VakilDesk
LEGAL MANAGEMENT MADE SIMPLER

Interview Assignment
Web Scraping using Python

Objective:

The objective of this assignment is to assess your proficiency in web scraping techniques using Python, including handling dynamic content and implementing robust error handling.

Task Description:

You are tasked with building a web scraping logic that extracts data from the links below and saves it to only **MongoDB or PostgreSQL** database.

Scrapping Link:

1. https://www.scrapethissite.com/pages/ajax-javascript/#2015
2. https://www.scrapethissite.com/pages/forms/
3. https://www.scrapethissite.com/pages/advanced/

Requirements:

1. **Scraping Dynamic Content:**
   a. Use only **"requests library"** to scrape content from the website.
   b. Extract every data which is available in the front-end of the above website.

2. **Data Persistence:**
   a. Design an appropriate database schema to store scraped data efficiently.
   b. Save the scraped data to **MongoDB or PostgreSQL** only.

3. **Robust Error Handling:**
   a. Implement error handling to handle timeouts, network errors, or unexpected changes in the website's structure.
   b. Log any errors encountered during scraping to a log file for debugging purposes.

4. **Performance Optimization (Optional):**
   a. Implement techniques to improve scraping performance, such as parallel scraping or caching mechanisms using celery etc.
   b. Optimize the scraping process to minimize the load on the target website's servers.

5. **Documentation:**
    a. Write clear and concise documentation explaining how to set up and run your scraping application.
    b. Include instructions on any dependencies required and how to install them.

6. **Submission Guidelines:**
    a. Submit your project as a **Git Repository Link only**.
    b. Develop & deliver the project within **4 Days** by sending Git Link on email.
    c. Include all instructions or documentation needed to run the project.

7. **Evaluation Criteria:**
    Your assignment will be evaluated based on the following criteria:

    a. **Functionality:** Does the scraping application successfully extract data from the target website?
    b. **Code Quality:** Is the code well-structured, readable, and follows best practices?
    c. **Error Handling:** Does the application handle errors properly and log them appropriately?
    d. **Data Persistence:** Is the scraped data stored efficiently in the database?
    e. **Performance Optimization (Optional):** Have you implemented any performance optimizations to improve scraping efficiency?
    f. **Documentation:** Is the setup and usage of the scraping application clearly documented?

Note: You are encouraged to use any libraries or tools you deem necessary to accomplish the task efficiently. Make sure to justify your choices in the documentation.