



Rapport de projet

Par *JAST*:

Sarah Santiago

Jimmy Fabre

Thuy-Trang Nguyen

Alban Richard

Sommaire :

1. Introduction

1.1 Pictu.....
1.2 Watch if.....
1.3 Présentation du groupe.....

2. Reprise du cahier des charges

3. Répartition des tâches

4. Réalisation du Projet

4.1 Algorithme.....
4.2 Communication.....
4.3 Données.....
4.4 Interface.....
4.5 Réseau.....
4.6 Site Web.....
4.7 Installateur / Lanceur.....

5. Nos impressions

6. Visualisation de l'historique de développement

7. Conclusion

1/Introduction

Ce rapport contient toutes les informations afin de comprendre et de découvrir le projet que nous avons réalisé au cours du second semestre de notre année de SPÉ à EPITA : Une application capable de proposer un contenu aux utilisateurs.

1.1 Pictu

Au départ, nous pensions réaliser une application proposant des images par rapport aux goûts de l'utilisateur. Cette application devait s'appeler "Pictu" et avoir ce logo ci-dessous:



Or, nous nous sommes rendu compte que nous préférions proposer un autre type de contenu, ce qui a donné Watch If...

1.2 Watch If

Cette application a pour but principal de proposer des films aux utilisateurs en fonction de leur goût. Nous souhaitons garder l'aspect “social” du projet en permettant aux utilisateurs de laisser des commentaires pour donner leur avis sur les différents films ou bien même d'en rajouter des nouveaux pas encore présents sur le site.

Notre application a comme logo:



Ce projet nous permettra de développer nos connaissances et notre savoir sur le langage C. Ainsi, nous pourrons mettre en œuvre ce que nous

apprenons à Epita autrement qu'avec des travaux de programmation et l'appliquer sur un projet qui nous tient à cœur.

1.3 Présentation du groupe

Sarah Santiago:

Étudiante à l'Epita, j'ai toujours aimé programmer. Je sens que ce projet nous permettra de développer nos connaissances notamment en langage C et nous permettra de faire de nouvelles découvertes. En tout cas, j'espère que nous réussirons à réaliser notre projet.

Jimmy Fabre:

Étudiant à l'Epita, j'ai toujours aimé découvrir de nouvelles choses. Ce projet est un moyen de repousser nos limites et de voir ce qu'on est capable de faire.

Thuy Trang Nguyen:

Étudiante à l'Epita, je pense que ce projet me donnera plus d'expérience dans le travail d'équipe et comment faire un projet informatique.

Alban Richard:

Étudiant à l'Epita, j'ai hâte de commencer ce projet. En effet, je trouve ce projet très intéressant dans la réalisation. En espérant que nous arriverons à nos fins et rendre un travail de qualité.

1/Reprise du cahier des charges

Le projet:

Notre projet consiste à créer une application permettant à l'utilisateur de consommer du contenu, comme des photos ou encore des vidéos, qui serait recommandé par rapport à ses goûts. En effet, il lui sera possible de se divertir en regardant des choses qu'il apprécie, et pourra aimer ce contenu afin d'avoir un contenu similaire par la suite. le principe algorithmique de ce projet et tout l'algorithme de recommandation qui va être mis en place pour que l'utilisateur ne tombe plus que sur du contenu qu'ils souhaitent aimer.

I) Présentation du projet :

Objectif du projet

Ce projet nous permettra de développer nos connaissances et notre savoir sur le langage C. Ainsi, nous pourrons mettre en œuvre ce que nous apprenons à Epita autrement qu'avec des travaux de programmation et l'appliquer sur un projet qui nous tient à cœur.

Réalisation du projet

Pour les outils au niveau de l'application, nous avons pensé à utiliser GTK et glade afin de réaliser l'interface

GTK :

GTK est un ensemble de bibliothèques logicielles, c'est-à-dire un ensemble de fonctions permettant de réaliser des interfaces graphiques. Cette bibliothèque a été développée originellement pour les besoins du logiciel de traitement d'images GIMP.

Etat de l'art

Les algorithmes de recommandation sont des outils informatiques qui utilisent des données pour prédire les préférences d'un utilisateur et recommander du contenu pertinent. Les algorithmes de recommandation sont largement utilisés dans de nombreux secteurs, tels que la musique, les films, les jeux, les produits et les services en ligne. L'état de l'art actuel des algorithmes de recommandation se base sur des méthodes telles que le filtrage collaboratif, la factorisation de matrices et l'apprentissage en profondeur. Ces algorithmes utilisent souvent des techniques d'apprentissage automatique pour améliorer les prédictions de préférences d'un utilisateur en utilisant des données telles que les habitudes de consommation et les données d'interaction.

Il existe plusieurs types d'algorithmes de recommandation, chacun ayant ses propres forces et limitations:

Filtrage collaboratif :

Ce type d'algorithme utilise les données d'interaction des utilisateurs pour recommander des éléments similaires à ceux qu'ils ont appréciés auparavant. Il peut être basé sur l'utilisateur (filtrage collaboratif utilisateur-utilisateur) ou sur l'élément (filtrage collaboratif utilisateur-élément).

Factorisation de matrices :

Ce type d'algorithme utilise une décomposition mathématique pour représenter les préférences des utilisateurs et les caractéristiques des éléments dans un espace de facteurs latents. Il peut ensuite recommander des éléments similaires en fonction de la similarité des facteurs latents.

Apprentissage en profondeur :

Ce type d'algorithme utilise des réseaux de neurones pour apprendre les relations entre les utilisateurs, les éléments et les préférences. Il peut également prendre en compte des facteurs supplémentaires tels que les métadonnées des éléments pour améliorer les recommandations.

Filtrage basé sur le contenu:

Ce type d'algorithme utilise les caractéristiques des éléments pour recommander des éléments similaires en fonction de la similitude du contenu. Il peut inclure des techniques telles que la reconnaissance d'images ou de sons pour extraire les caractéristiques des éléments.

Ces algorithmes peuvent être combinés et personnalisés pour s'adapter aux besoins spécifiques de chaque système de recommandation. Il est important de noter que le choix de l'algorithme dépend des données disponibles, des objectifs de recommandation et des contraintes en matière de temps et de coûts de calcul.

Choix du type d'algorithme

Les quatre types d'algorithmes présentés un peu plus tôt possèdent chacun des forces ainsi que des faiblesses distinctes, et c'est pourquoi nous cherchons à les combiner. Cette combinaison a pour but de tirer parti au maximum de leurs forces et de minimiser leurs défauts respectifs. Cette approche devrait en fin de compte permettre une meilleure performance générale avec le moins de pertes possibles en matière d'efficacité. Toujours avec ce même objectif en tête, certains des algorithmes choisis seront plus utilisés que d'autres, afin de permettre leur usage à leur plein potentiel.

II) Découpage des tâches :

Répartition des tâches

Répartition des tâches	Sarah Santiago	Jimmy Fabre	Thuy Tran Nguyen	Alban Richard
Algorithmes		❖	❖	
Communication			❖	❖
Données	❖			❖
Interface	❖	❖		
Réseaux	❖	❖		
Site Web	❖			❖
Installateur/ Lanceur			❖	❖

Legende : ❖ = Responsable ; ❖ = Suppleant

Détail des tâches

Design du site :

Nous créerons un site Web pour présenter notre projet avec des informations de contact et des moyens de se connecter à l'application.

En respectant les consignes du cahier des charges initial :

- Une présentation du projet (l'historique, les membres, la chronologie de réalisation, les problèmes rencontrés, les solutions envisagées),
- Les liens sur les sites (des membres, des logiciels, images, sons, librairies, applets et autres éléments que vous auriez pu utiliser),
- Un download du rapport, du projet et d'une version lite de celui-ci (sans toutes les choses inutiles à l'exécution du projet).

Cette partie sera réalisée par Sarah.

Interface :

Cette tâche consiste, à l'aide de glade et de la bibliothèque GTK, à créer une interface graphique plutôt que textuelle pour le confort de l'utilisateur.

Elle permettra de consommer le contenu recommandé et de lui donner une note.

Cette partie sera réalisée par Alban.

Algo de recommandation :

Nous allons réaliser un algorithme de recommandation qui est le fondement même du projet. Le filtrage collaboratif semble être un bon compromis entre complexité et efficacité.

Cette partie sera réalisée par Thuy-Trang.

Communication :

Le but de cette partie est de faire connaître notre projet pour potentiellement recevoir des conseils, des avis et des utilisateurs si nous décidons de le rendre public.

Nous nous servirons des plateformes connues comme Instagram, Facebook, Twitter ou encore Comeet.

Cette partie sera réalisée par Sarah.

Trouver les données à étudier :

Pour le filtrage collaboratif il nous faut tout d'abord une base de données à exploiter avec du contenu et des notes attribuées sur ce même contenu.

Il semble plutôt facile de trouver ce style de base de données sur internet. Par la suite, nous essayerons de convertir cette base de données en une base de données compatible avec le SQL.

Cette partie sera réalisé par Thuy-Trang et Jimmy.

Réseau :

Il va falloir mettre en place un serveur qui va permettre aux utilisateurs de communiquer premièrement avec la base de données et tous les autres utilisateurs. En effet, chaque note donnée par un utilisateur va permettre d'améliorer l'algorithme de recommandation.

Nous pensons faire un serveur avec le Protocole TCP vu en cours de Programmation.

Cette partie sera réalisée par Jimmy.

Tableau d'avancement

Répartition des tâches	1re soutenance	2e soutenance	3e soutenance
Algorithme	☆☆	☆☆	☆☆☆
Communication		☆	☆☆☆
Données	☆	☆	☆☆☆
Interface	☆	☆☆	☆☆☆
Réseaux	☆	☆	☆☆☆
Site Web	☆	☆	☆☆☆
Installateur / Lanceur			☆☆☆

Legende : ☆ = Tâche commencée ; ☆☆ = Tâche avancée ; ☆☆☆ = Tâche terminée

3/Répartition des tâches

Voici un rappel de la répartition des tâches de notre groupe:

Répartition des tâches	Sarah Santiago	Jimmy Fabre	Thuy-Trang Nguyen	Alban Richard
Algorithmme		❖	❖	
Communication		❖	❖	
Données		❖	❖	❖
Interface	❖			❖
Réseaux	❖	❖		
Site Web	❖			❖
Installateur/ Lanceur			❖	❖

Legende : ❖ = Responsable ; ❖ = Suppleant

Il a subi quelques modifications depuis le cahier des charges car certaines personnes ont préféré travailler sur d'autre parties ou bien avait besoin d'aide. Je pense au cas de l'algorithme où Jimmy a essayé d'aider Thuy-Trang à corriger les erreurs présentes.

4/Avancement du projet

4.1 Algorithme

Pour cette partie, nous avons réussi à construire un système de recommandation de films en utilisant la technique de filtrage collaboratif avec les principales étapes suivantes :

- Création d'une matrice utilitaire de notes entre les utilisateurs et les films.

```
#ifndef UTILITY_MATRIX_H
#define UTILITY_MATRIX_H

char* get_movie_names(int index_movie);

char* get_movie_genres(int index_movie);

void get_utility_matrix(double **utility_matrix);

void new_user_movies(double *newuser, int uid);
```

- Trouver la similarité du nouvel utilisateur (ou de l'utilisateur concerné) avec d'autres utilisateurs à l'aide de la similarité en cosinus centré (corrélation de Pearson).

```
double pearson_correlation(double *A, double *B, unsigned int size)
```

```
void calc_similarity(double *normalizeduser, double *normalized_matrix, double *similarity, int No_of_users, int No_of_movies)
```

-
- Le regroupement d'utilisateurs basé sur la similarité à l'aide de k signifie regroupement (K-means Clustering).

```
void kmeans(
            int dim,// dimension of data
            double *X,// pointer to data
            int n,// number of elements
            int k,// number of clusters
            double *cluster_centroid,// initial cluster centroids
            int *cluster_assignment_final // output
        );
```

- Prédiction de films à l'aide de la technique de filtrage collaboratif (factorisation matricielle de faible rang) basée sur les clusters obtenus à l'étape 3.

```
int make_prediction(
                    double *user, //new user ratings defiend as double *user = (double *)malloc(sizeof(double) * No_of_movies);
                    int *similar_users, //ids of all similar users based on clustering
                    int no_of_susers, //number of similar users
                    double *similarity, //similarity with all the existing user
                    double *utility_matrix, //utility matrix
                    int *recommended_movies, //list of recommended movies to be saved in
                    double *predicted_ratings, //predicted ratigs of those movies to be saved in
                    int No_of_movies
                ); //returns number of recommended movies
```

- Les 10 meilleurs films recommandés seront imprimés en ligne de commande.

```
void recommender(int uid){
```

Résultat :

```
[nix-shell:~/Projet/algo_SQL]$ ./a.out
        Menu
Please enter one of the following choices:
    1 Help
    2 Login
```

```
[nix-shell:~/Projet/algo_SQL]$ ./a.out
        Menu
Please enter one of the following choices:
    1 Help
    2 Login
1
Hello, following are the instructions to use this movie recommendation system!
To rate a movie:
1. Open movies.csv file
2. Note the movie id of movie you want to rate
3. Enter movie id and ratings
Note: Movie ratings are out of 5.

        Menu
Please enter one of the following choices:
    1 Help
    2 Login
2
Are you an existing user(Type '0') or a new user(Type '1'): 
```

```

Are you an existing user(Type '0') or a new user(Type '1'): 0
Enter your user id: 1
Movies that you have rated:
2.5 - Dangerous Minds Drama
3.0 - Dumbo Animation|Children|Drama|Musical
3.0 - Sleepers Thriller
2.0 - Escape from New York Action|Adventure|Sci-Fi|Thriller
4.0 - Cinema Paradiso (Nuovo cinema Paradiso) Drama
2.0 - Deer Hunter The Drama|War
2.0 - Ben-Hur Action|Adventure|Drama
2.0 - Gandhi Drama
3.5 - Dracula (Bram Stoker's Dracula) Fantasy|Horror|Romance|Thriller
2.0 - Cape Fear Thriller
2.5 - Star Trek: The Motion Picture Adventure|Sci-Fi
1.0 - Beavis and Butt-Head Do America Adventure|Animation|Comedy|Crime
4.0 - French Connection The Action|Crime|Thriller
4.0 - Tron Action|Adventure|Sci-Fi
3.0 - Gods Must Be Crazy The Adventure|Comedy
2.0 - Willow Action|Adventure|Fantasy
2.0 - Antz Adventure|Animation|Children|Comedy|Fantasy
2.5 - Fly The Drama|Horror|Sci-Fi|Thriller
1.0 - Time Bandits Adventure|Comedy|Fantasy|Sci-Fi
3.0 - Blazing Saddles Comedy|Western

Do you want to:
0. Rate movies(Type '0')
1. Show existing recommendations for your user id (Type '1')
2. Movies that you have rated (Type '2')
3. Exit(Type '3')
0
Enter movieid: 1
Enter rating: 2
Rate more(Type '0') or recommend(Type '1'): ■

```

Malgré tous nos efforts pour résoudre l'erreur, nous avons encore 1 dernière erreur non résolue pour afficher 10 films recommandés :

```

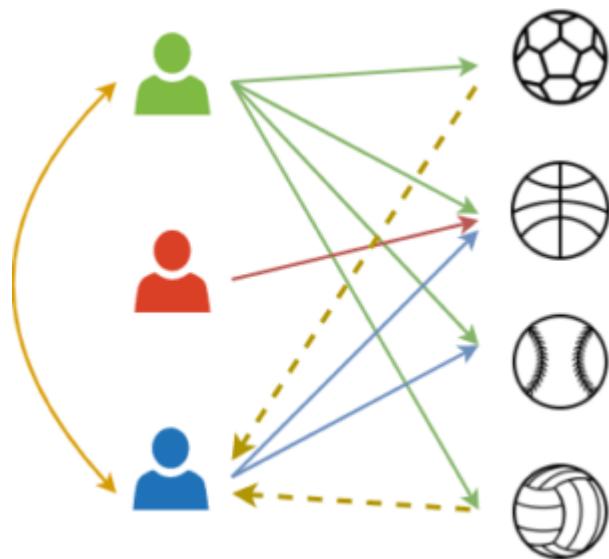
Do you want to:
0. Rate movies(Type '0')
1. Show existing recommendations for your user id (Type '1')
2. Movies that you have rated (Type '2')
3. Exit(Type '3')
1
WARNING: Empty cluster 0!
WARNING: Empty cluster 1!
WARNING: Empty cluster 2!
WARNING: Empty cluster 3!
WARNING: Empty cluster 4!
WARNING: Empty cluster 5!
WARNING: Empty cluster 6!
WARNING: Empty cluster 7!
WARNING: Empty cluster 8!
WARNING: Empty cluster 9!
WARNING: Empty cluster 10!
WARNING: Empty cluster 11!
WARNING: Empty cluster 12!
WARNING: Empty cluster 13!
WARNING: Empty cluster 14!
WARNING: Empty cluster 15!
Segmentation fault (core dumped)

```

Les connaissances que nous avons acquises pour mettre en œuvre le projet :

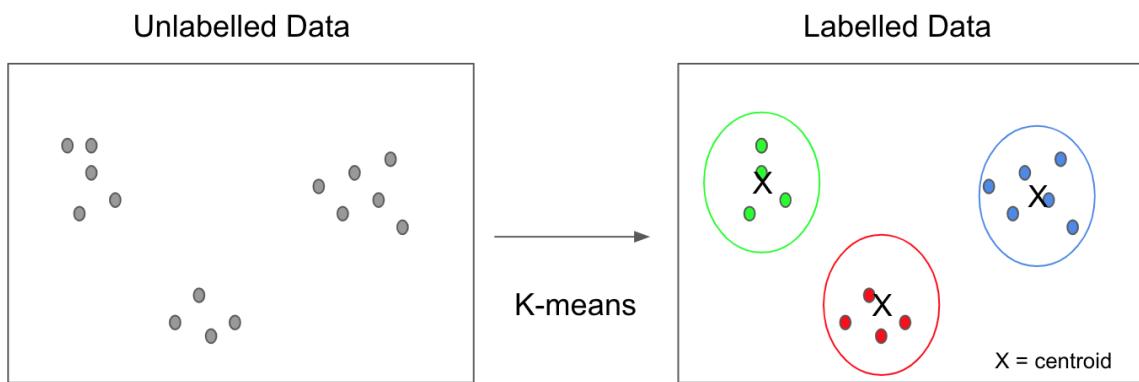
I) L'algorithme de filtrage collaboratif :

Le filtrage collaboratif est une technique qui peut filtrer les éléments qu'un utilisateur pourrait aimer sur la base des réactions d'utilisateurs similaires. Cela fonctionne en recherchant un grand groupe de personnes et en trouvant un plus petit ensemble d'utilisateurs ayant des goûts similaires à un utilisateur particulier. Il examine les éléments qu'ils aiment et les combine pour créer une liste classée de suggestions.



II) K signifie regroupement (K-means Clustering) :

K-means Clustering est une forme d'apprentissage non supervisé. Les scientifiques l'utilisent lorsqu'ils disposent d'un grand nombre de données non étiquetées (toute information sans groupes ou catégories définis.) K signifie que le but du regroupement est de rechercher des données pour différents groupes. La variable alphabétique "K" doit représenter chaque groupe. Contrairement à plusieurs algorithmes, celui-ci travaille de manière répétitive pour attribuer chaque point de données à un groupe K tout en considérant les caractéristiques dont il dispose.



Les points de données sont regroupés en fonction de la similarité de leurs caractéristiques:

- On peut utiliser le centre des K groupes pour étiqueter de nouvelles données.
- Les étiquettes de données d'entraînement sont attribuées à chaque grappe.

→ Par exemple : Une entreprise souhaite créer des politiques préférentielles pour différents groupes de clients en fonction de l'interaction entre chaque client avec cette entreprise (nombre d'années en tant que client, montant d'argent que les clients ont versé à l'entreprise). Entreprise, âge, sexe, ville, profession, etc). Disons que cette entreprise a beaucoup de données sur beaucoup de clients mais qu'il n'y a aucun moyen de diviser tous ces clients en un certain nombre de groupes/clusters différents. Si on posait cette question à une personne connaissant l'apprentissage automatique, la première méthode à laquelle elle penserait serait le clustering K-means. Après avoir divisé chaque groupe, les employés de cette entreprise peuvent choisir un ou plusieurs clients dans chaque groupe pour décider quel client groupe auquel correspond chaque groupe. Cette dernière partie nécessite une intervention humaine, mais la quantité de travail a été fortement réduite.

Le centroïde de chaque groupe est un ensemble de valeurs définissant les groupes résultants. L'analyse de la caractéristique du

centroïde peut aider à interpréter le type de groupe représenté par chaque grappe.

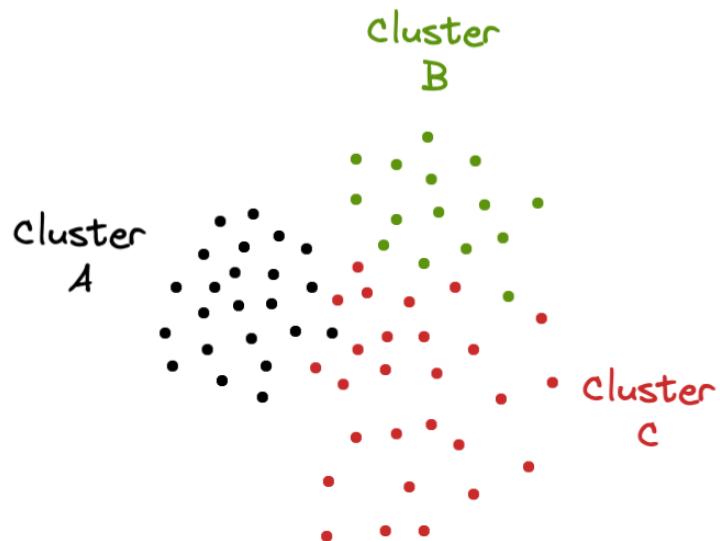
Il existe quelques limites à l'algorithme de clustering K-means

:

1. *Nous devons savoir combien de clusters ont besoin d'être regroupés*

Notez que dans l'algorithme ci-dessus, nous avons besoin de connaître la quantité K est le nombre de grappes. En fait, dans de nombreux cas, nous ne déterminons pas cette valeur.

2. *Les clusters doivent avoir à peu près le même nombre de points*



3. *Lorsqu'un cluster est à l'intérieur d'un autre cluster*

III) Le coefficient de corrélation de Pearson :

Le coefficient de corrélation de Pearson détermine tout changement dans une variable qui est influencée par l'autre variable liée. Le coefficient de corrélation de Pearson est influencé par le concept de covariance, ce qui en fait une meilleure méthode pour déterminer la relation et l'interdépendance entre les deux variables.

→ Exemple : l'âge d'un enfant augmente avec le temps. Au fur et à mesure que le temps passe, son âge commence à augmenter en années également.

Il existe généralement deux types de relations entre deux variables :

- Relation linéaire positive : lorsqu'une variable augmente, l'autre augmente également.

→ Exemple : à mesure que le nombre de jours augmente, la plante grandit (augmente) aussi.



-
- Relations linéaires négatives: lorsqu'une variable augmente, l'autre diminue.
→ Exemple : lorsqu'une voiture se rend à une destination, à mesure que la distance parcourue augmente, la distance jusqu'à la destination diminue.



IV)

Similarité en cosinus centré :

La similarité cosinus est une mesure de similarité entre deux vecteurs non nuls définis dans un espace de produit interne. La similarité cosinus est le cosinus de l'angle entre les vecteurs; c'est-à-dire qu'il s'agit du produit scalaire des vecteurs divisé par le produit de leurs longueurs. Il s'ensuit que la similarité cosinus ne dépend pas des grandeurs des vecteurs, mais

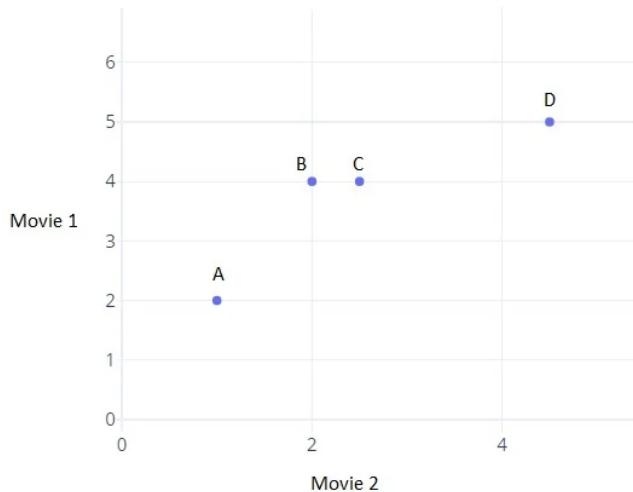
seulement de leur angle. Pour comprendre le concept de similarité, créons d'abord un jeu de données simple. Les données incluent quatre utilisateurs A, B, C et D, qui ont noté deux films. Les classements sont stockés dans des listes, et chaque liste contient deux nombres indiquant le classement de chaque film :

Les notes par A sont [1.0, 2.0]

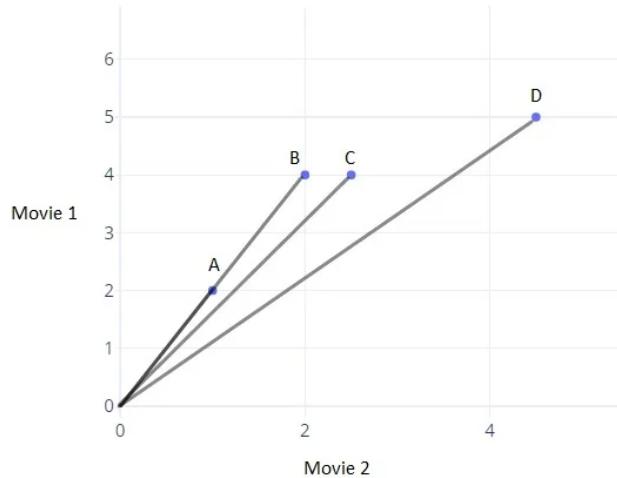
Les notes par B sont [2.0, 4.0]

Les notes par C sont [2.5, 4.0]

Les notes par D sont [4.5, 5.0]



Dans le graphique ci-dessus, chaque point représente un utilisateur et est tracé par rapport aux notes qu'il a attribuées à deux films.



Le graphique montre quatre lignes joignant chaque point à l'origine. Les lignes pour A et B coïncident, rendant l'angle entre elles égal à zéro. Vous pouvez considérer que si l'angle entre les lignes augmente, la similitude diminue et si l'angle est nul, les utilisateurs sont très similaires. Pour calculer la similarité à l'aide de l'angle, vous avez besoin d'une fonction qui renvoie une similarité plus élevée ou une distance plus petite pour un angle inférieur et une similarité inférieure ou une distance plus grande pour un angle supérieur. Le cosinus d'un angle est une fonction qui décroît de 1 à -1 lorsque l'angle augmente de 0 à 180. Vous pouvez utiliser le cosinus de l'angle pour trouver la similitude entre deux utilisateurs. Plus l'angle est élevé, plus le cosinus sera bas et donc, plus la similarité des utilisateurs sera faible. Vous pouvez également inverser la valeur du cosinus de l'angle pour obtenir la distance cosinus entre les utilisateurs en la soustrayant de 1. Notez que les utilisateurs A et B sont considérés comme absolument similaires dans la métrique de similarité cosinus malgré des notes différentes. C'est en fait un phénomène courant dans le monde réel, et les utilisateurs comme l'utilisateur A sont ce que vous pouvez appeler des évaluateurs difficiles. Un exemple serait un critique de cinéma qui attribue toujours des notes inférieures à la moyenne, mais le classement des éléments de sa liste serait similaire à celui des évaluateurs moyens comme B. Pour prendre en compte ces préférences individuelles des utilisateurs, vous devrez amener tous les utilisateurs au

même niveau en supprimant leurs préjugés. Vous pouvez le faire en soustrayant la note moyenne donnée par cet utilisateur à tous les éléments de chaque élément évalué par cet utilisateur. Voici à quoi cela ressemblerait:

- Pour l'utilisateur A, le vecteur de notation [1, 2] a la moyenne de 1,5. Soustraire 1,5 de chaque note vous donnerait le vecteur [-0,5, 0,5].
- Pour l'utilisateur B, le vecteur d'évaluation [2, 4] a la moyenne de 3.

En soustrayant 3 de chaque évaluation, vous obtiendrez le vecteur [-1, 1]. En faisant cela, vous avez changé la valeur de la note moyenne donnée par chaque utilisateur à 0. Essayez de faire la même chose pour les utilisateurs C et D, et vous verrez que les notes sont maintenant ajustées pour donner une moyenne de 0 pour tous les utilisateurs, ce qui les ramène tous au même niveau et supprime leurs préjugés. Le cosinus de l'angle entre les vecteurs ajustés est appelé cosinus centré. Cette approche est normalement utilisée lorsqu'il y a beaucoup de valeurs manquantes dans les vecteurs et que vous devez placer une valeur commune pour remplir les valeurs manquantes.

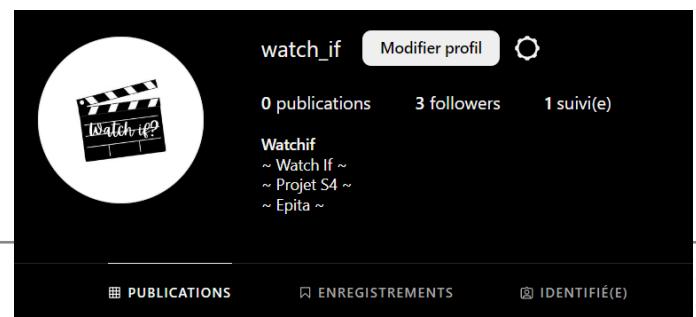
Remplir les valeurs manquantes dans la matrice des notations avec une valeur aléatoire peut entraîner des inexactitudes. Un bon choix pour remplir les valeurs manquantes pourrait être la note moyenne de chaque utilisateur, mais les moyennes originales des utilisateurs A et B sont respectivement de 1,5 et 3, et remplir toutes les valeurs vides de A avec 1,5

et celles de B avec 3 seraient en fait des utilisateurs différents. Mais après avoir ajusté les valeurs, la moyenne centrée des deux utilisateurs est 0, ce qui vous permet de saisir plus précisément l'idée que l'élément est supérieur ou inférieur à la moyenne pour les deux utilisateurs, toutes les valeurs manquantes dans les vecteurs des deux utilisateurs ayant la même valeur 0.

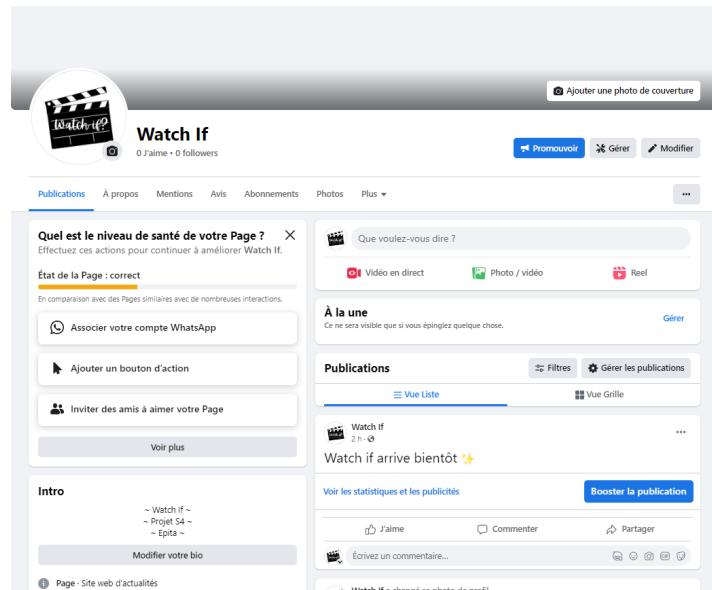
4.2 Communication

Pour cette partie, nous avons parlé de notre projet autour de nous et nous avons créé une page sur l'application CoMeet, réalisée par un étudiant d'Epita, qui permet de partager nos projets. Ainsi, les utilisateurs peuvent voir l'avancée du développement de l'application à travers des posts.

De plus, Sarah a créé des profils sur les réseaux sociaux représentant notre projet afin de pouvoir le promouvoir. En effet, il est possible de nous retrouver sur:



- instagram



- facebook



- twitter

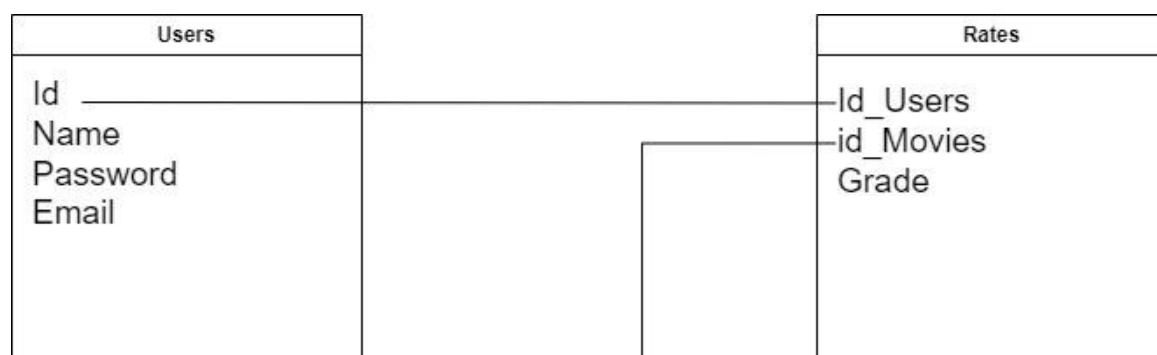
4.3 Données

Grâce à Thuy-Trang nous possédons un dataset de plus de 9000 films avec leurs genres et des notes sur 5 donner par des utilisateurs. Ce dataset va permettre d'entraîner l'algorithme de recommandation.

Jusqu'à présent toutes les données étaient stockées sous le format csv mais pour être en accord avec le serveur et faciliter les requêtes Jimmy a converti tous les fichiers dans une base de données SQL.

A l'aide d'un code en python, parser les données pour ensuite exécuter la requête a était très simple et rapide.

Voici une représentation graphique de la base de données, avec les 4 tables qui les composent:



4.4 Interface

Alban est celui qui a réalisé cette partie en intégralité, avec l'aide de Jimmy pour le design initial.

Nous avons choisi pour la partie design d'utiliser l'application Figma qui a pu nous permettre d'obtenir une idée claire de la direction vers laquelle nous souhaitions amener le projet.

Une fois cette direction établie, nous avons choisi d'utiliser l'application glade pour la création plus concrète de l'application. Ce choix est principalement dû au fait que la bibliothèque Gtk possède des moyens de directement récupérer les données de fichiers glade afin d'encoder les différentes fonctionnalités et éléments graphiques que nous souhaitions réaliser pour cette interface. Grâce à cela, notre travail fut grandement simplifié, malgré quelques bugs venant nous ralentir.

Alors que le travail avec glade nous permettait de créer rapidement boutons, étiquettes et tout autre élément d'interface dont nous avions besoin, la partie en C, éditée avec VSCode et VIM, consistait principalement à connecter ces différents éléments, que ce soit entre eux ou avec notre serveur et notre algorithme, pour obtenir un tout cohérent et fonctionnel.

Plus précisément, nous avons d'abord mis en place une fenêtre principale permettant de se connecter à son compte, chercher un film et voir toutes les informations le concernant. Nous avons ensuite ajouté plusieurs fenêtres secondaires, au nombre de quatre, servant chacune une fonction différente.

La première est la fenêtre de connexion. Comme son nom l'indique, elle permet de rentrer son nom d'utilisateur ainsi que son mot de passe afin d'accéder à son espace. Il y a également sur cette fenêtre un bouton permettant de rediriger vers la fenêtre de création de compte.

La fenêtre de création de compte, elle, comporte les espaces pour entrer le nom d'utilisateur voulu, le mot de passe ainsi que sa confirmation. Dans le cas où le nom d'utilisateur est déjà en usage, ou si le mot de passe et sa confirmation ne sont pas identiques, la fenêtre affichera un message d'erreur sous la forme d'un troisième popup.

La dernière fenêtre est un popup d'information, indiquant simplement que la déconnexion a été un succès.

Nous aurions voulu ajouter du code CSS à notre interface pour la rendre plus jolie mais les problèmes rencontrés sur les autres parties nous ont ralenti et nous avons préféré privilégier notre temps de travail sur les autres parties plus importantes que le design de l'application.

4.5 Réseau

Cette partie a entièrement été réalisée par Jimmy.

Nous avons tout d'abord choisi d'implémenter un serveur avec le protocole TCP vu durant nos cours de Programmation.

Comme expliqué dans le rapport de soutenance n°1 nous utilisions une fonction : `select()` qui permettait de surveiller une liste de descripteur de fichier dans le but de garder tous les utilisateurs connectés en permanence au serveur. Mais après réflexions et la mise en place des queues partagées pour attribuer des threads à chaque nouvelle connexion, nous nous sommes rendu compte que les utilisateurs avaient besoin de se connecter seulement pour récupérer des informations dans la database ou bien en envoyer.

Nous nous sommes donc rabattus sur une architecture plus simple où chaque nouvel utilisateur qui se connecte se voit attribuer un Thread, s'il y en a de disponible, et va pouvoir se connecter. Une fois connectés pour la première fois le client retient les informations et pour toute futur demande elles sont envoyées en amont des autres requêtes pour prouver au serveur que le client est bien connecté.

Le serveur communique de 4 manières différentes avec le client et inversement, au moment de la connexion un int est envoyé au serveur pour prévenir le serveur de quelle fonction nous avons besoin, par exemple 0 correspond au login. Ensuite les données utilisateur comme l'identifiant et le mot de passe transite à travers une structure de données appelé Users,

de même pour les Film et leurs informations dans une appelé Movie. Enfin ils peuvent aussi communiquer seulement avec des string par exemple pour le nom d'un film ou d'un utilisateur.

Chaque appel du client au serveur donne lieu à une requête SQL, dans ce projet nous utilisons Sqlite3. Nous avons acquis de nombreuses compétences dans ce domaine tel que l'utilisation de clé primaire et clé étrangère pour réaliser des JOIN entre 2 tables par exemple pour récupérer les genres d'un film ou bien les notes données par un utilisateur.

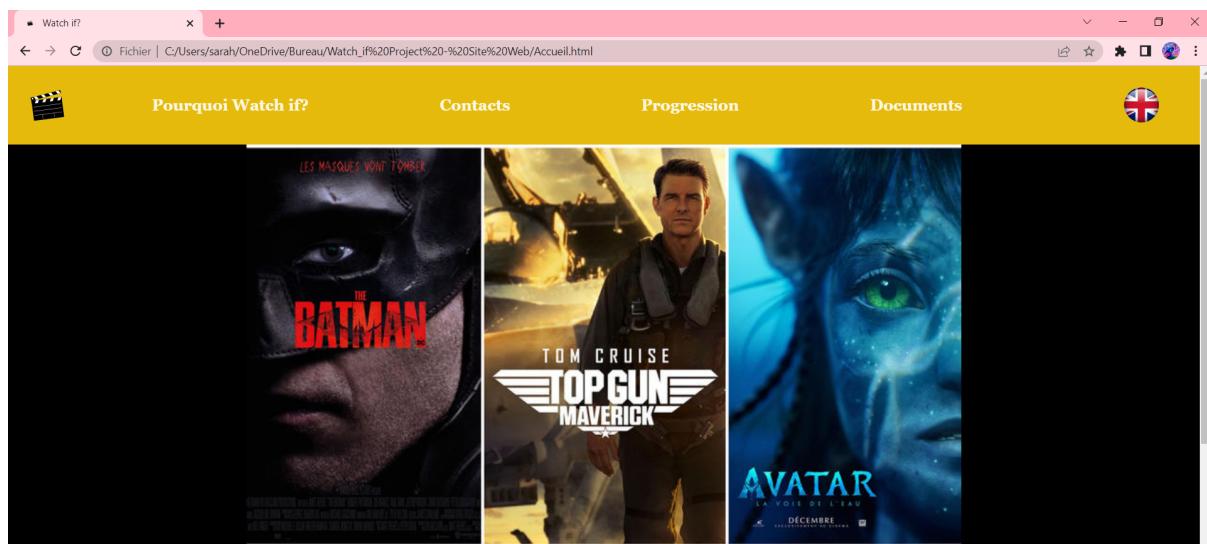
Nous sommes conscients que le serveur présente de nombreuses failles de sécurité mais pour ce projet qui restera privé nous avons décidé de ne pas trop nous pencher sur la question de la sécurité des données. Par exemple, les mots de passe sont stockés en "clair" dans la base de données c'est-à-dire qu'ils ne sont pas cryptés ce qui pourrait être un point à modifier pour un futur projet.

4.6 Site Web

Tout d'abord, avant de commencer à écrire des lignes de code pour le site web, Sarah, la responsable de cette partie a dû voir les bases de html et de css puisque c'est un sujet n'étant pas présent en cours. Ensuite, nous commençons enfin à écrire quelques lignes. Au début, nous cherchions au moins à avoir un site fonctionnel minimum donc nous utilisions seulement html.

Ensuite, nous avons rajouté du code css afin de faire la mise en page. Sans cela, notre site n'aurait aucune décoration, ce qui n'attirait pas le regard de l'utilisateur. Aussi, nous avons pensé utiliser JavaScript pour notre site, cependant, nous n'avons pas réussi à bien manier cet outil... Donc nous avons décidé d'utiliser seulement du code html et css.

Vous pouvez voir le résultat final juste en dessous:



Téléchargez l'application maintenant:

The screenshot shows the homepage of the Watch If? website. The header is yellow with the title "Pourquoi Watch if?" and a movie clapperboard icon. Navigation links include "Contacts", "Progression", "Documents", and a British flag icon. The main content area has a dark background with white text.

Pourquoi Watch if?

Dans le cadre de notre projet de S4, nous avons choisi d'étudier les algorithmes de recommandation et leur fonctionnement. Pour le début du projet nous n'avions pas vraiment une idée précise du contenu que nous voulions proposer aux utilisateurs mais nous avons trouvé une idée originale et intéressante.

C'est alors que nous avons réfléchi et avons trouvé que lorsque nous voulons voir un film, nous passons plus de temps à le chercher qu'à le regarder. C'est ainsi que nous avons décidé de créer Watch If.

Mais qu'est-ce que Watch If?

Cette application a pour but principal de proposer des films aux utilisateurs en fonction de leur goût. Nous souhaitons garder l'aspect "social" du projet en permettant aux utilisateurs de laisser des commentaires pour donner leur avis sur les différents films.

The screenshot shows the "Contacts" page of the Watch If? website. The layout is identical to the homepage, featuring a yellow header with the title "Pourquoi Watch if?", a movie clapperboard icon, and navigation links for "Progression" and "Documents".

Les développeurs



Sarah Santiago

Étudiante à l'Epita, j'ai toujours aimé programmer. Je sens que ce projet nous permettra de développer nos connaissances notamment en langage C et nous permettra de faire de nouvelles découvertes.



Jimmy Fabre

Étudiant à l'Epita, j'ai toujours aimé découvrir de nouvelles choses. Ce projet est un moyen de repousser nos limites et de voir ce qu'on est capable de faire.

The screenshot shows a web browser window with a yellow header bar. The header contains the text "Pourquoi Watch if?", "Contacts", "Progression", "Documents", and a British flag icon. The main content area has a title "La Progression" and a section titled "• L'algorithme". It includes a code snippet and a note about finding user similarity.

La Progression

• **L'algorithme**

Pour cette partie, nous avons réussi à construire un système de recommandation de films en utilisant la technique de filtrage collaboratif avec les principales étapes suivantes :

- Création d'une matrice utilitaire de notes entre les utilisateurs et les films.

```
#ifndef UTILITY_MATRIX_H
#define UTILITY_MATRIX_H

char* get_movie_names(int index_movie);

char* get_movie_genres(int index_movie);

void get_utility_matrix(double **utility_matrix);

void new_user_movies(double *newuser,int uid);
```

- Trouver la similarité du nouvel utilisateur (ou de l'utilisateur concerné) avec d'autres utilisateurs à l'aide

The screenshot shows a web browser window with a yellow header bar. The header contains the text "Pourquoi Watch if?", "Contacts", "Progression", "Documents", and a British flag icon. The main content area has a title "Les Documents" and two sections: "Le cahier des charges" and "Le rapport de la première soutenance". Each section includes a download button.

Les Documents

• **Le cahier des charges**

Ce document contractuel regroupe toutes les informations reliées à la réalisation de notre projet, notamment la description de notre projet, les ressources utilisées ainsi que l'organisation de notre travail.

[Télécharger le
cahier des charges
ici](#)

• **Le rapport de la première soutenance**

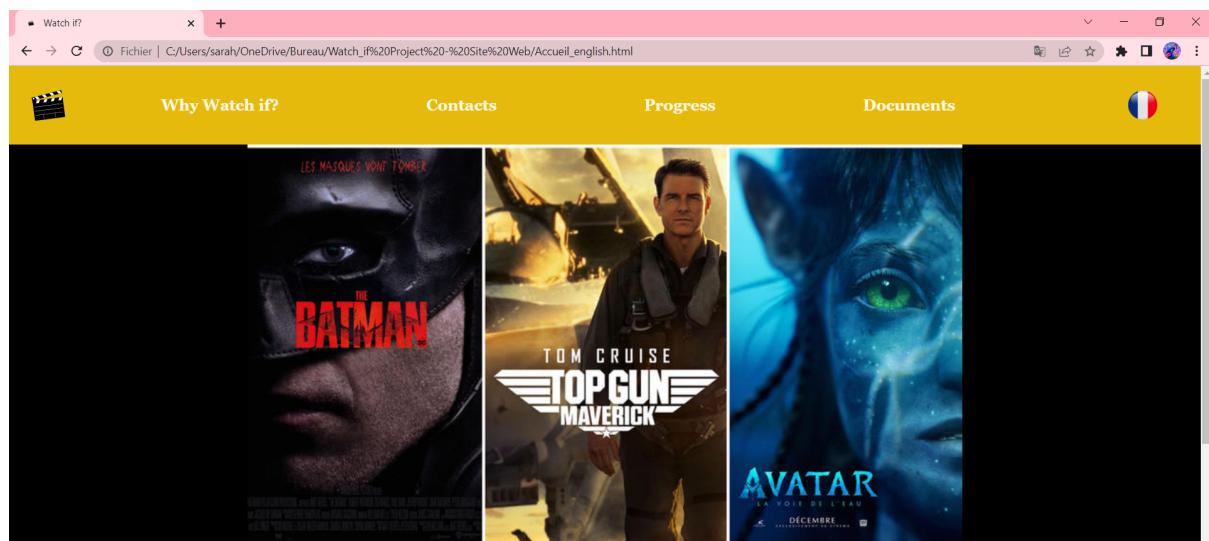
Ce rapport présente ce qui a été réalisé depuis la validation du cahier des charges pour la première soutenance. Il fait le bilan de ce qui a été fait, par qui et ce qui doit être fait pour la prochaine fois.

[Télécharger le
rapport de la
soutenance](#)

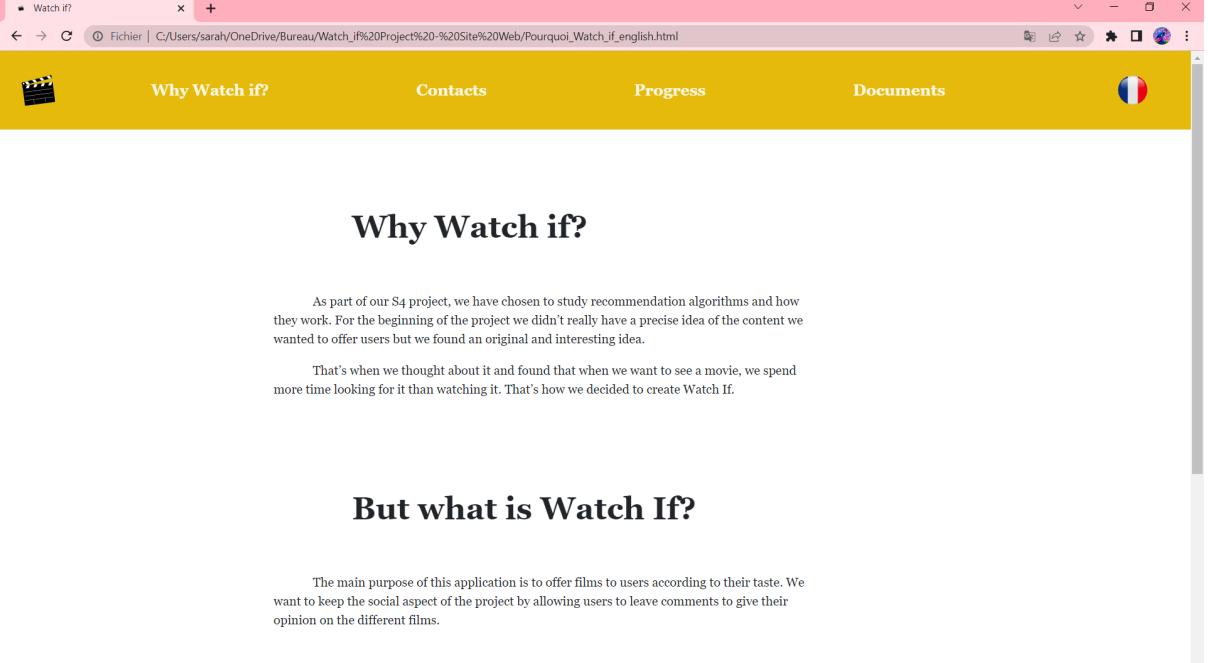
Comme il est possible de constater, nous avons opté pour la couleur jaune pour notre site, ce qui est une couleur très dynamique qui attire l'oeil. Ensuite, nous avons intégré à notre site plusieurs page afin de l'organiser et se repérer plus facilement dessus.

Nous pouvons y retrouver une brève présentation de Watch If mais aussi la présentation du groupe avec des liens permettant aux utilisateurs de nous contacter sur différentes plateformes. Nous pouvons également télécharger les documents en lien avec le projet comme le cahier des charges et les rapports.

De plus, nous avons aussi la possibilité de mettre le site en anglais afin de s'ouvrir à plus de monde même si pour l'instant notre application reste en français seulement et les documents ne sont pas traduits également.



Download the application now:



The screenshot shows a web browser window titled "Watch if?". The address bar indicates the file is located at "C:/Users/sarah/OneDrive/Bureau/Watch_if%20Project%20-%20Site%20Web/Pourquoi_Watch_if_english.html". The main content area has a yellow header with the title "Why Watch if?". Below the header, there is a paragraph of text and a small note about the creation of the project. A horizontal navigation bar below the header includes links for "Why Watch if?", "Contacts", "Progress", "Documents", and a French flag icon.

Why Watch if?

As part of our S4 project, we have chosen to study recommendation algorithms and how they work. For the beginning of the project we didn't really have a precise idea of the content we wanted to offer users but we found an original and interesting idea.

That's when we thought about it and found that when we want to see a movie, we spend more time looking for it than watching it. That's how we decided to create Watch If.

But what is Watch If?

The main purpose of this application is to offer films to users according to their taste. We want to keep the social aspect of the project by allowing users to leave comments to give their opinion on the different films.

Voici un lien pour se rendre sur notre site:

<https://peaceful-begonia-362ab7.netlify.app>

4.7 Installateur/Lanceur

Suite à de nombreux problèmes rencontrés, nous n'avons pas réussi à tout réunir afin de créer un installateur/lanceur malheureusement.

5/Nos impressions

5.1 Jimmy Fabre

Ce projet a été plus compliqué que je le pensais mais il m'a beaucoup apporté, j'ai adoré travailler sur le serveur, apprendre le SQL et pouvoir aider les autres sur leurs parties. Malheureusement il faudra plus de temps pour régler les nombreux bugs et pourquoi pas rendre le projet public.

5.2 Alban Richard

Prendre part dans ce projet était une bonne expérience, autant pour ce que nous avons appris que pour l'ambiance de groupe. Cela à été pour moi l'occasion d'apprendre à maîtriser la programmation d'interfaces ainsi que leur création en général. Je suis donc assez satisfait de ce que nous avons réalisé. Malgré quelques complexités de programmation ayant presque toutes été résolues, nous avons pu compléter une grande majorité de notre projet initial.

5.3 Sarah Santiago

Nous avons pu nous surpasser avec ce projet, nous avons pu également découvrir de nouvelles choses. Au départ, je trouvais ce projet très compliqué et complexe sachant qu'il faudrait apprendre beaucoup de nouvelles bases par nous-même. Finalement, nous avons réussi à avancer sur de nombreux points malgré le fait d'avoir rencontré de nombreuses difficultés. Le seul regret est de ne pas avoir réussi à réussir certains points. Cependant, je suis fière de nous pour ce que nous avons réussi à faire.

5.4 Thuy Trang Nguyen

Ce projet est incroyable, il m'a aidé à utiliser le langage C plus couramment. Bien qu'il ne s'agisse que d'un petit projet, il m'a permis d'acquérir plus d'aptitudes au travail d'équipe et d'acquérir plus d'expérience pour mener à bien un projet informatique. Bien qu'il ait été très difficile d'écrire le code car dans le processus d'écriture, nous avons rencontré beaucoup d'erreurs lors de la compilation et nous avons dû passer beaucoup de temps à les corriger.

6/ Visualisation de l'historique de développement

Voici le graphique qui représente les nombre de commits à réaliser tout le long du projet.



On peut voir que les commits sont souvent réalisés aux alentours des dates des soutenances.

Plus d'infos sur: github.com/MysteurJim/WatchIf/graphs/contributors

7/Conclusion

En définitive, nous pensons avoir globalement mené à bien notre projet, et ce malgré les difficultés que nous avons rencontré. En effet, ce projet nous a toujours tenu à cœur, et ce depuis le début. Il nous a également permis d'avoir une autre vision de la programmation. Aussi, nous avons réussi à le compléter dans les temps, avec une majorité des éléments voulus étant présents et fonctionnels. Nous avons maintenant atteint la fin de l'année, et nous pouvons affirmer que ce ne fut pas un voyage des plus faciles, il fut néanmoins très instructif.