Python Developer Assignment

In this assignment, you need to read data from multiple sources, structure them, and make them available by providing a simple REST API. Once you complete the assignment you'll have a simple RESTful interface using which users can interact and find information about different movies. Your application should support HTTP clients. People should be able to use CURL, Postman or any other REST client to interact with your web service.

Phase 1

You will start by scraping data from Wikipedia. Please have a look at the following page.

https://en.wikipedia.org/wiki/List_of_Academy_Award-winning_films

As you can see this page contains a list of academy award winning films. Clicking on any of the movies will take you to the movie details page. Your first task is to scrape data both from the link provided above, and from each individual movie link.

For each individual link, for example something like this (https://en.wikipedia.org/wiki/Once_Upon_a_Time_in_Hollywood), it is enough to parse only data from the right sidebar. You don't need to parse the whole body of the article.

Since html structure can be a bit vague, It is also fine if you are unable to parse one or two pieces of information. Your goal should be to implement this phase in a reasonable amount of time.

Don't forget to commit your work!

Phase 2

In this phase, you will start persisting the movie data into a database. You are free to choose the structure of your dataset.

You have the flexibility to use any ORM or write RAW SQL query or even use noSQL and go for document db. You can also go for in memory or file system based small databases. It is completely up to you. Just make sure you don't use too much time thinking about the schema or database structure.

Once you are done, please commit your work.

Phase 3

In this phase, you will provide a REST endpoint on top of your movie database. You are also free to choose how you would like to structure your interfaces. Just make sure it is user friendly.

Here are some example endpoints (you are not bound to implement like this).

- http://localhost:8080/movies?count=20&page=3 (a list of 20 movies)
- http://localhost:8080/movie/123/ (detail info about movie with id 123)

Phase 4 (Optional)

This phase is not mandatory for this assignment, but nice to have. In this phase you will add rating information with the movies. Please have a look at the content of this two URLs:

- https://school.cefalolab.com/assignment/python/movies.csv
- https://school.cefalolab.com/assignment/python/ratings.csv

Movie information is contained in the file `*movies.csv*`. Each line of this file after the header row represents one movie, and has the following format: **movieId, title, genres**

All ratings are contained in the file `*ratings.csv*`. Each line of this file after the header row represents one rating of one movie by one user, and has the following format: **userId, movieId, rating, timestamp**

Your final code must read the files from the given URLs and extract rating information for the matching movies that you already have in your database. You may need to consider both the movie name and year to identify the movie correctly. Then using these files, try to extract average rating and number of rating givers, and update your API to include this information.

Please note that the data from Phase 1 and this dataset may not be common. So you may miss rating information of some movies. That is fine.

Remember to commit your work.

Deliverables

The only deliverable of this assignment is the source repository with clear instructions about how to run your program.

Your program can be run in two phases. One for initialization, at which phase the data is fetched and parsed from different sources, and inserted into the database. Sample command,

```
$ python application.py parse
```

And then for running the rest api. Sample command,

```
$ python application.py serve
```

Special Instructions

- You must use Python to solve this problem.
- You must use Beautiful Soup for web scraping purpose.
- You are free to use any other third-party library or framework if you want. Just make sure that you have added the dependency of those to your repository and you have given the instruction on the readme file regarding how to run your program including the libraries.

- Your application should run from the command line. And you should provide a readme file regarding how to run it. The readme should contain any special instruction for installation and configuration of your application.
- Make sure your instructions are clear. You should include the specific version of the libraries, databases etc. Following your instruction, anyone should be able to setup your project in a new machine.
- Commit to GitHub from the initial setup. Make sure to provide access to **“cefalolab”** user to your code repository for code review purpose.
- Commit early and often. Don’t commit everything after finishing the assignment. By looking into your commit message, we will try to get an idea how you approached the problem.