**Jobs Analytics Source Code Improvements**

**Problem Statement:**

The current process of writing/maintaining codes (especially Beautiful Soup and Scrapy) is too repetitive and requires lot of local storage/computing power due to redundancy.

**Existing Solution:**

1. Writing same code skeleton again and again and just changing db parameters and crawling section (tags and attributes)
2. Managing all py file locally with same repeated code snippet
3. Running them locally/on VM on every phase run

**Proposed Solution:**

1. Identify selenium, no selenium site from companies list and set flag as selenium or non-selenium in DB
2. From non-selenium sites, prepare pattern and non-pattern companies list.
3. Identify common HTML elements from pattern and non-pattern companies scripts
4. Maintain tags in db
5. Write a template code that takes tags and attributes and runs the company
6. Make a UI to download py file, report directly for a given CDMSID
7. Manage and run the py code and the tags and attributes at db level so py file will be reduced from local
8. Phase Run will automate for all such cdmsids and burden on VM as well as time for phase run will reduce to a great extent
9. For error companies or new companies we can test in jobs tool before assign to developer.
10. We can add one scheduler to run all companies and we can generate companies status report. it will help full R&A Team

**Problem Statement:**

**The current process of writing/maintaining codes (especially Beautiful Soup and Scrapy) is too repetitive and requires lot of local storage/computing power due to redundancy.**

Note: Approximately the current ratio of Beautiful Soup/Scrapy Non-Pattern URLs vs Selenium and API Post method URL is **50:35:15**

- It includes copying and pasting the same code snippet repeatedly and just changing the required parameters like url, cdmsid, patternid and crawling part of the code. The rest of the lines generally remains same.

- The process(code snippet) is repetitive and always follows three major portions:

1. SQL call based on some parameters like cdmsid, patternid, url, dailyspider, pyresource to get other corresponding prameters like company name, Sno, etc. and also calling current TPhase.

2. Crawling part ---> Main Part that needs attention of developer the most

3. SQL insertion query after crawling is done.

**Solution:**

- The 1st and 3rd part only changes the parameters, the skeleton remains the same and hence can be made a method (decorator) allowing developers to get rid of that section of code.

- The second part in case of Beautiful Soup and Scrapy is also generally same hence can be made a different method to only take tags and attributes as value and doing the crawling under the hood.

**Benefits:**

- We can integrate this process to database which will keep tags attributes parameters and run the code at the db level rather than local level --> Developers only need to tweak in the parameters in the db based on their urls and will not need to manage any py file corresponding to that locally

- We can give a UI to that through which anyone (Our Team or R&A Team) can "RUN" or "Download" the cdmsid and test the data if required.

- The most important benefit will be automating Phase Run completely and just giving Error companies to developers to tweak tags and attributes. **This will reduce time for phase run and burden on VMs to a big extent keeping in mind that it will reduce 50% of Non-Pattern code files from local/VMs.**