Description:

In the context of your e-commerce microservices architecture, you need to implement a recommendation engine service that provides personalized product recommendations to users based on their browsing and purchase history. This is a common real-world problem in e-commerce, where personalized recommendations can significantly improve user engagement and sales.

Requirements:

Recommendation Algorithm: Choose and implement a recommendation algorithm (e.g., collaborative filtering, content filtering ..) to generate product recommendations for users.

Microservice Integration: Design the recommendation engine as a standalone microservice that can communicate with other microservices in your architecture. It should be able to fetch user data, product data, and transaction history from relevant microservices. (Let's assume that the same microservice can be used to store product and user data)

Data Storage: Determine how you will store and update user profiles and product information for generating recommendations. You can use a database or caching mechanism for this purpose.

API: Create a RESTful API for the recommendation engine service that accepts user IDs or other relevant data as input and returns a list of recommended products.

Personalization: Ensure that recommendations are personalized for each user. The service should take into account factors like user preferences, and purchase history.

Scalability: Design the recommendation engine service to be horizontally scalable, so it can handle increased load as the number of users and products grow.

Security: Design the API as it will be used on a production environment

Constraints :

- Use python as a programming language.
- Solution should be dockerized

You are free to make assumptions as long as you provide a clear explanation.

You will find attached to this challenge a dataset representing a purchase history for different users. Feel free to update the dataset if you feel the need to do so.