Senior Full Stack Developer – Home Assignment

Intro

Welcome to the home assignment for the Senior Full Stack Developer position at Nvidia! We're excited to see your skills in action and assess your abilities in creating a web tool for analyzing manufacturing data.

For this task, you are provided with a MongoDB collection that stores manufacturing data. The collection consists of documents, where each document represents information about a specific product that underwent various tests during the manufacturing process. Here are the key columns (fields) present in each document:

- PN (Part Number): This field represents the unique identifier or code assigned to each product or part.
- TEST_TYPE: This field denotes the name or type of the test performed on the product or part during the manufacturing process.
- PASS: This field contains a binary value (0 or 1) representing the test result. A value of 1 indicates a passed test, while 0 indicates a failed test.
- TEST_DATE: This field holds the date when the specific test was conducted.

Task Overview:

Your task is to create a web tool that enables users to interact with this manufacturing data and gain insights from it. The web tool should allow users to filter the data based on three criteria: TEST_DATE, PN, and TEST_TYPE. After applying the filters, the tool should display a yield-over-time chart (yield explanation below), and let the user choose the granularity of the time frame (buckets by hour/day/week/month). In addition, there should be an option to download the raw data to the user based on the filters.

Requirements:

1. Design a web tool that allows users to filter data based on TEST_DATE, PN, and TEST_TYPE.
2. Display a yield-over-time chart representing the pass rate (yield) of manufacturing tests over a selected time period.

3. Provide a button to download the raw data based on the applied filters.
4. To efficiently handle the raw data download – you are required to implement a custom database cursor to minimize memory usage.
   The custom cursor should work with large datasets without consuming excessive memory. This means the cursor should avoid loading the entire collection into memory at once and instead fetch data in smaller batches or chunks as needed.

   The custom cursor should support basic cursor operations such as next(), hasNext(), and rewind(). It should be designed to efficiently traverse the collection, fetching only the necessary data based on user filters for the raw data download.

Yield Explanation:

Yield is a manufacturing term that represents the ratio of good units produced to the total number of units attempted in a specific process or test. For the manufacturing data in this assignment, the yield will be calculated as the ratio of passed tests (where PASS is 1) to the total number of tests conducted for a particular TEST_DATE, PN, and TEST_TYPE.

Mongo DB collection:

The manufacturer data for the assignment is attached in two formats – csv and mongodump. To use the attached mongodump, please use the following commands –

```
tar -xzvf mongo_export.tar.gz
mongorestore --uri "mongodb://<URI>" --db nvidia --collection mnf_data
export/nvidiaDB/mnf_data.bson
```

Submission and Evaluation:

Please submit your well-organized and documented codebase along with a README file containing instructions on how to run the application locally. Ensure that your solution demonstrates efficient data retrieval using the custom MongoDB cursor while providing a user-friendly and visually appealing web tool.

In addition, please include a docker container with all your code and the run command. This will allow us to conveniently test your application in a Docker container, replicating the runtime environment without any hassle.

We look forward to reviewing your work and seeing how you tackle this exciting challenge!

If you have any questions or need further clarifications, don't hesitate to reach out.

Duration - 6 days to complete.

Good luck, and happy coding!