Scraper Service-Linked In

**Scrapper.py Functionality :**

1. db\_connect(properties):

This function takes properties which basically contains database parameters loaded from parameters.json file.

Returns: database connection object

Text

Description automatically generated

1. get\_all\_skills(connection)

Argument: connection object

Returns : List of all skill id’s and skill names from our skill\_master table.

Text

Description automatically generated

1. get\_resume\_skills(connection)

Argument: connection object

Returns : List of all resume id and resume skills from our resume\_skills table.

Graphical user interface, text

Description automatically generated

1. get\_emailing\_list(connection):

Argument: connection object returned by db\_connect function()

Returns: Returns a dictionary containing a particular *resume\_id , first name and corresponding email id* of the user to whom the resume belongs to.

Text

Description automatically generated

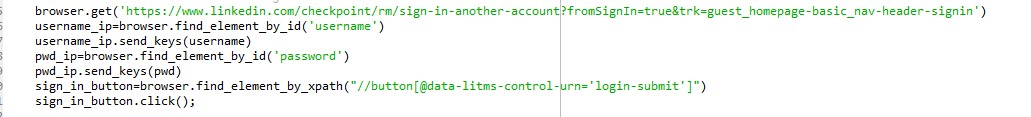
1. get\_job\_description(keyword,no\_of\_jobs\_to\_retrieve,data):

Arguments:

1. Keyword: takes the search query i.e which job profile is the user looking for to be used as filter later.
2. No\_of\_jobs\_to\_retrieve: Takes the number of jobs user wants to search and retrieve from each site .
3. data basically has information from parameters.json which has the password for the dummy linked in account.

Execution:

1. Login in to a dummy linked in profile.



1. After logging in the web browser takes us to the home page from which we are redirecting the browser via selenium to go the job listing page.

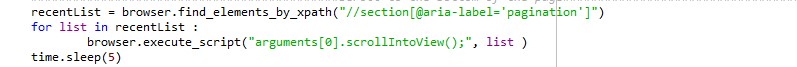


1. At the job listing page we enter the search query which is stored in our keyword variable.

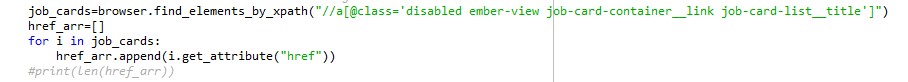
Text

Description automatically generated

1. After the search query is executed we scroll done to the end of the job listings page so that all possible job listings on the current page can be seen. Currently we are only retrieving jobs from the first page.



1. Now we retrieve all the job links from the job listing panel and store them in a list



1. Now for each of the job listing from that list we scrape all the relevant data such as responsibilities, skills and so on(usually written in list tags on each job posting) and storing it in a list. After that, we are converting all the entries in that list to a single string variable

Graphical user interface, text, application

Description automatically generated

1. This is then passed to the get\_user\_id\_to\_list\_of\_job\_ids() function.

Returns:

Final list containing email\_id and their corresponding job links is returned.

**Email service**

We are using the smtp protocol to send email containing the job links to each of the email id. Email body is organized by using html and customized email to a user.

Text

Description automatically generated

**keyword\_extraction\_modules.py**

This file uses four different functions:

1. get\_list\_of\_matched\_skills(description, total\_skills)

This function takes the description and matches with skills from the skills table. It returns the list of matches skills

Text

Description automatically generated

1. get\_dict\_with\_list\_of\_skills\_from\_description(links\_description\_dict, total\_skills)

This function creates a dictionary of job description and skills using the get\_list\_of\_matched\_skills function.

A screenshot of a computer

Description automatically generated with medium confidence

1. match\_both\_lists(list\_of\_skills\_in\_resume, list\_of\_skills\_in\_description, threshold, total\_skill\_count)

This functions returns true if the % of skills in the resume matches skills in the job description is greater than the threshold.

A screenshot of a computer

Description automatically generated with medium confidence

1. get\_user\_id\_to\_list\_of\_job\_ids(resume\_skills\_dict, links\_description\_dict, db\_connection, total\_skills, threshold)

This function returns the user ids and the corresponding job links the matches with the skills in the resume.

Text

Description automatically generated