Exercice Pratique : Collection de films

> ***Objectifs :***
> *Développer une application web simple de collection de films qui permet aux utilisateurs de consulter une liste films depuis une base de données.*
> *L'application sera composée en 2 parties : une **application web en Python** et une **base de données MySQL**.*
> *Vous utiliserez Docker et Docker Compose pour containeriser et orchestrer l'ensemble de l'application.*

Structure du projet

Base de données

- La base de données `MySQL` contiendra une table `movies` avec les colonnes suivantes :
  - `name` : Le nom du film.
  - `rating` : La note du film.

Application Web

- L'application web sera développée en `Python` avec le framework `Flask`.
- Elle fournira une interface web pour consulter la liste des films depuis la base de données.

Instructions

Partie 1 : Initialisation du projet

- Vous allez travailler avec une application de collection de films déjà préparée. Le code de l'application web sera fourni.
- Votre tâche consistera à dockeriser cette application et à la faire fonctionner localement sur votre machine.
- Un fichier d'initialisation de la base de données `database.sql` sera également fourni.
  - À vous de voir comment vous allez l'intégrer dans votre projet (plusieurs solutions possibles).

2. Configuration de l'Environnement Docker

- Vous devrez écrire un fichier `Dockerfile` pour l'application web en Python.
- Un fichier `docker-compose.yml` sera également nécessaire pour orchestrer les différents services.

3. Développement de l'Application Web

> - *L'application web sera développée en `Python` avec le framework `Flask`.*

- Elle fournira une interface web pour consulter la liste des films depuis la base de données.

Variables d'environnement nécessaires :

- **PORT** : Le port sur lequel le serveur web écoutera les requêtes HTTP.
- **DB_HOST** : L'adresse IP ou le nom d'hôte du serveur MySQL.
- **DB_PORT** : Le port sur lequel le serveur MySQL écoute les connexions.
- **DB_USER** : Le nom d'utilisateur de la base de données MySQL.
- **DB_PASSWORD** : Le mot de passe de la base de données MySQL.
- **DB_NAME** : Le nom de la base de données MySQL.

Configuration du projet :

1. Ce projet nécessite un environnement avec Python (3) installé.
2. Vous devrez installer les dépendances du projet avec `pip install -r requirements.txt`.

Lancement du serveur :

- Vous pouvez lancer le serveur avec la commande `python server.py`.

Choix de l'image de base :

Pour l'image de base Docker, vous pouvez utiliser l'image officielle de Python disponible. Cette image contient un environnement Python préconfiguré, ce qui simplifiera le processus de développement.

> ***Attention :*** *Les versions* `slim` *ou* `alpine` *de l'image Python ne contiennent pas tous les outils nécessaires pour installer les dépendances du projet (ex:* `mysql-client` *).*

4. Configuration de la Base de Données

> *Avant de lancer l'application web, il est crucial de s'assurer que la base de données est démarrée et correctement configurée (création de la table* `movies` *).*
> *Vous pouvez utiliser le fichier* `database.sql` *fourni pour initialiser et remplir la base de données.*

```
CREATE DATABASE IF NOT EXISTS movies;

USE movies;

CREATE TABLE IF NOT EXISTS movies
(
    name   VARCHAR(255),
    rating DOUBLE
);

...
...
```

Initialisation avec l'image MySQL

L'image Docker officielle de MySQL permet d'initialiser automatiquement la base de données en exécutant des scripts SQL au premier démarrage. Pour utiliser cette fonctionnalité, vous pouvez placer vos scripts d'initialisation dans un dossier et monter ce dossier dans le conteneur au chemin `/docker-entrypoint-initdb.d` .

Vous pouvez soit utiliser un volume Docker pour monter le dossier, soit créer une image personnalisée avec les scripts.

Ressources :

- L'application web sera fournie.
- Un script de création de la base de données vous sera également fourni pour vous aider à initialiser la base de données MySQL.
- Vous pouvez consulter la documentation officielle de Docker pour en apprendre davantage sur la création de Dockerfiles et la configuration de Docker Compose.
- Devhints Cheatsheet
  - Docker
  - Docker Compose
  - Dockerfile

Etapes :

- Commencez par étudier le schéma de la base de données fourni et familiarisez-vous avec les fonctionnalités de l'application.
- Préparez votre conteneur `MySQL` en utilisant l'image officielle de MySQL et en montant un volume pour initialiser la base de données avec le script fourni (ou autrement...).
- Écrivez les `Dockerfiles` pour l'application web, en vous assurant de bien spécifier les dépendances et les commandes de démarrage.
- Configurez le fichier `docker-compose.yml` pour définir les services nécessaires et les liens entre eux.
- Après avoir initialisé la base de données à l'aide du script fourni, lancez l'application avec la commande `docker-compose up` et testez son bon fonctionnement.