TP API 2

À ce stade, vous avez réussi à construire un data lake sur votre machine locale, qui est constitué de plusieurs dossiers et aussi de plusieurs table SQL.
Un des buts d'un data lake est l'exposition des données, afin de donner la possibilité aux acteurs externes d'utiliser les données qui sont stockées dans le data lake.
Cependant, les données doivent être protégées par des droits d'accès pour éviter la divulgation des données personnelles.

NB: in this assignment, your API will interact with your data lake

I- authentication and authorization

Question 1 - droits d'accès
Implémentez un mécanisme de gestion des droits. À travers des appels à endpoints, un utilisateur doit être capable de:
- Donner les droits à un utilisateur pour accéder à des données sur le file system des droits d'accès à une table.
- Supprimer les droits d'accès

Question 2 - authentification
Chaque utilisateur qui utilise l'API doit être authentifié

NB:
- Django fournit un moyen de gestion de l'authentification/token
- Dans ce TP, tout appel API doit être authentifié et autorisé pour accéder à une ressource.
- Chaque fois que votre API reçoit une requête, elle doit stocker: qui a eu accès, quand l'accès a-t-il eu lieu, la requête reçue lors de l'accès et le body de la requête

II - Data retrieval

Question 3 - retrieve all
For every data on the file system in the data lake, provide an endpoint to retrieve all data with pagination. Each time, the maximum number of messages to be retrieved is 10

Question 4 - projection
Provide you endpoints with the capability to select just a projection of the dataset

Question 5 - filtering
Provide when possible the possibility to filter the result by:
- Payment method
- Country
- Product category
- Status
- Amount
  - Could be greater
  - Could be lesser
  - Could be equal
- Customer rating
  - Could be greater
  - Could be lesser
  - Could be equal

III - Metrics

- Get money spent the last 5 minutes
- Get total spent per user and transaction type
- Get the top x product bought - x is an integer that should be passed as a parameter

IV data lineage, audit and logs

- Get a specific version of the stored data, when the versioning is allowed otherwise, return an error code
- Get Who queried or accessed a specific data/table data.
- Get the list of all the ressources available in my data lake

V advanced capabilities

- Full-text search across data. For instance I would like to get everything about a transaction_id or whatever. The endpoint should return all the informations found about the text passed - This end point takes as parameter the date from where I want to start the full search. Our data lake is not designed for this, so proposed a technology that can better suit this need
  - All the tables/resources where the data where found
  - RPC (remote procedure call) - add an endpoint that will trigger the training of a machine learning inferencing model - You can choose whatever the model you want to implement.
  - Re-push a transaction - Let's suppose you have a corrupted transactions in your data lake. Add an end point that can select that specific transaction from the data lake and push it to Kafka in the very beginning of the pipeline for processing - use the right endpoint to check if the new processing was rightfully done - When you push a product you should change the timestamp on fly
  - Repush all- push all historical product to the beginning of the pipeline

API doc

A good API should have a documentation to know how to use it. So document all the endpoints. The doc should include:

- Endpoint URL - description of what the endpoint does
- Input parameters - fields description
- Output parameters - fields description
- A sample query

Note: You can use swagger