README - LinkedIn Database Solutions

In this document, I will walk you through step-by-step how to run and test my final project. I will also discuss all the deliverables – some don't require any action and are simply there for reference. Please feel free to reach out via slack or email (aseoff@chapman.edu) if you have any questions or encounter any issues.

Application Instructions:

1. Within the 'Final' folder, make sure you are in the 'Code' folder in a terminal
2. Execute the following command in your terminal: **streamlit run app.py**
3. Test it out. Please refer to Deliverable #4 below for more detailed information about the application.

Deliverables:

1. **Final Requirements (pdf)**

   This pdf outlines all the project requirements and shows examples from the application of how I fulfilled each. It also includes the final schema.

2. **Final Schema (png)**

   This image depicts my final schema for my database that is used within the application.

3. **Final Paper (pdf)**

   This is a thorough overview of the entire project from the initial idea to execution. The paper outlines 1) The problem to address, 2) Related applications, 3) Elements of the solution, 4) Results, and 5) Discussion and Next Steps. It also contains an appendix which contains the final requirements and final schema (also attached separately)

4. **Code (Folder)**

   This folder contains two python programs: scraper.py and app.py.

   Scraper.py
   This is the program I created that allowed me to scrape real LinkedIn job data.

   Testing: Feel free to use the below link (or another you find on LinkedIn jobs) to input into the program. Name the file whatever you'd like (Ex: test(.csv))

Testing Link:
https://www.linkedin.com/jobs/search/?currentJobId=3362755547&distance=25&geoId=103644278&keywords=data%20science

When testing this program, a csv file should be created with the name you inputted. The file should contain information about 20 jobs you scraped from the link you entered.

Note: The number of links and number of jobs per link can (and does) change. I just set it to 1 (link) and 20 (jobs) for testing purposes and to speed things up on your end. Please let me know if you would like to see if work with more data. I'm happy to do so ☺

App.py

This is the streamlit (python) application. This is the entire front-end for the project. Please feel free to test all the features. I recommend trying out a variety of filters within the jobs tab. Notice how the query will update and utilizes aggregations and sub-queries depending on what you input. When you expand a job, you have the option to enter a note and save the job with the corresponding note. Please give this feature a try.

Once you have saved a job (or maybe a couple), head over to the saved jobs section and see if your saved jobs appear with the notes you inputted.

Note: There are a few ~weird~ things (that I'm not thrilled about with the program. Please know that I am aware and hope to eventually fix it all.

1. Sometimes main page will not update when you apply filters. For example, if you first search for jobs with a set of input filters, and then try to look at connections, the jobs will still display after pressing the "Apply Connections Filter". To correct this, you need to press the "Apply Connections Filter" twice (don't ask me why!). This has something to do with the streamlit auto refresh nature.

2. When attempting to modify or delete a saved job, the interface may not cooperate. Feel free to verify that notes/deletion was committed correctly to the database by either 1) rerunning the streamlit application or 2) checking in Data Grip with a SELECT * from saved_jobs query. This is something that has been driving me crazy, but I had to let it go for now. I will fix this!!!

3. Some filters are set to 'None'. Also drives me nuts and is a default feature with the streamlit multiselect widget. I'm sure there is a workaround, but that will have to be another problem for later. Please manually remove the 'None' option from the filters that display it initially.

5. **Data Manipulation**

   This folder contains two ipynb files that I used to manipulate the original scraped LinkedIn jobs data to create csv files for companies (based on jobs and connections) and connections (incorporating company id rather than company name). No need to do anything with this, but it is there so you can see my entire process.

6. **Sample Data**

   This folder contains sample data that has already been entered into the database. These csv files were created from the ipynb files explained above. No need to do anything with these files, since the data is already loaded in. Again, it is just there for reference.