
Programming Task: Real Estate Broker Collection

Objective: Build a script in **JavaScript** to collect up to **200 real estate broker records** listed on the website https://eservicesredp.rega.gov.sa/auth/queries/Brokerage. The collected data will be stored for future processing or analysis.

Task Explanation:
You are tasked with building a script using **JavaScript** to automate the collection of broker information from the specified website. The task involves the following main components:

1. **Automated Data Collection:**
   - Use **JavaScript** to access and scrape data from https://eservicesredp.rega.gov.sa/auth/queries/Brokerage.
   - Extract the list of **up to 200 real estate brokers** available on the website. This includes their names, contact details, license numbers, and any other publicly available information.
2. **Data Storage:**
   - Save the extracted data in a structured format such as **CSV** or **JSON** for easy processing.
   - Ensure the stored data is clean, consistent, and organized for analysis or future use.
3. **Handling Pagination (if applicable):**
   - If the website contains multiple pages of broker listings, ensure the script handles pagination but stops after collecting 200 records.
4. **Error Handling and Logging:**
   - Implement robust error handling to ensure the script continues running smoothly even if a specific page or broker's data is unavailable.
   - Include logging to track the progress and status of the data collection process, with a clear indication of any issues.

Deliverables:

1. **Complete JavaScript Script:**
   - Provide a fully functional JavaScript script that performs the required data collection.

   - Ensure the script is modular, readable, and well-commented.

2. **Collected Data File:**

   - Submit the data collected from the website in **CSV** or **JSON** format.
   - Ensure the data file contains **up to 200 records** with all relevant details about each broker.

3. **Documentation:**

   - A README file explaining:
     - How to set up and run the script locally.
     - The structure of the project.
     - Any dependencies or libraries used (e.g., Puppeteer, Axios, Cheerio, etc.).

4. **Demo-Ready Output:**

   - Ensure the script runs without errors and successfully collects the required data.
   - Provide a sample output file (CSV or JSON) to demonstrate the functionality of the script.