How to use the backend of the Pathfinder project

The backend runs completely off of Python 3.8, and will only work on Mac and Linux.

First, there are several libraries that the project is dependent on that need to be installed using pip3.

- Selenium
- bs4
- sqlalchemy
- psycopg2

Then, there is a big dependency that is difficult to deal with. The machine running the backend needs to have its public IP whitelisted by our proxy service. Since this program is intended in theory to be periodically run on a server, this wouldn't be a problem. Running it as a user, however, means that we would need to add your IP address to the whitelist in order for you to run the code with minimal errors. This is unfortunately an unavoidable part of using proxy servers for web scraping.

Once dependencies are dealt with, running the program is easy. All parts of the program are operated through web_scraper.py, which has the following options:

-p --pages: scrapes the number of pages equal to the proceeding number. Defaults to 0.

-j --jobs: scrapes UP TO the given number of jobs per page, defaults to 25.

-o --output: appends formatted data, as formatted by data_formatter.py, to an output file called 'formatted.txt.'

-u --upload: disables or enables data upload to the database. Defaults to true. Argument is <true> or <false>.

-h --head: If entered, will NOT run the browser in headless mode, making the window visible for the user.

--help: Prints a usage message.

More information is given in the web_scraper.py design doc. An example for how to run the script is the following:

```
$python3 web_scraper.py -p 2 -j 5 -h -o
```

This command will run the scraper to scrape two pages, 5 jobs per page, for every job listed in the job_list array. It will also put the web drivers in not headless mode, where the user can see browser activities. Finally, the -o outputs formatted data to a log to see how the formatted data works.

How to use the frontend of the Pathfinder project

To set up environment:

- cd frontend
- npm install

- cd backend
- npm install

Running Locally:

- If you want to see webapp and database connection running separately
  - cd frontend
  - npm start

  (in a new terminal)
  - cd backend
  - npm start

Else, in root folder:

- npm start

Tests:

- cd frontend
- npm test

Commands:

```
  "backend": "cd backend && npm start"
  "frontend": "cd frontend && npm start"
  "start": "(npm run backend) & (npm run frontend)"
  "build": "cd backend && npm run build"
  "test": "jest --runInBand --testTimeout=2000 --coverage"
  "test-start": "BROWSER=none react-scripts start"
```