**ETL Task**

The following task is designed to test your familiarity with Python 3 when it comes to writing simple and efficient data transformation pipelines. It covers the following topics: HTTP requests, HTML Parsing, grouping and aggregation of datasets, ZipFile processing and Excel Workbooks generation.

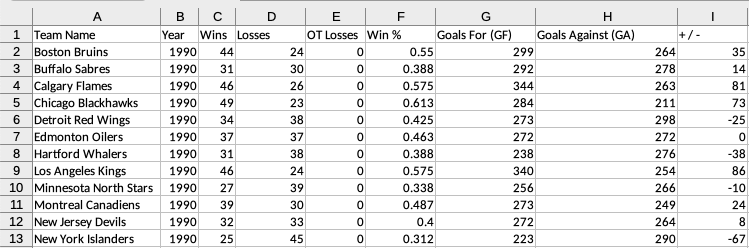
The task is to be completed in 3 to 5 hours. Write code that’s readable, efficient and *testable*. You’re expected to write unit tests for the core calculations of the program. You should deliver a solution that can be run and produces the desired output on the first try without crashing or producing unexpected results. The use of typing annotations are highly recommended.

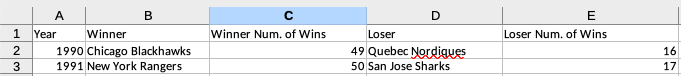
Make sure you understand the requirements, if you have any question please ask before you start coding. Don’t waste time.

Good luck!

**Write a Python program that:**

* Scrapes all Hockey Team Stats starting from this page: <https://www.scrapethissite.com/pages/forms/>. Make sure to get all data for all subpages.
* Transforms the collected data so that when run, the application produces the following outputs:
  1. One ZipFile containing all original HTML files collected from the site. The HTML files in the ZipFile should be named after their corresponding page number: 1.html, 2.html, … 24.html.
  2. One Excel File that contains exactly two sheets as follows:
     1. Sheet 1 Name: “NHL Stats 1990-2011”.
        + This sheet should contain all scraped rows in the exact same order as their appear on the website.
        + Example:



* + 1. Sheet 2 Name: “Winner and Loser per Year”.
       - This sheet should contain a summary of the team with the most number of wins and the least number of wins per each year in the dataset.
       - The expected columns are:
         * Year, Winner, Winner Num. of Wins, Loser, Loser Num. of Wins
       - Important:
         * For this exercise, *consider that the Loser is not the team with most number of games lost but with the least number of games won*.
         * It’s assumed that there’s only one winner and loser for each year. If there’s more, (because they’re tied with the same number of wins) take just one.
         * Example:

**Additional Requirements:**

1. The following libraries **cannot** be used:
   1. Pandas, Numpy, Scrapy.
2. The following libraries are recommended:
   1. BeautifulSoup 4
   2. AioHTTP
3. Write unit tests (**use Pytest**) for the core calculations of your program. Write in them in such a way that there’s no need to perform actual HTTP Requests in your tests.
4. If possible, make your solution run in under 10 seconds.
5. Type annotations and modularized code are expected but not required.

**Deliver your solution as a git repository in a ZipFile or as a link to a public repository that can be cloned. Important:**

* Include a README file with instructions on how to run your program.
* Include a requirements.txt file with all dependencies required to run your code.