# USER GUIDE

## Team AM-CJ-EV-SB Resume Checker

The Github repository for this project can be found at <https://github.com/SeanBarber-n9741275/Capstone-Project>

## Setup

### Required Modules

-datetime -enum

-Flask -functools

-hashlib -io

-json -nltk

-numpy -os

-pandas -pdfminer

-pickle -plotly

-pyLDAvis -sklearn

-SQLAlchemy -werkzeug

-yake

The server operates using python.

To set up the server, first open the console, navigate to the project folder and enter “pip install -r requirements.txt” - this will install all the required dependencies. Then run “python main.py” to start the server. Navigate to the provided URL to access the website.

## General Functionality

### Login

Users can register and log in to their accounts through the relevant pages available on the navbar. Logging in is required to use the resume checker as well as to access their profiles to see and compare their previous resumes and attempts.

### Resume Checker

Users who are logged in can select their chosen job type and upload their resume to receive feedback from the resume checker algorithm. This algorithm applies weightings to words based on their relevance to previously successful algorithms. This information is displayed in a bar graph for the users to clearly see which words have a bigger impact. The resume checker also gives a score to the resume, so the user has an easier to understand indication of its overall strength.

### Previous Attempt Comparisons

Users who are logged in are able to compare their previous resumes with their current attempt. This shows the results from both attempts side-by-side, allowing the user to clearly see which is superior, and which keywords had a stronger impact on the resume’s strength.

### Algorithm Explanation

The ranking of the resumes is determined by an algorithm, which is takes the information within the dataset, and using natural language processing produces a model. The algorithm ranks each word by its occurrence across all the resumes in the dataset, as well as inversely weighting them, so that less-frequently-occurring words are weighted higher. The model is the used the compare the keywords within the algorithm being tested and combining their relative weights to produce an overall weight for each keyword, and an overall weight for the document. This allows the resumes to be ranked.

## Areas for Future Extension

During the development of the artefact, a number of features were found that were either unable to be finished within the available timeframe, or were out-of-scope for the project. However, these can be considered for any future extensions of the project;

* Highlighting and colour coding words on the user’s resume to display clearly and in a visual manner the location of issues on the
* Sourcing of a more sophisticated data set to for building the model, as the current is not deep enough for the needs of the project
* Allowing the dataset to be populated using the user’s submitted successful resumes
* Giving more curated feedback to the user based on the job type of their resume
* Providing more specific feedback on how to improve the score of one’s resume
* Utilisation of more advanced types of analysis beyond keyword search to analyse resumes
* Functionality to inform employers of their own hiring biases and how to mitigate them
* An ability to adapt to and display the current industry “buzzwords” regarding hiring
* Add more explanations regarding the nature of design biases and their impacts