Backend/Data Assignment

Consider you are working for a game analytics company and have been tasked to create a data explorer data analysts can run analysis. As an enterprising engineer at that company you decide to solve this problem on your own. Your mission if you chose to accept is to implement the below three parts.

- Api
  - Api to support the data explorer
  - An api to take a link to the csv data and store it.
- DB
  - Store the csv data.
  - Clickhouse or Postgres
- Deployment
  - Create a docker image so that code/db can be deployed anywhere.
  - Deploy the system to a **free tier of any cloud provider** for analysts to try.

You can use any online resource at your disposal like google, chatgpt for completing the assignment but make sure you understand what you are doing. We would ask you to explain your choices in next rounds.

Judging criteria

You will be judged on these elements in order.

- Completeness of solution.
  - Deployed solution with the 2 apis - 6 pts
  - Apis work as expected - 6 pts
- Completeness of instructions and documentation you write.
  - Instructions about local and remote setup - 5 pts
- Code readability and maintainability.
  - Code Quality - 5 pts
  - Tests - 3 pts
- Performance of the solution.
  - No red flags on this - 2 pts
- Cost effectiveness.
  - No red flags on this - 2pts
- Minimal UI to try things - 2 pts

- You will not be judged on how pretty the UI is. Just the functionality. It could just be simple text boxes, buttons and a table.

Bonus points

- Authentication - 1pt
- Aggregate searches - 3pts
- Date searches - 1 pt

Submission

- Language for the code has to be python. This is what you will be using primarily at Segwise, hence we want to see how comfortable you are. Choice of web framework is yours.
- DB to use has to be clickhouse or postgres. Both have their pros and cons for the use case. You decide and lets us know the reasoning. You can decide the data models.
- Submit **link to the github project**. No zip files in google drive or email would be accepted. **Add shobhit@segwise.ai as collaborator** to your github project. Any documentation or instructions you want to provide should be in a README in the repo itself.
- Also submit the **link to the deployed solution** in README itself.
- Please also submit sample api request/response for the apis in the README file.
- README should also contain information on how to run the submission locally.
- Along with this, also submit the cost of running your system in production 24x7 for 30 days, assuming one file upload and 100 queries a day. Please add this to README as well.
- Submission needs to be made within 48 hours of being given the assignment or as communicated in the email**.** Sorry but because we want to hire ASAP, we would not be able to extend deadline.

API

- Uploading csv endpoint
  - This endpoint should accept a single csv file link as input. The csv link will be open to public and does not require any authentication. The file could be as big as 50 MB in size.
  - All queries would be run assuming this data is saved.
  - The exact api design and how you want to take input params and values is your choice.
  - Columns as well as data types and format in final csv will remain same as the sample csv. Sample csv has 100 rows to give you an idea about the data.
  - sample csv file for testing: https://docs.google.com/spreadsheets/d/e/2PACX-1vSCtraqtnsdYd4FgEfqKsHMR2kiwqX1H9uewvAbuqBmOMSZqTAkSEXwPxWK_8uYQap5omtMrUF1UJAY/pub?gid=1439814054&single=true&output=csv
  - NOTE: Final csv file for testing might not be a google drive link, so do not make that assumption. You can assume it will be a direct link to the csv file but it will have same schema as the file provided here for testing.

    - BONUS: if you can add some simple authentication so not everybody can make a call.
- Data explorer endpoint
    - This endpoint should accept parameters with values to query the data from csv. The default view should be to show all data stored. The service should support query by any field in the csv. For numerical field, exact match should work. For string fields, response should contain anything that matches the input as a substring. For eg:
        - For a age, age=20 would mean return all records where age is exactly 20.
        - But for string field like a name, substring match should work. So name=Raj should match Raj, Rajesh, Rajan …. and so on.
        - Date fields should have exact match like numbers.
    - The exact api design and how you want to take input params and values is your choice.
    - BONUS: if you can allow greater than and less than searches for dates.
    - BONUS: if you can add some simple authentication so not everybody can make a call.

DB

- Save data in a clickhouse or a postgres.
- You are allowed to use the cloud version of the dbs if you can get a free trial anywhere but remember to cancel it after the assignment review is done. Or bundle the free version with your solution.
- You are encouraged to decide and come up with creative data model for best performance and insights given the data. The data schema will remain the same for testing. So if you want to hyper optimise for this schema, feel free to do so. The solution does not have to be generic.

Deployment

- Create a docker image for your submission. Commit the Dockerfile in the repo with rest of the code.
- Deploy your solution on a free tier of any cloud provider. Make sure you are not getting charged for it. There will be no reimbursement or cloud credits provided.

NOTE: If anything is not clear or ambiguous, make a reasonable assumption and proceed. Please mention the assumptions in the README file.

Time is of the essence. Happy building.