

User Manual

Project Name: ResumeAnalyzer

Authors: Darragh McGonigle (18316121), Stephen McAleese
(18921756)

Supervisor: Michael Scriney

Introduction

ResumeAnalyzer is an online web application designed to analyze resumes and display useful information about the software engineering job market in Ireland. ResumeAnalyzer consists of the following main features:

- **Home page:** the home page of the web application shows the resume upload button and the sidebar which can be used to access the other pages.
- **Results page:** shows the resume analysis dashboard after a resume has been uploaded and analyzed.
- **Reports page:** contains useful report cards which contain information about the software engineering job market such as the most in-demand skills and the most common years of experience requirements, job post locations, and soft skill requirements.
- **Tree page:** the tree page shows the sub-roles within the software engineering field. When a resume is uploaded, the page shows which roles match the uploaded resume best.

Running the app locally

First, git clone the repository from

<https://gitlab.computing.dcu.ie/mcgonid3/2022-ca400-mcgonid3-mcalees2/-/tree/master>

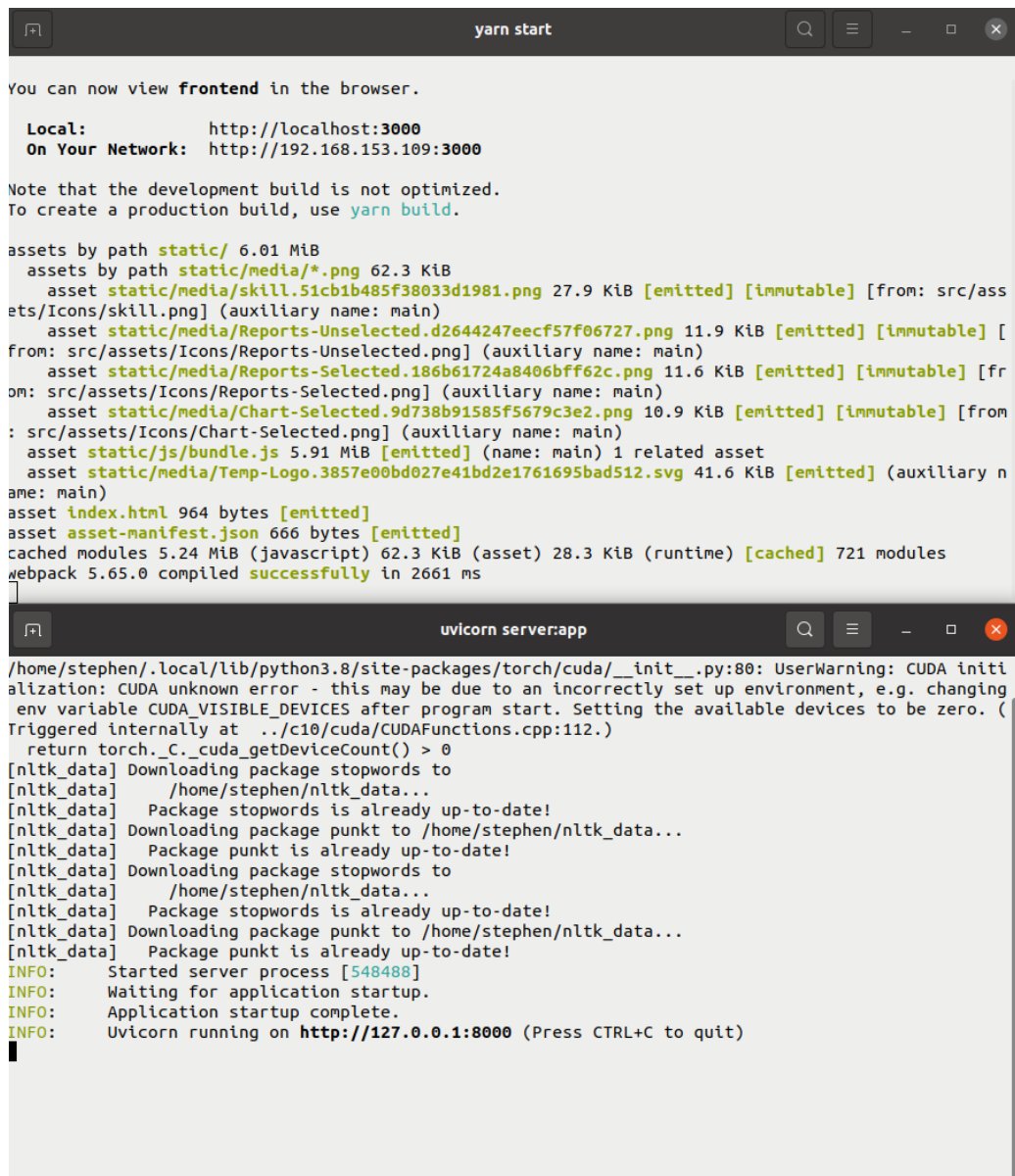
1. Set up the backend server

- Navigate into the directory '2022-ca400-mcgonid3-mcalees2/src/backend'
- Run the command **pip install -r requirements.txt** (Requires Python3 and Pip)
- Next, run the command **uvicorn server:app** to start the backend server*

2. Set up the frontend app

- Open a new terminal
- Navigate to the directory '2022-ca400-mcgonid3-mcalees2/src/frontend'
- Run the command **yarn** to install all dependencies (Requires node and yarn to be installed)
- Run the command **yarn start** to start the frontend server application which should open automatically in the web browser at the web address 'http://localhost:3000'
- Before you use the app, open the file 'frontend/src/config.js' and make sure the 'apiURL' variable is set to 'http://localhost:8000'.

****Note:** The app uses a remote database by default. If you want to use your own remote database, you will need to set one up locally use PostgreSQL and change the database connection string in 'src/backend/database.py'. Once the local database has been created, run the script **python3 populate-database.py** to populate the database with job post and job post analysis data.*



```

yarn start

You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.153.109:3000

Note that the development build is not optimized.
To create a production build, use yarn build.

assets by path static/ 6.01 MiB
  assets by path static/media/*.png 62.3 KiB
    asset static/media/skill.51cb1b485f38033d1981.png 27.9 KiB [emitted] [immutable] [from: src/assets/Icons/skill.png] (auxiliary name: main)
    asset static/media/Reports-Unselected.d2644247eecf57f06727.png 11.9 KiB [emitted] [immutable] [from: src/assets/Icons/Reports-Unselected.png] (auxiliary name: main)
    asset static/media/Reports-Selected.186b61724a8406bfff62c.png 11.6 KiB [emitted] [immutable] [from: src/assets/Icons/Reports-Selected.png] (auxiliary name: main)
    asset static/media/Chart-Selected.9d738b91585f5679c3e2.png 10.9 KiB [emitted] [immutable] [from: src/assets/Icons/Chart-Selected.png] (auxiliary name: main)
    asset static/js/bundle.js 5.91 MiB [emitted] (name: main) 1 related asset
    asset static/media/Temp-Logo.3857e08bd027e41bd2e1761695bad512.svg 41.6 KiB [emitted] (auxiliary name: main)
  asset index.html 964 bytes [emitted]
  asset asset-manifest.json 666 bytes [emitted]
cached modules 5.24 MiB (javascript) 62.3 KiB (asset) 28.3 KiB (runtime) [cached] 721 modules
webpack 5.65.0 compiled successfully in 2661 ms

uvicorn server:app

/home/stephen/.local/lib/python3.8/site-packages/torch/cuda/__init__.py:80: UserWarning: CUDA initialization: CUDA unknown error - this may be due to an incorrectly set up environment, e.g. changing env variable CUDA_VISIBLE_DEVICES after program start. Setting the available devices to be zero. (Triggered internally at ../c10/cuda/CUDAFuncions.cpp:112.)
  return torch._C._cuda_getDeviceCount() > 0
[nltk_data] Downloading package stopwords to
[nltk_data]   /home/stephen/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /home/stephen/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   /home/stephen/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /home/stephen/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
INFO: Started server process [548488]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)

```

Figure 1: the two terminals needed to run the web app locally

How to use the app

1. Open the web app

To open the web app, open a browser such as Google Chrome, type the following domain into the search bar, and press enter: <http://www.resumeanalyzer.xyz>

After navigating to the web app, the home screen of the web app should be shown.

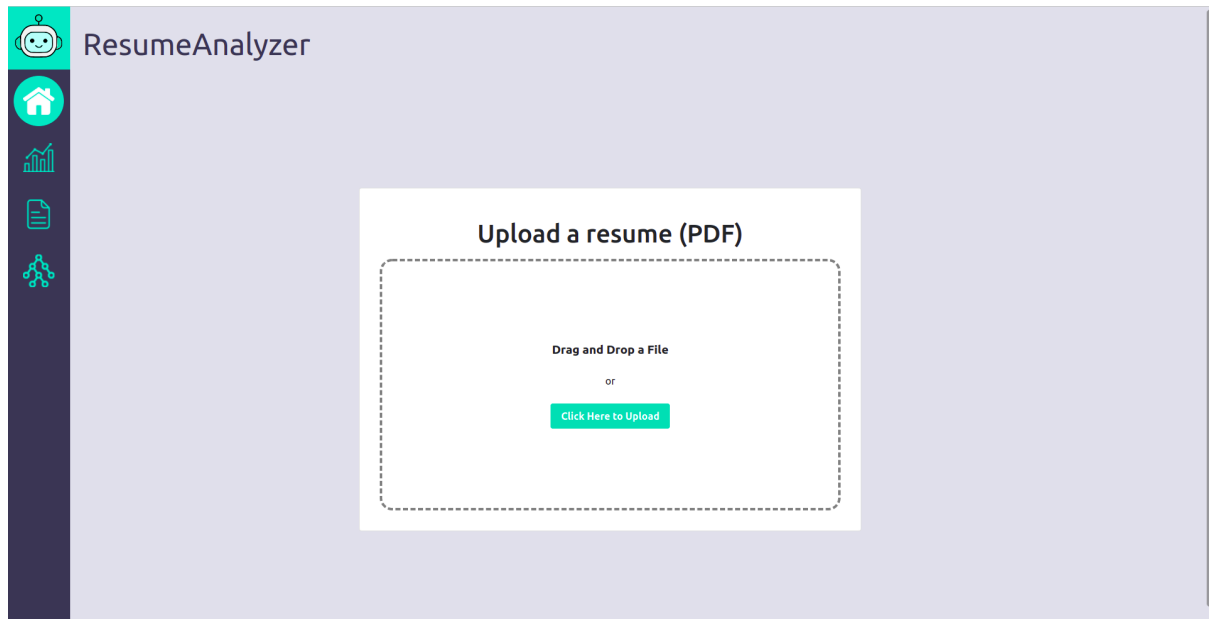


Figure 2: the home page of the web app

2. Upload a resume

A resume can be uploaded for analysis from the home screen of the web app. The web app accepts resumes in a PDF format and with a size of less than 5MB. A resume can be uploaded by clicking the green resume upload button which opens the file explorer. When a resume has been selected, clicking enter causes the resume to be uploaded. Alternatively, a resume can be uploaded by dragging the resume file and dropping it over the white resume upload area.

Once the resume has been uploaded, the button text will change to "submit" and its color will turn blue. If you would like to upload a different resume, you can reset the web app by reloading the page before uploading a new resume.

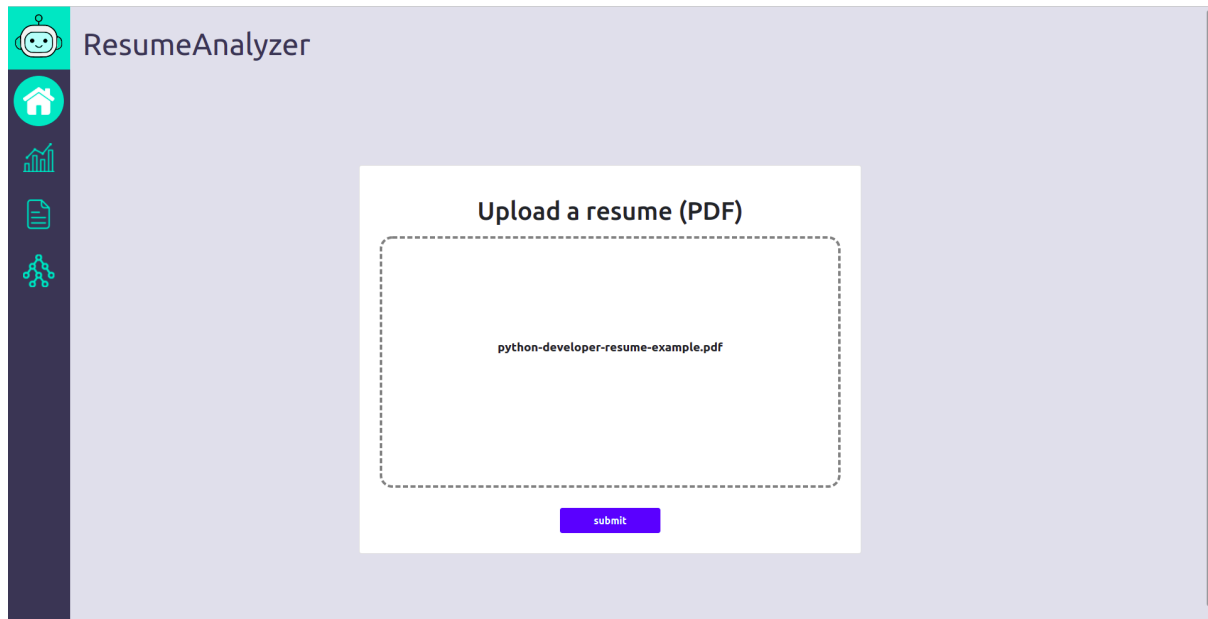


Figure 3: the home page after uploading a resume file

Clicking the submit button causes the resume to be uploaded to the server for analysis. After clicking the button, a loading screen will be shown for a few seconds. When the resume has been analyzed the web app will automatically show the results page.

3. View the results page

The results page shows several analysis panels:

- **Resume skills:** a list of skill keywords that were found in the uploaded resume.
- **Resume score:** an overall score for the resume. The score is composed of two sub-scores: a skill score and a length score.
 - **Length score:** the length score is highest for resumes that have an average length. Very short or very long resumes receive a lower score.
 - **Skill score:** the skill score is calculated as a weighted sum of the skills found in the resume and the score is between 0 and 100. This means that skills that have above-average skill counts boost the skill score more than rarer skills. The skill score is calculated for a particular role and the role can be changed with the dropdown on the score panel.
- **Resume:** a panel that displays the contents of the resume that was uploaded.
- **Skill frequencies:** a panel containing a doughnut chart and a table showing how many job posts each skill keyword was found in.
- **Skill recommendations:** shows skills related to the skills already found in your resume that you might want to learn.
- **Matching jobs:** shows a list of job posts where the requirements match the skills in your resume. The jobs are ranked in descending order with the best-matching jobs at the top.

Most of the information for the analysis cards comes from a large collection of job posts. This information is combined with information from the uploaded resume to analyze the resume. Some of the cards have a question mark. Hovering over a question mark displays more information about the card.

An analysis panel can be moved by dragging the top of the panel and dropping it somewhere else in the dashboard.

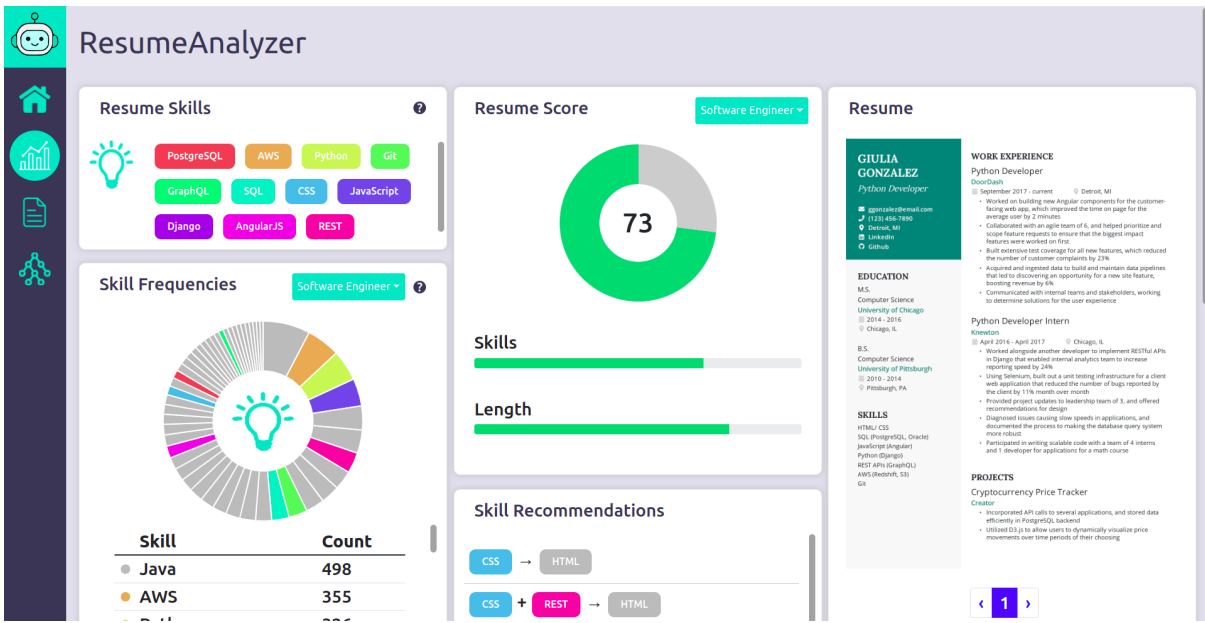


Figure 4: resume analysis results page (1)

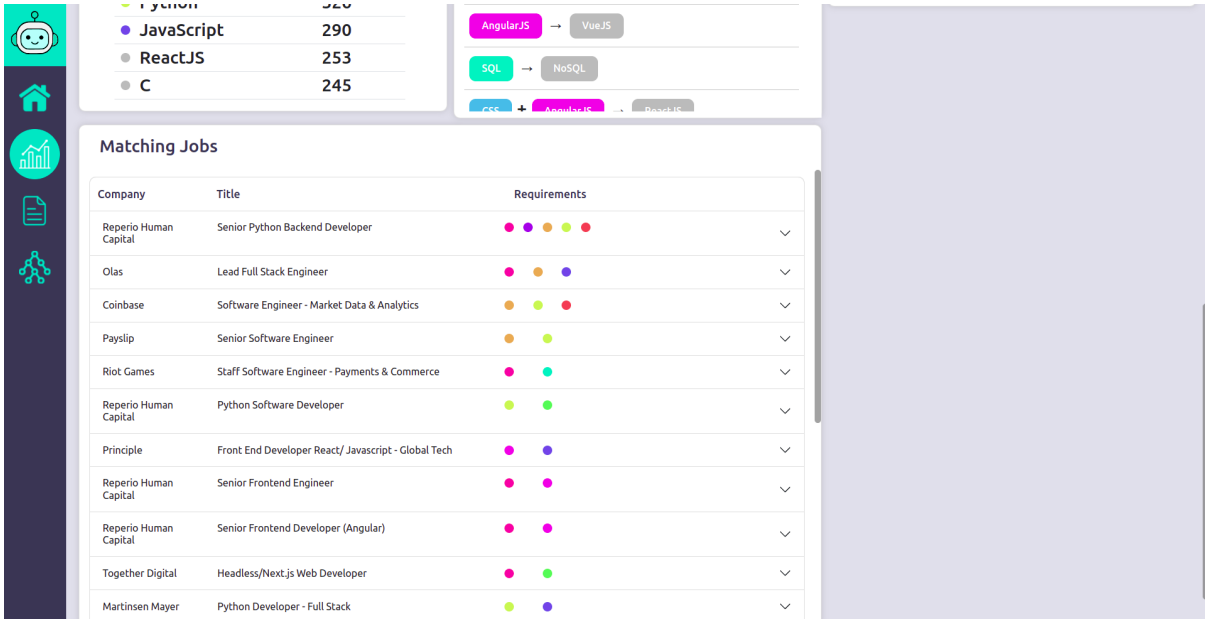


Figure 5: resume analysis results page (2)

4. View the reports page

The reports page can be accessed by clicking on the report icon in the sidebar.

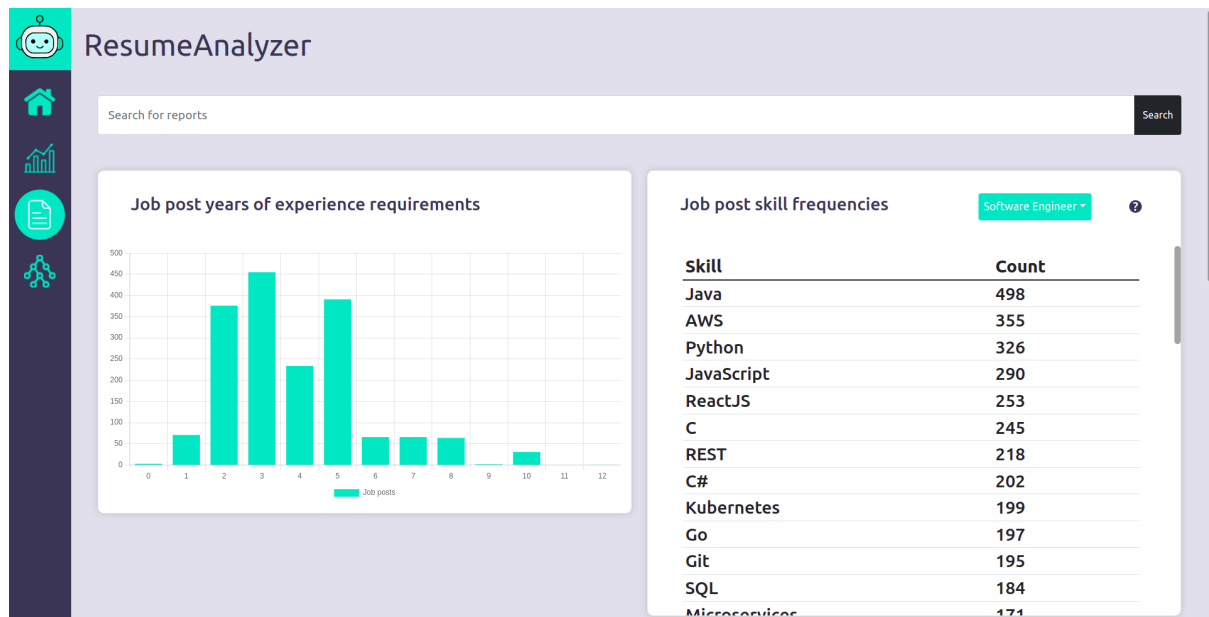


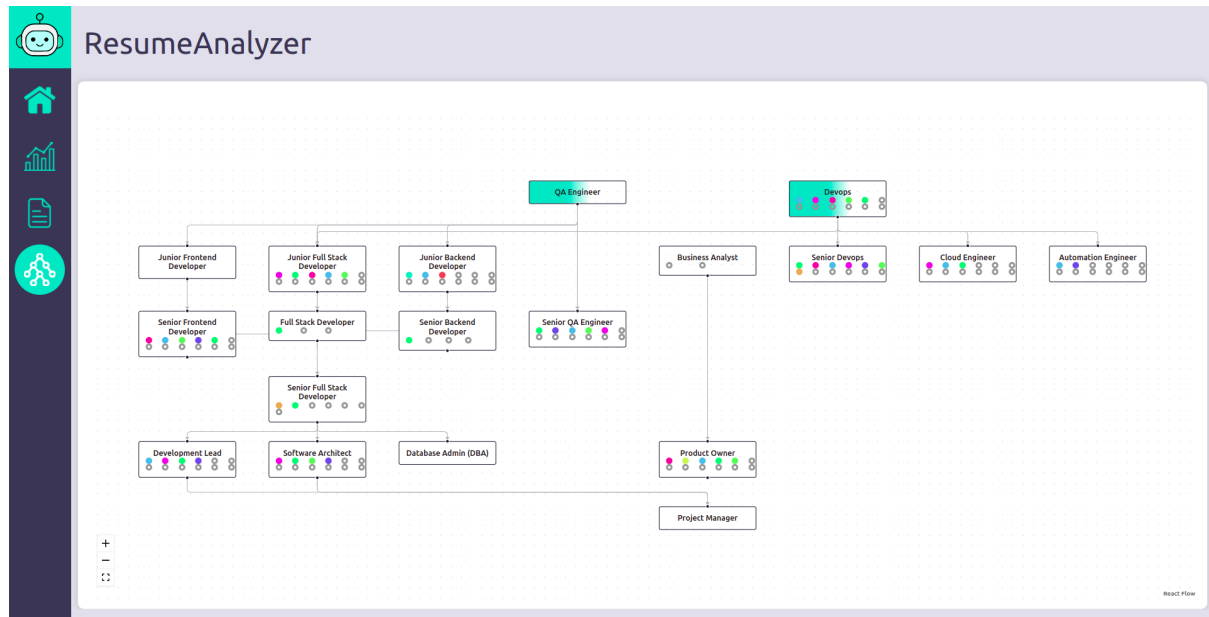
Figure 6: reports page

The reports page shows panels that show useful information about the software engineering job market in Ireland. Whereas the resume analysis page contains information from both resumes and job posts, the reports page only contains information from job posts. Therefore, it can be accessed via the sidebar without uploading a resume. The reports page contains panels including:

- **Years of experience required:** a bar chart showing a distribution of the years of experience required by the job posts. The x-axis shows years of experience requirement values from 0 to 12 and the bars show how many job posts were found for each year of experience requirement.
- **Job post skill frequencies:** a table that shows the number of job posts each skill keyword was found in. Higher values indicate that the skill keyword was found in more job posts.
- **Job post locations:** a bar chart showing the number of job posts that were found in each location.
- **Job post skill frequencies distribution:** shows the same information as the job post skill frequencies table but in a bar chart format instead.
- **Job post soft skill frequencies:** shows the most common soft skill keywords found in the job posts collection.

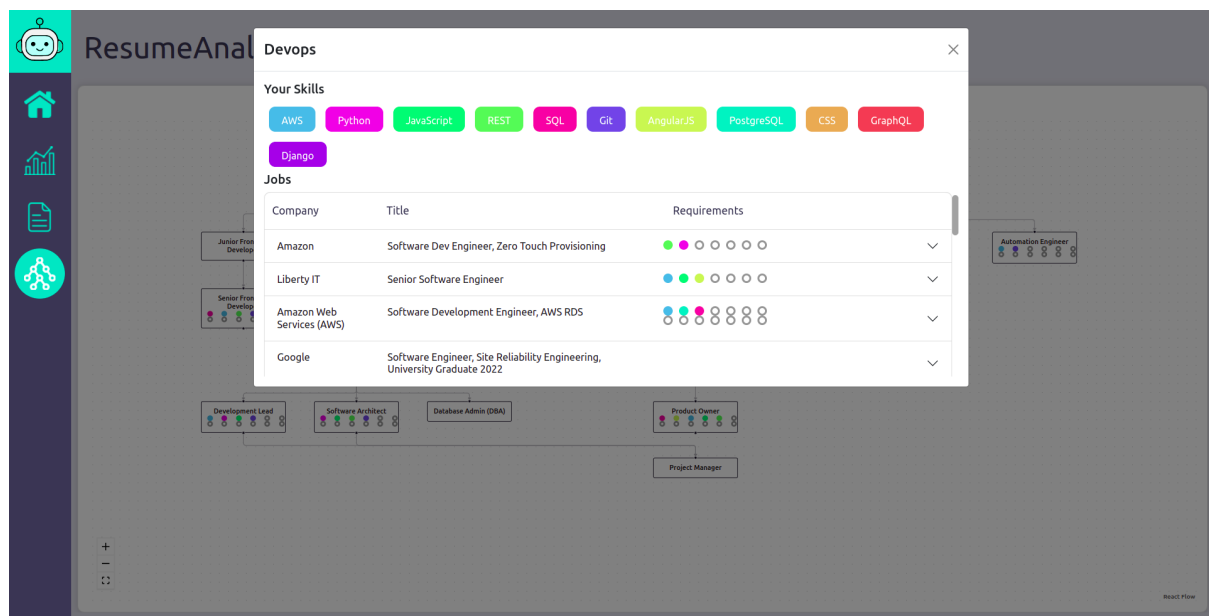
5. View the tree page

The tree page can be opened by clicking on the tree icon in the sidebar.



The tree page shows a graph of several software engineering career roles. After a resume has been uploaded, a green highlight is shown which indicates how qualified you are for certain career roles based on your resume and the skills found in it.

Clicking on one of the nodes in the graph, causes the node modal to be shown.



When the modal for a particular node in the career graph is opened, it shows a list of job posts that are related to the job title. Each node shows a list of job posts that match that role.