QODE assignment interview for Data Engineer (crawling focus)

Description

This project is designed as a part of the interview process for a Data Engineer position at QODE, with a focus on web crawling. The main codebase resides in the `dags` and `include` directories.

The project utilizes Docker for containerization, ensuring a consistent and reproducible environment across different platforms. It also uses the Astronomer CLI, a command-line interface that allows you to run Apache Airflow DAGs (Directed Acyclic Graphs) in an isolated environment.

The `dags` directory contains the Airflow DAGs, which define the workflows for the data extraction and processing tasks. The `include` directory contains the Python scripts that perform the actual web crawling.

Flow

The `candidate.py` script defines a Directed Acyclic Graph (DAG) for an Apache Airflow workflow. The DAG is designed to extract, transform, and load (ETL) candidate data from two sources: CareerViet and MyJobVN.

1. **Define the DAG**: The candidate DAG is defined using the @dag decorator. It is scheduled to run daily, starting from January 1, 2023, and does not catch up on any missed runs.
2. **Initialize the database table**: The first task in the DAG, `init_db`, is an instance of the `SQLExecuteQueryOperator`. It executes a SQL query to create the candidates table in the PostgreSQL database if it doesn't already exist.
3. **Extract candidate data**: The script then calls the `extract_careerviet` and `extract_myjobvn` functions to extract candidate data from the CareerViet and MyJobVN sources, respectively. These tasks are implemented using the `trio` library, which is a Python library for async I/O and structured concurrency. The first site uses `selenium` to interact with web elements, while the second site uses `BeautifulSoup` for parsing HTML.

   > For demonstration purposes, I limited only scrape 30 first pages of each website. You can change the `MAX_PAGE` variable in the `include/tasks/careerviet.py` and `include/tasks/myjobvn.py` file to `None` to scrape all pages
4. **Transform candidate data**: The extracted data is then transformed using the `transform_careerviet` and `transform_myjobvn` functions.
5. **Load candidate data**: Finally, the `load` task is called with the transformed candidate data from both sources. This task processes the data, saves it to a CSV file, and loads it into the database.

Data Schema

Column	CareerViet	MyJobVN

Column	CareerViet	MyJobVN
`name`	Title case	Title case
`workplace`	Extract first line	Standardized location name
`updated_at`	Replacing Vietnamese phrases with datetime objects	Converted to specific datetime format
`experience`	Replacing Vietnamese phrases with numeric values	Extracting numeric value and converting to integer
`salary`	Converting salary values to Vietnamese Dong (VND)	Converting salary string to integer, considering different currencies
`literacy`	Replacing Vietnamese literacy levels with English equivalents	Standardized to a predefined set of values
`source`	Added 'CareerViet'	Added 'MyJobVN'

Output Data Schema Description

- `name`: Names of the candidates.
- `workplace`: Workplace information of the candidates.
- `updated_at`: Date when the candidate's information was last updated.
- `experience`: Experience of the candidates in years.
- `salary`: Salary expectations of the candidates, expressed in Vietnamese Dong (VND)
- `literacy`: Literacy level of the candidates.
- `source`: Source of the data.

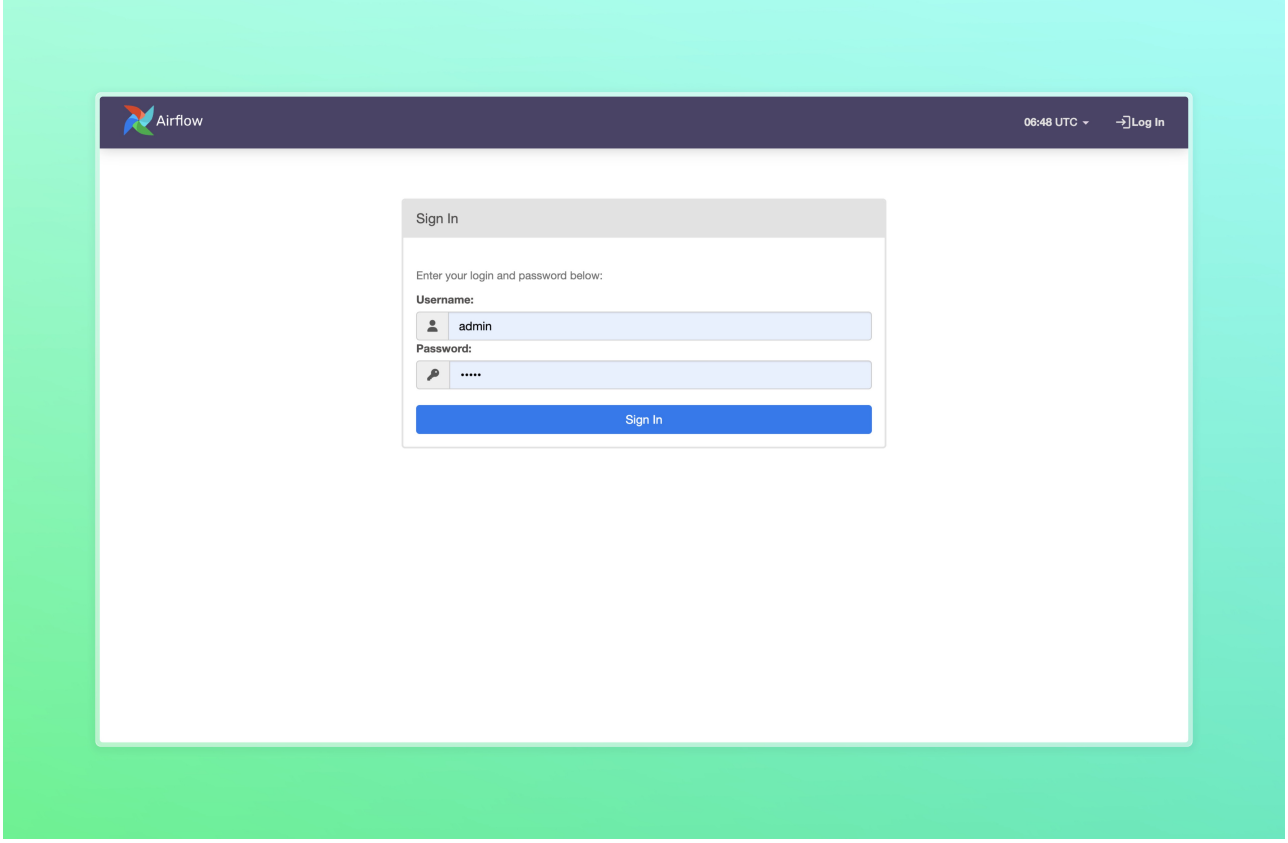
How to run

1. Install Docker
2. Install Astro CLI
3. Run in terminal

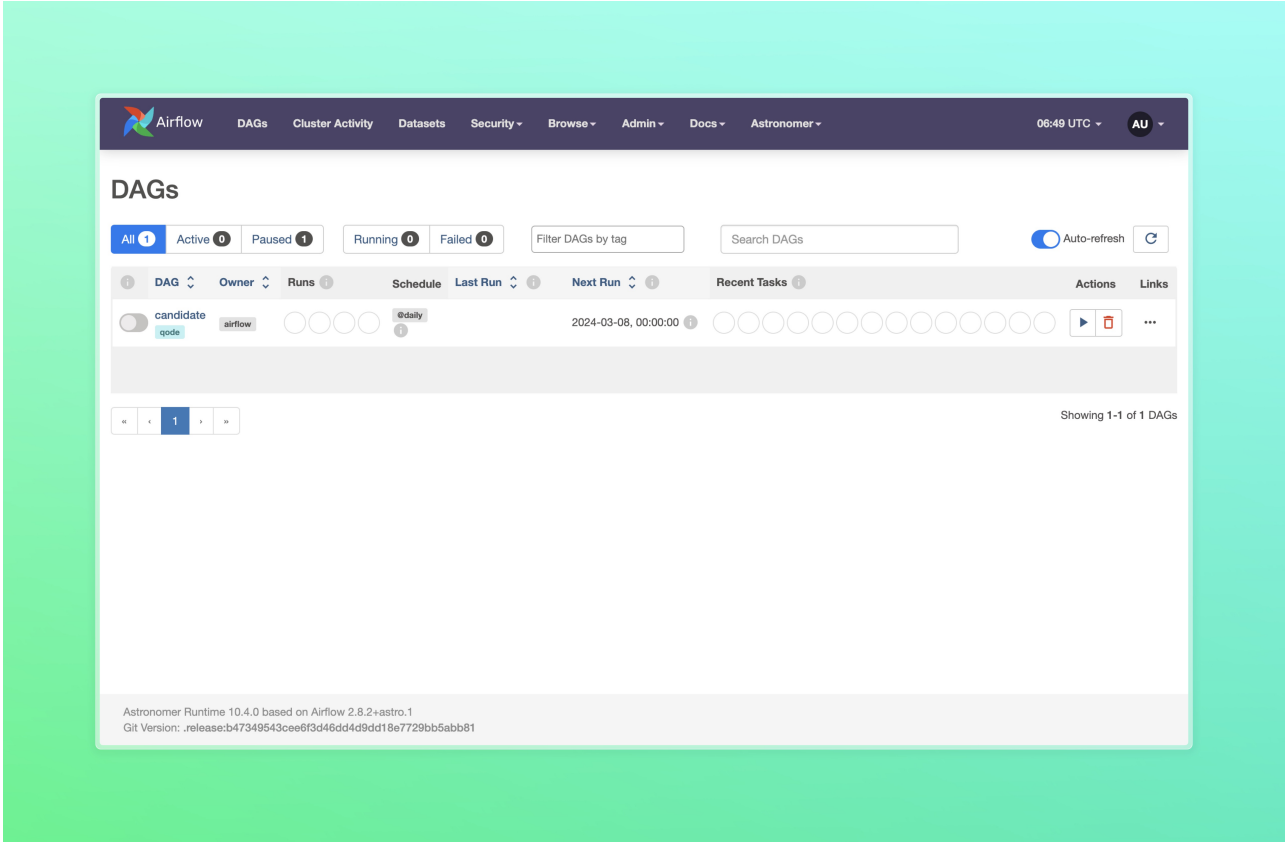
```
astro dev start
```

> Make sure port **8080**, **5432**, **4444** are not used by other processes.

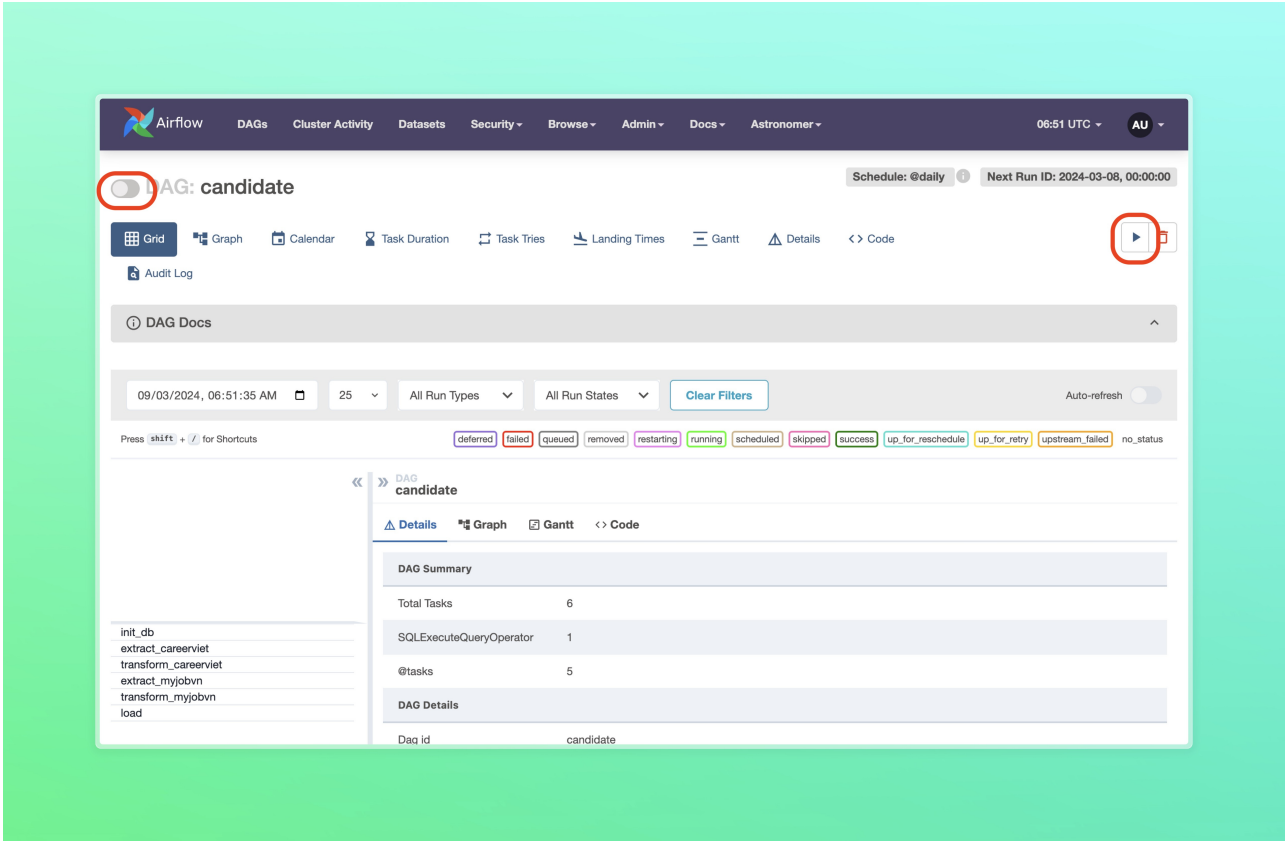
4. Open your browser and go to `http://localhost:8080` to see the Airflow UI, the username and password are both `admin`.



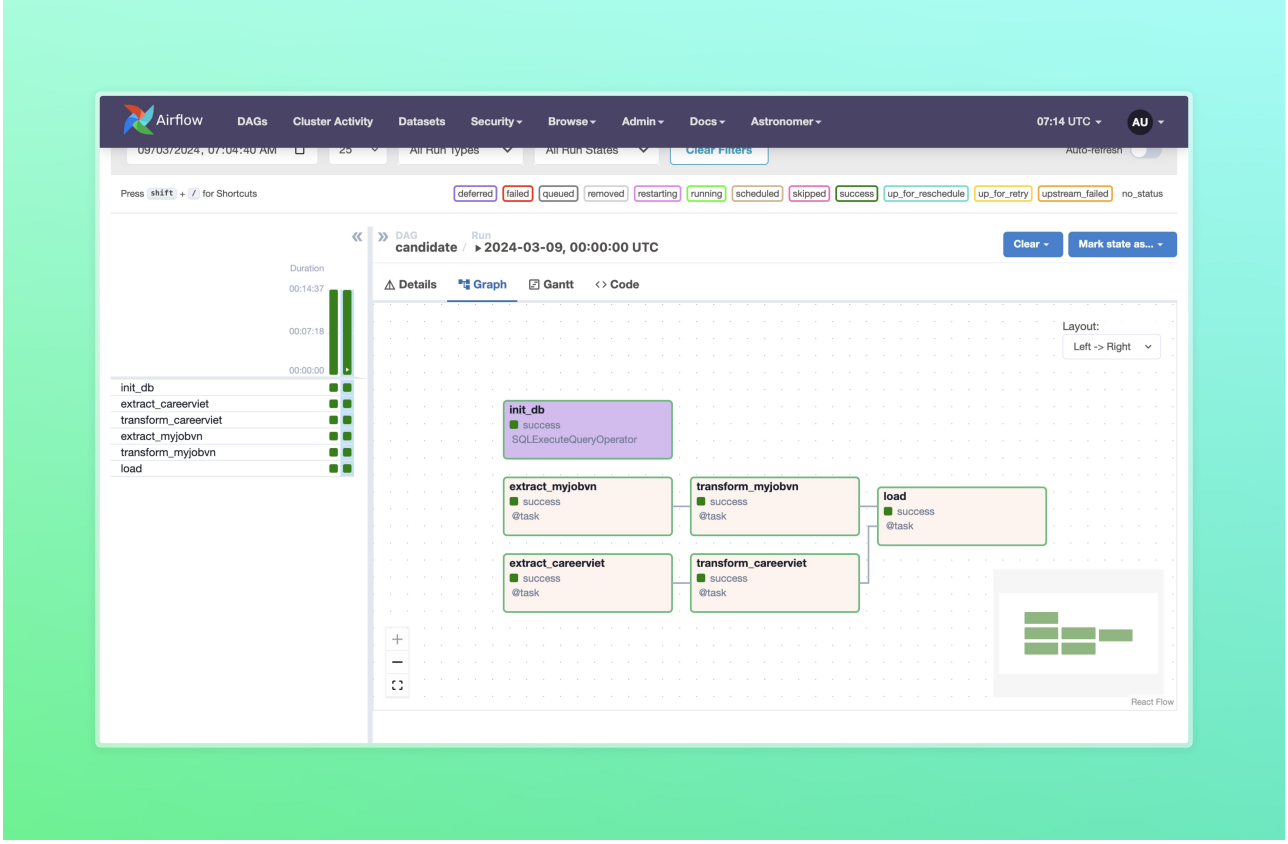
5. Click 'candidate' DAG to go to the DAG's page.



6. Turn on the DAG's switch to enable the DAG. Then click 'Trigger DAG' to start the scraping process.



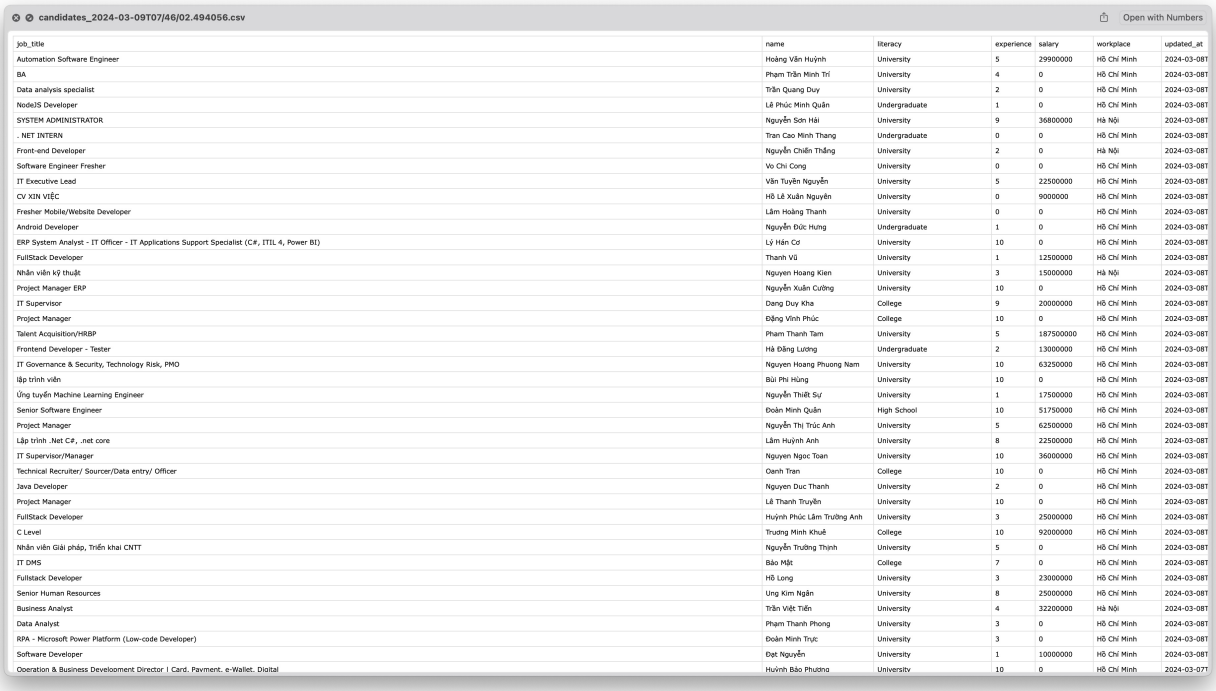
7. You can see the progress of the scraping process by clicking on the 'candidate' DAG and then clicking on the **Graph** tab.



8. After the scraping process is finished, you can see the CSV file in the `outputs` directory. You can also see the data in the PostgreSQL database with

    - Host `localhost`

- Port `5432`
- Username `postgres`
- Password `postgres`
- Database `postgres`



candidates_2024-03-09T07/46/02.494056.csv — Open with Numbers

job_title	name	literacy	experience	salary	workplace	updated_at
Automation Software Engineer	Hoàng Văn Huỳnh	University	5	29900000	Hồ Chí Minh	2024-03-08T
BA	Phạm Trần Minh Trí	University	4	0	Hồ Chí Minh	2024-03-08T
Data analysis specialist	Trần Quang Duy	University	2	0	Hồ Chí Minh	2024-03-08T
NodeJS Developer	Lê Phúc Minh Quân	Undergraduate	1	0	Hồ Chí Minh	2024-03-08T
SYSTEM ADMINISTRATOR	Nguyễn Sơn Hải	University	9	36800000	Hà Nội	2024-03-08T
. NET INTERN	Tran Cao Minh Thang	Undergraduate	0	0	Hồ Chí Minh	2024-03-08T
Front-end Developer	Nguyễn Chiến Thắng	University	2	0	Hà Nội	2024-03-08T
Software Engineer Fresher	Vo Chi Cong	University	0	0	Hồ Chí Minh	2024-03-08T
IT Executive Lead	Văn Tuyên Nguyễn	University	5	22500000	Hồ Chí Minh	2024-03-08T
CV XIN VIỆC	Hồ Lê Xuân Nguyên	University	0	9000000	Hồ Chí Minh	2024-03-08T
Fresher Mobile/Website Developer	Lâm Hoàng Thanh	University	0	0	Hồ Chí Minh	2024-03-08T
Android Developer	Nguyễn Đức Hưng	Undergraduate	1	0	Hồ Chí Minh	2024-03-08T
ERP System Analyst - IT Officer - IT Applications Support Specialist (C#, ITIL 4, Power BI)	Lý Hán Cơ	University	10	0	Hồ Chí Minh	2024-03-08T
FullStack Developer	Thanh Vũ	University	1	12500000	Hồ Chí Minh	2024-03-08T
Nhân viên kỹ thuật	Nguyen Hoang Kien	University	3	15000000	Hà Nội	2024-03-08T
Project Manager ERP	Nguyễn Xuân Cường	University	10	0	Hồ Chí Minh	2024-03-08T
IT Supervisor	Dang Duy Kha	College	9	20000000	Hồ Chí Minh	2024-03-08T
Project Manager	Đặng Vĩnh Phúc	College	10	0	Hồ Chí Minh	2024-03-08T
Talent Acquisition/HRBP	Pham Thanh Tam	University	5	187500000	Hồ Chí Minh	2024-03-08T
Frontend Developer - Tester	Hà Đăng Lương	Undergraduate	2	13000000	Hồ Chí Minh	2024-03-08T
IT Governance & Security, Technology Risk, PMO	Nguyen Hoang Phuong Nam	University	10	63250000	Hồ Chí Minh	2024-03-08T
lập trình viên	Bùi Phi Hùng	University	10	0	Hồ Chí Minh	2024-03-08T
Ứng tuyển Machine Learning Engineer	Nguyễn Thiết Sư	University	1	17500000	Hồ Chí Minh	2024-03-08T
Senior Software Engineer	Đoàn Minh Quân	High School	10	51750000	Hồ Chí Minh	2024-03-08T
Project Manager	Nguyễn Thị Trúc Anh	University	5	62500000	Hồ Chí Minh	2024-03-08T
Lập trình .Net C#, .net core	Lâm Huỳnh Anh	University	8	22500000	Hồ Chí Minh	2024-03-08T
IT Supervisor/Manager	Nguyen Ngoc Toan	University	10	36000000	Hồ Chí Minh	2024-03-08T
Technical Recruiter/ Sourcer/Data entry/ Officer	Oanh Tran	College	10	0	Hồ Chí Minh	2024-03-08T
Java Developer	Nguyen Duc Thanh	University	2	0	Hồ Chí Minh	2024-03-08T
Project Manager	Lê Thanh Truyền	University	10	0	Hồ Chí Minh	2024-03-08T
FullStack Developer	Huỳnh Phúc Lâm Trường Anh	University	3	25000000	Hồ Chí Minh	2024-03-08T
C Level	Trương Minh Khuê	College	10	92000000	Hồ Chí Minh	2024-03-08T
Nhân viên Giải pháp, Triển khai CNTT	Nguyễn Trường Thịnh	University	5	0	Hồ Chí Minh	2024-03-08T
IT DMS	Bảo Mật	College	7	0	Hồ Chí Minh	2024-03-08T
Fullstack Developer	Hồ Long	University	3	23000000	Hồ Chí Minh	2024-03-08T
Senior Human Resources	Ung Kim Ngân	University	8	25000000	Hồ Chí Minh	2024-03-08T
Business Analyst	Trần Việt Tiến	University	4	32200000	Hà Nội	2024-03-08T
Data Analyst	Phạm Thanh Phong	University	3	0	Hồ Chí Minh	2024-03-08T
RPA - Microsoft Power Platform (Low-code Developer)	Đoàn Minh Trực	University	3	0	Hồ Chí Minh	2024-03-08T
Software Developer	Đạt Nguyễn	University	1	10000000	Hồ Chí Minh	2024-03-08T
Operation & Business Development Director I Card, Payment, e-Wallet, Digital	Huỳnh Bảo Phương	University	10	0	Hồ Chí Minh	2024-03-07T

Difficulties

- To make sure Selenium can run on your machine, I used Selenium Grid in Docker by writing `docker-compose.override.yml`. By default, Selenium Grid has maximum sessions is one, and a timeout is 300, this is not enough for scraping efficiently and quickly, so I need to override those values by defining environment variables. Also, I need to config `networks` to work with my Airflow
- At first, I could not run concurrently in the `extract` task, after some investigation, I found out that Selenium Webdriver methods are not designed to work with async, they block operations that don't release control back to the event loop until they're done. So I use `trio` library to run the WebDriver. You can find the block code of the solution in the `include/tasks/careerviet.py` file.

```
driver = await trio.to_thread.run_sync(
        webdriver.Remote,
        "http://selenium:4444",
        True,
        None,
        webdriver.EdgeOptions(),
    )
```