**Take-Home Interview Assessment**

This assessment consists of a main challenge and an optional bonus challenge.

**MAIN CHALLENGE: Enhanced Web Scraper**

**What we are looking for**:

* + **Develop** a robust Python web scraper that efficiently retrieves product information and customer reviews from the attached list of product pages (see ‘page-urls.csv’).
  + **Extraction:** Extract relevant product information (title, brand, price, ratings/reviews count, description) and customer review details (reviewer name, rating, review title, review body, date of review, verified buyer status, any additional details). Pay attention to consistency and usability.
  + **Advanced Scraping Techniques:** We want to see advanced features to handle dynamic content, complex website structures, and unexpected changes. We expect the web scraper to be resilient.
  + **Concurrency:** Implement concurrent scraping using appropriate techniques. Implement mechanisms to avoid overloading the target website and ensure ethical scraping practices.
  + **Data Storage:** Store the extracted product information and customer review data in DuckDB (<https://duckdb.org/>).
  + **Design:** Create a well-structured and maintainable codebase. You will be required to extend your codebase during a live interview session.

**Deliverables:**

* **Python Project:** You will deliver a fully working Python project.
* **Instructions:** We intend to run your solution, so please provide instructions for setting up the environment and running your project.
* **Key Decisions:** Please highlight key design decisions and trade-offs you made as you worked through your solution. Be sure to include a discussion on how you would scale this solution, for example, effective use of concurrency.

**OPTIONAL BONUS: Customer Review Data Processing**

Analyze the provided data to compare average ratings for product categories between a product details dataset and customer reviews dataset:

* **Product Detail:** ~2 million rows in CSV format, available at '<https://www.kaggle.com/datasets/asaniczka/amazon-uk-products-dataset-2023>'
* **Customer Reviews:** ~41 million rows in Parquet format, available at '<https://datasets-documentation.s3.eu-west-3.amazonaws.com/amazon_reviews/amazon_reviews_2015.snappy.parquet>'

**Requirements:**

* **Data Query:** Display the average rating difference for each category across both datasets. Feel free to normalize product categories as you see fit for consistent comparisons.
* **DuckDB Integration:** Utilize the same DuckDB instance used in the main challenge for both datasets.
* **Scalability Considerations:** The use case for the query is that it will be used on a dashboard and served through an API. ***You don't have to build out an API***. However, the expectation is that the data will grow substantially in time, and as such the query should be as efficient as possible.

**Deliverables:**

* **Python Project:** You will deliver a fully working solution that calculates the average rating difference for each category.
* **Instructions:** We intend to execute your solution, so please provide any relevant instructions.