Assignment for a Software Engineer position July 2022

https://github.com/mixmixmix/soft_assignment

A server application, among other functionalities, needs to analyse the data gathered by another system about cases of animal diseases. Such tasks are usually done by making queries to a database or analysing data files in a tabular format that are synchronised with your server.

In this assignment, your task is to write and document a command line program (in any programming language and format you prefer) that will be used to extract key information from the provided tabular files.

Instructions:

1. All the source code and documentation need to be committed to a public repository on Github (https://github.com, you might need to create an account if you don't have one already) and the link to this repository has to be sent by email to **musa.bayoh@fao.org** by Sunday 31st July 2300 (11PM)

2. You are free to choose any non-legacy technology, your code needs to run in the command line terminal on any modern linux server, or windows/macos desktop.

3. Provide instructions (a README file) how to deploy and use your program including installation of necessary frameworks and libraries. You should always assume that a person deploying it has less experience than you do; the manual needs to be clear and professional. Quality of documentation is part of the assignment mark.

3. Provide many comments in your source code explaining your logic in detail. Imagine that you want to hand over the program to a less experienced colleague for further maintenance. Clarity of your source code and how easy it is to understand your thinking is an important aspect of this assignment.

4. Five files in the repository https://github.com/mixmixmix/soft_assignment are an example of input (*.csv files) and expected output (*.json). Your program will be tested on `data_cases_1.csv` and `data_cases_2.csv` and unseen examples from the same data source.
The input file `data_*.csv` is a list of reports from one of 5 different localities (2 health centres and three villages) on cases of sickness and death of animals. Each report contains the following information:

- 'uuid' - a unique identifier of each record
- 'datetime' - date and time of report submission
- 'species' - animal species
- 'number_morbidity' - number of sick animals
- 'disease_id' - ID of the diseases as in disease_list.csv file
- 'number_mortality' - number of deceased animals
- 'total_number_cases' - total number of cases (both deceased and sick)
- 'location' - name of one of five location from which report originates

Tasks:

There are five main tasks A-E. Even if you are not able to complete all the tasks, make sure your program is functioning and well documented. You can use any technology stack you prefer, however you need to provide clear instructions on how to deploy your program.

A. Your program reads in two input csv files, `data_cases_1.csv` with records of cases of animal diseases, and `disease_list.csv` that contains names of the diseases. Your first task is to extract some summary statistics from those files to produce the output in a valid json file. See provided example `indicators_1.json`). Make sure your output matches the example, and your program works correctly on `data_cases_2.csv` as well.

B. Names of input and output files should be specified as command line arguments.

C. Provide a README file explaining how to deploy and use the program

D. Sometimes the input file is corrupted. Analyse the problem with the file and enhance your program so that it can correctly analyse `data_cases_corrupted.csv` (output as per `indicators_corrupted.yml`)

E. Enhance your program to output more advanced indicators as in the `advanced_indicators_1.json` example file.