

Rapport de projet intégré

Application Web de recommandation de films

Réalisé par :

Bouawda Zakaria
Ouahabi Reda
Moukrim Hamza
Naciri Yassine

Encadré par :

Mme BELOUADHA Fatima-Zahra

Table Des Matières

Table Des Matières	3
I-Introduction.....	4
II-Présentation et analyse de l'application	5
1) Présentation de l'application	5
2) Acteurs de l'application	5
3) Spécifications fonctionnels	5
4) Diagramme des cas d'utilisations	7
5) Description des cas d'utilisations	8
6) Diagramme de classes	10
III-Réalisation et mise en œuvre	11
1) Solutions proposées	11
2) Langages utilisés.....	12
3) Logiciels utilisés.....	14
4) présentation de l'algorithme de recommandation.....	16
5) Interfaces.....	19
IV-Conclusion	25

I-Introduction

Un grand nombre de systèmes de recommandation existent dans divers domaines. Leur objectif est de filtrer et d'adapter les informations pour chaque utilisateur.

Les systèmes de recommandation ont été utilisés afin de faire face au problème de surcharge et de richesse d'informations disponibles notamment à travers le Web ou les e-services. Les systèmes de recommandation visent à proposer à un utilisateur actif une ou des recommandations d'items susceptibles de l'intéresser. Ces recommandations peuvent concerner un article à lire, un livre à commander, un film à regarder, un restaurant à choisir, etc.

Les méthodes généralement utilisées pour le calcul de la recommandation sont soit basées sur le contenu soit sur la similarité de l'utilisateur avec les autres utilisateurs (approches collaboratives). Dans ce mémoire nous proposons un système de recommandation des films basé sur le filtrage collaboratif, qui permet de calculer la similarité entre service Web sémantique, dans le but de générer des recommandations de films pour un utilisateur cible.

II-Présentation et analyse de l'application

1) Présentation de l'application

L'application doit permettre aux utilisateurs de consulter des films à travers une page d'accueil. Ils peuvent aussi choisir de consulter les films par classement ou par genre.

Un utilisateur peut créer un compte et s'authentifier pour bénéficier de plus de fonctionnalités. Parmi ces fonctionnalités, on trouve l'ajout de commentaires et l'ajout d'un films aux favoris (que l'utilisateur peut consulter plus tard).

Le classement d'un film est déduit de la note moyenne que les utilisateurs lui ont donnée. Un utilisateur peut aussi consulter des recommandations de films à travers une page « recommandations ». Les recommandations d'un utilisateur sont basées sur ses films favoris.

2) Acteurs de l'application

Les acteurs de l'application sont :

- Utilisateur anonyme
- Client

3) Spécifications fonctionnels

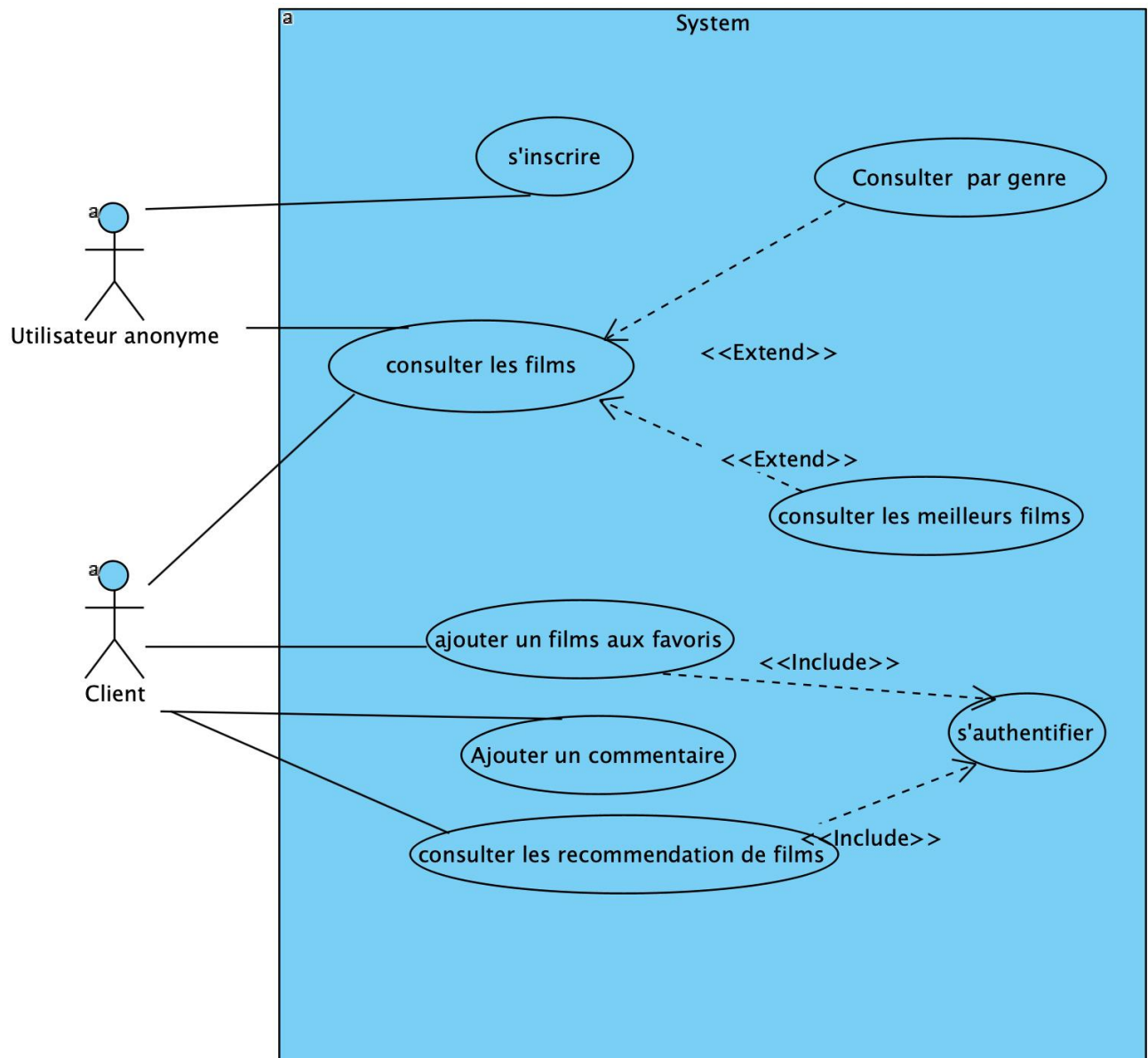
L'application doit permettre aux clients de:

- S'authentifier
- Consulter les films
- Consulter les films par genre
- Consulter les films par classements
- Ajouter des films au favoris
- Ajouter des commentaires
- Consulter des suggestions de films basée sur leurs films favoris

L'application doit permettre aux utilisateurs anonymes:

- S'inscrire
- Consulter les films
- Consulter les films par genre
- Consulter les films par classements

4) Diagramme des cas d'utilisations



5) Description des cas d'utilisations

a) S'inscrire

Use Case	S'inscrire
Description	Un utilisateur s'inscrit pour profiter de toutes les fonctionnalités de l'application
Acteur principal	Utilisateur anonyme
Acteur secondaire	-
Préconditions	-
Post conditions	Un compte utilisateur est créé pour l'utilisateur

b) Consulter les films

Use Case	Consulter les films
Description	Un utilisateur anonyme ou un client consulte la liste des films disponibles
Acteurs principaux	Utilisateur anonyme - client
Acteur secondaire	-
Préconditions	-
Post conditions	L'utilisateur a pu consulter la liste des films

c) Ajouter un film aux favoris

Use Case	Ajouter un film aux favoris
Description	Un client sélectionne un film et l'ajoute à sa liste des favoris
Acteur principal	Client
Acteur secondaire	-
Préconditions	Un client est authentifié dans le système et consulte un film.
Post conditions	Le film est ajouté à la liste des favoris de l'utilisateur

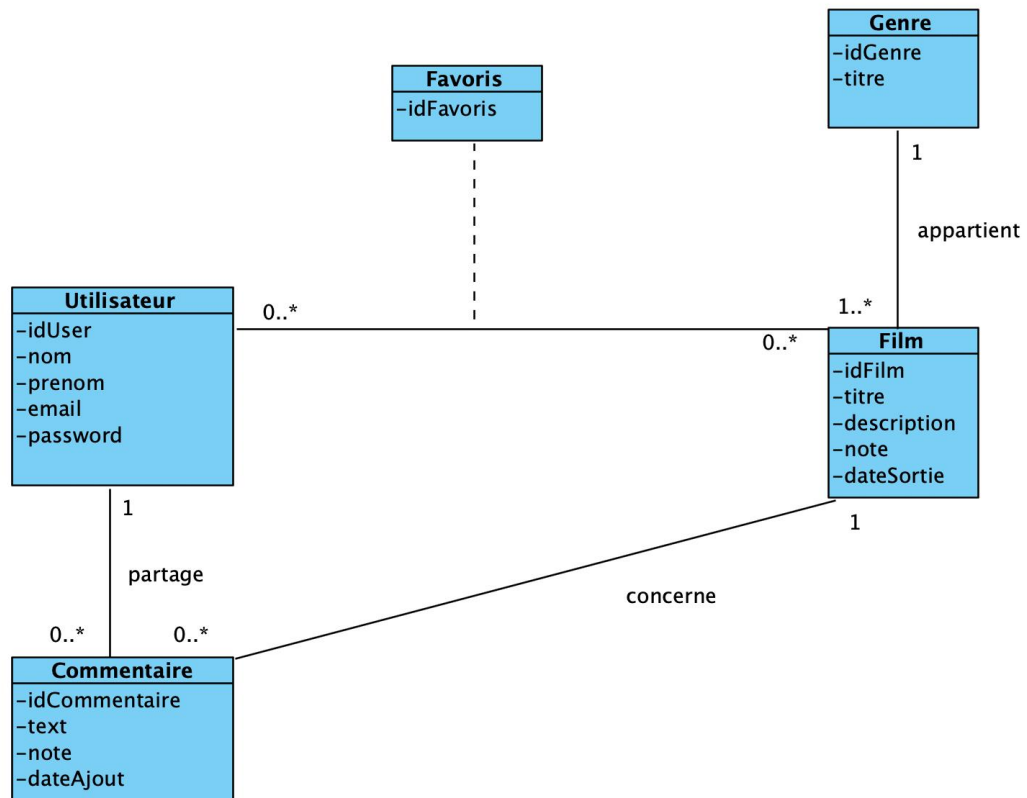
d) Ajouter un commentaire

Use Case	Ajouter un film aux favoris
Description	Un client sélectionne un film et partage un commentaire sur le film et donne une note
Acteur principal	Client
Acteur secondaire	-
Préconditions	Un client est authentifié dans le système et consulte un film.
Post conditions	Le commentaire est ajouté et est visible aux autres utilisateurs La note moyenne du film est actualisée Le classement des films est actualisé

e) Consulter les recommandations de films

Use Case	Consulter les recommandations de films
Description	Un client souhaite voir des suggestions de films basées sur ses films favoris
Acteur principal	Client
Acteur secondaire	-
Préconditions	Un client est authentifié dans le système
Post conditions	Le client a pu consulter la liste des films suggérés

6) Diagramme de classes



III-Réalisation et mise en œuvre

1) Solutions proposées

L'application est sous forme d'une application web. Nous avons opté pour une architecture client-serveur pour sa sécurité et la centralisation des ressources.

Pour la partie serveur, nous avons choisi de la développer à l'aide de node.js. La partie client est développée à l'aide de Angular et la base de données est réalisée avec mongoDB :

a) Node.js

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'événements, et implémente sous licence MIT les spécifications CommonJS.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des serveurs webs tels que Nginx ou Apache.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.



b) Angular

Angular est un framework côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés.

Angular est une réécriture complète d'AngularJS, cadriciel construit par la même équipe. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités.



C)MongoDb

Apparue au milieu des années 2000, MongoDB est une base de données NoSQL orientée document. Elle est utilisée pour le stockage de volumes massifs de données.

Contrairement à une base de données relationnelle SQL traditionnelle, MongoDB ne repose pas sur des tableaux et des colonnes. Les données sont stockées sous forme de collections et de documents.

Les documents sont des paires de valeurs / clés servant d'unité de données de base. Les collections quant à elles contiennent des ensembles de documents et de fonctions. Elles sont l'équivalent des tableaux dans les bases de données relationnelles classiques.



2) Langages utilisés

- A) HTML5: est un langage de base pour la création de site internet, il sert à structurer votre document. D'autres langages peuvent s'ajouter lors de la conception, mais tous les sites web contiennent du HTML, langage essentiel dans le développement web. Constitue le corps des pages.

HTML



- B) CSS : Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en html. (css, 2021)

CSS



- C) Python: Python (prononcé /pi.t5/) est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.



- D) TypeScript : Le TypeScript est un langage de programmation développé par Microsoft en 2012. Son ambition principale est d'améliorer la productivité de développement d'applications complexes. C'est un langage open source, développé comme un sur-ensemble de Javascript. Tout code valide en Javascript l'est également en TypeScript. (introduction à typescript, 2021)



3) Logiciels utilisés

B) Google Chrome

Un navigateur web propriétaire développé par Google depuis 2008, basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS. (chrome, 2021)



C) VsCode

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. (VsCode, 2021)



D) Postman

Postman sert à exécuter des appels HTTP directement depuis une interface graphique. Vous pourrez simplement choisir l'URL, la méthode HTTP (le plus souvent GET, POST, PUT, PATCH et DELETE), les headers, les query params et dans certains cas le body de la requête. (postman, 2021)



E) Visual paradigm

Visual Paradigm est une application complète de conception et de gestion UML pour le développement informatique d'entreprise qui prend en charge C#. NET, VB.NET, et C++.NET. (visual paradigm, 2021)



F) PyCharm

PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.



4) présentation de l'algorithme de recommandation

Nous construisons un modèle de système de recommandation simple et également un autre système de recommandation basé sur le contenu. Bien que ces modèles soient loin d'être conformes aux normes de l'industrie en termes de complexité, de qualité ou de précision.

Les systèmes de recommandation font partie des applications les plus populaires de la science des données aujourd'hui. Ils sont utilisés pour prédire la "note" ou la "préférence" qu'un utilisateur donnerait à un article. Presque toutes les grandes entreprises technologiques les ont appliquées sous une forme ou une autre. Amazon l'utilise pour suggérer des produits aux clients, YouTube l'utilise pour décider quelle vidéo lire ensuite en lecture automatique, et Facebook l'utilise pour recommander des pages à aimer et des personnes à suivre.

De plus, pour certaines entreprises comme Netflix, Amazon Prime, Hulu et Hotstar, le modèle commercial et son succès tournent autour de la puissance de leurs recommandations. Netflix a même offert un million de dollars en 2009 à quiconque pourrait améliorer son système de 10 %.

Il existe également des systèmes de recommandation populaires pour des domaines tels que les restaurants, les films et les rencontres en ligne. Des systèmes de recommandation ont également été développés pour explorer les articles de recherche et les experts, les collaborateurs et les services financiers. YouTube utilise le système de recommandation à grande échelle pour nous suggérer des vidéos en fonction de notre historique. Par exemple, si nous regardons beaucoup de vidéos éducatives, cela suggérerait ces types de vidéos.

Mais quels sont ces systèmes de recommandation ?

De manière générale, les systèmes de recommandation peuvent être classés en 3 types :

Recommandeurs simples : offrez des recommandations générales à chaque utilisateur, en fonction de la popularité et/ou du genre du film. L'idée de base derrière ce système est que les films qui sont plus populaires et acclamés par la critique auront une plus grande probabilité d'être aimés par le public moyen. Un exemple pourrait être IMDB Top 250. Et en fait, dans la partie **Trending**, qu'on va vous montrer par la suite, c'est exactement un système de recommandation simple avec les films les mieux classés !

Recommandeurs basés sur le contenu : suggérez des éléments similaires en fonction d'un élément particulier. Ce système utilise des métadonnées d'éléments, telles que le genre, le réalisateur, la description, les acteurs, etc. pour les films, pour faire ces recommandations. L'idée générale derrière ces systèmes de recommandation est que si une personne aime un article particulier, elle aimera également un article qui lui est similaire. Et pour le recommander, il utilisera les métadonnées des éléments passés de l'utilisateur. Un bon exemple pourrait être YouTube, où en fonction de notre historique, il nous suggère de nouvelles vidéos que nous pourrions potentiellement regarder.

Moteurs de filtrage collaboratifs : ces systèmes sont largement utilisés et tentent de prédire la note ou la préférence qu'un utilisateur donnerait à un élément en fonction des notes et des préférences passées des autres utilisateurs. Les filtres collaboratifs ne nécessitent pas de métadonnées d'élément comme leurs homologues basés sur le contenu. Recommandeurs simples.

Comme décrit dans la section précédente, les recommandations simples sont des systèmes de base qui recommandent les meilleurs éléments en fonction d'une certaine métrique ou d'un score. Dans cette section, nous allons créer un clone simplifié des 250 meilleurs films IMDB à l'aide des métadonnées collectées à partir d'IMDB.

Voici les étapes impliquées :

Décidez de la métrique ou du score sur lequel évaluer les films.

Calculez le score de chaque film.

Triez les films en fonction du score et affichez les meilleurs résultats.

À propos de l'ensemble de données

Les fichiers de l'ensemble de données contiennent des métadonnées pour les 45 000 films répertoriés dans l'ensemble de données Full MovieLens. L'ensemble de données se compose de films sortis au plus tard en juillet 2017. Cet ensemble de données capture des éléments tels que la distribution, l'équipe, les mots-clés de l'intrigue, le budget, les revenus, les affiches, les dates de sortie, les langues, les sociétés de production, les pays, le nombre de votes TMDb et les moyennes des votes.

Ces points de fonctionnalité pourraient être potentiellement utilisés pour former nos modèles d'apprentissage automatique pour le filtrage de contenu et collaboratif.

Ce jeu de données se compose des fichiers suivants :

`movies_metadata.csv` : ce fichier contient des informations sur environ 45 000 films présentés dans l'ensemble de données Full MovieLens. Les fonctionnalités incluent les affiches, les arrière-plans, le budget, le genre, les revenus, les dates de sortie, les langues, les pays de production et les entreprises.

`keywords.csv` : contient les mots-clés de l'intrigue du film pour nos films MovieLens. Disponible sous la forme d'un objet JSON "stringified".

`credits.csv` : contient des informations sur les acteurs et l'équipe de tous les films. Disponible sous la forme d'un objet JSON "stringified".

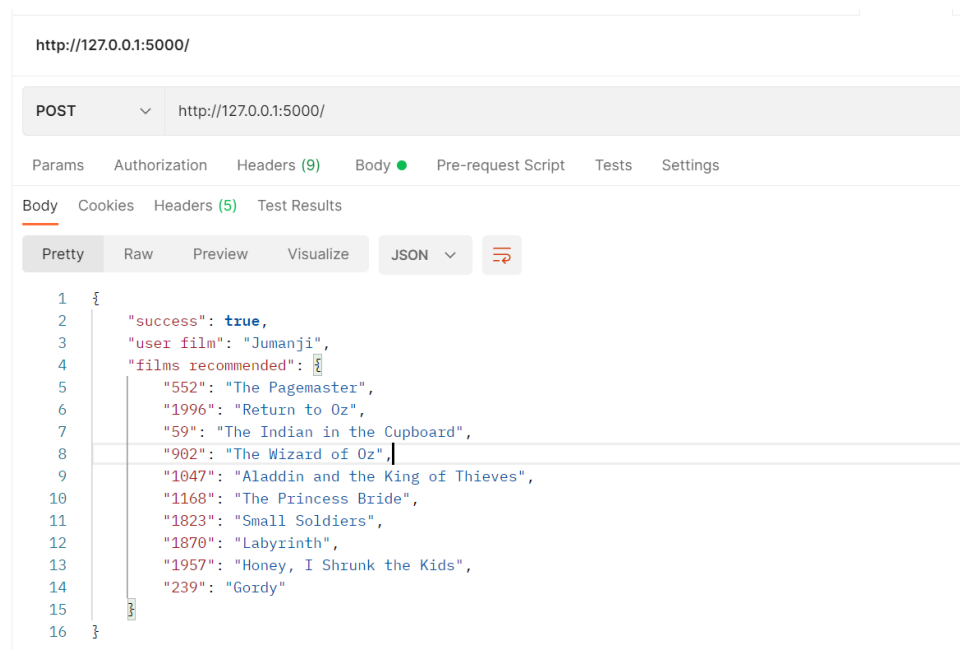
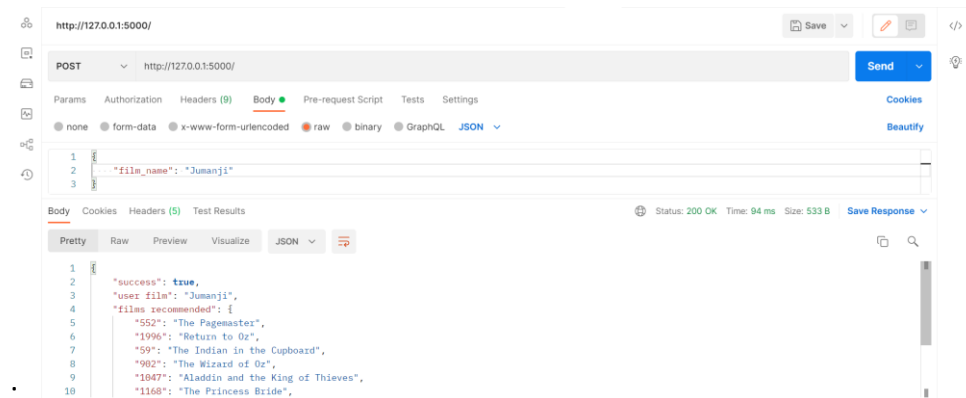
`links.csv` : ce fichier contient les identifiants TMDb et IMDB de tous les films présentés dans l'ensemble de données Full MovieLens.

`links_small.csv` : contient les identifiants TMDb et IMDB d'un petit sous-ensemble de 9 000 films de l'ensemble de données complet.

`ratings_small.csv` : le sous-ensemble de 100 000 évaluations de 700 utilisateurs sur 9 000 films.

L'ensemble de données Full MovieLens comprend 26 millions d'évaluations et 750 000 applications de balises, provenant de 270 000 utilisateurs sur les 45 000 films de cet ensemble de données. Il est accessible depuis le site officiel de GroupLens.

Alors on teste le content_based recommandation system, en envoyant le nom d'un film 'Jumanji', et il va nous répondre avec les 10 films, qui ont une relation avec ce film, en appliquant les algorithmes du machine Learning (train the model using Term Frequency-Inverse Document Frequency (TF-IDF)) et NLP sur les données et les métadonnées qu'on a dans le dataset:



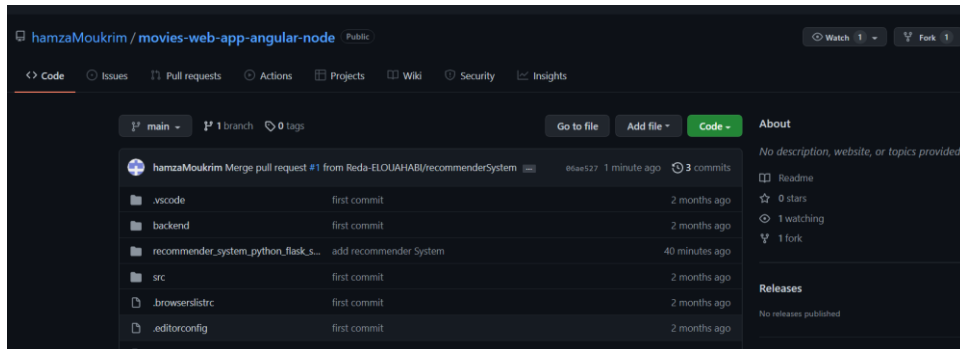
```

* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Jumanji
data      b'\r\n      "film_name": "Jumanji"\r\n}' <class 'pandas.core.series.Series'>
127.0.0.1 - - [17/Jan/2022 19:47:10] "POST / HTTP/1.1" 200 -

```

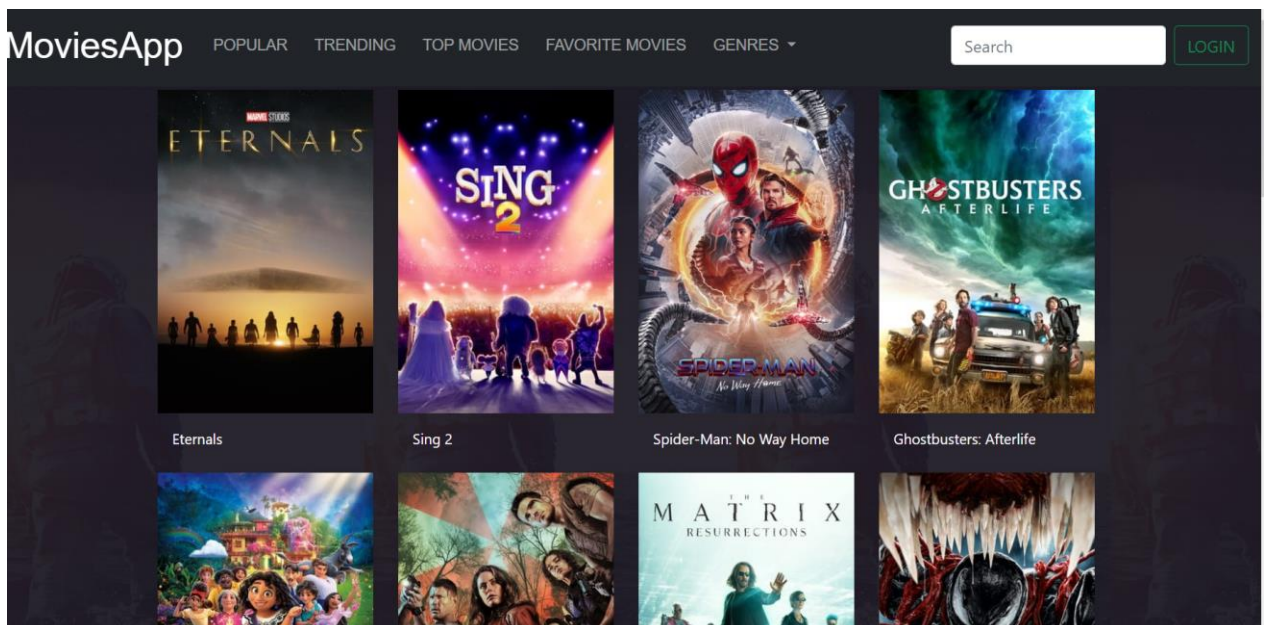
Pour le code source de toute l'application y inclus le système de recommandation, vous le trouverez sur :

https://github.com/hamzaMoukrim/movies-web-app-angular-node/tree/main/recommender_system_python_flask_sklearn

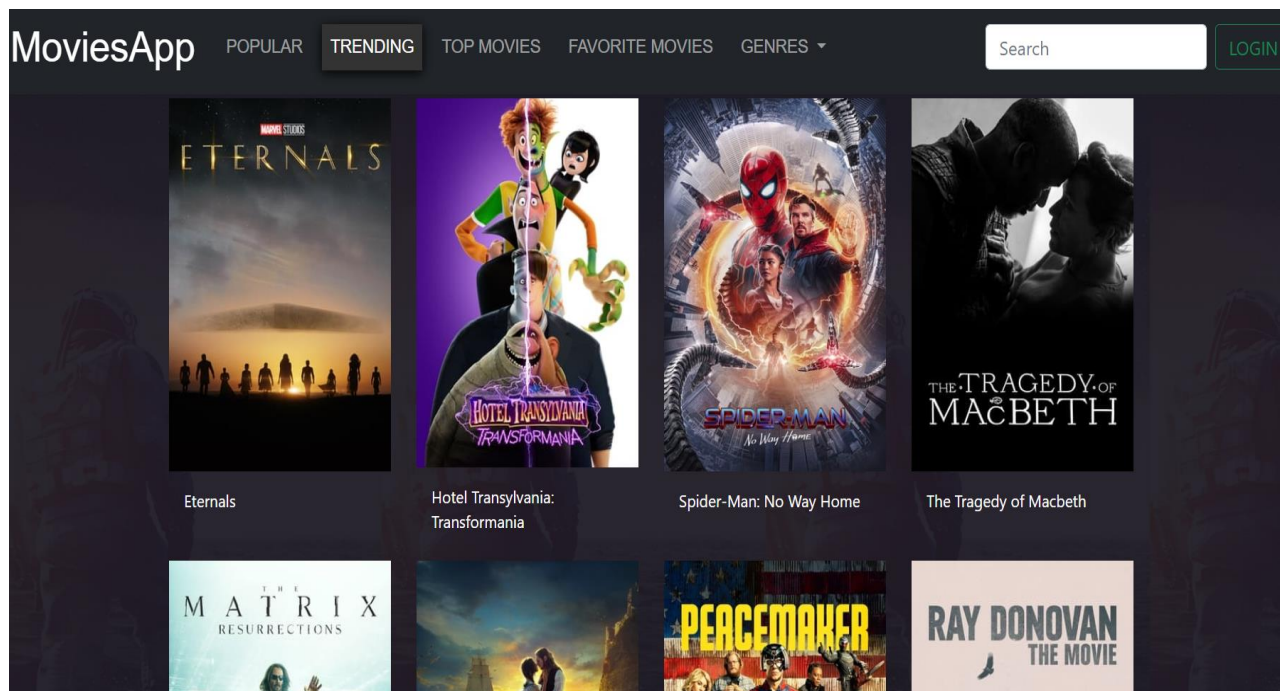


5) Interfaces

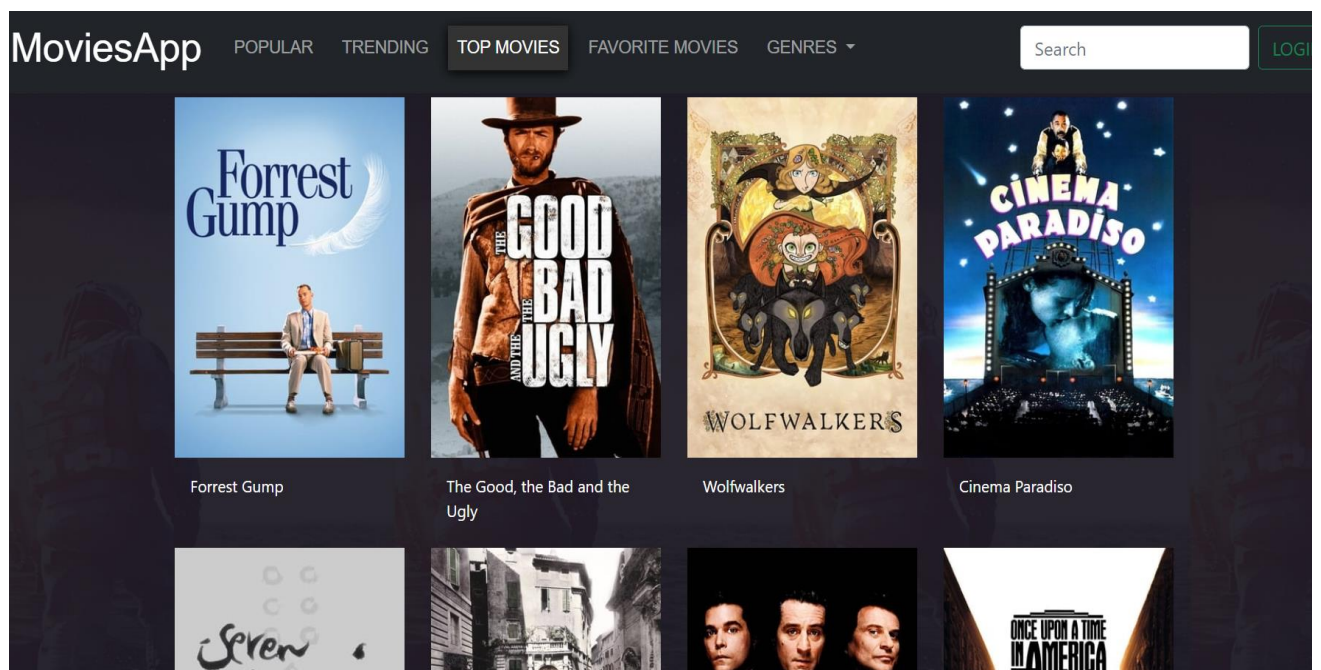
On trouve dans la première page de notre application deux cas : si l'utilisateur anonyme l'application affiche les films qui sont populaire et les plus pertinents. Et si l'utilisateur déjà fait le login, alors on va afficher à cet utilisateur les films recommandés à lui, en se basant sur la liste des favoris de cet utilisateur.



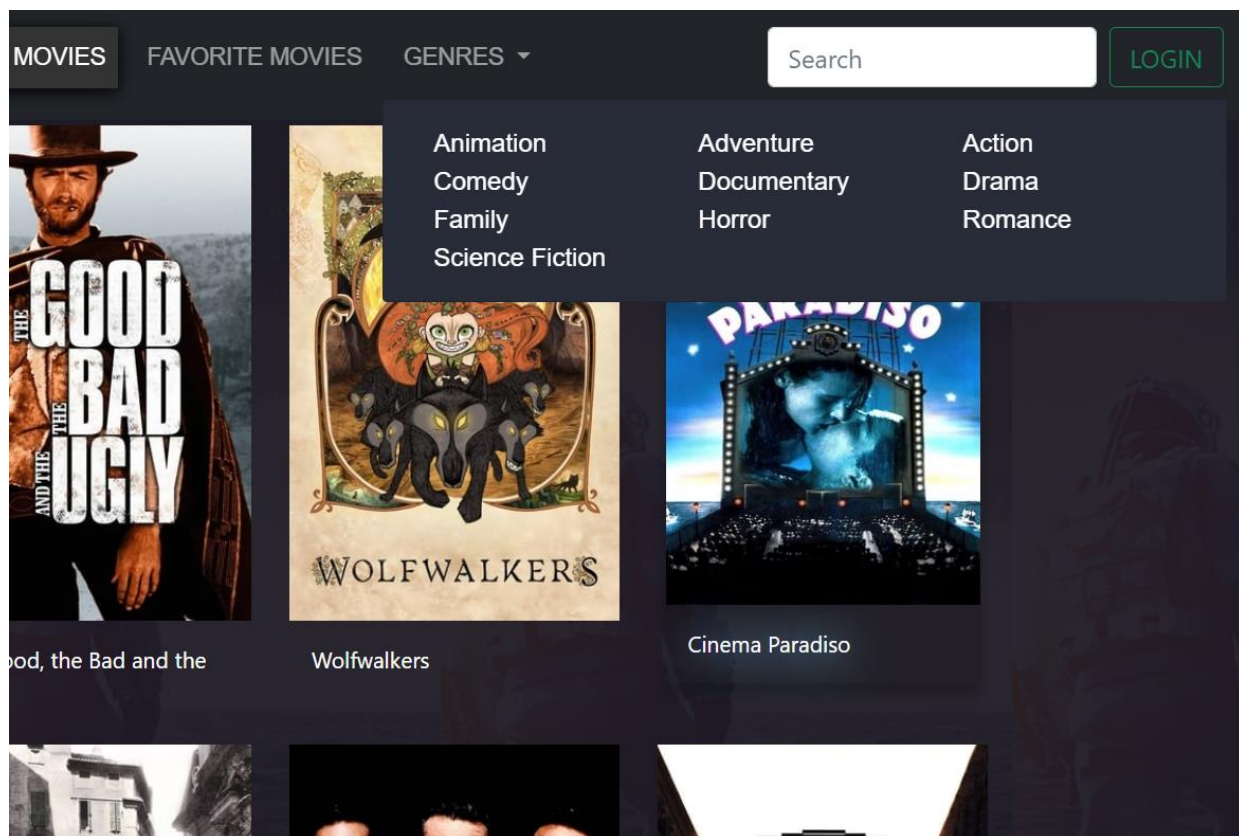
L'utilisateur soit anonyme ou non peut voir aussi la liste des films qui sont en **tendance** :



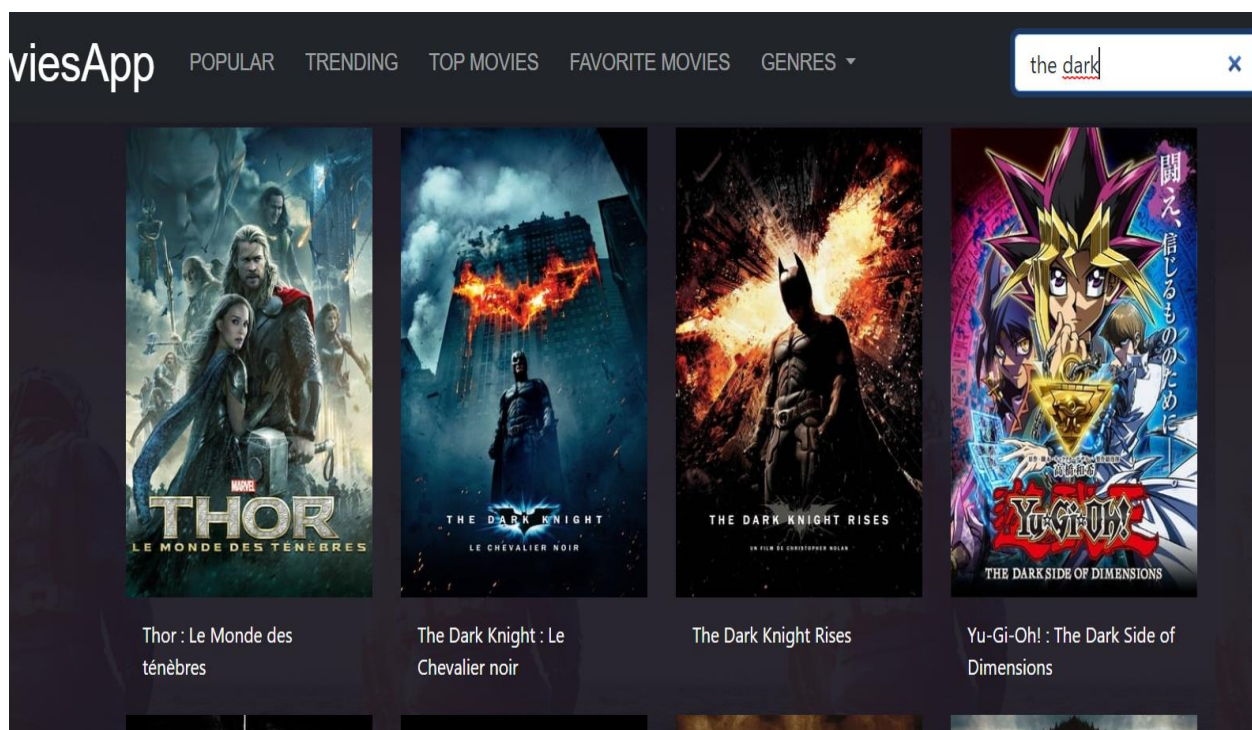
Et aussi les meilleurs films de tous les temps :



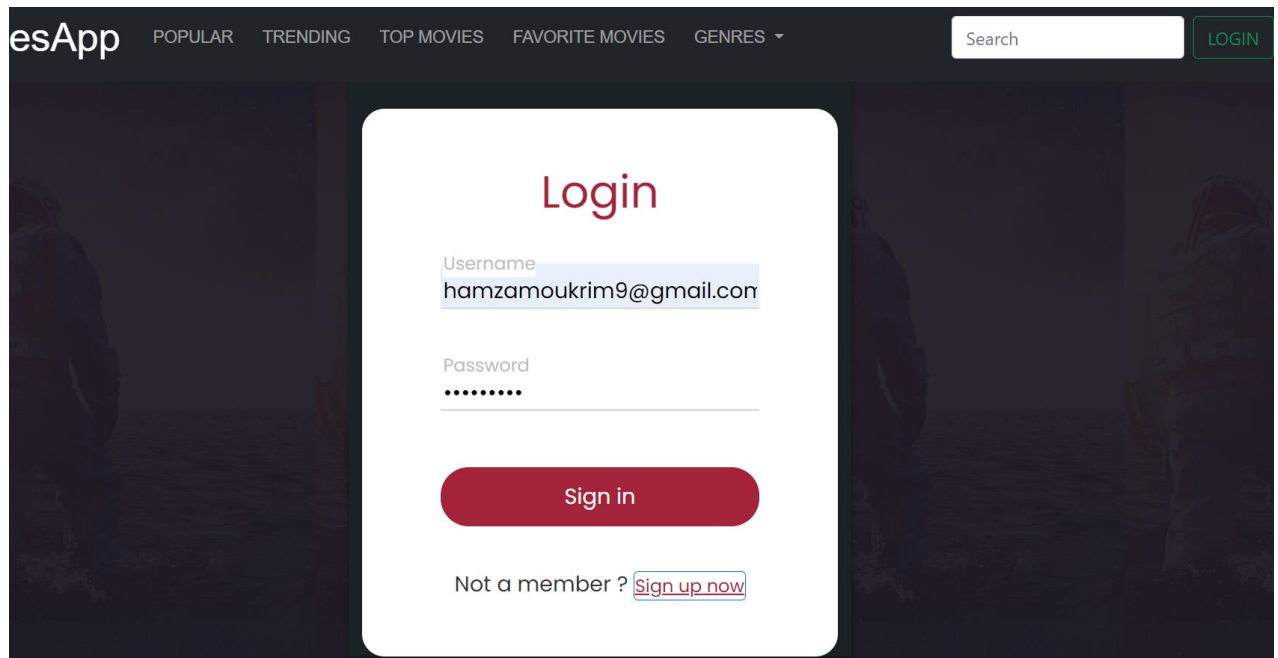
Il peut aussi filtrer les films selon les types des films, comme les films d'animation, de science-fiction, Drama...



Chercher un film :



Il peut voir également la liste de ces films favoris, mais à condition s'il a déjà fait le login dans notre application, sinon il doit faire le login si la un compte :



The image shows a login form for 'esApp'. The header includes the logo 'esApp' and navigation links: POPULAR, TRENDING, TOP MOVIES, FAVORITE MOVIES, and GENRES. A search bar and a LOGIN button are also present. The login form itself is centered and contains the title 'Login', a Username field with the value 'hamzamoukrim9@gmail.com', a Password field with masked characters, a 'Sign in' button, and a link 'Sign up now' for users who are not members.

esApp POPULAR TRENDING TOP MOVIES FAVORITE MOVIES GENRES

Search LOGIN

Login

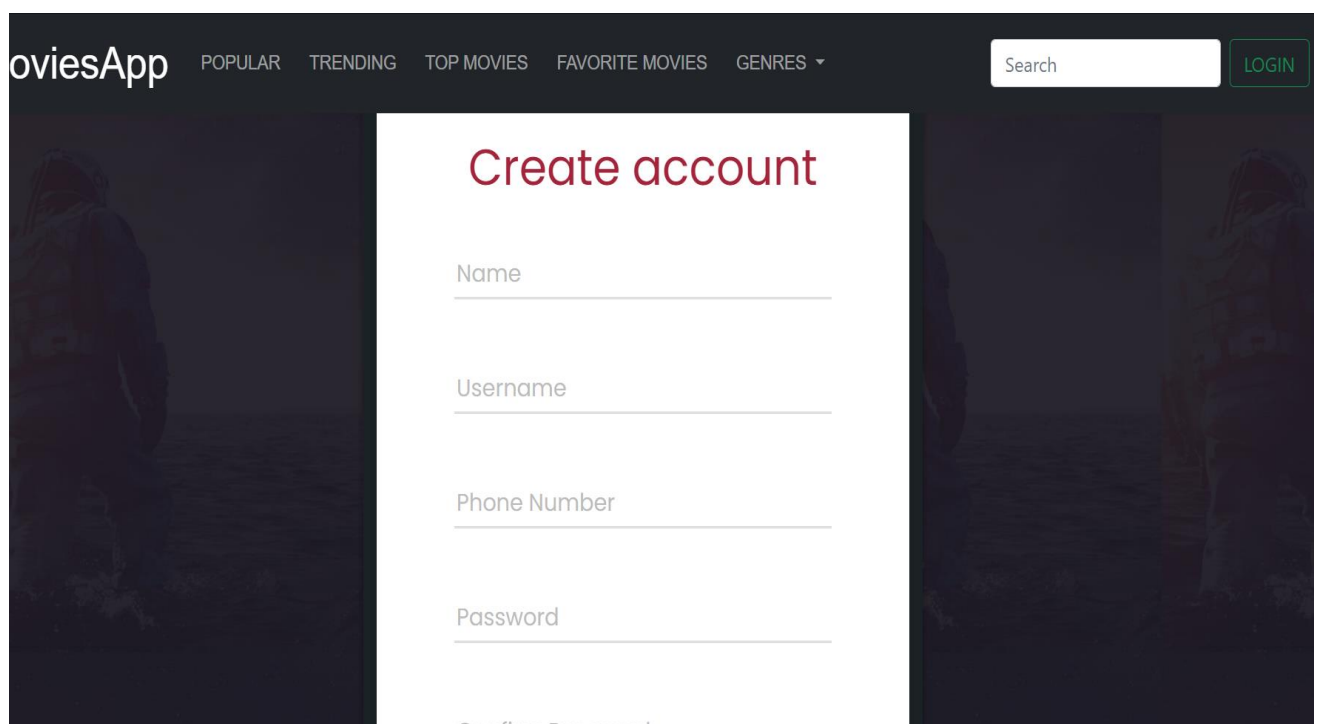
Username
hamzamoukrim9@gmail.com

Password
.....

Sign in

Not a member ? [Sign up now](#)

Sinon il peut créer un nouveau compte :



The image shows a 'Create account' form for 'moviesApp'. The header includes the logo 'oviesApp' and navigation links: POPULAR, TRENDING, TOP MOVIES, FAVORITE MOVIES, and GENRES. A search bar and a LOGIN button are also present. The form contains fields for Name, Username, Phone Number, and Password, followed by a 'Confirm Password' field.

oviesApp POPULAR TRENDING TOP MOVIES FAVORITE MOVIES GENRES

Search LOGIN

Create account

Name

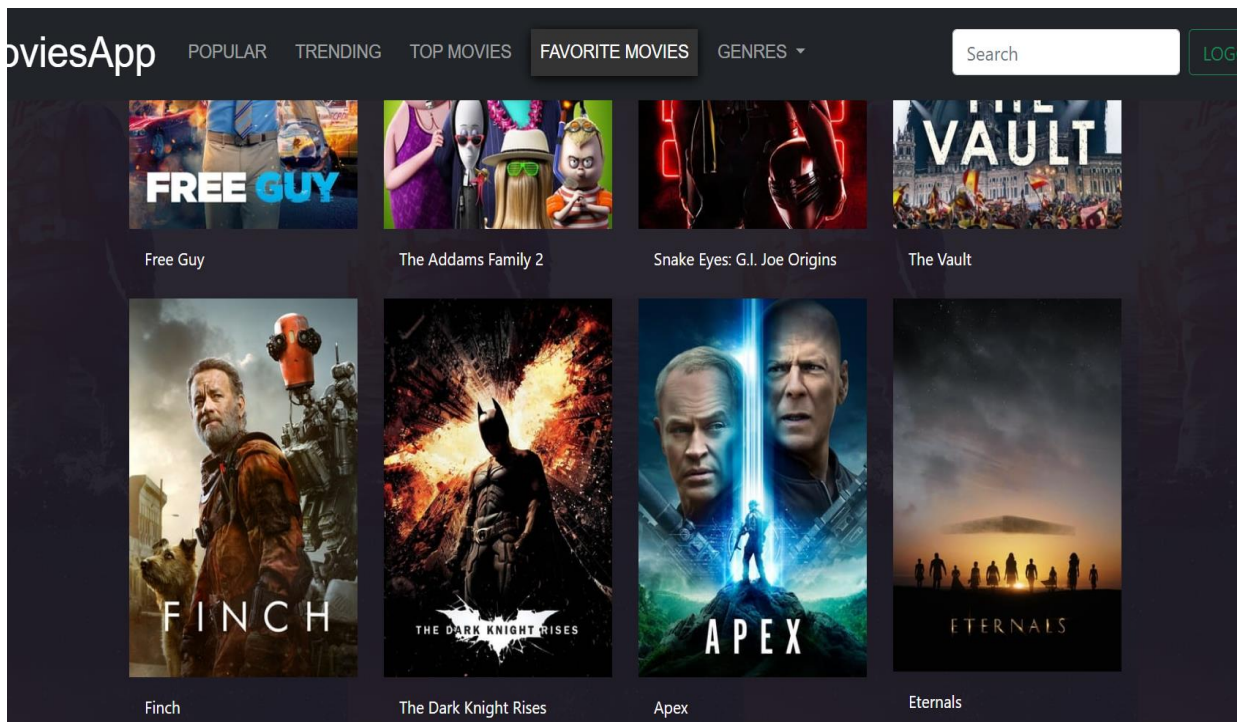
Username

Phone Number

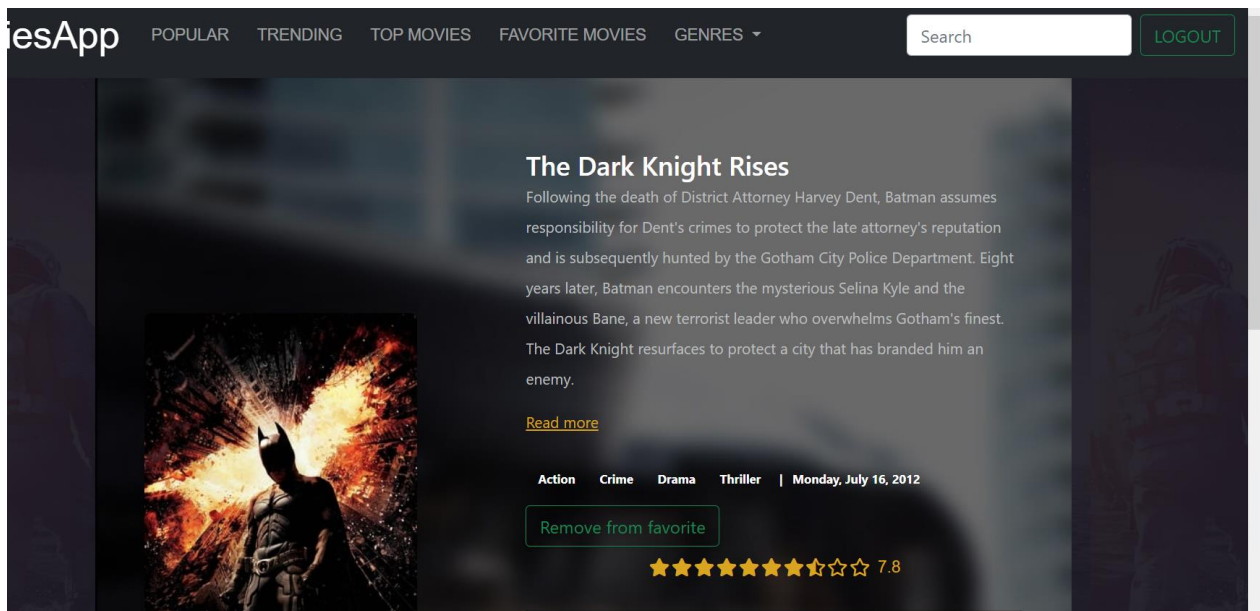
Password

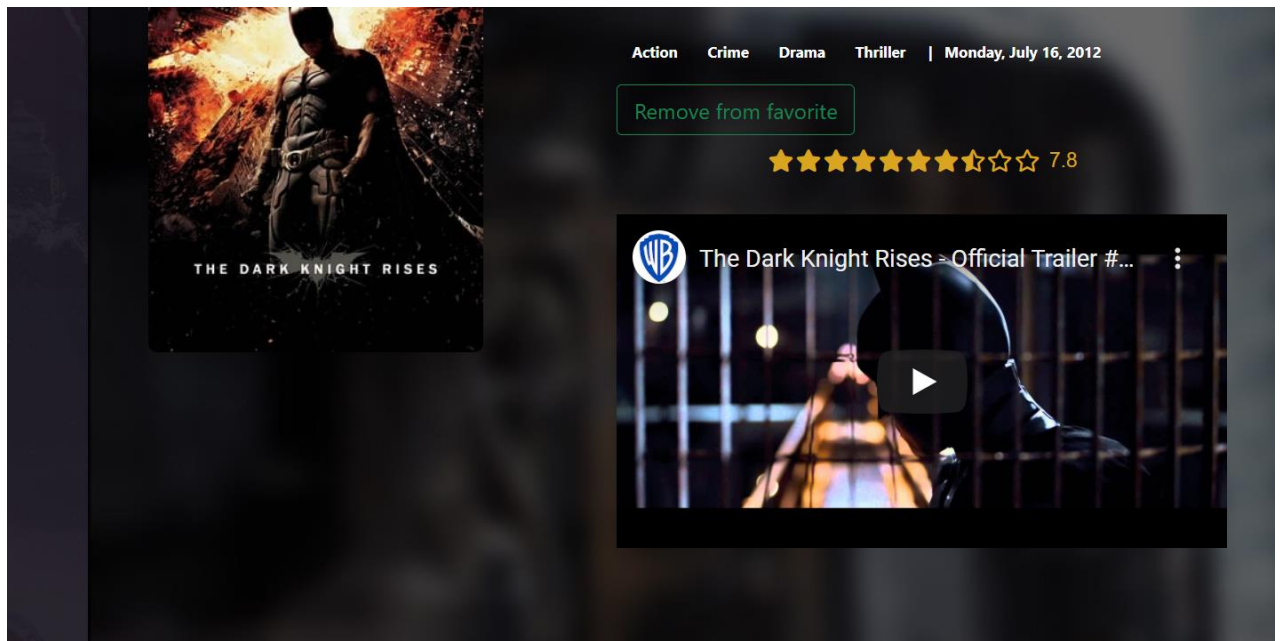
Confirm Password

Et voilà l'interface de la page de ces films favoris :

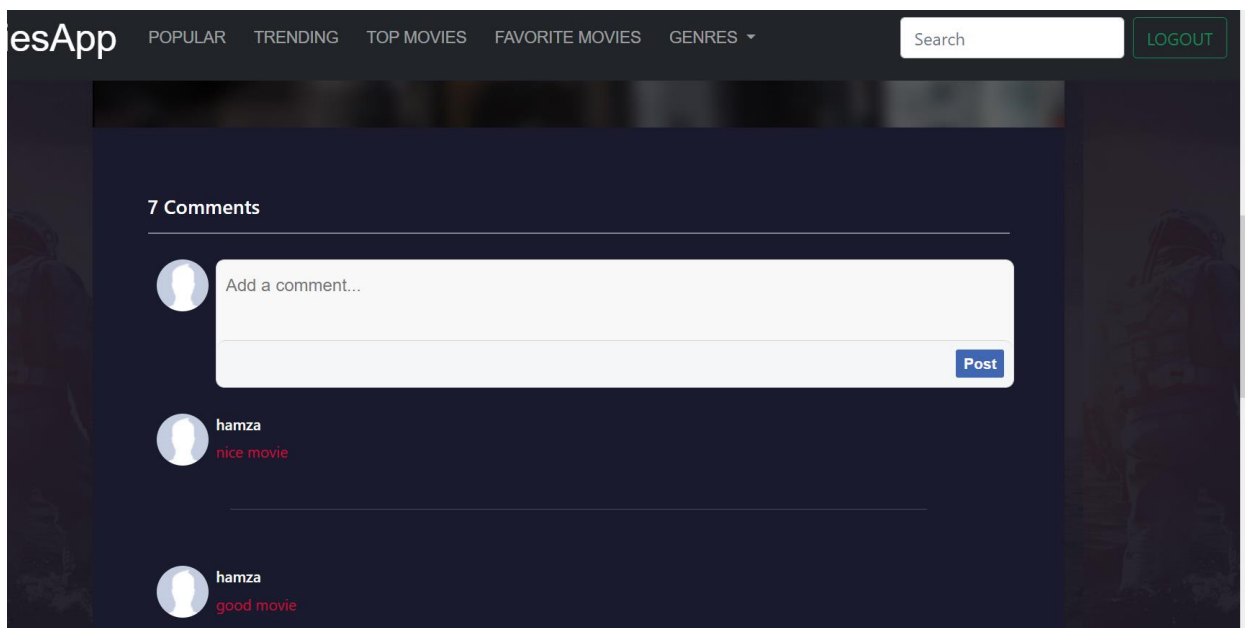


Il peut voir aussi les détails de chaque film, comme la description et le trailler, rating, aussi il peut ajouter ou enlever un film a sa liste des favoris :





Et enfin il peut commenter un film :



IV-Conclusion

Les systèmes de recommandation automatique sont devenus, à l'instar des moteurs de recherche, un outil incontournable pour tout site Web focalisé sur un certain type d'articles disponibles dans un catalogue riche, que ces articles soient des objets, des produits culturels (livres, films, morceaux de musique, etc.), des éléments d'information (news) ou encore simplement des pages (liens hypertextes). L'objectif de ces systèmes est de sélectionner, dans leur catalogue, les items les plus susceptibles d'intéresser un utilisateur particulier.

Nous avons donné une vue générale sur ce domaine en introduisant la notion de recommandation de films basant sur le filtrage collaboratif.