**Assignment-Data Scientist**

**Assignment: Web Scraping**

**Objective:**

Create a Python script to automate the process of scraping **all available transaction documents for Wakad for the year 2023** from the given site. The process should include handling form submissions, scraping transaction data, handling pagination, and downloading HTML files of transaction documents.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Steps to Complete the Assignment:**

**1. Automate Form Filling -**

Navigate to - <https://pay2igr.igrmaharashtra.gov.in/eDisplay/>

Use **Selenium** to automate the form filling process.

(i) Form Details:

- **Year** : 2023

- **Taluka** : Haveli

- **Village** : Wakad

- **Article Name** – करारनामा

- Iterate over all **available Property Numbers.**

- Automate **Captcha** filling to enable seamless form submission.

Handle the **“No Data Found”** scenario gracefully by checking if results are available.

**2. Scrape Transaction Document**

- For each transaction displayed on the page, extract the following details:

- Doc Num(Document Number)

- SRO Code

- Year

- Identify the link to the index document present in the last column of each row.

**3. Download Index Files**

- Simulate clicking the link in the last column for each transaction to access the document page.

- Download the **HTML content** of the document page and save it locally using the following naming convention:

- **{DocNum}\_{SROCode}\_{Year}.html**

- **Example**: 12345\_567\_2023.html

**4. Handle Pagination**

- Each page shows 10 transactions.

- Implement pagination handling to iterate through all available pages and process all transactions for each property number.

**5. Organize and Store Files**

- Save all the **downloaded HTML files** in a designated folder.

- Ensure files are named properly as per the given format: {DocNum}\_{SROCode}\_{Year}.html

**Deliverables:**

1. **Python Script**: A well-structured Python script with clear comments explaining each step of the process.

2. **Downloaded Files** : A folder containing the downloaded HTML files named as **{DocNum}\_{SROCode}\_{Year}.html.**

**Assessment Criteria:**

**1. Script Functionality:**

- The script should correctly automate form submissions, handle pagination, and download the required files.

- It should handle cases where "No Data Found" appears gracefully.

**2. File Organization:**

- All files should be saved in a designated folder.

- Each file should be named according to the format: {DocNum}\_{SROCode}\_{Year}.html

**3. Error Handling:**

- Handle cases where there are no transaction results for a property.

- Handle site timeouts, failed requests, or any other errors that may occur during scraping.

**4. Code Quality:**

- The code should be clean, well-structured, and properly commented.

- Functions should be modular, and repetitive code should be avoided.

**Additional Challenge: Request Limit Handling :**

- The site has a request limit of **10 requests/day**.

- Candidates should implement methods to handle this limitation. Such as - **Session Persistence**, **Proxy Rotation** etc