



# Documentation



## Resume Ranker

Web site created using create-react-app

R <https://ccibt9.me>

The Job Finder is a web app that allows recruiters to post multiple job openings and correspondingly rank resumes of the applicants. The applicants can apply on the job portal and get further insights through the dynamic job cards available therein.

The Recruiter has a general as well as a job-specific dashboard, through which they can get deeper insights into the real-time statistical data of the applicants.

## ▼ Features

- Rank based on cosine similarity
- Ability to rank based on the following criteria:
  - Domain experience, Number of job switches, Organizations worked for, Keyword matches, Going by academic document profiles
- The ability to generate reports with custom filters
- Search engine to display details based on criteria provided

## Resources

Aa	Name
	<a href="#"><u>Quick Start</u></a>
	<a href="#"><u>Technical Specs</u></a>
	<a href="#"><u>Metrics and Code Quality</u></a>
	<a href="#"><u>API Documentation</u></a>
	<a href="#"><u>Client Documentation</u></a>
	<a href="#"><u>Ranking Algorithm</u></a>
	<a href="#"><u>Operational Handbook</u></a>



# Quick Start

## Using Docker

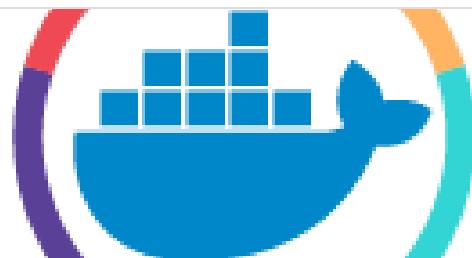
For convenience, the project is shipped with a `docker-compose` file which is used by Docker's compose tool for container orchestration. This file contains a specification to get all components of the project up and running using a single command.

- Make sure you have Docker installed. If not, install Docker using the instructions for your platform found [here](#).

### Install Docker Engine

Docker Desktop for Linux Docker Desktop helps you build, share, and run containers easily on Mac and Windows as you do on Linux. We are excited to share that Docker Desktop for Linux is now GA.

 <https://docs.docker.com/engine/install/>



- Clone the repository.

```
$ git clone https://github.com/RameshSankarS/TEAM-B4  
$ cd TEAM-B4
```

- Start the containers. (The `-d` flag is to run the containers in the background.)

```
$ docker compose up --build -d
```

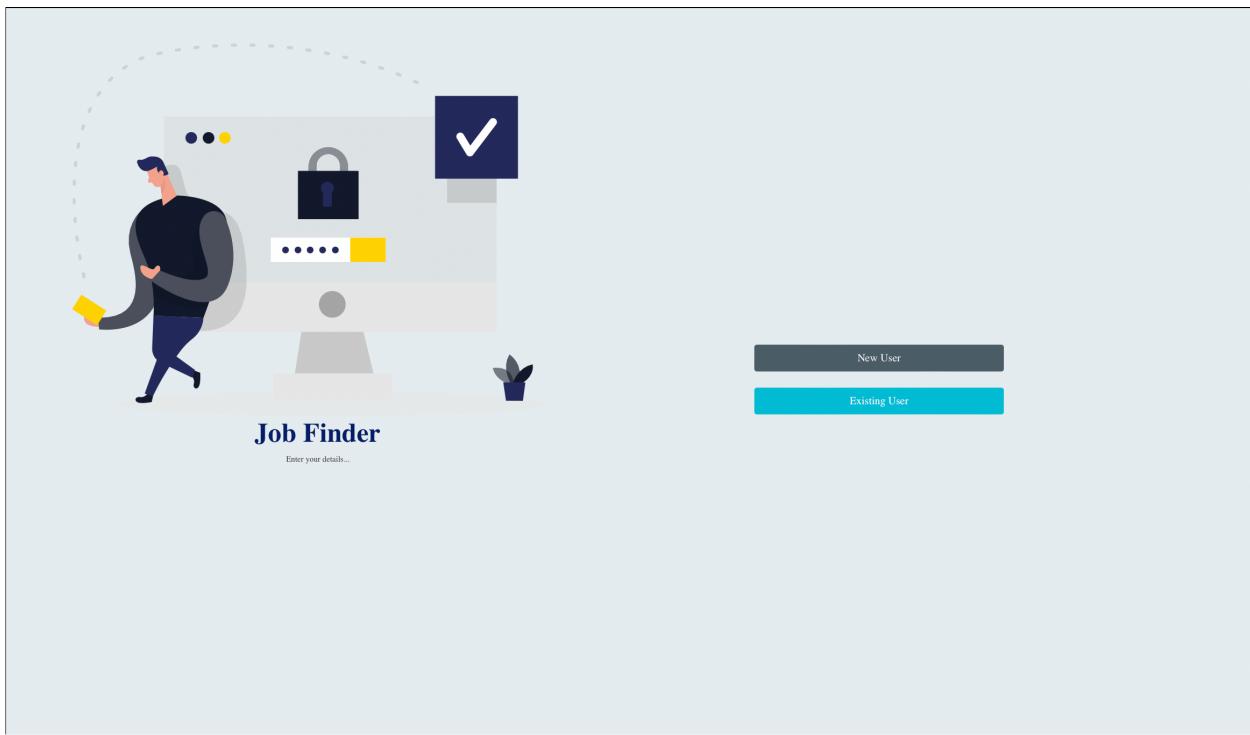
- Wait for the client to build.

```
$ docker compose logs client -f
```

You should get logs similar to this:

```
...
ranking_algo_client | File sizes after gzip:
ranking_algo_client |
ranking_algo_client |   397.95 kB  build/static/js/main.3154da2d.js
ranking_algo_client |   19.86 kB   build/static/css/main.f0be168c.css
ranking_algo_client |
ranking_algo_client | The project was built assuming it is hosted at /.
ranking_algo_client | You can control this with the homepage field in your package.json.
ranking_algo_client |
ranking_algo_client | The build folder is ready to be deployed.
ranking_algo_client | You may serve it with a static server:
ranking_algo_client |
ranking_algo_client |   npm install -g serve
ranking_algo_client |   serve -s build
ranking_algo_client |
ranking_algo_client | Find out more about deployment here:
ranking_algo_client |
ranking_algo_client |   https://cra.link/deployment
ranking_algo_client |
```

- Once the client is built, you are good to go. Open <https://localhost:1337/> in a browser. You should be greeted with a page that looks like this



You're good to go. Play around with the interface and try to break things :)



# Technical Specs

The project is divided into three main components.

- The Server
- The Client
- The ML models

## The Server

The server mainly depends on:

- NodeJS (18.0.0)
- ExpressJS (express@4.18.1)

## The Client

The client is a web app written using:

- NodeJS (v18.0.0)
- ReactJS (react@18.2.0)

## The ML models

The ML models are written in Python using PyTorch and exposed by Celery workers.

- Python (3.10)
- celery (5.2.7)
- numpy (1.22.3)
- pandas (1.4.2)

- PyTorch (1.11.0)
- HuggingFace Transformers (4.21.)
- nltk (3.7)
- spacy (3.3)

## Other Services

The project uses MongoDB as it's primary data source. RabbitMQ is used to enable communication between the server and the celery workers.

- MongoDB (5.0)
- RabbitMQ (3.7.28)



# Metrics and Code Quality

## Code Formatting and Linting

Code quality is important. To ensure standardisation of the code and meeting quality metrics throughout the project we've used linters and formatters.

Language	Tools
Python	- flake8 (linter) - black (code formatter)
JavaScript	- JSLint (linter) - prettier (code formatter)

These tools are registered as `pre-commit` hooks to the git repository. Everytime a patch is added to the code, the hooks are fired and the code is formatted and linted by appropriate tools.

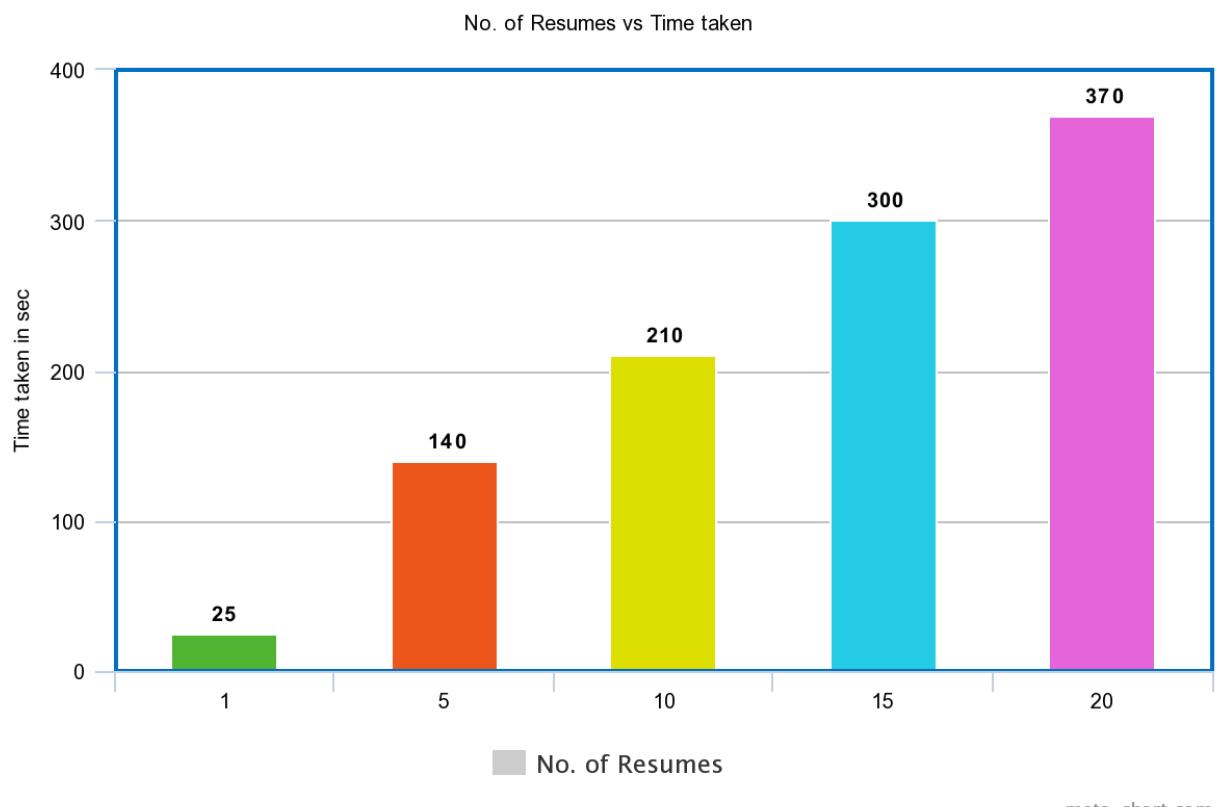
## Code Testing

To ensure the code works under all\* conditions, we've written an exhaustive test suite consisting of unit tests for all components.

Language	Tools
Python	- unittest
JavaScript	- jest

## Load Metrics

No. of Resumes Vs Time taken in sec



## Results

Accuracy of the Spacy Named Entity Recognition model fine-tuned with Roberta Transformers is 96%

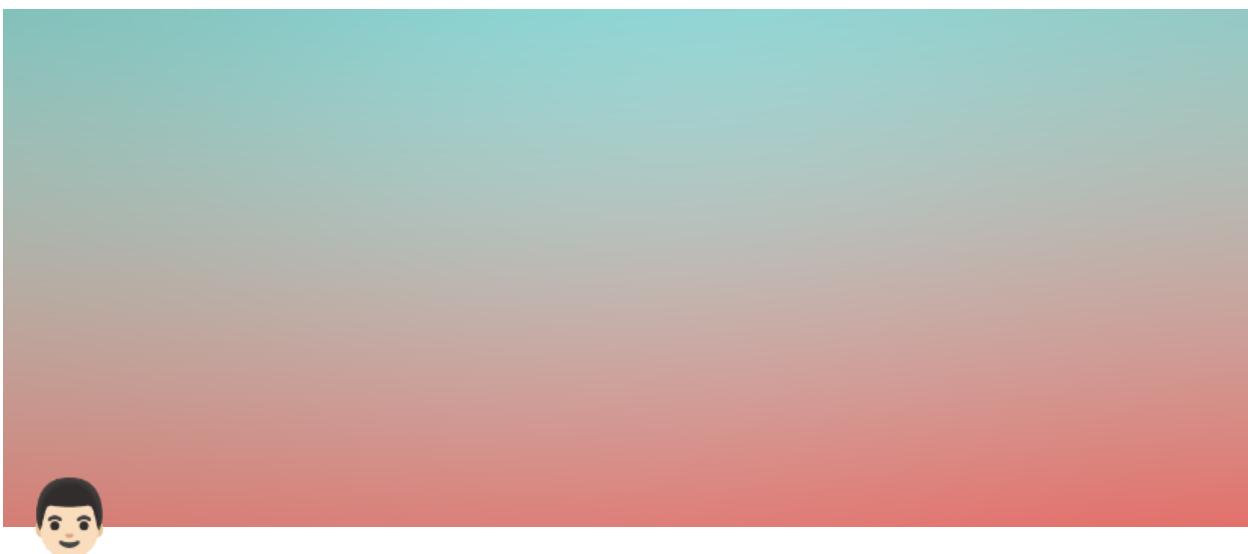
All the testcases were passed in the ML side and WebDev side too



# API Documentation

The server is documented using a collection on Postman. The collection is live and gets updated with every API change.

<https://www.getpostman.com/collections/70d28fddf08813da291c>



# Client Documentation

## Description

A web app that allows recruiters to post multiple job openings and correspondingly rank resumes of the applicants. The applicants can apply on the job portal and get further insights through the dynamic job cards available therein.

The Recruiter has a General as well as a job-specific dashboard, through which they can get deeper insights into the real-time statistical data of the applicants.

## Code Overview

Code structure and the overview of the client can be identified at

```
https://github.com/RameshSankarS/TEAM-B4/tree/main/client
```

# Testing Results

## Client testing coverage

### All files

60.85% Statements 698/1147 50.14% Branches 335/678 54.39% Functions 167/307 62.27% Lines 600/1092

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲	Statements	Branches	Functions	Lines
src	100%	7/7	100%	0/0
src/components	0%	0/0	0%	0/0
src/components/Appliedcards	94.11%	16/17	75%	6/6
src/components/CalculateTime	82.35%	14/17	70%	1/1
src/components/Createdcards	100%	5/5	100%	2/2
src/components/Dashboard/ApplicantTable	76.92%	20/26	83.33%	5/10
src/components/Dashboard/chart	100%	4/4	100%	1/1
src/components/Dashboard/chart/BarChart	100%	12/12	100%	4/4
src/components/Dashboard/chart/FunctionalLineChart	100%	10/10	83.33%	3/3
src/components/Dashboard/chart/MainGraph	100%	28/28	97.72%	1/1
src/components/Dashboard/chart/PieChart	100%	3/3	100%	1/1
src/components/Dashboard/chart/ScatterChart	100%	4/4	100%	1/1

## Server testing coverage

### All files

85.57% Statements 860/1005 56.7% Branches 110/194 79.85% Functions 111/139 86.22% Lines 845/980

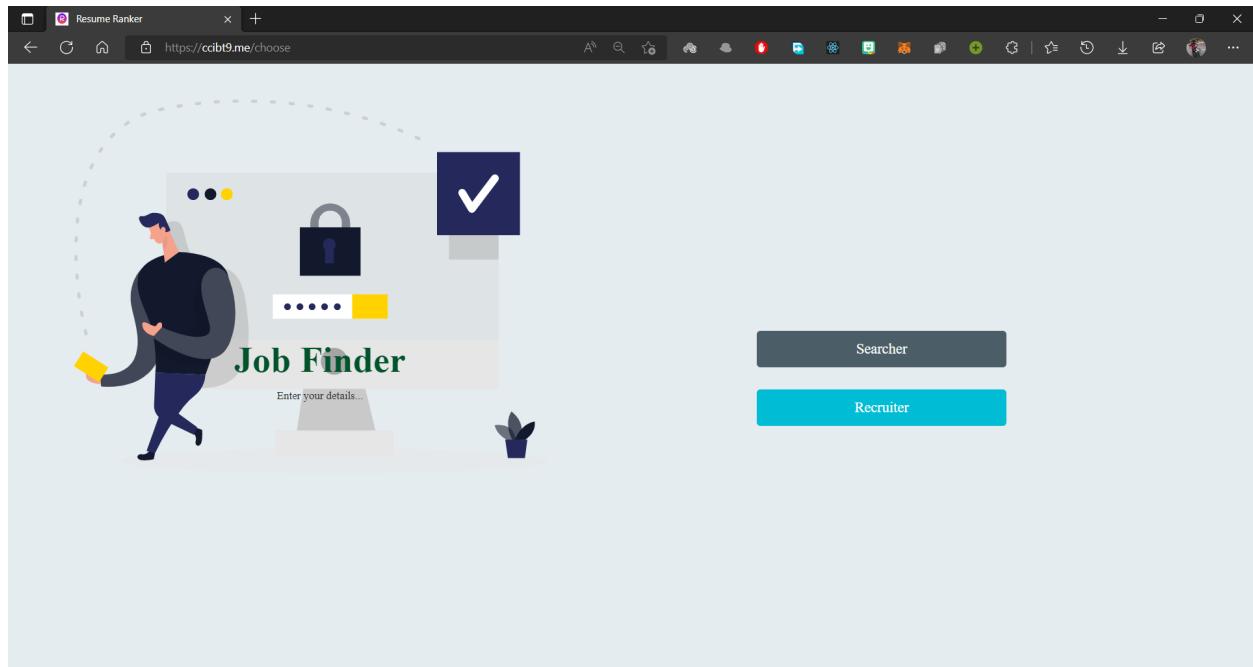
Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

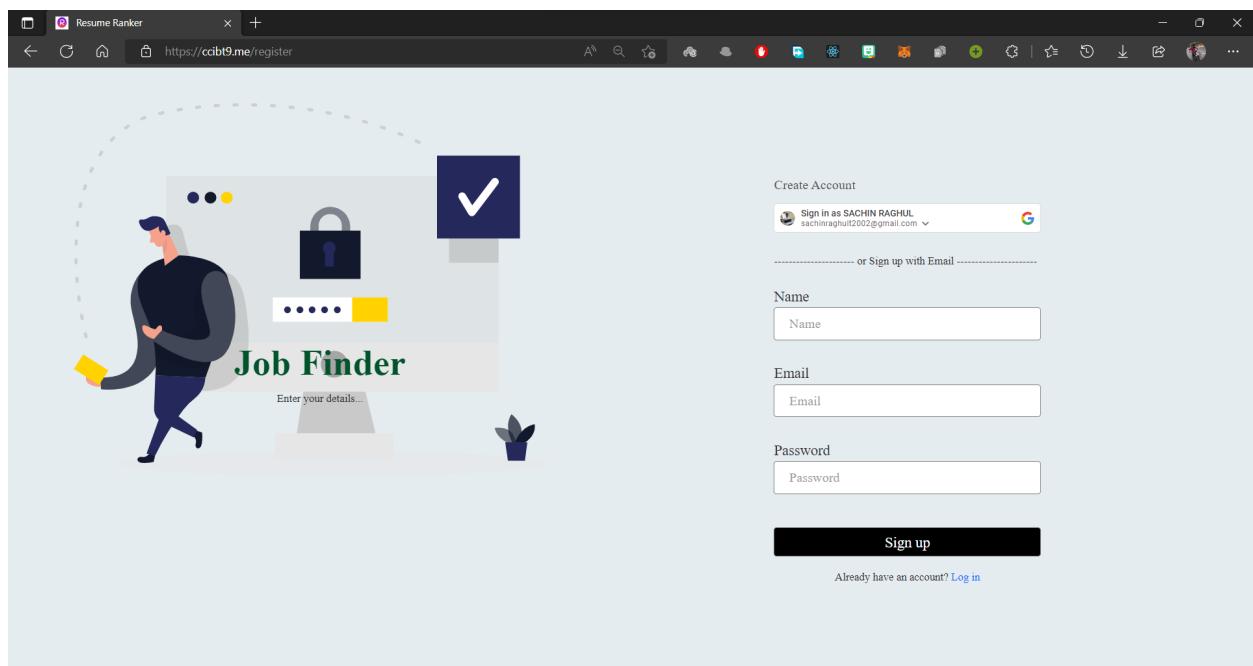
File ▲	Statements	Branches	Functions	Lines
src	97.36%	37/38	100%	0/1
src/api	85.2%	691/811	50.69%	85.6%
src/celery	100%	6/6	100%	6/6
src/data	100%	6/6	100%	6/6
src/db	93.33%	14/15	50%	93.33%
src/file_upload	63.63%	7/11	50%	63.63%
src/file_upload/drive	38.88%	7/18	100%	38.88%
src/log	100%	11/11	62.5%	11/11
src/middleware	76.92%	20/26	66.66%	20/24
src/models	96.82%	61/63	72.22%	59/59

# HOME

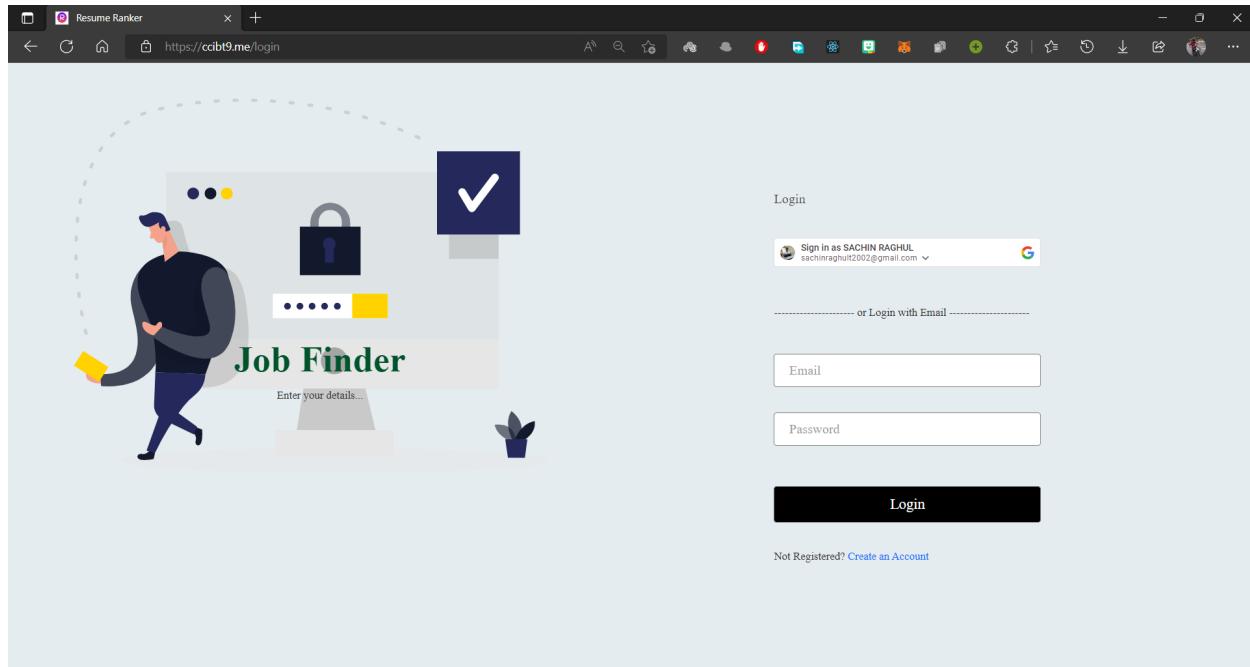
## Welcome



## Recruiter

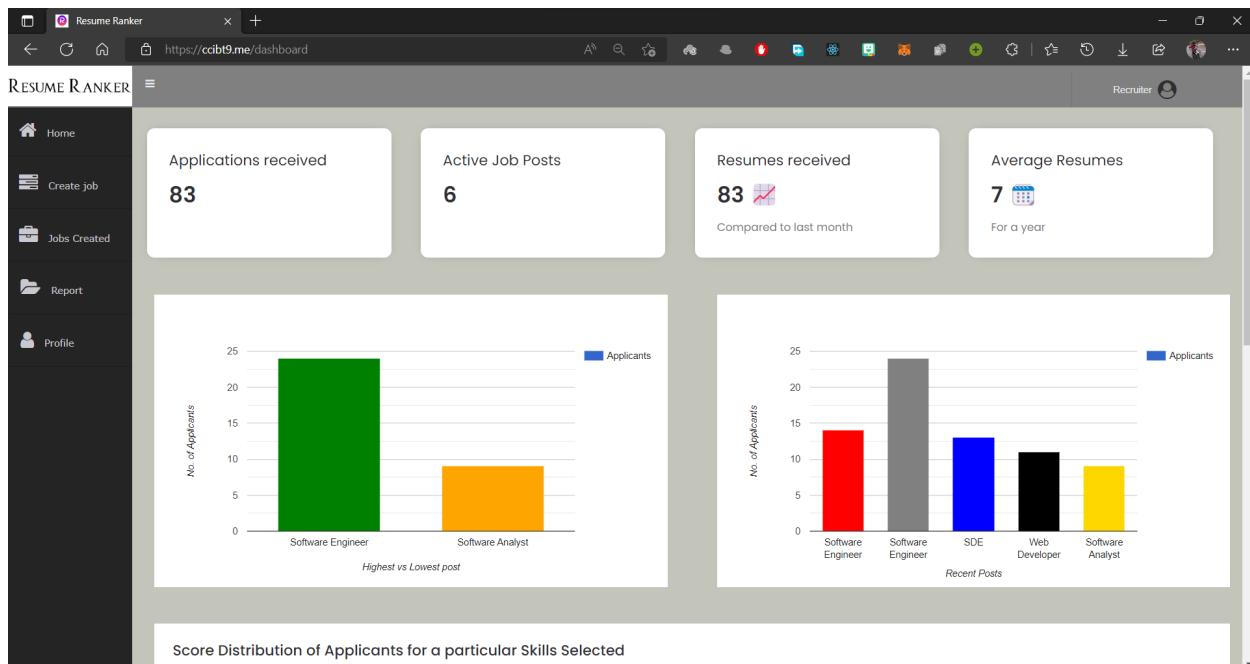


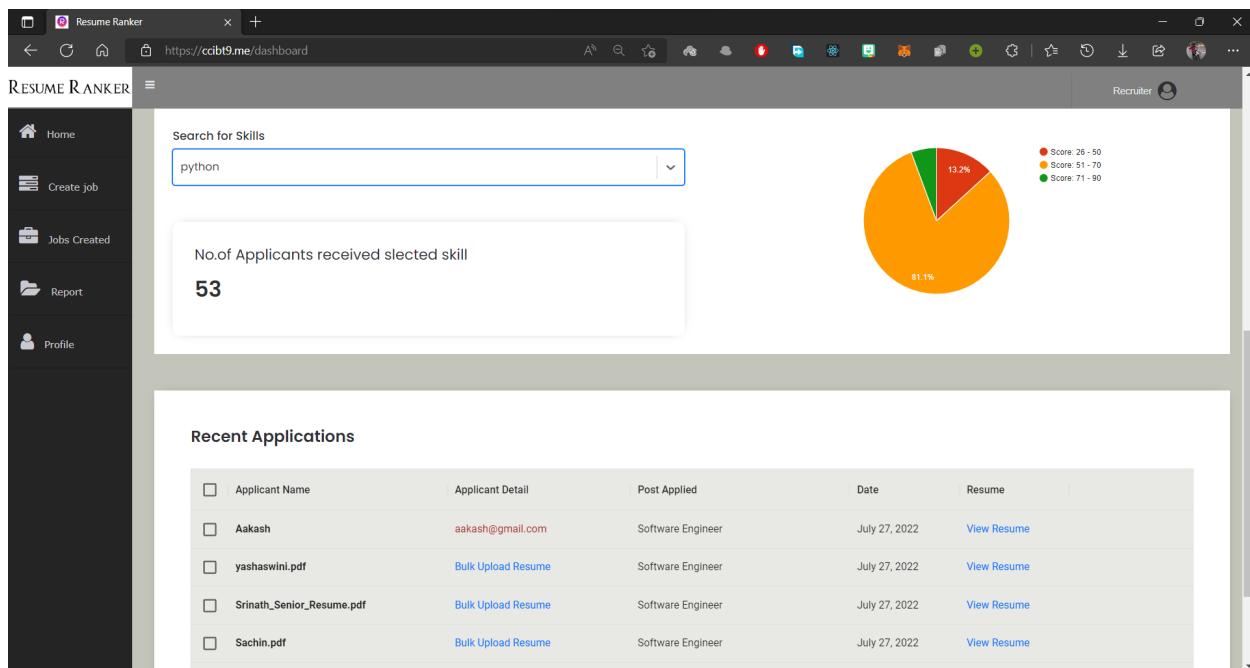
# Login



# RECRUITER

## Landing Dashboard





## Job Creation Form

The screenshot shows the Job Creation Form at <https://ccbt9.me/create-job-post>. The left sidebar includes Home, Create job, Jobs Created, Report, and Profile. The main form is titled 'APPLICATION FORM' and contains the following fields:

- \*Job Title: Job Title input field
- \*Job desc: Job Description input field
- \*Experience: Qualified Experience input field
- \*Keywords: Keyword(s) input field containing 'keyword' and an 'Add a Keyword' button
- \*Skills: Skill(s) input field and Priority(s) input field (0 - 100) with an 'Add a Skill' button
- Select Tags: Full Time and Part Time checkboxes

On the right side of the form, there is a large graphic with the text 'JOB VACANCY' and a red arrow pointing downwards.

## Jobs Created

The screenshot shows the 'Your Jobs' section of the Resume Ranker application. On the left is a sidebar with navigation links: Home, Create job, Jobs Created (which is selected and highlighted in blue), Report, and Profile. The main area is titled 'Your Jobs' and contains a heading 'Active Job Posts'. There are five job post cards displayed:

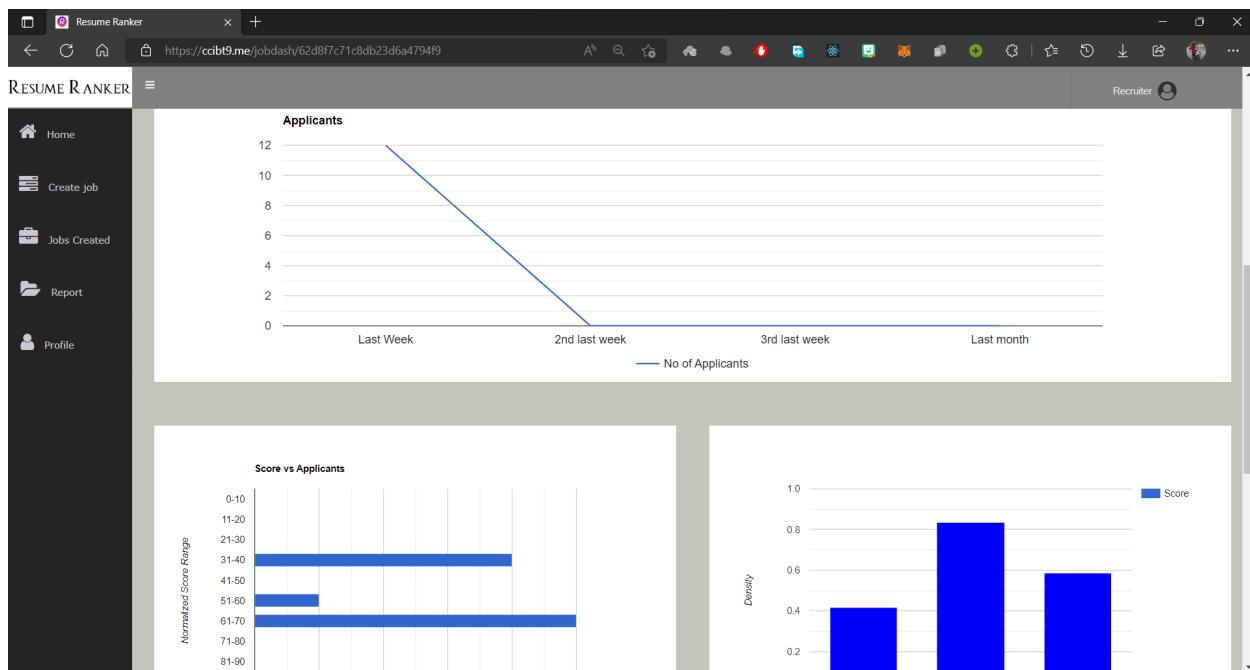
- Software Engineer** (July 27, 2022) - Preview image shows a person at a desk. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Participants in a formal... Buttons: VIEW DASHBOARD, RANKED RESUMES.
- Software Engineer** (July 25, 2022) - Preview image shows a person running with a briefcase. Description: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Buttons: VIEW DASHBOARD, RANKED RESUMES.
- SDE** (July 25, 2022) - Preview image shows a yellow sign that says 'WE ARE HIRING'. Description: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Buttons: VIEW DASHBOARD, RANKED RESUMES.
- Web Developer** (July 22, 2022) - Preview image shows a person at a desk. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Buttons: VIEW DASHBOARD, RANKED RESUMES.
- Software Analyst** (July 21, 2022) - Preview image shows a person at a desk. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Buttons: VIEW DASHBOARD, RANKED RESUMES.

## Job Specific Dashboard

The screenshot shows the 'Job Specific Dashboard' for a 'Software Engineer' position. The top right corner has a 'Bulk Resume Upload' button. The dashboard includes the following information:

- Software Engineer** - created at 21-07-2022 and passes deadline by 31-07-2022
- Description**: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs.
- Skills Required**: Java, Python, c++
- Number of Applications**: 12 applicant received till now
- Status of Job post**: Active

On the right side of the dashboard is a recruitment advertisement with a magnifying glass over a resume icon and the text 'WE ARE HIRING'.



## Bulk Upload Resume

The screenshot shows the 'Bulk Upload Resume' page for a 'Software Engineer' position, created on July 27, 2022. The left sidebar is identical to the dashboard. The main content includes: 1) A job thumbnail with the text 'WE ARE HIRING'. 2) Job status buttons: 'Internship' and 'Full Time'. 3) A 'Deadline' box showing 'Sunday 31 JULY 2022'. 4) A 'Description' section with a paragraph about Wells Fargo job vacancies. 5) A 'Qualified Experience' section with a detailed bullet-pointed list of requirements. 6) A 'Keywords' section at the bottom.

The screenshot shows the 'Skills Required' section with the following data:

Skill	Count
Java	25
Python	20
ML	15
Web Dev	20
DBMS	18

The 'Number of Applications' section indicates 14 applications received till now.

An 'Upload Bulk resume' form is shown, with a file input field containing 'resumes.zip' and a 'Browse' button. A reCAPTCHA checkbox is checked, and a progress bar shows '46%' completion of the upload.

## Ranking Table

The screenshot shows the 'Ranking Resumes' section with the following data:

User	Status	Action	Score	Score Distribution
Sachin.pdf	In Review	Resume	0.66	0.19 0.11 0.11 0.10 0.08 0.07 Skill Exp Org Job Key Inst
Bhavika_Jain_resume.pdf	In Review	Resume	0.61	0.17 0.13 0.07 0.10 0.07 0.06 Skill Exp Org Job Key Inst
Muralikrishnan_Resume.pdf	In Review	Resume	0.60	0.19 0.12 0.07 0.10 0.06 0.05 Skill Exp Org Job Key Inst
Akash's Resume.pdf	Accepted	Resume	0.60	0.17 0.06 0.15 0.08 0.08 0.06 Skill Exp Org Job Key Inst
Aakash	In Review	Resume	0.60	0.17 0.06 0.15 0.08 0.08 0.06 Skill Exp Org Job Key Inst

The screenshot shows the 'Resume Ranker' application interface. On the left is a dark sidebar with navigation links: Home, Create job, Jobs Created, Report, and Profile. The main area has a title 'An easier way to search and sort'. A search bar contains 'java', with a dropdown menu showing 'javascript' and 'java'. Below the search bar is a section titled 'Rank By' with several filter checkboxes: Experience (checked), Organisation (checked), Keywords (unchecked), Job switches (unchecked), Skill Sets (checked), and Institution (unchecked). To the right of the search bar is a table titled 'User' with three rows of resume data. Each row includes a file icon, a resume name, a status (In Review or Accepted), an action button (Resume or Decline), a score (e.g., 0.66, 0.60, 0.60), and a horizontal bar chart showing score distribution across categories: Skill, Exp, Org, Job, Key, and Inst.

## Preprocessed Profile

The screenshot shows the 'Resume Ranker' application interface with a title 'Preprocessed Profile'. On the left is a dark sidebar with navigation links: Home, Create job, Jobs Created, Report, and Profile. The main area displays a resume profile for 'Aakash'. It includes sections for 'Applicant Name' (Aakash), 'Applicant Email' (akash@gmail.com), 'Scores' (Overall: 0.60, Skills: 0.17, Experience: 0.06, Organisation: 0.15, Job Switch: 0.08, Keyword: 0.08, Institution: 0.06), and 'College Names' (Vishwas Senior Secondary School Hisar, Malviya Nagar Institute of Technology). There is also a 'Company Names' section listing companies like Hp, Jpmorgan chase, Oracle, Qualcomm, and a 'Skills' section listing various technical skills. To the right of the profile details is a large blue circular placeholder image for a profile picture, with 'Accept' and 'Reject' buttons below it.

# Report Generation

The screenshot shows the 'Report Generation Dashboard' on a web browser. On the left is a dark sidebar with navigation links: Home, Create job, Jobs Created, Report (which is selected), and Profile. The main area has a light gray background. At the top, there's a header bar with the title 'Report Generation Dashboard'. Below it is a white box labeled 'Report Generation with filter'. Inside this box, there's a dropdown menu titled 'Select filter to be applied ▾' with the following options: 'None', 'Job **Specific** posts reports', 'Filter job posts report based on **Title**', 'Active job posts report', 'Job posts report between a **Time** frame', and 'Filter job posts report based on **Skill**'. To the right of the dropdown are two teal-colored buttons: 'Apply Filter' and 'Download Report'.

This screenshot shows the same 'Report Generation Dashboard' after a filter has been applied. The sidebar and the 'Report Generation with filter' box are identical to the first screenshot. However, the main content area now displays a 'Report Card' section. The 'Post Title' is listed as 'Software Engineer' and the 'Number of Applicants' is '12'. Below this is another section titled 'Top Applicant Score Distribution Details'.

RESUME RANKER Recruiter

## Report Card

**Post Title : Software Engineer**  
Number of Applicants : 12

### Top Applicant Score Distribution Details

Rank	Name	Score	Score Distribution					
			Skill	Exp	Org	Job	Key	Inst
1	Muralikrishnan_Resume.pdf	0.70	0.25	0.11	0.07	0.10	0.10	0.07
2	yash.pdf	0.68	0.25	0.08	0.07	0.10	0.11	0.07
3	Sachin's Resume.pdf	0.65	0.18	0.11	0.09	0.10	0.11	0.06
4	simran.pdf	0.63	0.18	0.07	0.14	0.10	0.10	0.05

RESUME RANKER Recruiter

4	simran.pdf	0.63	0.18	0.07	0.14	0.10	0.10	0.05
5	srijarko.pdf	0.61	0.16	0.07	0.11	0.10	0.10	0.07

### Distribution of Top Applicants Score

Resume	Skill	Exp	Org	JobSwitch	Keyword	Institute
Sachin's Resume.pdf	0.25	0.11	0.07	0.10	0.10	0.07
srijarko.pdf	0.18	0.07	0.14	0.10	0.10	0.05

### Scatter Distribution of Applicants over Score

Score	Applicants
0.25	0
0.50	4
0.75	6

## Downloadable Report

The screenshot shows a web browser window for 'Resume Ranker' at the URL <https://ccibt9.me/report#>. On the left, a 'Print' dialog box is open, showing settings for 'Microsoft Print to PDF', 'Copies 1', and 'Pages All'. The main content area displays a 'Report Card' for a 'Post Title : Software Engineer' with 'Number of Applicants : 12'. The card includes a section titled 'Top Applicant Score Distribution Details' with a table:

Rank	Name	Score	Score Distribution
1	Muralikrishnan_Resume.pdf	0.70	Skill: 0.25, Exp: 0.11, Org: 0.07, JobSwitch: 0.10, Keyword: 0.10, Institute: 0.07
2	yash.pdf	0.68	Skill: 0.25, Exp: 0.08, Org: 0.07, JobSwitch: 0.10, Keyword: 0.11, Institute: 0.07
3	Sachin's Resume.pdf	0.65	Skill: 0.18, Exp: 0.11, Org: 0.09, JobSwitch: 0.10, Keyword: 0.11, Institute: 0.06
4	simran.pdf	0.63	Skill: 0.18, Exp: 0.07, Org: 0.14, JobSwitch: 0.10, Keyword: 0.10, Institute: 0.05
5	srijarko.pdf	0.61	Skill: 0.18, Exp: 0.07, Org: 0.11, JobSwitch: 0.10, Keyword: 0.10, Institute: 0.07

Below this is a chart titled 'Distribution of Top Applicants Score' with a legend: Skill (blue), Exp (orange), Org (yellow), JobSwitch (purple), Keyword (green), and Institute (light blue).

## Profile Settings

The screenshot shows a web browser window for 'Resume Ranker' at the URL <https://ccibt9.me/profile>. On the left, a sidebar menu includes 'Home', 'Create job', 'Jobs Created', 'Report', and 'Profile'. The main area displays a profile for a user named 'Recruiter' (Software Engineer). The profile details include:

- Account Details: 8222078733, rec@gmail.com, sachinraghult, MG Street.
- Contact Details: LinkedIn ID: sachinraghult.

To the right, there is an 'Edit' form with fields for:

- Username: Recruiter
- Linkedin Id: sachinraghult
- Phone: 8222078733
- Address: MG Street
- DOB: dd-mm-yyyy

Buttons for 'Update' and 'Edit' are present at the bottom of the form.

# View Resume

A screenshot of a resume viewed in a browser window. The resume is for Akash bhatnagar, a current student at Malviya Nagar Institute of Technology (MNIT) Jaipur, pursuing a B.Tech in Computer Science. It highlights technical proficiency in C++, C, Data Structures and Algorithms, and the MERN stack. The resume includes sections for EDUCATION (B.Tech in Computer Science from MNIT), INTERMEDIATE (Vishvas Senior Secondary School Hisar), MATRICULATION (O.P. Jindal Modern School Hisar), WORK EXPERIENCE (JPMorgan Chase), and PERSONAL PROJECTS (Pizza Ordering website, Blog website, Todo-list, College management system). Skills listed include DBMS, Machine Learning, NLP, Python, Java, Pytorch, React, MongoDB, MySQL, Data Structures and Algorithms, C++, C, HTML, CSS, Node.js, and Javascript.

# SEARCHER

## Job Feed

A screenshot of a job search interface titled "Job Feed". The interface features a sidebar with navigation links: Home, Applications, Job Feed, Bookmarks, and Profile. The main area displays four job listings in a grid format:

- Software Engineer**: Deadline July 31, 2022. Status: In Review. Application status: Applied. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Participates in a formal internship program with a duration of at least 8 – 10 weeks. Internship includes...  
Internship Full Time  
Apply Now Contact
- Game Developer**: Deadline July 31, 2022. Status: In Review. Application status: Applied. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Participates in a formal internship program with a duration of at least 8 – 10 weeks. Internship includes...  
Internship Full Time  
Apply Now Contact
- SDE**: Deadline July 31, 2022. Status: Accepted. Application status: Accepted. Description: Design and build innovative technologies in a large distributed computing environment and help lead fundamental changes in the industry. Create solutions to run predictions on distributed  
Internship Full Time  
Applied Contact
- Software Engineer**: Deadline July 31, 2022. Status: Accepted. Application status: Accepted. Description: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs  
Internship Full Time  
Accepted Contact

# Job Application

The screenshot shows a web browser window for 'Resume Ranker' with the URL <https://ccibt9.me/apply/62e04cbad4a80bdbb1f76410>. The page displays a job listing for a 'Game Developer'. The job status is 'In Review'. The deadline is listed as Sunday, 31, 2022. The job type is 'Internship' or 'Full Time'. The description section notes that it's a formal internship program with a duration of at least 8 – 10 weeks. It includes details about performing various assignments to become familiar with the organization and gain basic work experience. The 'Qualified Experience' section lists requirements such as a Bachelor's degree in Engineering, strong English skills, and the ability to work with multiple priorities. The 'Skills Required' section lists Java, Python, ML, Web Dev, and DBMS.

The screenshot shows the same 'Resume Ranker' application window, now at the stage where a resume is being uploaded. A file named 'Sachin's Resume.pdf' is selected in the 'Upload Your resume' input field. A 'Browse' button is visible next to the input field. Below the input field is a reCAPTCHA verification box with the text 'I'm not a robot' and the reCAPTCHA logo. A blue progress bar at the bottom indicates the upload status with the text 'Please wait! Uploading your resume...' and a green progress bar showing 100% completion.

# Jobs Applied

The screenshot shows a web browser window for 'Resume Ranker' at the URL <https://ccibt9.me/my-applications>. The interface includes a sidebar with links for Home, Applications, Job Feed, Bookmarks, and Profile. The main content area is titled 'My Applications' and displays two job application cards.

**Job Application 1:** SDE (applied 22 hours ago). Status: In Review. Job Tags: Internship, Full Time. Deadline: July 31, 2022. Description: Check out latest 2022 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. Participates in a formal internship program with a duration of at least 8 – 10 weeks. Internship includes performing various assignments to become familiar with the organization and gain basic work experience. A 'View Application' button is present.

**Job Application 2:** SDE (applied 2 days ago). Status: Accepted. Job Tags: Internship, Full Time. Deadline: July 31, 2022. Description: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. A large orange banner on the right says 'WE ARE HIRING! JOIN OUR TEAM'.

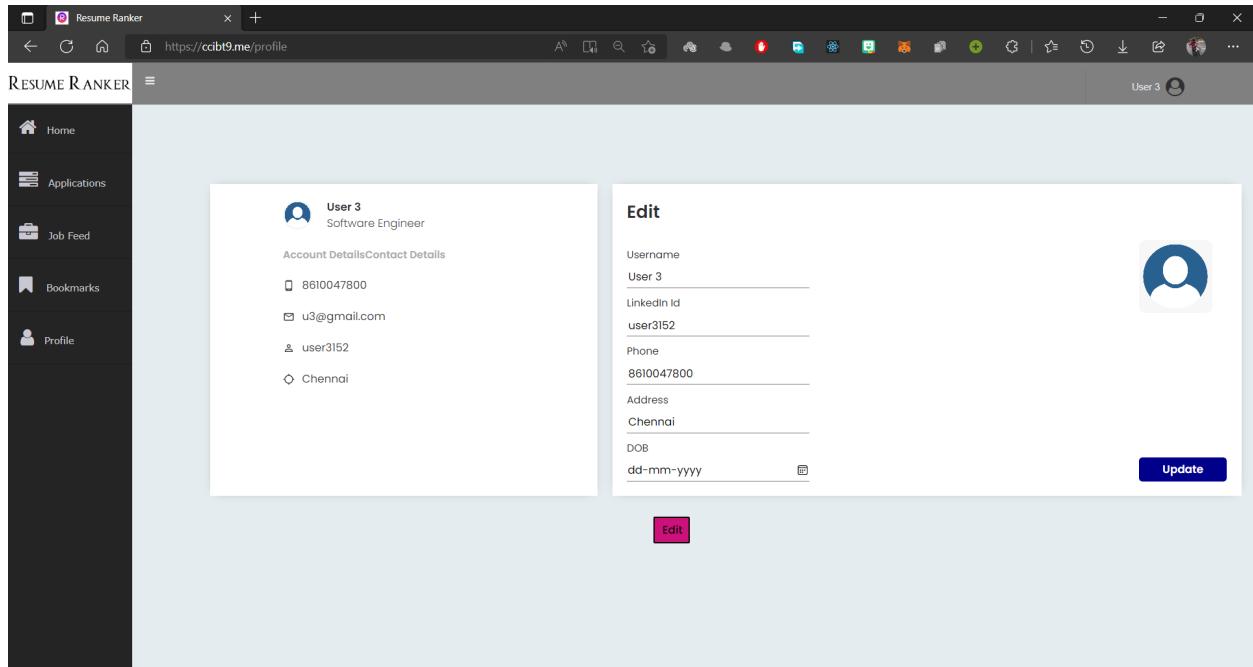
# Bookmarks

The screenshot shows a web browser window for 'Resume Ranker' at the URL <https://ccibt9.me/bookmarks>. The sidebar and layout are identical to the 'My Applications' page. The main content area is titled 'My Bookmarks' and displays two bookmarked job listings.

**Bookmark 1:** SDE (Accepted). Status: Accepted. Job Tags: Internship, Full Time. Deadline: July 31, 2022. Description: Check out latest 2020 Wells Fargo job vacancies in India. Get details on salary, company and location. Apply quickly to various Wells Fargo jobs. A blue 'Applied' button is present.

**Bookmark 2:** SDE (Open). Status: Open. Job Tags: Internship, Full Time. Deadline: July 31, 2022. Description: Design and build innovative technologies in a large distributed computing environment and help lead fundamental changes in the industry. Create solutions to run predictions on distributed systems with exposure to innovative technologies at incredible... A blue 'Apply Now' button is present.

# Profile Settings



## References

- <https://reactjs.org/docs/faq-state.html>
- <https://www.npmjs.com/>
- <https://reactjs.org/docs/hooks-reference.html>
- <https://mui.com/material-ui/getting-started/overview/>
- <https://jestjs.io/docs/getting-started>
- <https://www.react-google-charts.com/>



# Ranking Algorithm

## File Structure

The main ML setup is made up by `ranker/` directory, including:

- Main Ranking Algorithm module `algo/`
- ML-Backend Integration modules:
  - `db_utils/` for utility functions and
  - `celery_tasks/` for running tasks in the background
- Training notebooks directory `notebooks/`
- Test files for each criterion `tests/`
- Download script for weights and nltk data `weights/`
- Main ranking file on ML side `test.py`
- Dockerfile to start the containers including `celery`
- ML side Unit Tests `unitests.py`

Running the `main.sh` file via CLI will start the ranking algorithm and return a table of ranked resumes

## Quick Start

- Create a directory inside `ranker/` that will contain all resume files that you want to be ranked, move all resumes inside the directory, and check if the files have been properly moved

```
$ mkdir sample/  
$ cd sample  
$ mkdir resumes/  
$ cd resumes/  
$ ls
```

- Install all dependencies via

```
$ pip install -r requirements.txt
```

- Run the shell script to download the weights, run unit tests and the main ranking file via

```
$ ./main.sh
```

**Note :** In case you are not using the path mentioned in the guidelines above, please consider changing the path to the resume directory locally, on the `test.py` file as

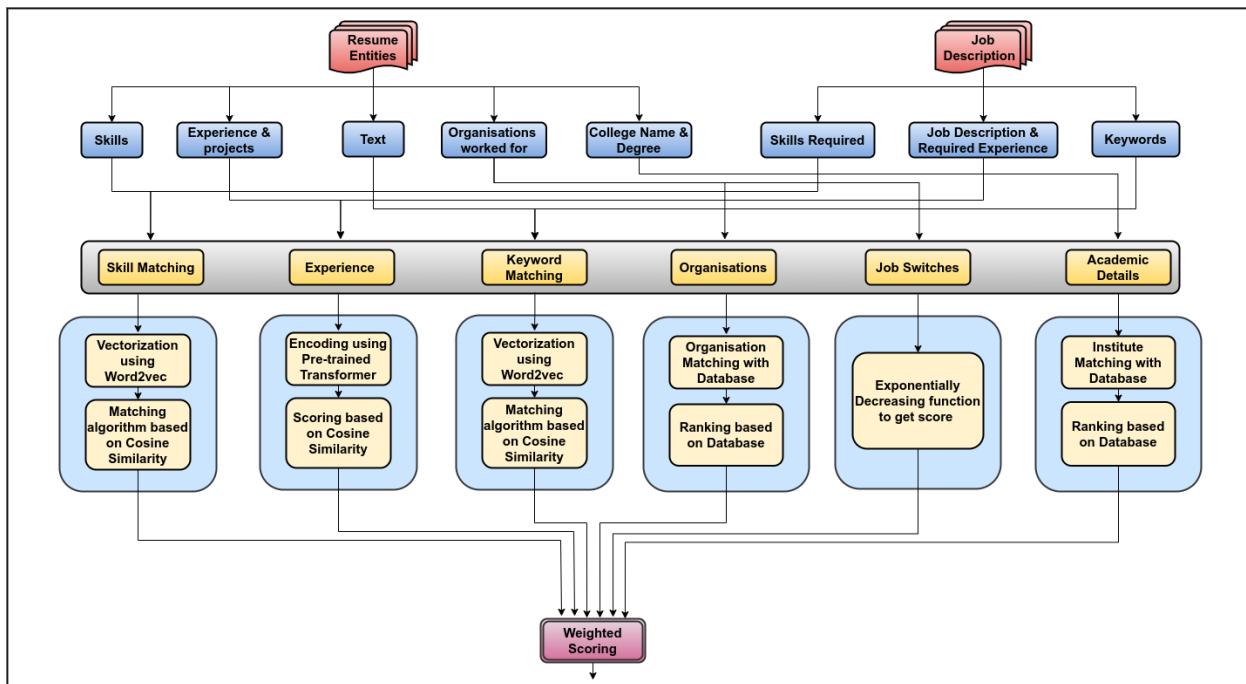
```
resume_dir = "<path/to/resume/directory>"  
resumes = os.listdir(resume_dir)
```

Check out the [Training Notebooks](#) for training details of Fuzzyset, NER and Word2Vec.

## Ranking Algorithm

1. [Workflow Diagram](#)
2. [Arguments passed to the Ranking Algorithm](#)
3. [Code Structure and Main Files](#)
4. [Working](#)
5. [Import](#)
6. [References](#)

### Workflow Diagram



### Arguments passed to the Ranking Algorithm

- **Resume Entities:** Extracted from Resumes uploaded by Job Searchers

```
{  
    "skills": [],  
    "college_name": [],  
    "experience": [],  
    "company_names": [],  
    "total_experience": [],  
    "jd_key_words": [],  
    "degree": None,  
    "text": None,  
}
```

- **Job Description:** Fetched from UI, to be uploaded by Job Recruiters

```
{
    "title": "SDE",
    "desc": "Individual Contributor Role focused on developing RIA application using ML",
    "exp": "Bachelor's degree in Engineering. Good spoken and written English skills to effectively communicate technical concepts. Strong",
    "skills": [{"skill": "Python", "value": 75}, {"skill": "machine-learning", "value": 25}],
    "keywords": [{"keyword": "Python"}, {"keyword": "ML"}],
    "tags": ["Internship"]
}
```

## Code Structure and Main Files

- Main Modules

```
|---- algo
    |---- parser (module for parsing the resumes)
        |---- constants.py (lists of subsections/names/stopwords/etc.)
        |---- utils.py (utility functions called by parser)
        |---- parser.py (main class for parsing)
    |---- criteria (module for assigning scores based on each criteria)
        |---- academicCriteria.py (scoring based on academic institutes)
        |---- domainCriteria.py (scoring based on domain experience)
        |---- jobSwitchCriteria.py (scoring based on no. of job switches)
        |---- keywordCriteria.py (scoring based on keyword matching bw resume and jd)
        |---- organizationCriteria.py (scoring based on organizations worked in)
        |---- skillRankCriteria.py (scoring based on skill match with jd)
    |---- database (folder storing data regarding orgs, skills, and academic institutions)
        |---- organizations.txt (file storing organization rankings)
        |---- Engineering.csv (file storing engg clg rankings)
        |---- Management.csv (file storing mgmt clg rankings)
        |---- skills.csv (file storing skillsets)
```

- Helper Modules and Files

```
|---- algo
    |---- ner (called in parser)
        |---- spacyNer (uses spacy ner fine-tuned with roberta transformers for extraction of college names)
            |---- config.py
            |---- ner.py
    |---- utils.py (file containing utility functions)

|---- config (module containing file ids for pre-trained weights)
    |---- weights_download.json
|---- weights (module for downloading trained ner, word2vec, and other helper models)
    |---- download.py (download script for weights download from gdrive via gdown)
```

- Main Scoring Files

```
|---- algo
    |---- ranker.py (file returning weighted average of scores based on individual criterion, makes function calls to all main modules)
    |---- test.py (file calling the ranker module and returning ranks of a bunch of resumes in a table by a simple sorting mechanism of scores a
```

## Working

### High Level

- Arguments (resume\_path, jobdescription) are passed within the `../test.py` file to the function called using `Scorer()` object from `ranker.py`
- All objects from `parser` and `criteria` module are initialized

- All entities are extracted from the resumes using the `ResumeParser()` object
- Scores for each criterion are calculated and stored using objects created from `criteria` module, passing corresponding entities keys and jobdescription keys as arguments
- Each score is multiplied by their weights and stored in a list
- Overall score is calculated by taking the mean of the weighted scores and appended to the beginning of the list
- This list is returned to the `../test.py` file
- The returned list is sorted in descending order based on the overall scores for each resume and displayed in a table

## Details of Each Criterion

- **academicCriteria:**

```

- College name and Degree are passed as arguments
- Fuzzysets is applied to extract a generalized name for the institutions
- Institute matching with database depending upon the Degree (Engineering/Management)
- Database has the names of the institutes ordered by rank, hence (index of institute)+1 gives the rank
- Scoring is based on this index passed in a mathematical function: 1000 * math.exp(-(ranks) / 400)
- Normalized score (using sigmoid function) is returned

```

- **domainCriteria:**

```

- Experience and Job Description is passed as arguments
- Encoding of both arguments is performed using pre-trained transformers
- Scoring is based on "cosine similarity"
- Mean score is returned

```

- **jobSwitchCriteria:**

```

- List of organizations is passed as argument
- Length of the list of organizations is calculated to find the number of job switched
- Scoring is based on a hand coded exponentially decreasing function
- Normalized score is returned

```

- **keywordCriteria:**

```

- Text extracted from the resume and Keywords given in Job Description are passed as arguments
- More keywords are extracted from the job description and stored
- Vectorization is performed using "word2vec"
- In case word2vec results in KeyError, "tfidf" is used
- Scoring is based on matching using "cosine similarity"
- Normalized score is returned

```

- **organizationCriteria:**

```

- List of organizations is passed as argument
- Fuzzysets is applied to extract a generalized name for the organization
- Organization matching with database
- Database has the names of the organizations ordered by rank, hence (index of institute)+1 gives the rank
- Scoring is based on this index passed in a mathematical function: 1000 * math.exp(-(ranks) / 400)
- Normalized score (using sigmoid function) is returned

```

- **skillRankCriteria:**

```
- Skills extracted from resume and weighted skills mentioned in the jobdescription are passed as arguments
- Vectorization is performed using "word2vec"
- In case word2vec results in KeyError, "tfidf" is used
- Scoring is based on matching using "cosine similarity"
- Score calculated for each is multiplied with its given weight and added
- Normalized score is returned
```

## Import

The `ranker.py` file is called in the main test file to get the final scoring of the resumes based on which ranks are calculated. The class from `ranker.py` can be imported into `./test.py` and initialized by:

```
from algo import Scorer
scorer = Scorer()
```

## References

### References

Aa	Reference Type	≡ Reference Link
<a href="#">Datasets</a>		• <a href="#">Resume Dataset</a> • <a href="#">Resume Corpus</a> • <a href="#">Online Job Posting</a>
<a href="#">Blogs</a>		• <a href="#">NLP made easy using SpaCy</a> • <a href="#">Keyword Extraction with NLP</a> • <a href="#">Keyword Extraction Process with NLP</a> • <a href="#">tf-idf vs word2vec vs BERT</a> • <a href="#">Information Extraction from CV</a>
<a href="#">Libraries</a>		• <a href="#">spacy</a> • <a href="#">pyresparser</a> • <a href="#">resume-parser</a> • <a href="#">transformer</a> • <a href="#">fuzzyset</a>
<a href="#">Research Papers</a>		• <a href="#">Duplicate Resume Checker</a> (Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks) • <a href="#">Resume Ranker IOSRJEN</a>
<a href="#">Notebooks &amp; Pre-trained weights</a>		• <a href="#">Guide to NLP in SpaCy</a> • <a href="#">Google word2vec</a> • <a href="#">Google Word2Vec pretrained weights</a>

- Check out the documentation for parser initialization [here](#).
- Check out the documentation for fuzzyset file generation [here](#).
- Check out the documentation for each criteria initialization [here](#).