**DELAWARE SCRAPING PROJECT**

**Goal: -**

To extract comprehensive data from all bid categories and their individual details pages.

**Objective:** -

The main aim is to use any library for extracting data from the Website which includes the following details.

* **From Main Table (for each bid in Open, Recently Closed, Awarded):**
  + Contract Number
  + Contract Title
  + Open Date
  + Deadline Date
  + Agency Code
  + UNSPSC
  + Current Bid Status (Open, Recently Closed, Awarded)
* **From Bid Detail Page (for each bid):**
  + Full Bid Title/Header
  + Solicitation Ad Date
  + Deadline for Bid Responses
  + Contact Name (if available)
  + Contact Email (if available)
  + List of all Supporting Bid Document URLs
  + Any other text information is present on the detail page.

**Approach: -**

* The main intuition is to use the Selenium Library, which extracts the details using html, present on the site. You just have to provide the path of the tag to extract the details.
* When one row is extracted, it starts extracting from cell-wise that begins with category columns to UNSPSC columns and afterwards it enters the pop-up window which contains the link of the documents.
* For the pagination, we used the tag (id, class) through which we were able to identify where the next page is present or not.
* Error Handling: While extracting the data if some fields were not present, we simply put the “NA” word in it.
* To stop the Program currently we have included (Ctrl+C) button which provides a sudden exit.
* While running the Script, if due to some error, it gets stopped somewhere and when we rerun it will start from that point where it stops early rather than starting.
* Create a function which basically modularizes the overall code.
* Added the logging part, which basically keeps the record of everything.

**Tools Used: -**

* Selenium

a. Options

b. Services

c. chrome Driver

d. by

e. Action Chains

f. EC

g. Web Driver

* Openpyxl
* Os
* Log
* Web Driver
* Time
* Padas

**Problem faced: -**

* At First the code was able to fetch only the data up to page 1st , it was not giving the data from the other pages.
* While extracting the data, the inner data which have the documents, and their links are not accessible.
* The browser window was not running when the code was running.
* Whenever the code stops between the execution, it must re-run from the start. Also, the data that is available in the excel file is overwritten.
* Getting the Xpath of the links.

**Conclusion: -**

Hence, we conclude that with the help of a different Library the data is being extracted from the Delaware Website in the exact form. The script is robust enough to handle dynamic content loading, pop-ups (if any appear upon clicking a bid), and variations in page structure. It also manages error handling for network issues, pages not found, or elements not loading correctly. All the data is saved in the excel file in an organized and readable format.