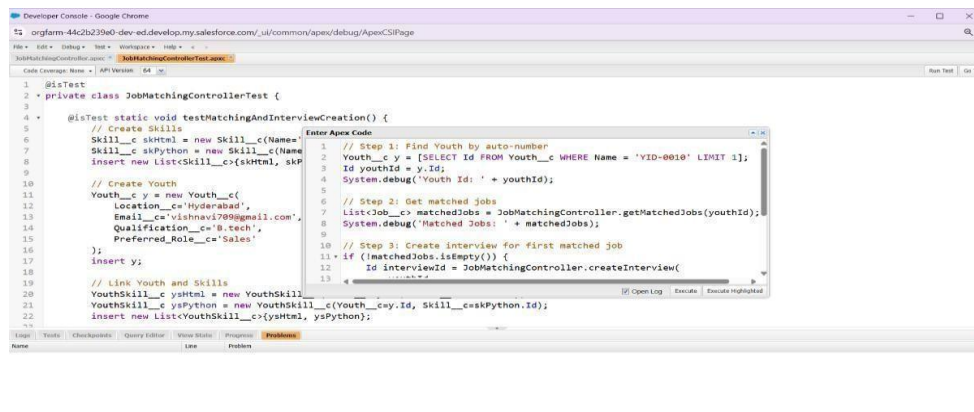
- Creates test Skill__c, Youth__c, YouthSkill__c, Job__c, JobSkill__c.
- Tests getMatchedJobs() → asserts correct number of jobs.
- Tests createInterview() → asserts interview Id is not null



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.

Setup | Home | Object Manager

Q: jobs

Environments

- Jobs
 - Apex Flex Queue
 - Apex Jobs
 - Background Jobs
 - Bulk Data Load Jobs
 - Scheduled Jobs**

Didn't find what you're looking for? Try using Global Search.

SETUP Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

Percentage of Scheduled Jobs Used: 1%
You have currently used 1 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the [Lightning Platform Apex Limits](#) topic.

View: All Scheduled Jobs | Create New View

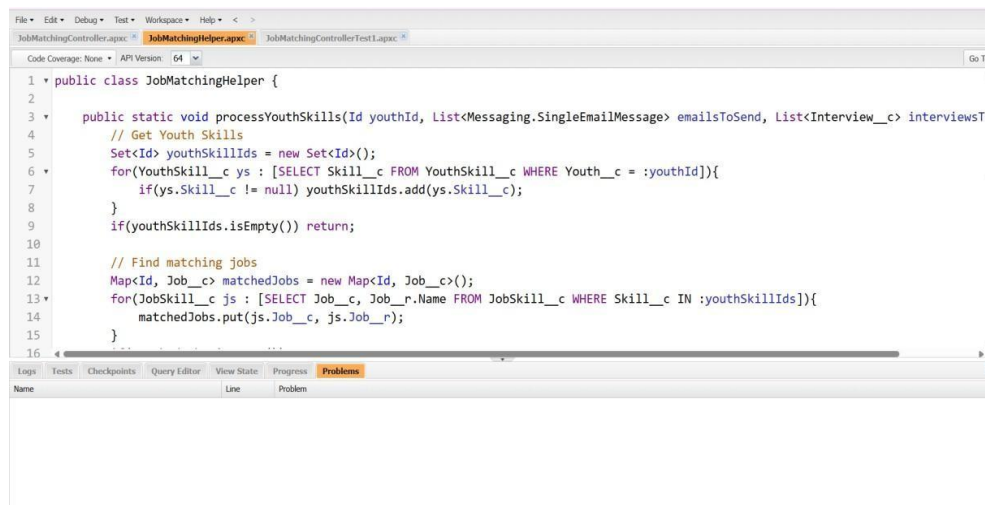
| Action | Job Name | Submitted By | Submitted | Started | Next Scheduled Run | Type | Cron Trigger ID |
|--------------------------|--|------------------------|--------------------|--------------------|---------------------|--|-------------------|
| Manage Del Pause Job | AutoInterviewSchedulerJob | AutoInterviewScheduler | 9/19/2025, 7:48 PM | | 9/20/2025, 10:00 AM | Scheduled Apex | 08egl.000000CMcTV |
| Del | Metalytics Data Loader Job for Org: 000G0.000000ACp2 | User_Integration | 8/27/2025, 9:18 AM | 9/19/2025, 4:53 AM | 9/20/2025, 4:53 AM | Autonomous Data Loader Job | 08egl.000000An2KX |
| | 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 | 08egl.000000An2KV |
| | 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 | 08egl.000000An2KV |

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**.



```
1 public class JobMatchingHelper {
2
3     public static void processYouthSkills(Id youthId, List<Messaging.SingleEmailMessage> emailsToSend, List<Interview__c> interviewsToCreate) {
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     }
17 }
```

Phase 6: User Interface Development

Step1:Lightning App Builder

- Go to setup -> App Manager -> New Lightning App.
- Name the app CareerBridge.
- Add navigation items: Home, Youth, Job, Interviews, Reports, Dashboards
- Assign to profiles and Finish.

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.


App Details

*App Name ⓘ
SkillMatch

*Developer Name ⓘ
SkillMatch

Description ⓘ
Enter a description...

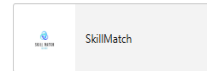
App Branding

Image ⓘ


Primary Color Hex
Value ⓘ
#0070C2

Org Theme Options
☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview



Lightning App Builder | App Settings | Pages | CareerBridge | ? Help

App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items**
- User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Create

- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Approval Submission Details

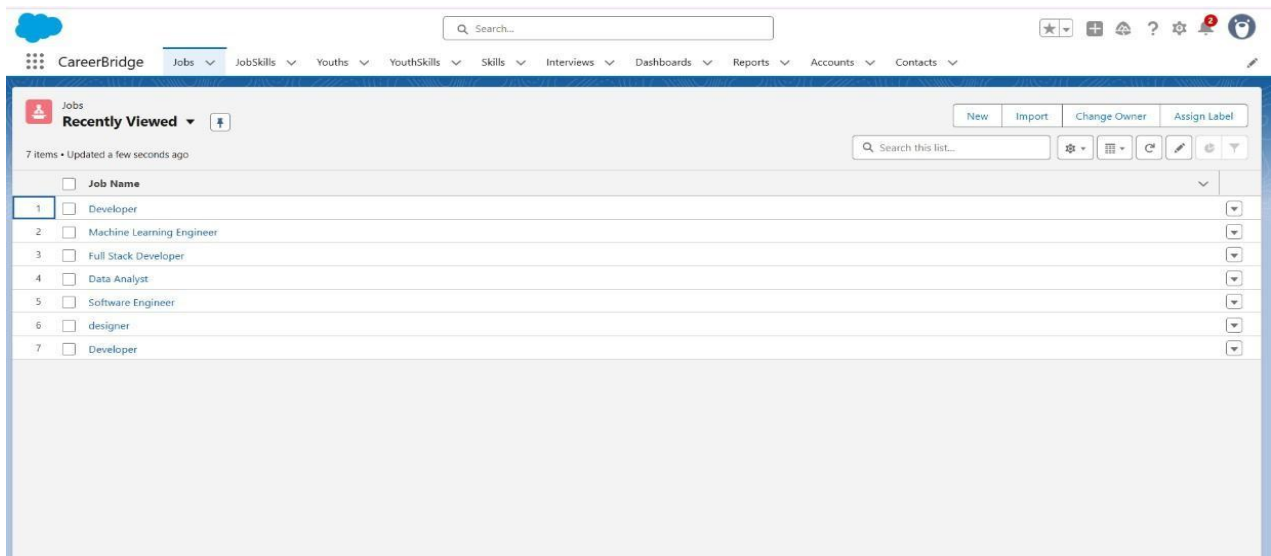
Selected Items

- Jobs
- JobSkills
- Youths
- YouthSkills
- Skills
- Interviews
- Dashboards
- Reports
- Accounts
- Contacts

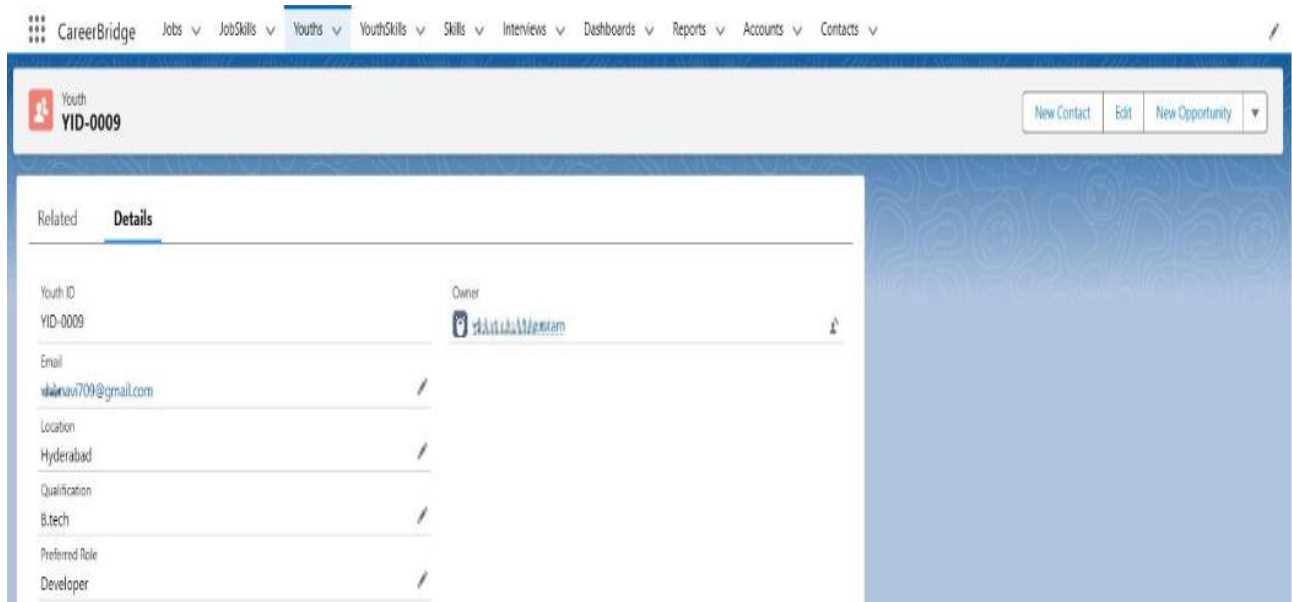
Step 2: Record Pages

- Custom layouts for specific object records (**Youth, Job, Interview, Skill**).
- You can decide what **fields, related lists, and components** appear.

Youth Record Page: Show youth details, skills, and recommended jobs.



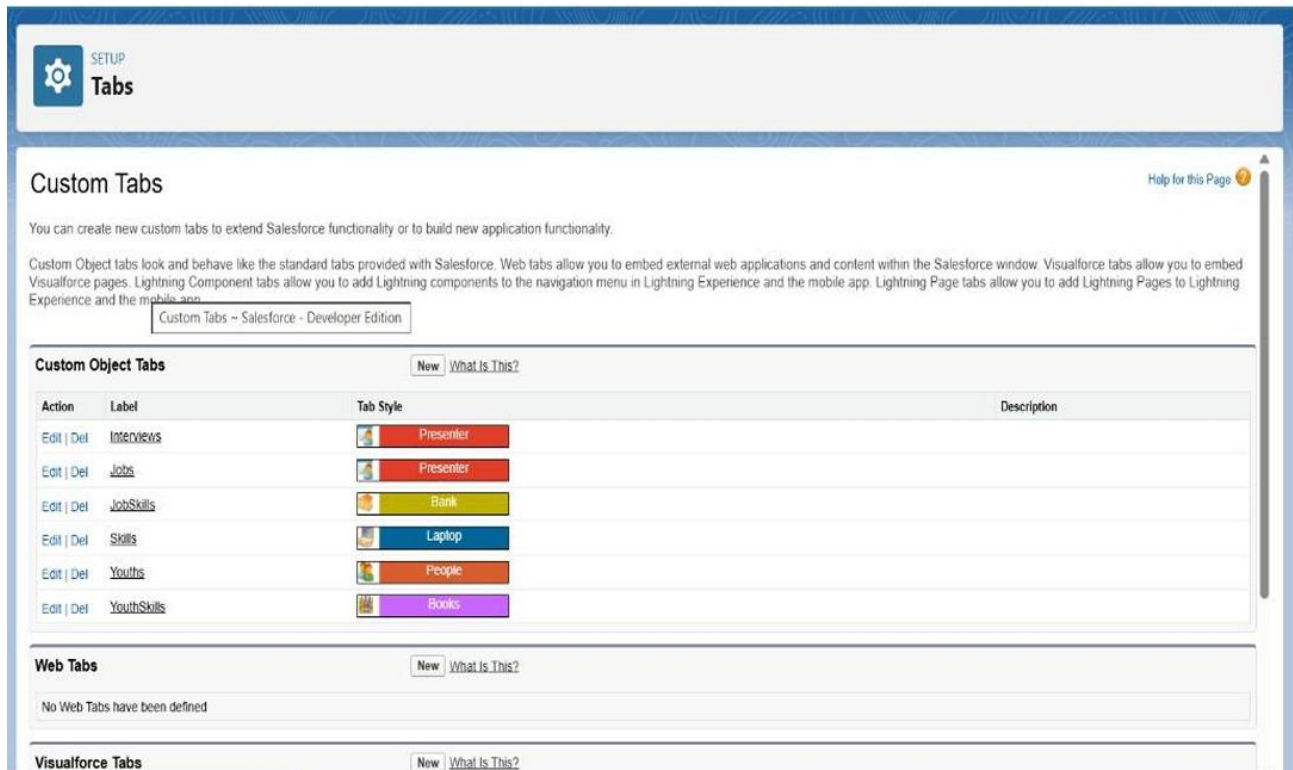
Job Record Page: Show job details, required skills, and linked inter views



- **Interview Record Page:** Show candidate, job, date, and status.

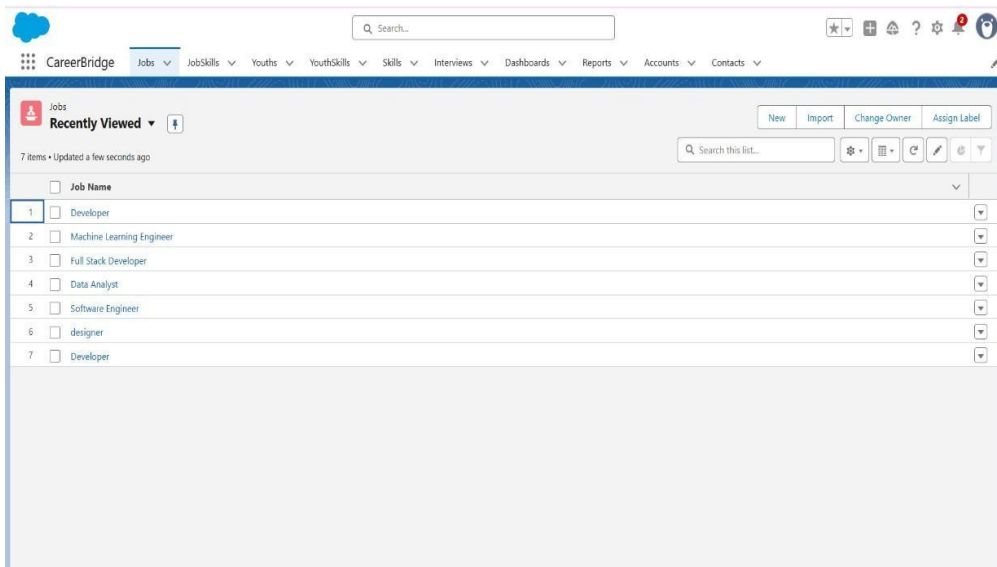
Step 3: Tabs

- Setup → **App Manager** → **Edit CareerBridge** → **Navigation Items**.
- Add or reorder tabs (**YouthSkill, Youth, Jobs, Interviews, JobsSkill, Skill**).
- **Save**.



Step 4: Home Page Layouts

- The home page can be customized for different profiles (e.g., Youth, Admin).
- Provides dashboards, tasks, and quick actions.
 - **CareerBridge Example:**
 - For Youth: Show registered jobs, upcoming interviews, and assigned skills.
 - For Admin: Display KPIs like total youth registered, jobs posted, and interviews scheduled.



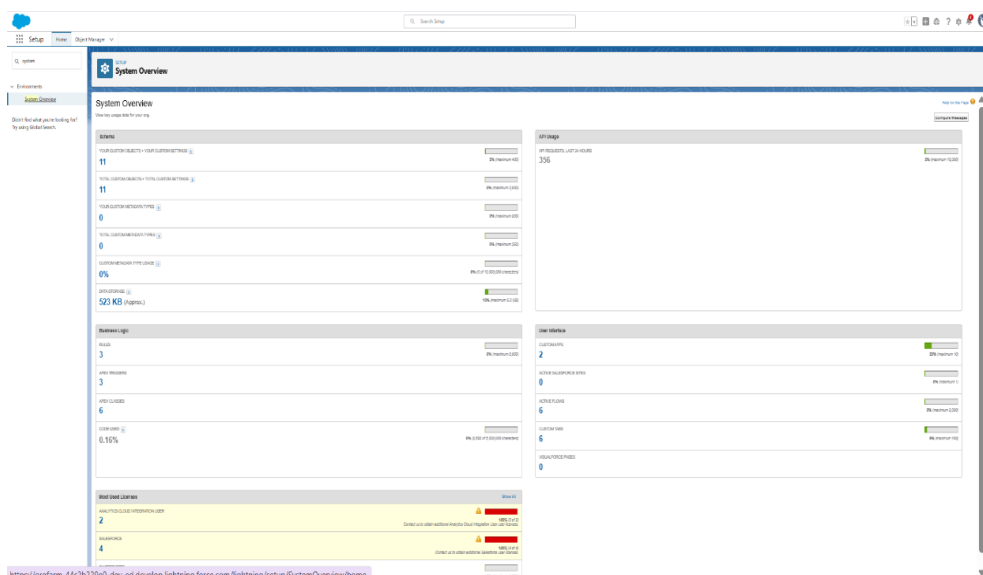
Phase 7: Integration & External Access

1. API Limits

- Salesforce tracks API calls made by external apps or integrations.
- In CareerBridge, all actions (Youth registration, Job posting, Flows, Emails) happen

inside Salesforce.

- **No external API calls are used**, so API Limits **do not affect your project**.
- You can check usage in **Setup → System Overview**, but nothing needs to be configured.



Phase 8: Data Management & Deployment

Step 1: Duplicate Management

Preventing duplicate records is critical, especially for **Youth** and **Job** objects.

A. Matching Rules

- Define the criteria to identify duplicates.
- **Steps:**
 1. Go to **Setup** → **Matching Rules** → **New**.
 2. Example: Match Youth_c.Email_c.
 3. Define criteria (e.g., exact match, case-insensitive) → **Save** → **Activate**.

The screenshot shows the Salesforce 'Matching Rules' configuration page. The left sidebar contains a search bar and navigation links for 'Data', 'Duplicate Management', and 'Matching Rules'. The main content area is titled 'Edit Rule Youth Email Match' and includes 'Save' and 'Cancel' buttons. The 'Rule Details' section shows the 'Object' as 'Youth', 'Rule Name' as 'Youth Email Match', and 'Unique Name' as 'Youth_Email_Match'. The 'Matching Criteria' section allows defining fields to compare, with 'Email' selected and 'Exact' chosen as the matching method. A 'Match Blank Fields' checkbox is checked. The bottom of the page has 'Save' and 'Cancel' buttons.

B. Duplicate Rules

- Decide what happens when duplicates are detected.
- **Steps:**
 1. Go to **Setup** → **Duplicate Rules** → **New**.
 2. Apply your **Matching Rule** to the object (e.g., Youth__c).
 3. Choose **Action**:
 - **Block** → Prevents user from creating duplicates.

- **Alert** → Warns the user, but allows creation.

4. Activate the rule.

The screenshot shows the Salesforce Setup interface. On the left, the navigation menu includes 'Setup', 'Home', and 'Object Manager'. Under 'Data', there are options for 'Duplicate Management', 'Duplicate Error Logs', 'Duplicate Rules' (which is selected), and 'Matching Rules'. The main content area is titled 'Duplicate Rules' and shows the configuration for a rule named 'Prevent Duplicate Youth Email'. The 'Rule Details' section includes fields for 'Rule Name' (Prevent Duplicate Youth En), 'Description', 'Object' (Youth), and 'Record Level Security' (Enforce sharing rules). The 'Actions' section specifies 'Action On Create' and 'Action On Edit' as 'Block', and an 'Alert Text' of 'Use one of these records?'. The 'Matching Rules' section shows 'Compare Youths With' set to 'Youths'.

Step 2: Data Backup

Always back up before importing or deploying, in case something goes wrong.

- **Steps:**
 1. Go to **Setup** → **Data Export** → **Export Now**.
 2. Choose objects to backup (e.g., Youth__c, Job__c, Interview__c).
 3. Download the zip files.

The screenshot shows an Excel spreadsheet with a data backup for the 'Youth__c' object. The spreadsheet has columns for 'Id', 'OwnerId', 'IsDeleted', 'Name', 'CreatedDate', 'CreatedBy', 'LastModifiedDate', 'LastModifiedBy', 'SystemMod Email__c', 'Location', 'Qualification', and 'Preferred Role__c'. The data is organized into rows, with the first row being a header and subsequent rows containing individual records. The 'Id' column contains long alphanumeric strings, and the 'Name' column contains names like 'Vishnavi B. tech', 'Vishnavi B. tech', 'Vishnavi B. tech', 'Vishnavi B. tech', and 'Vishnavi B. tech'. The 'Preferred Role__c' column contains roles like 'Developer', 'Sales', 'Developer', 'Sales', and 'Developer'.

| OwnerId | IsDeleted | Name | CreatedOn | CreatedBy | LastModified | SystemMo | Location | Description | Required | Status | c |
|-----------------------|-----------|--------------|-----------|------------|--------------|------------|-----------|-------------|----------|--------|---|
| a07gl.0000.005gl.0000 | 0 | Java Devel | ##### | 005gl.0000 | ##### | 005gl.0000 | Hyderabad | Developer | 2 | Open | |
| a06gl.0000.005gl.0000 | 0 | Developer | ##### | 005gl.0000 | ##### | 005gl.0000 | Hyderabad | Required E | 2 | Open | |
| a06gl.0000.005gl.0000 | 0 | designer | ##### | 005gl.0000 | ##### | 005gl.0000 | Hyderabad | | 1 | Open | |
| a06gl.0000.005gl.0000 | 0 | Software E | ##### | 005gl.0000 | ##### | 005gl.0000 | Bangalore | | 0 | Open | |
| a06gl.0000.005gl.0000 | 0 | Data Anal | ##### | 005gl.0000 | ##### | 005gl.0000 | Chennai | | 2 | Open | |
| a06gl.0000.005gl.0000 | 0 | Full Stack I | ##### | 005gl.0000 | ##### | 005gl.0000 | Pune | | 1 | Open | |
| a06gl.0000.005gl.0000 | 0 | Machine L | ##### | 005gl.0000 | ##### | 005gl.0000 | Delhi | | 1 | Open | |

AutoSave

Interview_c.csv

Read...

Saved to this PC

Search

FileHomeInsertDrawPage LayoutFormulasDataReviewViewHelp

CutCopyFormat PainterClipboard

Calibri11A⁺A⁻BBIUFont

Wrap TextMerge & CenterAlignment

GeneralConditional FormattingStylesFormat as TableCell StylesInsertDeleteFormatCells

AutosumFillClearEditingSort & FilterFind & SelectAdd-insAdd-ins

POSSIBLE DATA LOSS

Some features may be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it as an Excel file format.

Don't show again

Save As...

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
|----|-----------------------|---------|-----------|-------|------------|-----------|--------------|----------|-----------|------------|------------|----------------------|-------------|----------|----------|---|---|---|---|---|---|---|---|
| 1 | Id | OwnerId | IsDeleted | Name | CreatedOn | CreatedBy | LastModified | SystemMo | Interview | Status_c | Candidate | Job_c | Interview | Feedback | Result_c | | | | | | | | |
| 2 | a07gl.0000.005gl.0000 | 0 | IN-0001 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Completed | a05gl.0000 | a06gl.0000 | 005gl.0000 | Candidate | Rejected | | | | | | | | | |
| 3 | a07gl.0000.005gl.0000 | 0 | IN-0002 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Completed | a05gl.0000 | a06gl.0000 | 005gl.0000 | Candidate | Rejected | | | | | | | | | |
| 4 | a07gl.0000.005gl.0000 | 0 | IN-0003 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Scheduled | a05gl.0000 | a06gl.0000 | 005gl.00000071j8QAE | | | | | | | | | | | |
| 5 | a07gl.0000.005gl.0000 | 0 | IN-0004 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Scheduled | a05gl.0000 | a06gl.0000 | 005gl.00000071wM6QAI | | | | | | | | | | | |
| 6 | a07gl.0000.005gl.0000 | 0 | IN-0006 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Completed | a05gl.0000 | a06gl.0000 | 005gl.00000071j8QAE | Shortlisted | | | | | | | | | | |
| 7 | a07gl.0000.005gl.0000 | 0 | IN-0005 | ##### | 005gl.0000 | ##### | 005gl.0000 | ##### | Completed | a05gl.0000 | a06gl.0000 | 005gl.00000071wM6Q | Qualified | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | | | | | | | | | |

Interview_c

Ready

Accessibility: Unavailable

Phase 9: Reporting, Dashboards & Security Review

Goal:

- Track progress through reports and dashboards
- Protect sensitive data from unauthorized access
- Ensure stakeholders see the right information in real time

Step 1: Reporting

Reports help track Youth, Jobs, and Interviews effectively.

A. Create Reports

1. Go to **App Launcher** → **Reports** → **New Report**
2. Select the report type: e.g., **Interview** or **Custom Report**
3. Add fields:
 - Candidate Name (Youth__c Name)
 - Job Title (Job__c Name)
 - Interview Date (Interview_Date__c)
 - Status (Status__c)
4. Save & Run the report

B. Useful Reports

- **Interviews by Status** → Group by Status__c (Shows Scheduled vs Completed interviews)
- **Interviews by Result** → Group by Result__c (Shows Qualified, Shortlisted, Rejected outcomes)
- **Interviews Over Time** → Group by Last Interview Date (Shows trends or frequency of interviews)

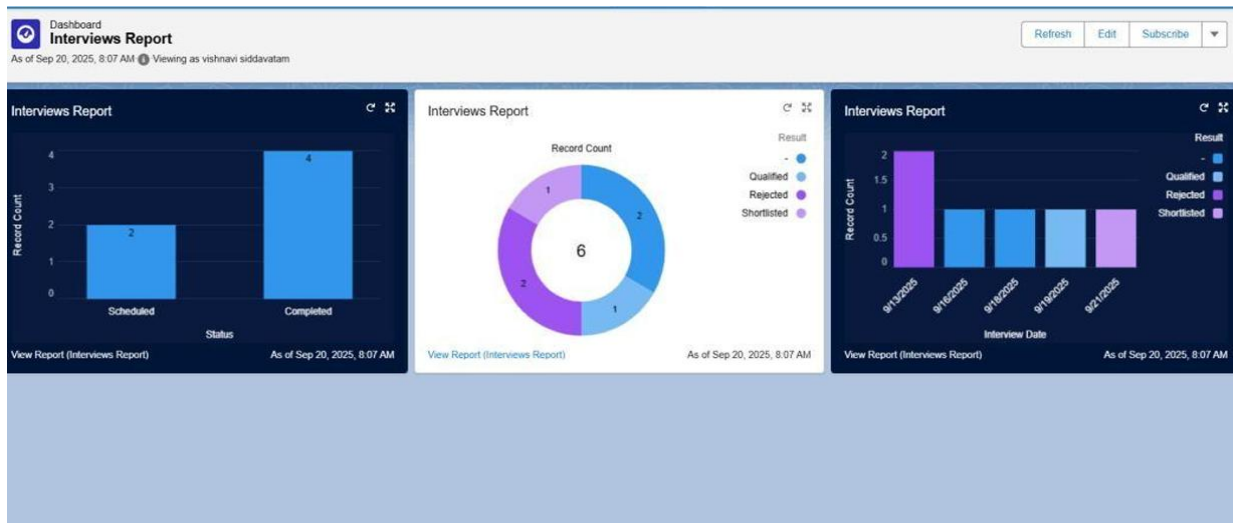
| Status | Result | Interview Date | Candidate Name | Job |
|---------------|-----------------|----------------|----------------|-----------|
| Scheduled (2) | - (2) | 9/16/2025 (1) | YID-0009 | Developer |
| | Subtotal | | | |
| | 9/16/2025 (1) | YID-0001 | Java Developer | |
| | Subtotal | | | |
| | Subtotal | | | |
| Completed (4) | Qualified (1) | 9/16/2025 (1) | YID-0009 | Developer |
| | Subtotal | | | |
| | Subtotal | | | |
| | Projected (2) | 9/13/2025 (2) | YID-0009 | designer |
| | Subtotal | | | |
| | Subtotal | | | |
| | Shortlisted (1) | 9/21/2025 (1) | YID-0009 | designer |
| | Subtotal | | | |
| | Subtotal | | | |
| | Total (6) | | | |

Step 2: Dashboards

Dashboards visually display metrics from reports.

A. Create Dashboard

1. Go to **App Launcher** → **Dashboards** → **New**
2. Add components:
 - **Metric:** Count of Completed Interviews
 - **Bar Chart:** Interviews by Status
 - **Line Chart:** Interviews per Week
 - **Table:** Recent Interviews
3. Enable Dynamic Dashboards → Set “View Dashboard As = Dynamic”

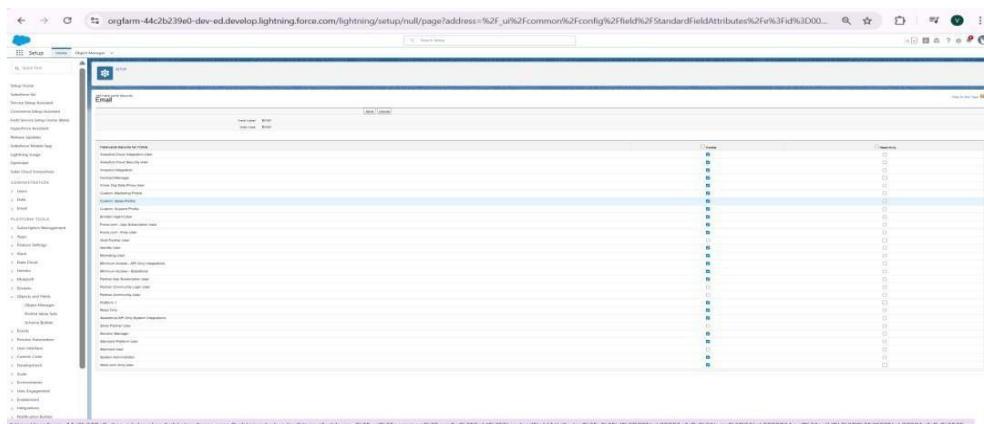


Step 3: Security

Protect sensitive data (like Youth emails, interview feedback) and control who can see what.

A. Field-Level Security (FLS)

1. Setup → Object Manager → Youth_c → Fields → Email_c → Set Field-Level Security



2. Hide fields from standard users or users without permission

B. Sharing & Org-Wide Defaults (OWD)

1. Setup → Sharing Settings
2. Ensure Youth_c = Private

C. Audit Trail

1. Setup → View Setup Audit Trail
2. Track unauthorized changes or admin activity

SETUP

View Setup Audit Trail

The last 20 entries for your organization are listed below. You can [download](#) your organization's setup audit trail for the last six months (Excel .csv file).

View Setup Audit Trail

| Date | User | Source Namespace Prefix | Action | Section | Delegate User ? |
|---------------------------|-------------------------------------|-------------------------|---|------------------|-----------------|
| 11/4/2025, 9:10:42 PM PST | korrasunilkumar04509@agentforce.com | | Added standard button override on Youths: View (Lightning Page null) | Custom Objects | |
| 11/4/2025, 9:09:51 PM PST | korrasunilkumar04509@agentforce.com | | Created Lightning Page: Youth | Lightning Pages | |
| 11/4/2025, 8:56:02 PM PST | korrasunilkumar04509@agentforce.com | | Created custom app SkillMatch | Custom Apps | |
| 11/4/2025, 8:56:02 PM PST | korrasunilkumar04509@agentforce.com | | Created Lightning Page: SkillMatch UtilityBar | Lightning Pages | |
| 11/4/2025, 8:54:37 PM PST | korrasunilkumar04509@agentforce.com | | Deleted custom app SkillMatch | Custom Apps | |
| 11/4/2025, 8:54:37 PM PST | korrasunilkumar04509@agentforce.com | | Deleted Lightning Page: SkillMatch UtilityBar | Lightning Pages | |
| 11/4/2025, 8:54:09 PM PST | korrasunilkumar04509@agentforce.com | | Created Lightning Page: SkillMatch UtilityBar | Lightning Pages | |
| 11/4/2025, 8:54:09 PM PST | korrasunilkumar04509@agentforce.com | | Created custom app SkillMatch | Custom Apps | |
| 11/4/2025, 3:49:39 AM PST | korrasunilkumar04509@agentforce.com | | New Youths validation rule "Email_Required" | Validation Rules | |
| 11/4/2025, 3:40:30 AM PST | korrasunilkumar04509@agentforce.com | | Activated flow with Name "Interview Completed" and Unique Name "Interview_Completed" | Flows | |
| 11/4/2025, 3:40:16 AM PST | korrasunilkumar04509@agentforce.com | | Created flow with Name "Interview Completed" and Unique Name "Interview_Completed" | Flows | |
| 11/4/2025, 3:37:47 AM PST | korrasunilkumar04509@agentforce.com | | Activated flow with Name "Interview Scheduled Flow" and Unique Name "Interview_Scheduled_Flow" | Flows | |
| 11/4/2025, 3:37:43 AM PST | korrasunilkumar04509@agentforce.com | | Deleted flow version #1 "Interview Scheduled Flow" for flow with Unique Name "Interview_Scheduled_Flow" | Flows | |
| 11/4/2025, 3:37:42 AM PST | korrasunilkumar04509@agentforce.com | | Created flow version #1 "Interview Scheduled Flow" for flow with Unique Name "Interview_Scheduled_Flow" | Flows | |
| 11/4/2025, 3:32:42 AM PST | korrasunilkumar04509@agentforce.com | | Created flow with Name "Interview Scheduled Flow" and Unique Name "Interview_Scheduled_Flow" | Flows | |
| 11/4/2025, 3:30:52 AM PST | korrasunilkumar04509@agentforce.com | | Activated flow with Name "Welcome Email Template" and Unique Name "Welcome_Email_Template" | Flows | |
| 11/4/2025, 3:30:48 AM PST | korrasunilkumar04509@agentforce.com | | Deleted flow version #1 "Welcome Email Template" for flow with Unique Name "Welcome_Email_Template" | Flows | |
| 11/4/2025, 3:30:43 AM PST | korrasunilkumar04509@agentforce.com | | Created flow version #1 "Welcome Email Template" for flow with Unique Name "Welcome_Email_Template" | Flows | |
| 11/4/2025, 3:24:39 AM PST | korrasunilkumar04509@agentforce.com | | Deactivated flow with Name "Welcome Email Template" and Unique Name "Welcome_Email_Template" | Flows | |
| 11/4/2025, 3:24:39 AM PST | korrasunilkumar04509@agentforce.com | | Deleted flow version #1 "Welcome Email Template" for flow with Unique Name "Welcome_Email_Template" | Flows | |

[Download setup audit trail for last six months \(Excel .csv file\)](#)

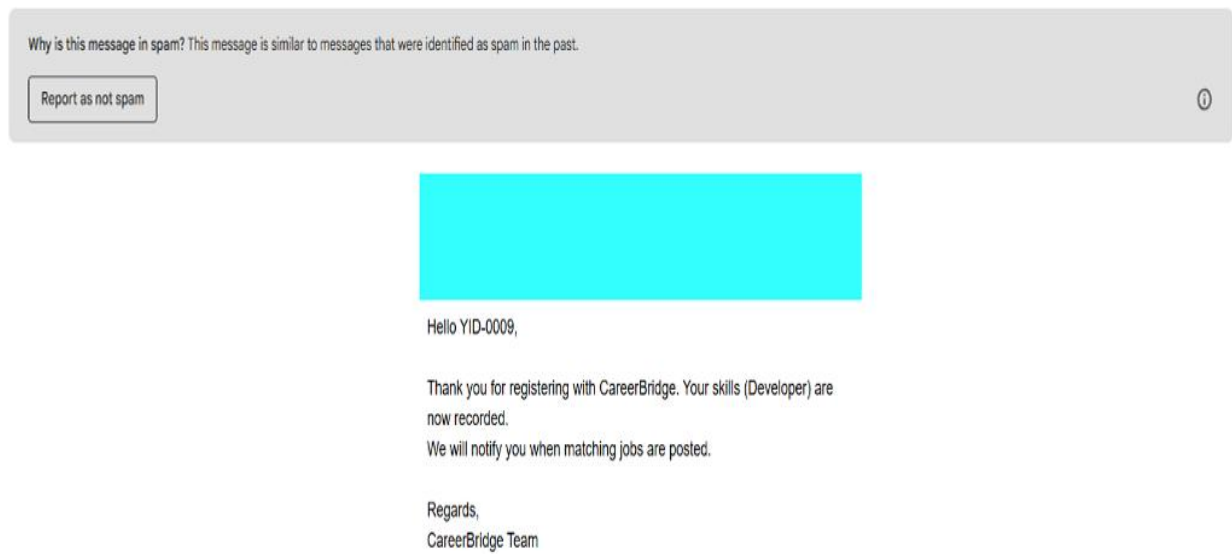
[develop.my.salesforce-setup.com/lightning/setup/SecurityEvents/home](#)

Phase 10: Quality Assurance Testing

Test Case 1 – Youth Registration → Welcome Email

- **Use Case / Scenario:** Youth registers on the platform.
- **Test Steps (Input):**

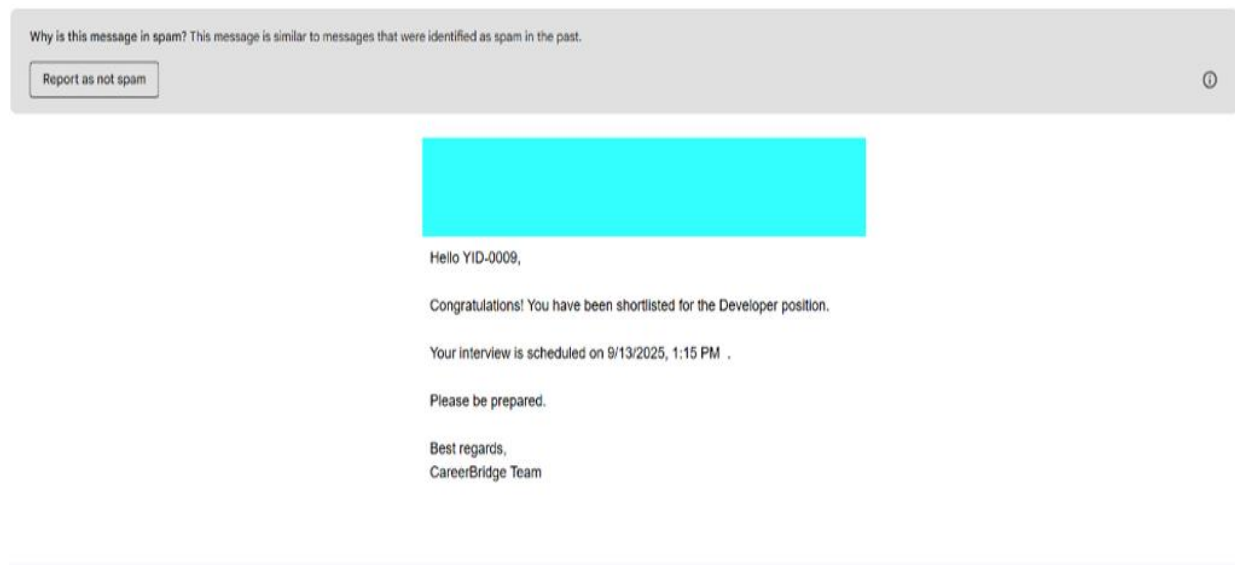
- Fill Youth form → Name = Vishnavi, Email = vishnavi@gmail.com, Skills = Java, Python.
- **Expected Result:**
 - Youth record saved in **Youth__c** object.
 - **Welcome Email** triggered:
“Hello , thank you for registering. Your skills (Python) are now recorded.”



Test Case 2 – Employer Posts Job → Skills Match → Interview Scheduled Email

- **Use Case / Scenario:** Employer posts job requiring skills.
- **Test Steps (Input):**

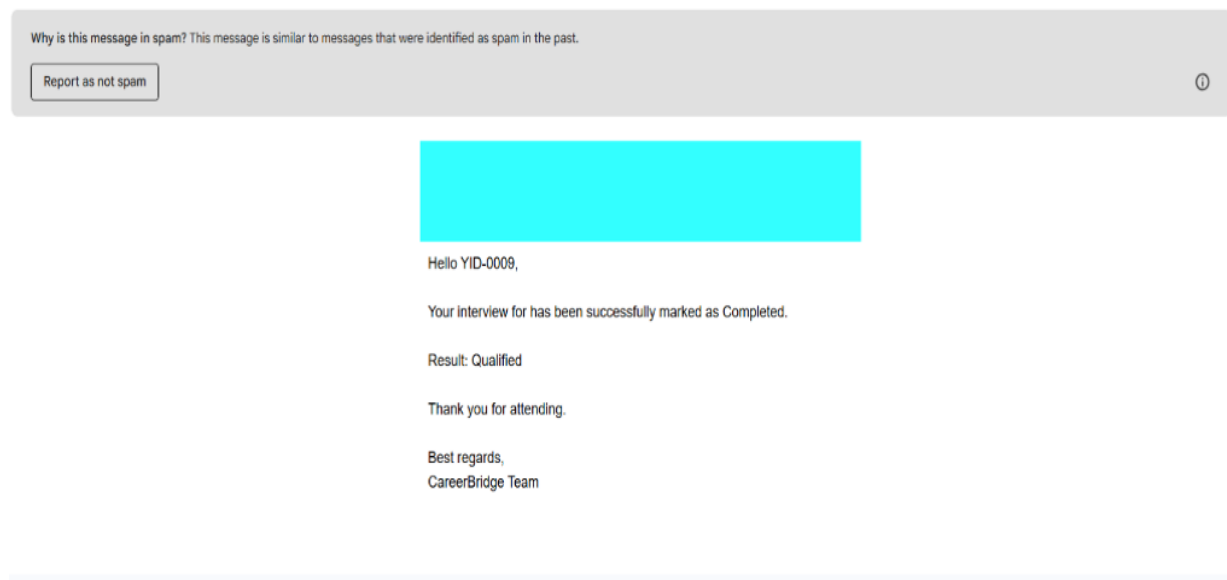
- Job Title = Java Developer, Location = Bangalore, Required Skill = Java.
- **Expected Result:**
 - Job record created in **Job__c** object.
 - System checks youth skills → Match found with Vishnavi.
 - **Interview__c record** auto-created with Status = Scheduled.
 - **Interview Scheduled Email** triggered:
“Hello Vishnavi, congratulations! You are shortlisted for Java Developer.
Interview is scheduled on [Date & Time].”



Test Case 3 – Youth Attends Interview → Interview Completed Email

- **Use Case / Scenario:** Interview is completed.

- **Test Steps (Input):**
 - Coordinator updates Interview record → Change Status__c = Completed.
- **Expected Result:**
 - Status changes to Completed.
 - **Interview Completed Email** triggered:
“Hello Vishnavi, your interview for Java Developer has been marked as Completed. Thank you for attending.”



Test Case 4 – Validation Rules (Error Handling)

- **Use Case / Scenario:** Youth tries to register without Email or Skills.
- **Test Steps (Input):**

- Leave Email blank → Save.
- Leave Location blank → Save.
- **Expected Result:**
 - System displays validation error messages:
 - ISBLANK(Email__c) → “Email is required.”
 - ISBLANK(Location__c) → “Location must be entered.”

The screenshot displays the CareerBridge user interface. At the top, there is a navigation bar with the CareerBridge logo and a search bar. Below this is a sub-navigation bar with tabs for Jobs, JobSkills, Youths, YouthSkills, Skills, Interviews, Dashboards, Reports, Accounts, and Contacts. The main content area shows a form for a Youth record with ID YID-0009. The form includes fields for Email, Location, Qualification, and Preferred Role. The Email field is highlighted with a yellow border, and the Location field contains the text 'Hyderabad'. A red error message box is overlaid on the form, stating 'We hit a snag.' and 'Review the errors on this page.' The error message is associated with the Email field, indicating that the email is required.

Conclusion

The CareerBridge CRM project was successfully implemented and tested across all phases. In Phase 10, Quality Assurance Testing validated that every Salesforce feature—record creation, flows, triggers, automatic emails, and validation rules—worked as expected.

The end-to-end flow ensures:

- Youth registration captures complete details and sends a **Welcome Email**.
- Employers post jobs, and the system automatically matches required skills with youth profiles.
- Matching results trigger interview scheduling along with an **Interview Scheduled Email**.
- Once interviews are completed, coordinators update the status, and the system sends an **Interview Completed Email**.
- Validation rules prevent saving incomplete or incorrect data, ensuring accuracy.

This testing confirms that the platform is:

- **Reliable** – All workflows run without errors.
- **Automated** – Reduces manual work for NGOs and coordinators.
- **User-friendly** – Provides transparency to youth, NGOs, and employers.

Final Outcome: CareerBridge CRM successfully connects youth skills to jobs, supports NGOs in managing opportunities, and helps employers find the right candidates efficiently