Hi,

Thank you for taking the time and effort to solve and code the following assignment. By doing this, we will be able to determine if your coding skills align with our needs.

The assignment involves parsing the attached html documents and saving their data into a database. After that, we ask you to analyze the documents for discrepancies and save them in the database.

We split the assignment into two parts for ease and better direction:

**Task 1: Data layer creation and parsing**  
Create a simple Data layer for MongoDB. It should be able to perform basic crud operations on a MongoDB database. It’s your choice how to structure it and you are free to add more functionality as needed.

Create a code to save the following information for each document: document ID, title, header, body, footer, country of creation, and date of creation. We also would like you to set up the needed code for storing in the DB the discrepancies information, including ID, document\_id, discrepancy\_type, location in the document, and discrepancy details.

To this end, please implement a class named “Parser” with a function named “parse” that receives a folder of files and extracts the information for each document.

We suggest you create a free account on MongoDB or any other available MongoDB-like server for development purposes. In the documentation please specify what collections you created on the DB.   
  
**Task 2: Discrepancy finder**  
Based on the parsed data in the DB find (and save to the DB) the following discrepancies:

* The header is shorter than a given (N) characters or missing
* The date in the footer is beyond a given date (D) or missing.
* The total sum of the first row in the table is higher than a given value (SUM).

To this end, please implement a class named “DocumentValidator” with a function “validate” that returns a tuple of the following structure: `(ValidationStatus, Dict)`. where `ValidationStatus` is an enum with the following values: `VALID`, `INVALID`, `ERROR`, `NOT\_FOUND`, `NOT\_PROCESSED`, and `Dict` is a dictionary that contains specific information about the validation result e.g. references to specific fields that are invalid, etc

**General instructions:**  
The code should be written in Python 3.9+, well-structured, well-documented, configurable, and easy to read. The architecture of the solution should be easily extendable and able to support future types of documents and discrepancies. If possible use suitable design patterns for example: `Strategy`, `Factory`, `Chain of Responsibility`. Please explain why this design pattern was selected and what are its pros and cons.

To submit the assignment please share with us a GitHub repository with the solution.