

# CS 5200 - Final Report

## On Campus Job Portal

**Group members:** Ravi Shankar Shankara Narayanan, Sibi Vishtan Thirukonda

**Group name:** ThirukondaSSankara NarayananR

An on-campus job portal is a platform that allows students to find part-time employment opportunities within their university or college campus. Here is a project description for an on-campus job portal:

**Project Title:** On-Campus Job Portal

**Project Objective:** To create a user-friendly platform that connects students with on-campus part-time job opportunities and helps employers find qualified candidates.

### **Features:**

1. Job Search: Students can search for job postings based on various criteria, such as job category, pay rate, and location.
2. Job Posting: Employers can post job openings on the platform, including details such as job description, qualifications required, and pay rate.
3. Application Management: Students can easily apply for jobs on the platform and track the status of their applications.
4. Employer Management: Employers can manage their job postings and applications, as well as track the progress of their hiring process.

### **Entities:**

- EmploymentSeekers (students who are seeking on campus employment).
- Address(the address where the student lives, a student can have multiple addresses like permanent and temporary address.)
- Employer ( person/ organization that needs employee students to work)
- HiringTeam ( Team with hiring manager, recruiter who helps the employers to hire students and each team has a portal admin to manage the job posting in the portal)
- JobPostings (the postings done by the hiring team with the requirements from the employer).
- Level\_desc(this table gives us the difficulty level of the job)
- Applications ( applications submitted by student side)
- College ( posting of the job from the college)

## **Pages:**

Our application has various pages, the user can navigate to these pages to perform the desired operation.

### **Student or Employment Seeker Side:**

- Student login page ( authentication page where student have to provide unique username and password in order to access the portal)
- Postings page ( it is the main page and has links to redirect to different pages say posting details, filters to be used with the different categories and classify the data)
- Posting details page ( has details for the posting with all the information say employer, salary and etc. with an apply button in the page)
- Application page ( has the required fields for the student to fill details and upload resume)
- Application tracking page ( has the applications student applied to).

### **Hiring Team Side:**

- Admin team login page ( authentication page where admin team have to provide their unique username and password in order to access the portal).
- Create new application page ( The hiring team will have access to post or create a new application from this page)
- Update status page ( If any updates regarding application would be updated from Hiring Team Side).

## **Tools Used:**

- MYSQL (SQL)
- IDE : Spyder and pycharm
- Operating System : MacOS, Windows

## **Motivation:**

- Convenience for students: By having a job portal on campus, students can easily search and apply for jobs that are available on campus. This saves them time and effort from having to travel off campus to search for employment.
- Access to exclusive opportunities: Many jobs on campus are only available to students, and having a job portal can make it easier for students to find and apply to these exclusive opportunities.

- Integration with campus resources: An on-campus job portal can be integrated with other campus resources, such as career services and academic departments, to provide students with a more comprehensive job search experience.
- Cost-effective for employers: Posting jobs on an on-campus job portal can be a cost-effective way for employers to recruit students. This can result in more job opportunities being available to students.
- Increased engagement and retention: By providing students with access to on-campus job opportunities, they may be more likely to feel engaged and invested in the campus community. This can lead to increased retention rates for the institution.

### **Why does this project domain interests us ?**

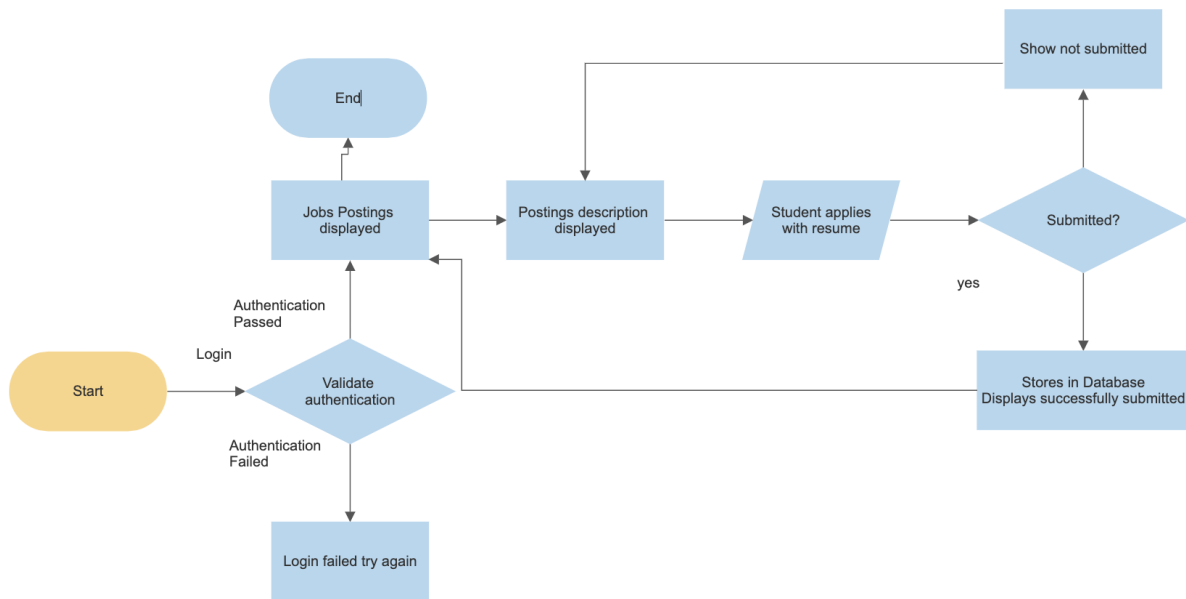
Many students rely on part-time jobs to support themselves while attending college or university. An on-campus job portal provides a convenient and centralized platform for students to find employment opportunities within their campus community.

An on-campus job portal project can be customized to fit the specific needs and requirements of a particular campus community. This offers an opportunity to tailor the project to the unique needs of the community.

### **Why SQL storage ?**

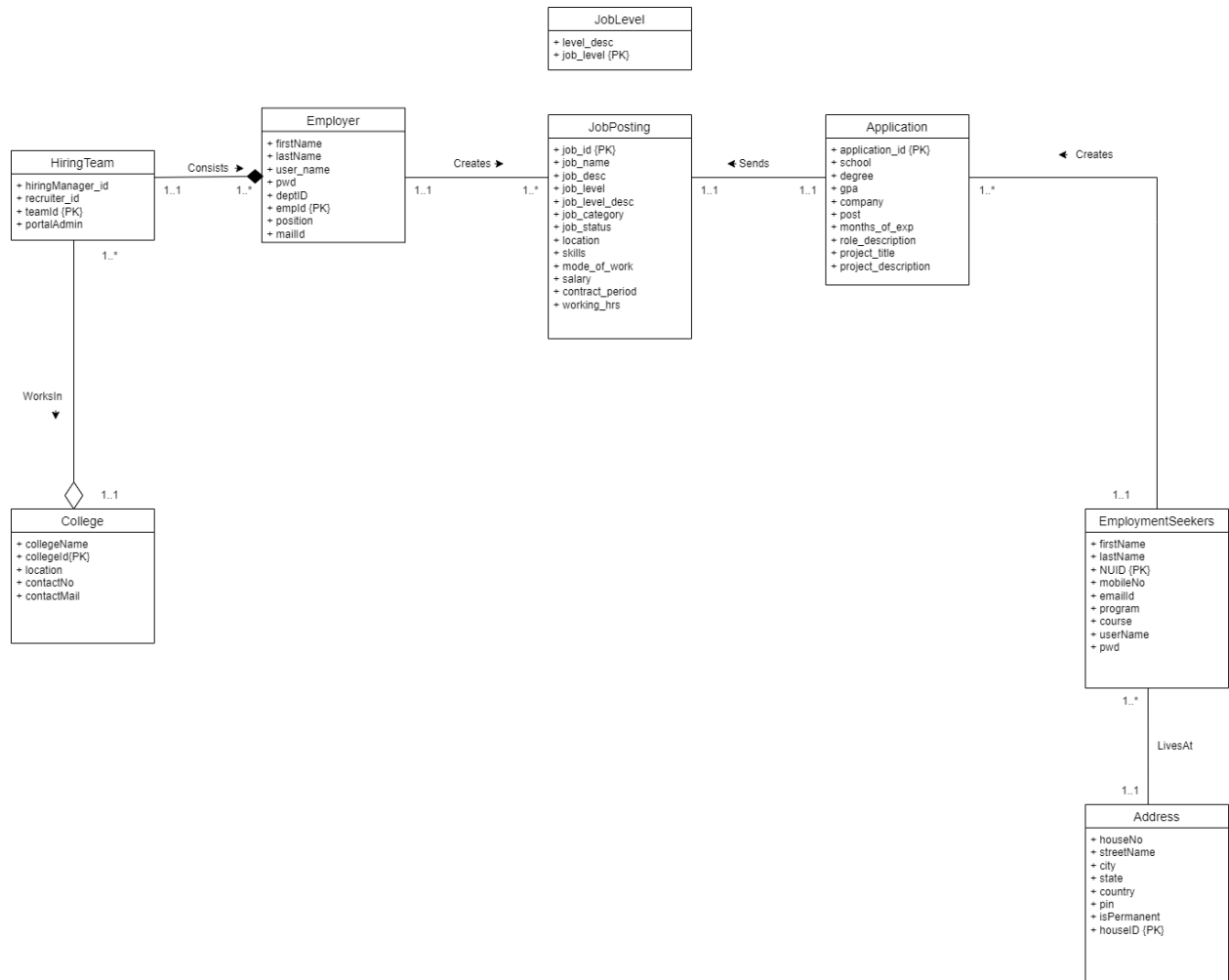
- SQL has a simple syntax that is easy to understand and use, making it accessible to beginners and experts alike.
- SQL can be integrated with other programming languages, such as Python and Java, to enhance its functionality and provide more robust solutions.

## Flow chart:

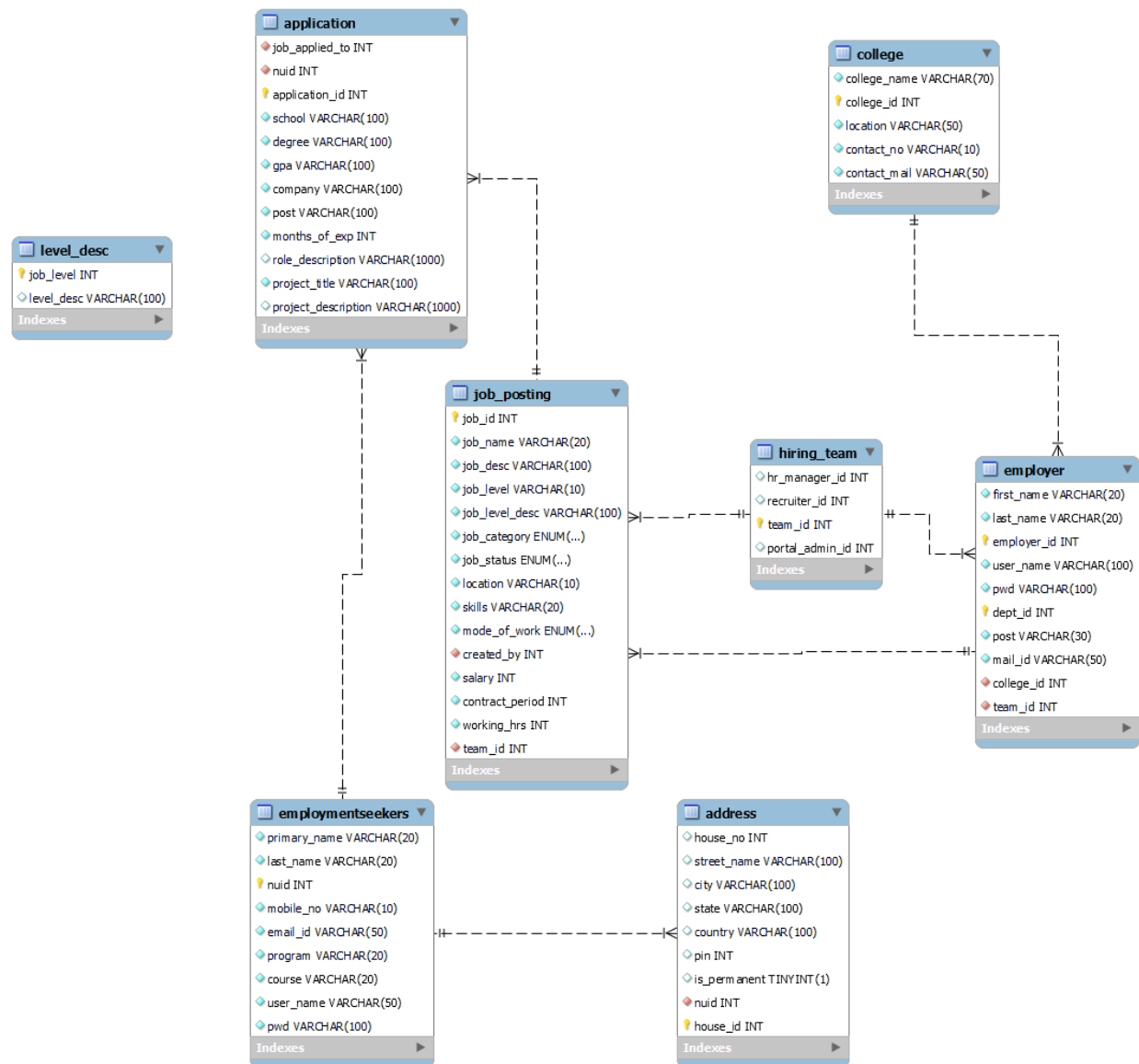


## UML diagram:

- One employment seeker will apply for more than 1 position.
- Each hiring team is associated with a college.
- One hiring team can have multiple employers(admin) and each employer can post multiple job posting



## Reverse Engineered Database:



## Packages:

- random
- MySQL connector

## **Tables Created:**

### **Tables Created:**

#### **1. employmentseekers**

- Primary\_name
- Last\_name
- Nuid
- Mobile\_no
- Email\_id
- Program
- Course
- User\_name
- pwd

#### **2. application**

- Nuid
- job\_id
- Job\_id
- School
- Degree
- Gpa
- Company
- Role
- Months\_of\_exp
- Role\_description
- Project\_title
- Project\_description
- Application\_id

#### **3. job\_posting**

- Job\_id
- Job\_name
- Job\_desc
- Job\_category
- Job\_status
- Location
- Skills
- Mode\_of\_work

- Created\_by
- Salary
- Contract\_period
- Working\_hrs
- team\_id

**4. employer**

- First\_name
- Last\_name
- Employer\_id
- User\_name
- Pwd
- Dept\_id
- Post
- Mail\_id
- College\_id
- team\_id

**5. college**

- College\_name
- College\_id
- Location
- Contact\_no
- contact\_mail

**6. address**

- House\_no
- Street\_no
- City
- State
- Country
- Pin
- Is\_permanant
- Nuid
- house\_id

**7. level\_desc**

- Job\_level
- level\_desc

**8. Hiring team**

- Hr\_manager\_id



- Recruiter\_id
- Team\_id
- portal\_admin\_id

### **CRUD operations covered:**

In this project we covered all four CRUD operations

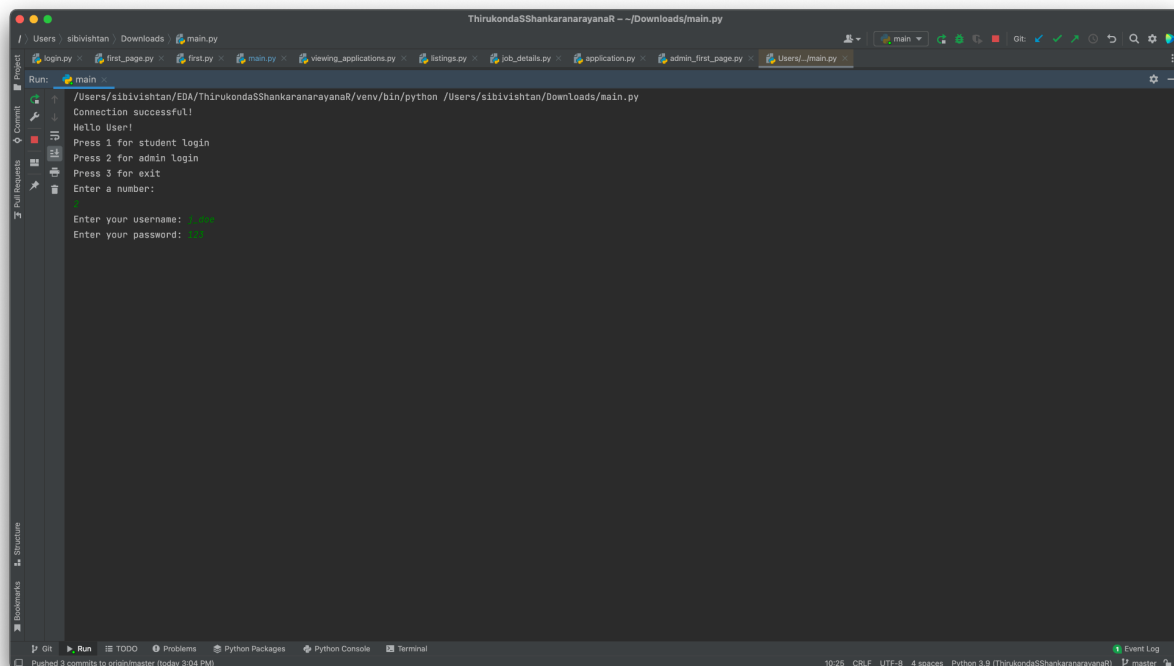
Create: The Admin creates a new job opening and the student or employment seeker applies for that job by creating a new application

Read: The job created by the admin is read and displayed when the students views the available jobs.

Update: The admin can update the job status from open to closed when a suitable candidate is found.

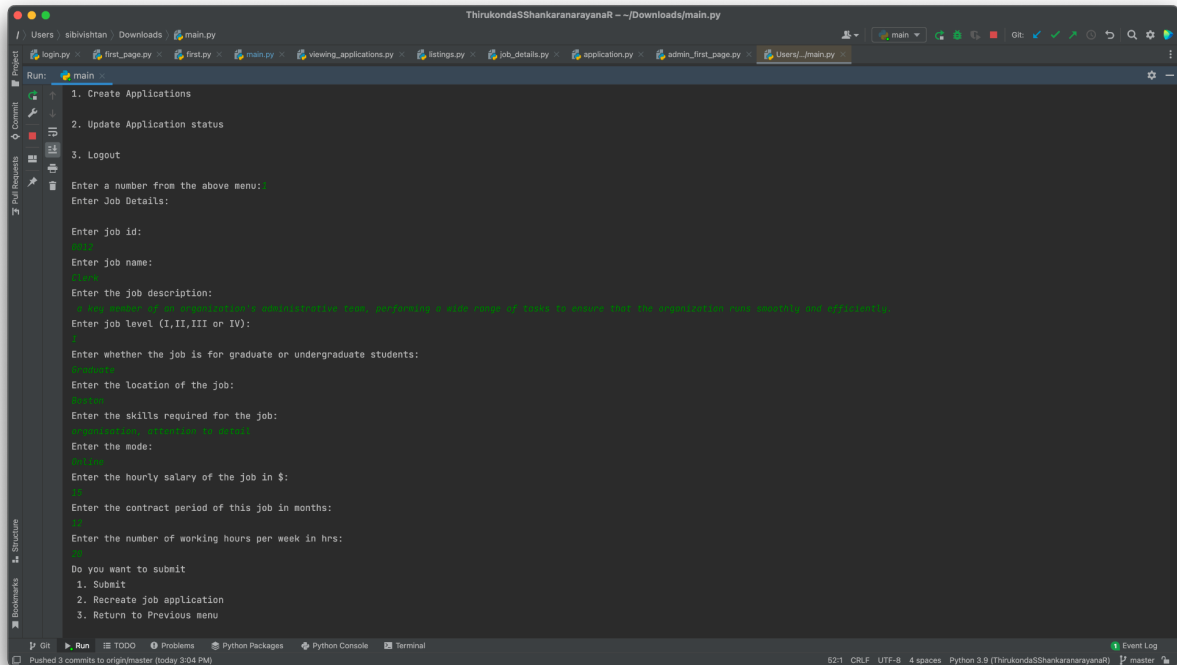
Delete: The student can withdraw (delete an job application) whenever needed.

### **Admin Side Execution:**



```
ThirukondaSSHankaranarayanaR -- ~/Downloads/main.py
/Users/sibivishtan/Downloads/main.py
Run: main
/Users/sibivishtan/EDA/ThirukondaSSHankaranarayanaR/venv/bin/python /Users/sibivishtan/Downloads/main.py
Connection successful!
Hello User!
Press 1 for student login
Press 2 for admin login
Press 3 for exit
Enter a number:
Enter your username: j_doe
Enter your password: 123
```

Admin login is first selected, which prompts the user to input. Here,username j\_doe with the password 123 is already inserted in order to give access to the admin operations to execute.



```
ThirukondaShankaranarayananR - ~/Downloads/main.py
Users | sibivishan | Downloads | main.py
login.py | first_page.py | first.py | main.py | viewing_applications.py | listings.py | job_details.py | application.py | admin_first_page.py | Users/_main.py
Run: main
1. Create Applications
2. Update Application status
3. Logout
Enter a number from the above menu:
1
Enter Job Details:
Enter job id:
12345
Enter job name:
Software Engineer
Enter the job description:
Software Engineer - responsible for developing and maintaining the company's software products
Enter job level (I,II,III or IV):
I
Enter whether the job is for graduate or undergraduate students:
graduate
Enter the location of the job:
Remote
Enter the skills required for the job:
Programming, Problem Solving
Enter the mode:
Full-time
Enter the hourly salary of the job in $:
50000
Enter the contract period of this job in months:
12
Enter the number of working hours per week in hrs:
40
Do you want to submit
1. Submit
2. Recreate job application
3. Return to Previous menu
```

Once logged in it will ask whether to **create** a new application or update status of application. When the admin user chooses to create a new application he has to enter all the necessary details for the job. Once it is done, the admin has options to submit the created job opening to server, or can recreate a job opening or return to previous menu.



And if the admin wants to close an existing job opening, admin can select option 2. This will prompt the user to enter the job id to update. Then a procedure will be called to update the status from open to closed.

1 • **SELECT \* FROM oncampus.job\_posting;**

100% 1:1

**Result Grid** Filter Rows: Search Edit: Export/Import:

	job_id	job_name	job_desc	job_level	job_level_desc	job_category	job_status	location	skills	mode_of_work	created_by	salary	contract_peri...	working_hrs	team_id
12	clerk	clerk	I	Easy Job	Graduate	Closed	Boston	organized	Online	1	15	12	20	2	

job\_posting 1 Apply Revert

**Action Output**

	Time	Action	Response	Duration / Fetch Time
524	22:37:58	INSERT INTO hiring_team(hr_manager_id,recruiter_id,team_id,portal_admin_id) VALUES ( 1000, 0001, 1, 121);	(2000, 000...	3 row(s) affected Records: 3 Duplicates: 0... 0.00044 sec
525	22:37:58	INSERT INTO employer(first_name, last_name, employer_id, user_name, pwd, dept_id, post, mail_id, college_id, team_id) V...	3 row(s) affected Records: 3 Duplicates: 0...	0.00043 sec
526	22:37:58	INSERT INTO employmentseekers(primary_name, last_name, nuid, mobile_no, email_id, program, course, user_name, pwd) VALU...	3 row(s) affected Records: 3 Duplicates: 0...	0.00046 sec
527	22:37:58	INSERT INTO address(house_no, street_name, city, state, country, pin, is_permanent, nuid, house_id) VALUES (6, 'Cardington Str...	3 row(s) affected Records: 3 Duplicates: 0...	0.00043 sec
528	22:37:58	INSERT INTO level_desc(job_level, level_desc) VALUES (1, 'level 1 Salary Range \$10-\$15'), (2, 'level 2 Salary Range \$12-...	5 row(s) affected Records: 5 Duplicates: 0...	0.00032 sec
529	23:00:02	SELECT * FROM oncampus.job_posting LIMIT 0, 1000	1 row(s) returned	0.0011 sec / 0.000011...

Query Completed

In the screenshot above, the status of the job has been closed.

## Employmentseeker side Execution:

Hello User!

Press 1 for student login

Press 2 for admin login

Press 3 for exit

Enter a number: 1

Enter your username:

The user can navigate back to the homepage by following the instructions on the prompt. TO login as student , we press 1. A username and password is already inserted into the employmentseeker table. Use username: jackie and pwd: 987 to log in as student.

```
Enter a number: 1
Enter your username: jackie
Enter your password: 987
Login successful!

Welcome jackie!

1. View Listings
2. Withdraw your Applications
3. logout
Enter an option:
```

A welcome message along with the options available to the employment seekers will be displayed.

```
Enter an option: 1
(1, 'Clerk')
Enter a job ID to view details OR Enter 0 to go back: 1
Job ID: 1
Job Name: Clerk
Job Description: Clerk Position
Job Level: II
Job Category Level 2 Salary Range $12-$16
Job Location Graduate
Job Skills Open
Mode of Work Boston
Posted by Organization
Salary Offline
level of work 1
Working hours 15

Would you like to
1.Apply
2.Go Back
Option:
```

To apply for a job, the employment seeker must select 1 to view the job opening , all jobs ever created by admin will be displayed. Only the job id and job name will be displayed. The user will then enter the job id of the desired job to view it's details.

After displaying the details, the user is yet again prompted with two options: For applying or to go back.

```

title of project : ML
Description of Project : Find inference
Review your responses :
Application ID: 919
Job ID: 1
NUID: 123456789
School: Khoury
Degree: CS
GPA: 4

Company: Deloitte
Role: Analyst
Months of Experience: 18
Role Description: Automate data readings

Projects
Project title ML
Project description Find inference

Do you want to submit
1. Submit
2. Go Back
Option:

```

On selecting on apply option, the user will be asked to enter basic details. Give short descriptions for each questions. After all details are filled, the user can review the details entered and select submit option.

```

Do you want to submit
1. Submit
2. Go Back
Option:1

Submitted successfully.

Press any key to return to previous menu :

```

The application details are now inserted into the application table into the database

The screenshot shows a database application window with a tab labeled 'application'. The SQL query entered is 'SELECT \* FROM oncampus.application;'. The results are displayed in a 'Result Grid' table with 12 columns and 1 data row. The columns are: job\_applied\_to, nuid, application\_id, school, degree, gpa, company, post, months\_of\_exp, role\_description, project\_title, and project\_description. The data row contains the values: 1, 123456789, 919, Khoury, CS, 4, Deloitte, Analyst, 18, Automate data readings, ML, and Find inference. The table has a light blue header and alternating row colors. On the right side of the window, there are buttons for 'Result Grid', 'Form Editor', and a search icon.

job_applied_to	nuid	application_id	school	degree	gpa	company	post	months_of_exp	role_description	project_title	project_description
1	123456789	919	Khoury	CS	4	Deloitte	Analyst	18	Automate data readings	ML	Find inference

To withdraw the application the user must select go back option till he reaches the page with the options for student to apply and withdraw applications.

```

Welcome jackie!

1. View Listings

2. Withdraw your Applications

3. logout

Enter an option: |

```

From this page, the user can withdraw an application by entering 2.

```

Active Applications

Job_id, NUID, Application id, School, Degree, GPA, Company, Post, Months of Exp, Role Description, Project title, Project Description

(1, 123456789, 919, 'Khoury', 'CS', '4', 'Deloitte', 'Analyst', 18, 'Automate data readings', 'ML', 'Find inference')

1. Withdraw any applications
2. Go Back
Selection:

```

On entering 2 all active applications for that user will be displayed.

```

1. Withdraw any applications
2. Go Back
Selection: 1

Enter the application ID to Withdraw : 919

Application Withdrawn successfully

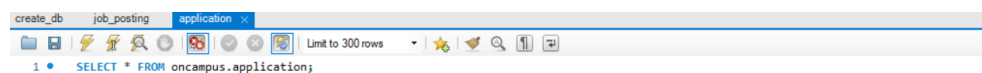
Active Applications

Job_id, NUID, Application id, School, Degree, GPA, Company, Post, Months of Exp, Role Description, Project title, Project Description

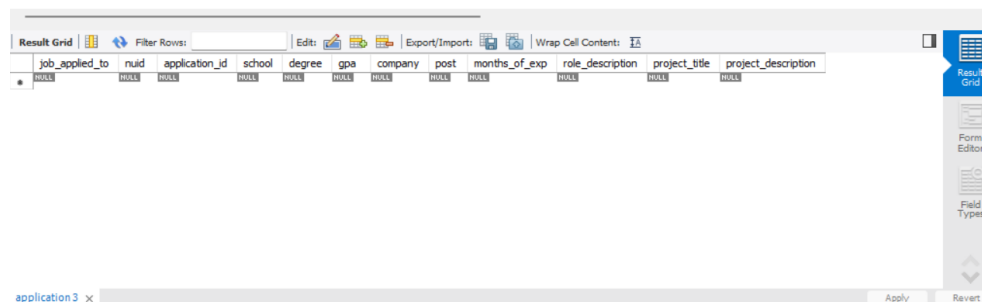
1. Withdraw any applications
2. Go Back
Selection:

```

Enter the application you want to delete in the prompt.



A screenshot of a terminal window with three tabs: 'create\_db', 'job\_posting', and 'application'. The 'application' tab is active. The terminal shows a SQL query: `SELECT * FROM oncampus.application;` The query is preceded by a prompt `1 •`. The terminal interface includes standard icons for file operations and a 'Limit to 300 rows' dropdown.



A screenshot of a database application interface. The top section shows a 'Result Grid' with columns: 'job\_applied\_to', 'nuid', 'application\_id', 'school', 'degree', 'gpa', 'company', 'post', 'months\_of\_exp', 'role\_description', 'project\_title', and 'project\_description'. Each column has a 'NULL' value in the first row. The interface includes a 'Filter Rows' section, an 'Edit' button, and an 'Export/Import' button. On the right side, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom, there is a status bar with 'application3 x', 'Apply', and 'Revert' buttons.

From the above screenshot it is evident that the application has been withdrawn.

### **Conclusion:**

The on-campus job portal provides a convenient way for students to find part-time jobs on campus, and for employers to find qualified candidates. It helps to streamline the hiring process and can be used as a tool for building community within the campus.

### **How to run the program:**

This project is executed in terminal using CLI, make sure you have python installed in your computer and link python to environment variable so that python program can be executed from terminal.

(to install python follow this link : <https://www.python.org/downloads/>).

As the modules will be mostly available in basic python there is no need for separate installation of dependencies. But if any issues faced.

Try :

- Pip install os
- Pip install random

Then open a terminal or powershell from the folder where the main.py program is located, and run'

**\$ python .\main.py**

Once done it will open the program with greetings. Follow the prompts correctly to execute the program.

### **Note :**

- Should not give extra values while running in the program the parameter values or it would produce an error.
- Should not give any mismatching datatype to the passing parameter values.
- We have functions to restrict the user from entering wrong values but it is recommended to follow the prompt precisely for a hassle-free experience.