Projet Webfloox

Présenté par Mohammed, Naoufel et Pierre

Sommaire

01 Description du projet

Base : Projet Netfloox

Login et compte : L'utilisateur doit se connecter ou s'enregistrer.

Avis utilisateur : L'utilisateur peut noter les films qu'il a vu.

Confirmation de l'utilisateur : L'utilisateur peut valider ou non les recommandations proposées.

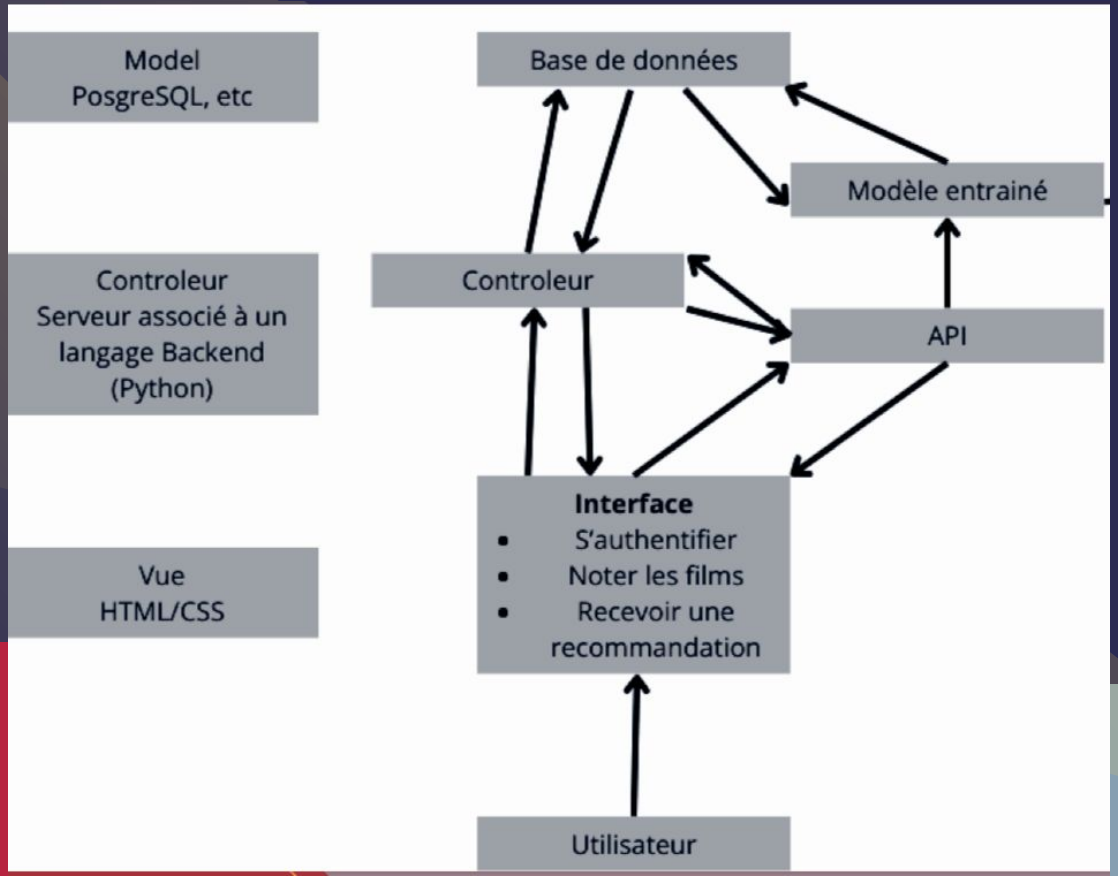
Base : Projet Netfloox

Recommandation : Nouvelle recommandation selon les notes fournies, qui sont comparées à celles d'autres utilisateurs

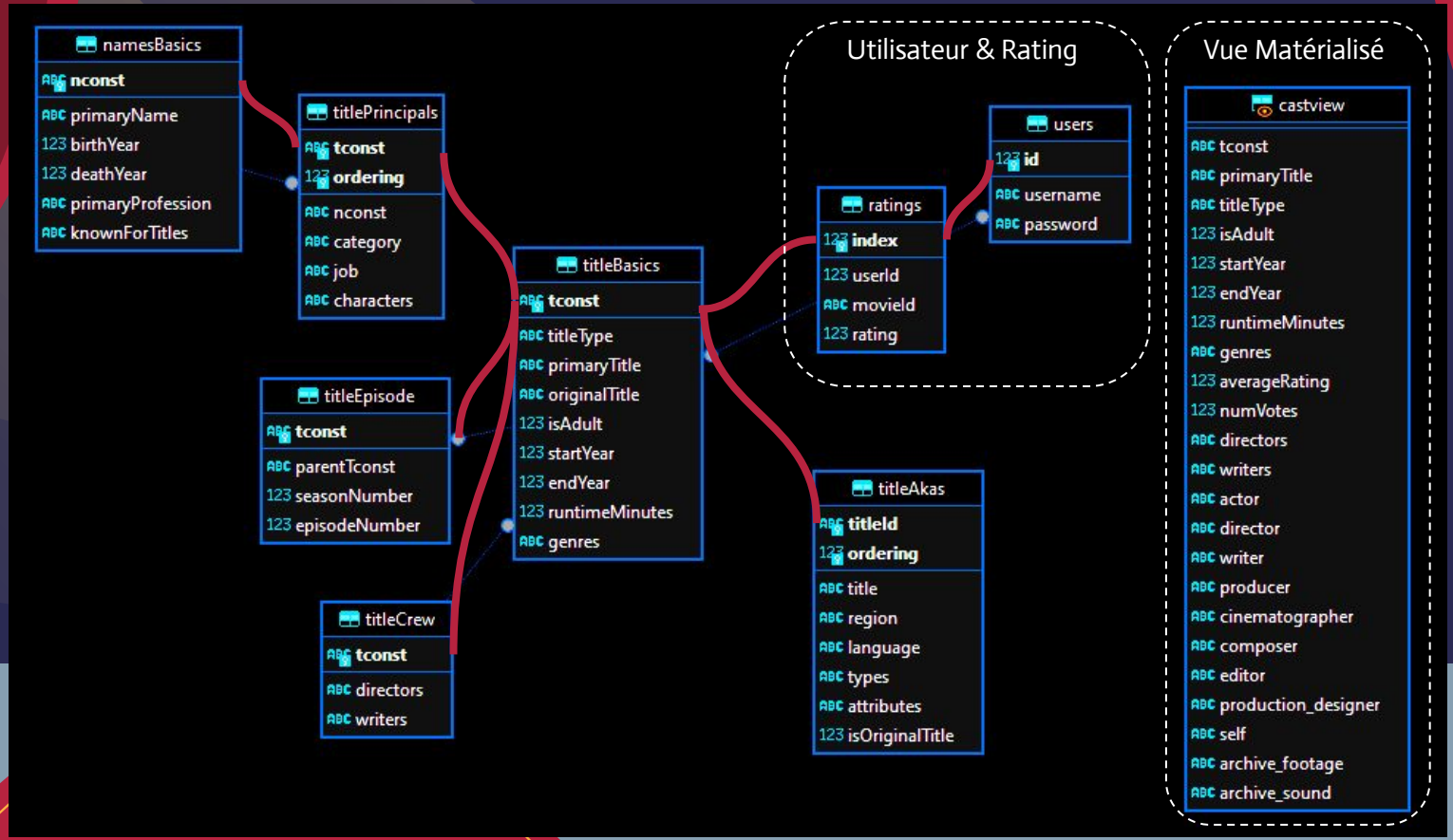
Résumé : Rajouter un résumé du film lorsque l'on clique dessus.

02 Architecture du projet

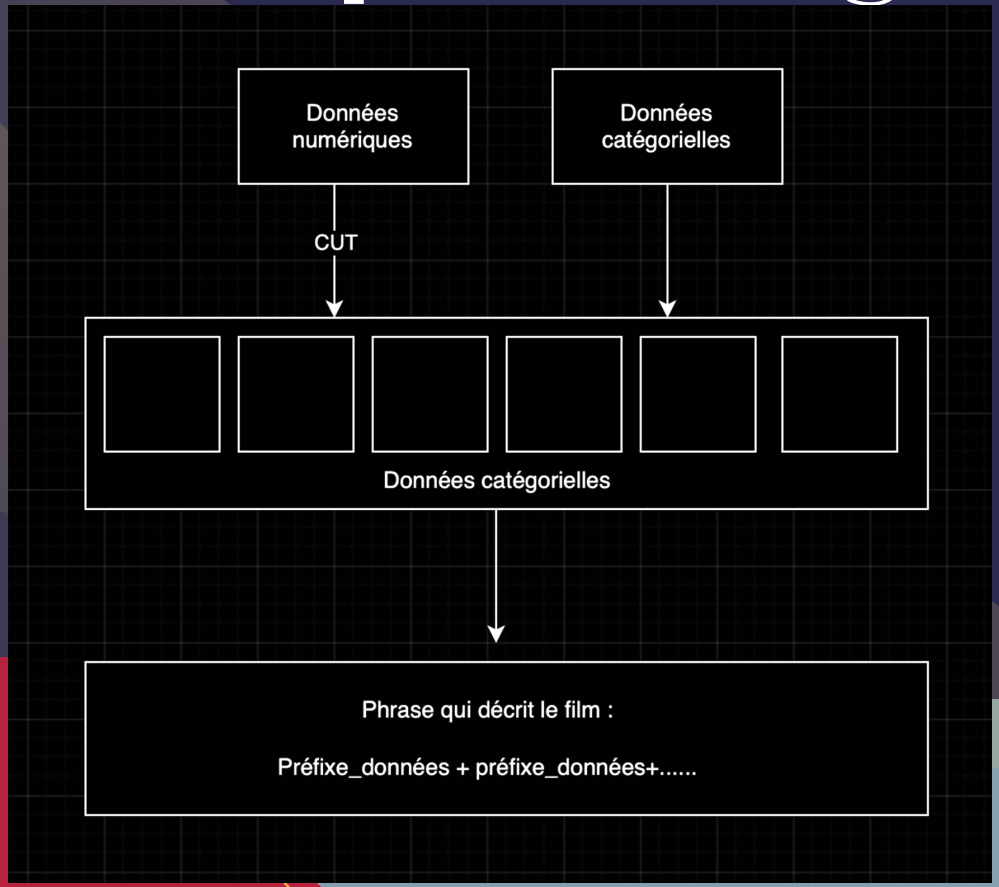
MVC


Model
PosgreSQL, etc
Controleur
Serveur associé à un langage Backend (Python)
Vue
HTML/CSS
Base de données
Modèle entrainé
Controleur
API
Interface
S'authentifier
Noter les films
Recevoir une recommandation
Utilisateur

Base de données



Préprocessing

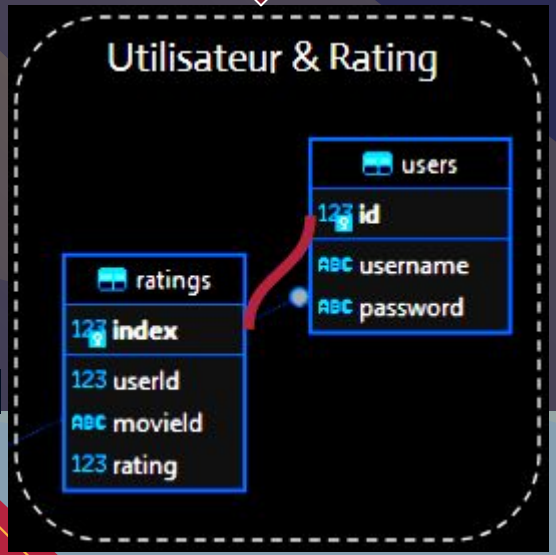


Classes ↔ ORM

USER

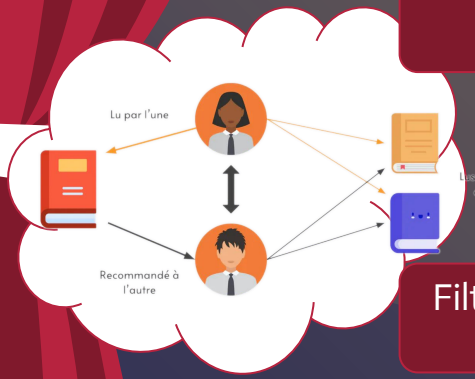
SQLAlchemy

RATING

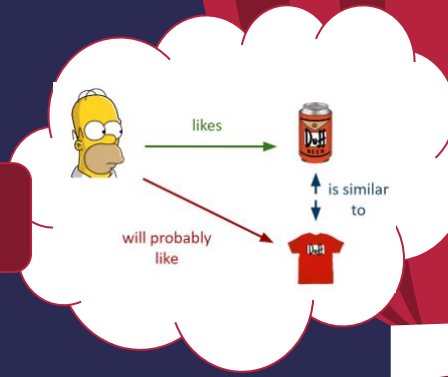



Flask
03
API
vs
FastAPI

Algorithme de recommandation



Similarité basée sur la description d'un film
(Cosinus Similarity)



Filtrage collaboratif basé sur la similarité avec d'autres utilisateurs
(KNN)

Mixage des recommandations selon l'ordre d'apparition

Flask

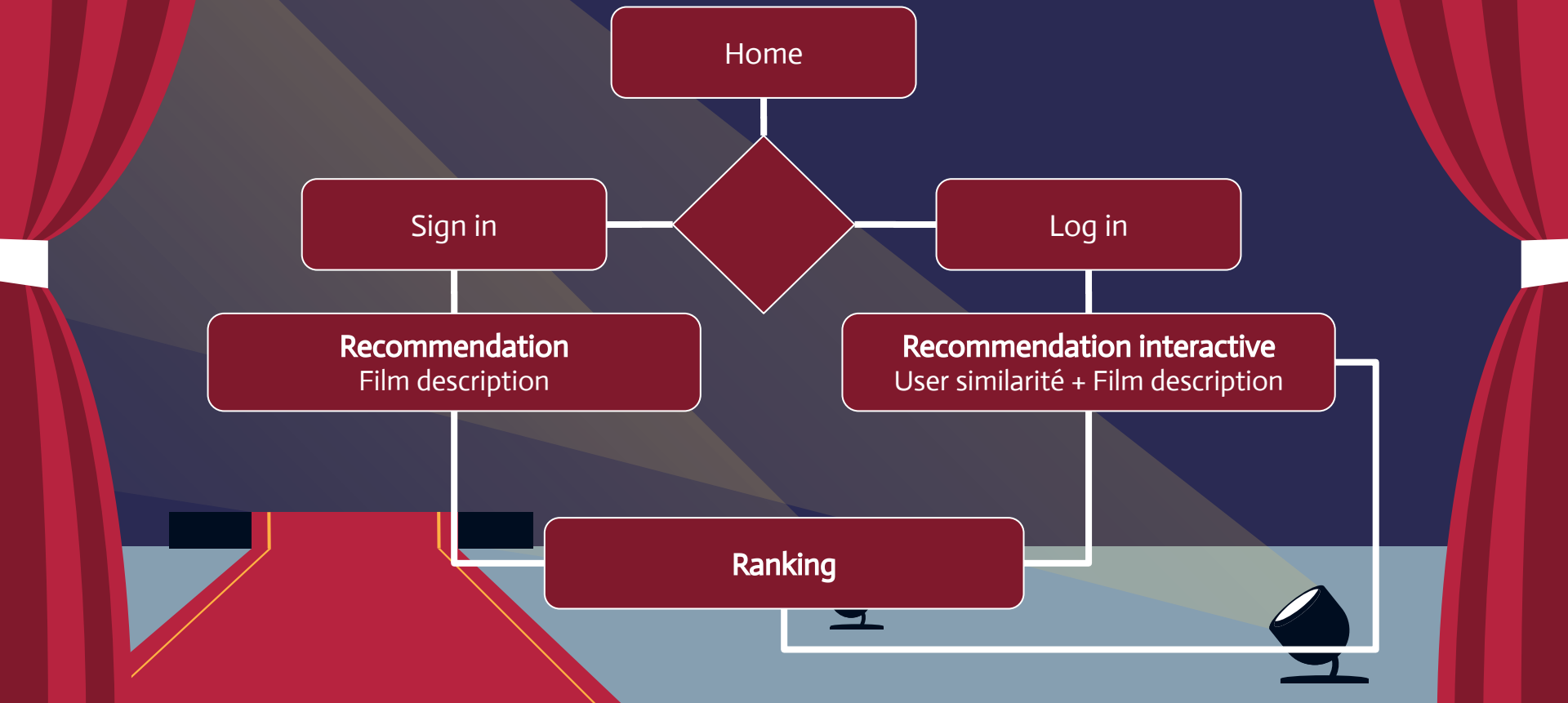
- Simple et minimaliste.
- Plus établi avec une grande communauté.
- Idéal pour les petites applications et les projets rapides.

FastAPI

- Meilleure performance.
- Plus difficile d'utilisation
- Convient aux applications nécessitant une haute performance.


And the winner is :
Flask

Arborescence des vues



Tests unitaires

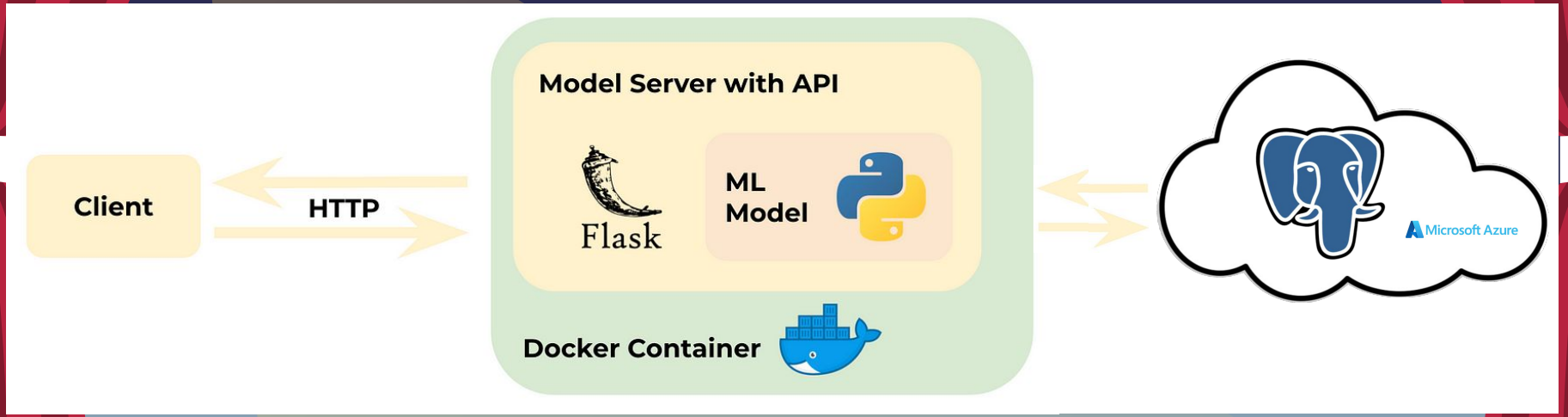
test_tmdb.py > TestMoviePosterUrl > test_get_movie_poster_url

```python
import unittest
from tmdb import get_movie_poster_url

class TestMoviePosterUrl(unittest.TestCase):

    tabnine: test | explain | document | ask
    def test_get_movie_poster_url(self):
        tmdb_id = 'tt00000'
        expected_result = {'image': '/static/image/nopicture.jpg',
                           'synop': 'Aucune description est disponible',
                           'video': ''}
        result = get_movie_poster_url(tmdb_id)
        self.assertEqual(result, expected_result)

if __name__ == '__main__':
    unittest.main()
```

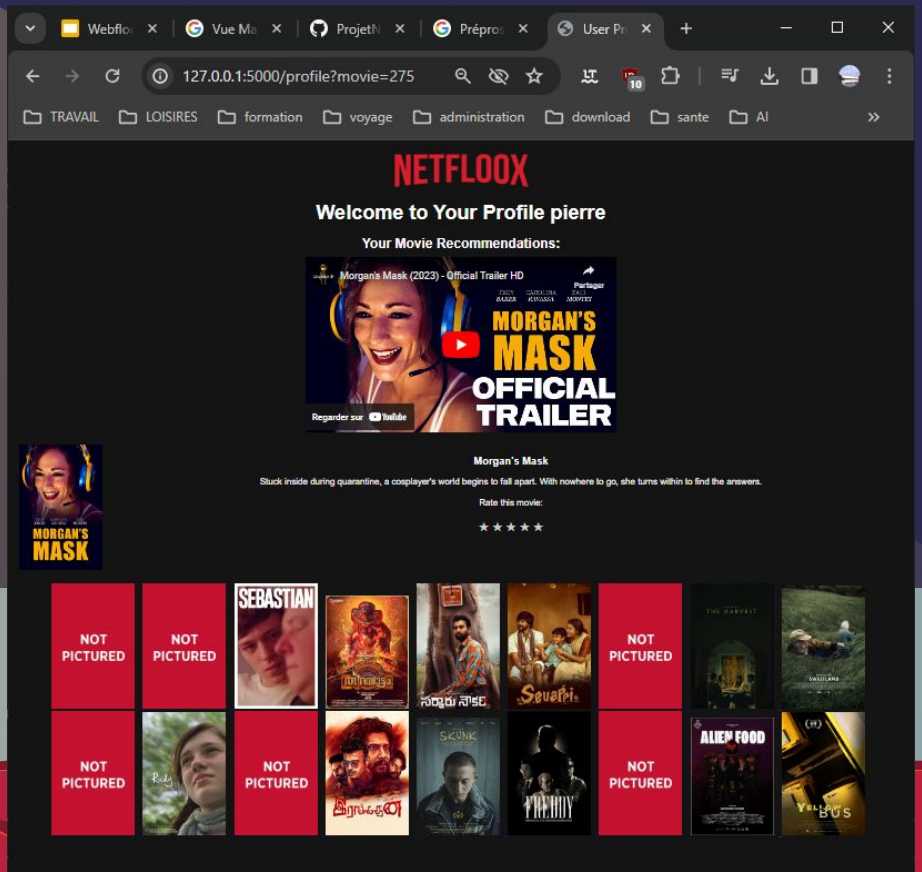
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(.venv) C:\partage\code\Webfloox_Groupe3>c:/partage/code/Webfloox_Groupe3/.venv/Scripts/python.exe
.
----------------------------------------------------------------------
Ran 1 test in 1.070s

OK
```

Déploiement



Application



04 Démo Application