



Dossier de projet

Titre professionnel Concepteur
Développeur d'applications Niveau II

Nicolas BARAN

Sommaire

Introduction	p.4
Liste des compétences du référentiel couvertes par le projet	p.5
1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.6
2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	p.7
3. Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.8
Résumé du projet	p.10
Cahier des charges	p.11
1. Besoin et objectifs	p.11
2. Fonctionnalités clés	p.12
3. Charte graphique	p.12
4. Maquette	p.13
Gestion de projet	p.15
Spécifications fonctionnelles du projet	p.17
1. Inscription et authentification	p.17
2. Découverte de films	p.18
3. Page des Favoris	p.19
4. Match de films	p.20
5. Page des Posts	p.21
6. Page « À propos »	p.23
7. Page « Profil »	p.25
8. Arborescence du site	p.27

Sommaire

Spécifications techniques du projet	p.28
1. React	p.28
2. Node.JS	p.28
3. MySQL	p.29
Réalisations personnelles	p.30
1. Front End	p.30
2. Back End	p.36
3. Base de données	p.43
Présentation du jeu d'essai	p.48
Description de la veille sur les vulnérabilités de sécurité	p.55
Description d'une situation de travail ayant nécessité une recherche	p.56
Intégration et déploiement continus	p.58
Evolutions potentielles	p.59

Introduction

Après une dizaine d'années passées au service de l'Armée de Terre, et passionné depuis toujours par le secteur des nouvelles technologies, j'ai fait le choix d'entreprendre une reconversion professionnelle dans le développement web en 2022.

J'ai ainsi choisi d'intégrer l'Ada Tech School pour un cursus de formation de 21 mois visant à acquérir le Titre Professionnel « Concepteur et Développeur d'applications Niveau II », les 9 premiers mois de formation à temps plein à l'école puis une année d'alternance en entreprise.

J'ai réalisé mon alternance au sein de l'entreprise Withings, connue pour ses produits électroniques grand public axés sur la santé et le bien-être. La marque est reconnue pour ses produits de santé innovants tels que les pèse-personnes intelligents, les trackers d'activité, les tensiomètres et les montres intelligentes. Withings a été un pionnier dans l'intégration de technologies de surveillance de la santé dans des dispositifs conviviaux pour un usage quotidien. Leurs produits se connectent généralement à une application mobile, permettant aux utilisateurs de suivre leurs données de santé au fil du temps et de gagner des perspectives sur leur bien-être.

Pendant la durée de mon alternance, j'ai travaillé pour l'équipe Application, plus précisément côté Assurance Qualité, en charge de la validation et de la stabilité de l'application de santé Withings. Je suis notamment responsable de la recherche de bugs et de la réalisation de tests de non-régression sur les plateformes iOS/Android.

Ce dossier a vocation à présenter le projet sur lequel j'ai travaillé au cours de la 2^{ème} année de formation, à compter de mars 2023, au rythme d'une journée par semaine en dehors du temps en entreprise : CineMatch. Ce projet a été guidé par ma passion pour le septième art et par l'envie d'adresser et de chercher à résoudre les problématiques du quotidien grâce aux nouvelles technologies.

Enfin, ces 2 dernières années, et la réalisation de ce projet, m'ont permis de conforter mon appétence pour le numérique, de développer de nouvelles compétences et d'acquérir les fondamentaux dans le développement web que j'espère mettre à profit de nombreux autres projets dans les années à venir.

Liste des compétences du référentiel couvertes par le projet

N° fiche AT	Activités types	N° Fiche CP	Compétences professionnelles
1	Composer et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	1 2 3 4 5	Maquetter une application Développer une interface utilisateur de type desktop Développer des composants d'accès aux données Développer la partie front-end d'une interface utilisateur web Développer la partie back-end d'une interface utilisateur web
2	Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	6 7 8	Concevoir une base de données Mettre en place une base de données Développer des composants dans le langage d'une base de données
3	Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	9 10 11 12 13 14 15	Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement Concevoir une application Développer des composants métier Construire une application organisée en couches Développer une application mobile Préparer et exécuter les plans de tests d'une application Préparer et exécuter le déploiement d'une application

1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

1.1 Maquetter une application

Le maquettage d'une application est une étape cruciale dans son développement. Pour ce faire, j'ai utilisé l'application Figma pour créer des mockups détaillés. Ces mockups ont été élaborés en tenant compte des cas d'utilisation préalablement définis, garantissant ainsi une représentation fidèle des fonctionnalités de l'application.

De plus, j'ai veillé à respecter les principes de sécurité dans la conception de l'interface utilisateur, afin d'assurer la protection des données et la convivialité de l'application pour les utilisateurs. En combinant une méthodologie de conception précise avec les outils appropriés, je m'efforcerais de créer une expérience utilisateur optimale et sécurisée.

1.2 Développer une interface utilisateur de type desktop

CineMatch est une application desktop conçue pour améliorer l'expérience cinématographique des utilisateurs, qui propose une interface utilisateur intuitive adaptée aux ordinateurs de bureau. À travers une navigation aisée, les utilisateurs peuvent "matcher" des films selon leurs préférences, favorisant la découverte de contenus partagés par la communauté.

La conception visuelle de l'application, centrée sur l'ergonomie et l'expérience utilisateur, assure une interaction fluide et agréable. CineMatch garantit une performance optimale, avec des chargements rapides et une utilisation efficace des ressources système. L'accessibilité est également au cœur du projet, avec une application facilement navigable par tous les utilisateurs.

1.3 Développer des composants d'accès aux données

Pour accéder aux données, j'ai créé des composants en utilisant MySQL et MySQL Workbench pour établir la connexion à la base de données. En parallèle, j'ai employé Sequelizer pour définir les modèles de données.

Cette combinaison d'outils m'a permis de mettre en place une architecture robuste pour gérer efficacement les données de l'application, assurant ainsi une intégration fluide et une gestion optimale des informations.

1.4 Développer la partie front-end d'une interface utilisateur web

CineMatch offre une interface graphique développée avec React, assurant une réactivité optimale sur différents appareils. Cette interface a été conçue en conformité avec les maquettes initiales, garantissant une cohérence visuelle et une expérience utilisateur homogène.

La combinaison de React et du respect des maquettes permet d'offrir aux utilisateurs une navigation fluide et une interface attrayante pour explorer le contenu cinématographique.

1.5 Développer la partie back-end d'une interface utilisateur web

L'API de l'application a été mise en œuvre en utilisant Express, un framework JavaScript côté serveur. Les composants serveur ont été soigneusement développés pour garantir la sécurité de l'application, en mettant en place des mesures de protection essentielles. Cette approche permet de prévenir les vulnérabilités potentielles et d'assurer la fiabilité de l'application dans le traitement des données et des requêtes, renforçant ainsi la confiance des utilisateurs.

La conception et le développement de la persistance des données impliquent l'intégration rigoureuse des meilleures pratiques de sécurité. Cela garantit la protection des données stockées, prévient les vulnérabilités et assure la confidentialité des informations sensibles, contribuant ainsi à la robustesse et à la fiabilité de l'application.

2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

2.1 Concevoir une base de données

La base de données a été créée avec MySQL, offrant une solide base de stockage des données. Les modèles ont été élaborés en utilisant Sequelize, simplifiant ainsi la gestion et la manipulation des données de manière efficace et cohérente. Cette combinaison de technologies permet une intégration fluide de la base de données dans l'application.

2.2 Mettre en place une base de données

La mise en place de la base de données a été réalisée avec succès grâce à l'utilisation de MySQL Workbench, facilitant ainsi la création et la gestion des tables et des schémas de données de manière efficace. Cette solution garantit une structure de base de données robuste pour l'application.

2.3 Développer des composants dans le langage d'une base de données

J'ai développé une série de composants de base de données structurée et robuste qui constituent le socle de l'application. Les modèles tels que `Users.js`, `Movies.js`, `Likes.js`, `LikedMovies.js`, `Posts.js`, et `Comments.js` ont été méticuleusement conçus pour capturer et refléter l'essence des interactions utilisateur.

Chacun de ces modèles représente une entité distincte avec des attributs et des relations bien définies qui correspondent à la logique métier de l'application.

Le modèle `Users.js` gère les données d'inscription et d'authentification, tandis que `Movies.js` stocke les informations des films. `Likes.js` et `LikedMovies.js` tracent les préférences des utilisateurs, permettant ainsi des recommandations personnalisées et une fonction de matching sociale.

`Posts.js` sert de plateforme pour les discussions et partages d'opinions, et `Comments.js` permet aux utilisateurs d'interagir davantage en commentant les publications.

3. Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

3.1 Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

J'ai pris en charge la gestion intégrale du projet CineMatch, en assumant l'ensemble des responsabilités normalement réparties au sein d'une équipe. Après avoir défini les objectifs du projet, j'ai élaboré un calendrier précis, jalonné d'échéances claires, pour structurer mon travail et garantir le respect des délais. L'utilisation de Git comme système de versionning a été essentielle pour maintenir l'organisation et l'intégrité du code.

Ce choix m'a permis de suivre mes avancements, de revenir sur d'anciennes versions si nécessaire et de travailler sur différentes fonctionnalités en parallèle sans risque de conflit. GitHub a servi de plateforme centrale pour héberger le code source, tout en facilitant le suivi des issues et des améliorations grâce à ses fonctionnalités de gestion de projet.

Cette méthodologie de travail, bien que menée en solo, a emprunté aux meilleures pratiques collaboratives de développement logiciel et a permis un déroulement fluide et structuré du projet.

3.2 Concevoir une application

Après avoir conceptualisé l'idée, j'ai adopté une démarche exploratoire auprès de mes proches (famille, amis, collègues, etc.) avec une diversité de profils (âge, sexe, niveau d'appétence pour le cinéma, etc.) pour évaluer leur disposition à utiliser une telle application au quotidien.

J'ai également effectué une analyse du paysage existant pour comprendre les concurrents potentiels. J'ai notamment identifié les applications suivantes : Scénographe, Kast, ou encore Teleparty qui permettent de regarder des films à plusieurs en simultané mais qui ne règle pas la problématique du choix du film.

En combinant ma démarche exploratoire et en identifiant les cas d'utilisation, j'ai pu définir les fonctionnalités nécessaires pour répondre aux besoins du public cible tout en intégrant des mesures de sécurité essentielles pour garantir la protection des données des utilisateurs.

3.3 Développer les composants métier

Les composants métiers de l'application sont étroitement alignés sur les besoins des utilisateurs. Ils incluent des fonctionnalités essentielles telles que l'inscription, l'authentification, la possibilité d'aimer des films, la publication de posts et de commentaires, ainsi que la consultation des messages d'autres utilisateurs.

Cette approche centrée sur l'utilisateur garantit que l'application offre une expérience pertinente et engageante, en répondant aux besoins et aux attentes de manière efficace. Les composants métiers sont conçus pour créer une interaction conviviale tout en permettant aux utilisateurs de participer activement à l'application.

3.4 Construire une application organisée en couches

CineMatch est une application structurée en trois couches distinctes : la couche de données, la couche applicative et la couche visuelle.

Ces trois composantes essentielles sont interconnectées pour fournir une expérience utilisateur complète. L'interface graphique offre une interaction conviviale, la base de données stocke les informations, tandis que la couche applicative (back-end) assure le transfert fluide des données entre le front-end et la base de données, garantissant ainsi le bon fonctionnement de l'application dans son ensemble.

3.5 Préparer et exécuter les plans de tests d'une application

Des tests unitaires ont été mis en place avec Jest, ainsi que des tests end-to-end grâce à Cypress.

Résumé du projet

(Version FR)

Quel couple ne s'est jamais retrouvé le soir devant son écran à parcourir sans fin des listes de films sur des plateformes de streaming sans savoir par où commencer et sans réussir à se mettre d'accord ? Avec CineMatch cette époque est révolue, choisir un film devient aussi ludique que de le regarder !

Cette application est née de la fusion entre une passion pour le septième art et la volonté de simplifier le quotidien de nombreux foyers en apportant une solution interactive et communautaire à une véritable problématique contemporaine récurrente : le choix du film du soir !

Avec CineMatch, les utilisateurs sont invités à exprimer leur intérêt pour les films, qui leur sont suggérés de manière aléatoire, grâce à une fonctionnalité intuitive : un swipe à gauche pour passer au film suivant, un swipe à droite si le film les intéresse. Et lorsque deux utilisateurs, ou plus, montrent un intérêt mutuel pour une œuvre cinématographique, c'est un match !

Pour faciliter le choix des utilisateurs sur les films proposés, CineMatch leur permet d'accéder facilement aux bandes annonces de ces derniers. Par ailleurs, l'application se caractérise également par sa capacité à favoriser l'interaction sociale : les utilisateurs peuvent non seulement partager leurs impressions à travers la rédaction de posts, mais également de commentaires, enrichissant ainsi l'expérience utilisateur et concourant à créer une véritable communauté de cinéphiles.

Que vous soyez fan de comédies romantiques, de films d'actions palpitants, de thrillers haletants, ou que vous souhaitiez sortir de votre zone de confort en explorant de nouveaux genres et styles cinématographiques, CineMatch est la solution qu'il vous faut pour trouver et partager vos films préférés. En définitive, cette application se veut avant tout pratique, par le gain de temps généré, mais aussi ludique et collaborative par ses fonctionnalités.

(Version EN)

What couple hasn't found themselves in the evening, endlessly scrolling through lists of movies on streaming platforms, unsure where to start and unable to agree on the movie they want to watch? With CineMatch, those days are over and choosing a movie becomes as enjoyable as watching it!

This app was born from a passion for cinema and the desire to simplify the daily lives of many households by providing an interactive and communal solution to a recurring issue: the choice of the evening's movie!

With CineMatch, users are invited to express their interest in movies, which are suggested to them randomly, using an intuitive feature: swipe left to move to the next movie, swipe right if interested. When two or more users show mutual interest in a movie, it's a match!

In order to facilitate users' choice among the proposed films, CineMatch gives them an easy access to trailers. The app is also characterized by its ability to promote social interaction: users can not only share their impressions through posts but also through comments, thus enriching the user experience and contributing to creating a genuine community of cinema-lovers.

Whether you're a fan of romantic comedies, action movies, gripping thrillers, or you want to step out of your comfort zone by exploring new genres and cinematic styles, CineMatch is the solution you need to find in order to find and share your favorite films. To sum up, this app aims to be practical by helping you saving time, but also fun and collaborative through its features.

Cahier des charges

1. Besoin et objectifs du projet

Besoin

L'origine de Cinematch trouve ses racines dans une expérience personnelle partagée par de nombreux cinéphiles, y compris moi-même. A maintes reprises, j'ai eu des discussions animées avec ma femme, des amis ou des proches sur nos choix de films pour la soirée et ai constaté une frustration commune récurrente : la difficulté à parvenir à un consensus parmi la multitude d'options disponibles.

En effet, si les plateformes de streaming offrent une pléthore de films, les utilisateurs sont laissés à eux-mêmes pour naviguer à travers cette abondance de contenus sans jamais trouver le film parfait pour la soirée. Le constat partagé est le suivant : l'expérience de recherche de films qui devait être un moment de détente se transforme en une véritable épreuve (sentiment de perte de temps, incapacité à trouver le consensus, etc.), les utilisateurs passant souvent plus de temps à rechercher un film qu'à en regarder un.

Pour répondre à cette problématique, j'ai décidé d'imaginer une solution novatrice pour le processus de sélection du film et pour réduire la frustration et le temps passé associés.

Objectifs

Cinematch, en tant que solution visant à répondre à l'expression du besoin exprimé plus haut, se doit de répondre à plusieurs objectifs spécifiques :

- **L'accessibilité et la simplicité d'utilisation**

Cinematch vise à simplifier le processus de sélection de films en offrant une application facile à utiliser avec une interface utilisateur intuitive et responsive. De plus, la comptabilité sur divers appareils, y compris des smartphones, des tablettes et des ordinateurs de bureau, doit pouvoir être garantie pour offrir une expérience utilisateur optimale.

- **La découverte de nouveaux films**

Cinematch encourage la découverte de nouveaux films en exposant les utilisateurs à un large éventail de propositions cinématographiques. En effet, l'application permet aux utilisateurs de découvrir de nouveaux films de manière engageante et intuitive grâce à l'implémentation d'un mécanisme de suggestion aléatoire qui tire parti d'une API externe pour présenter une variété de films.

- **L'évaluation et la sélection des films**

Cinematch permet aux utilisateurs d'exprimer leurs préférences, envies ou désintérêts cinématographiques de manière interactive grâce à une fonctionnalité de « swipe », simplifiant ainsi le processus de sélection. Par ailleurs, en stockant les films aimés par l'utilisateur dans une liste de favoris, Cinematch vise à simplifier l'accès à ces films pour une consultation ultérieure.

- **La création d'interaction sociale**

Cinematch intègre une fonctionnalité de match qui met en lumière les films populaires au sein de la communauté. En effet, l'application permet aux utilisateurs de voir quelles préférences de films ils partagent avec d'autres et de les connecter à travers ces intérêts communs.

- **Le partage de retours d'expérience**

Cinematch offre une section de commentaires pour que les utilisateurs puissent discuter des films et partager leurs opinions.

Cibles

CineMatch est destiné à un large éventail d'utilisateurs, de 18 ans et plus, qui apprécient le cinéma et pour les plus novices cherchant à découvrir de nouveaux films de façon interactive et sociale.

2. Fonctionnalités clés

Les principales fonctionnalités clés de l'application CineMatch sont les suivantes :

- **Création de compte** : Création d'un nouvel utilisateur dans la base de données
- **Connexion à ce compte** : L'utilisateur peut se connecter pour accéder aux fonctionnalités de l'application.
- **Découverte des films** : Les utilisateurs peuvent "matcher" (aimer) ou non les films proposés.
- **Enregistrement des favoris** : Les films aimés sont stockés dans une section « Favoris » pour chaque utilisateur.
- **Matchs de films** : Une section dédiée pour les films appréciés par plusieurs utilisateurs, indiquant un intérêt commun.
- **Interactions sociales** : Possibilité d'ajouter des commentaires et des posts sur les films pour favoriser les échanges entre utilisateurs.

3. Charte graphique

L'application étant à destination des cinéphiles, l'objectif était de retrouver dans la charte graphique de l'application les codes du cinéma : fauteuils rouge, toile blanche, murs noirs.

Pour la recherche graphique j'ai utilisé Figma pour maquetter mon application et utilisé le site Realtime Colors pour choisir les couleurs primaires et secondaires. Le primaire concerne les CTA et les sections principales, et le secondaire concerne les boutons et les cartes d'informations moins importants.

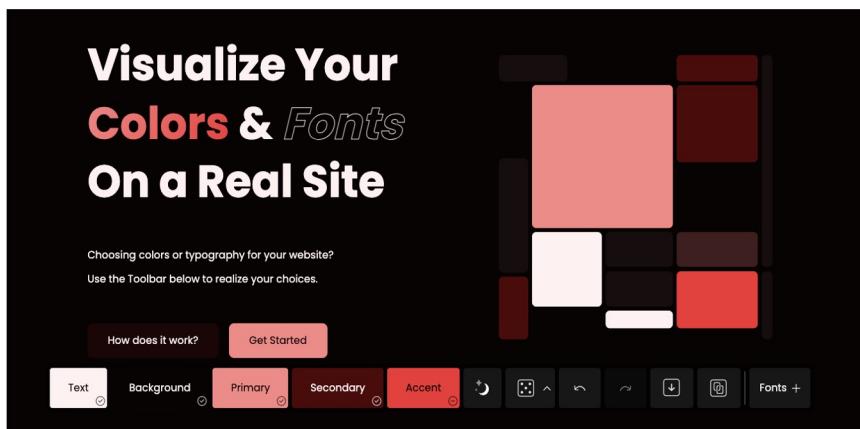
Concernant l'association de couleurs le choix s'est porté sur celle-ci :

The screenshot shows the Realtime Colors website interface. On the left, there's a title 'Colors & Fonts On a Real Site' with a subtitle 'Choosing colors or typography for your website? Use the Toolbar below to realize your choices.' Below this are two buttons: 'How does it work?' and 'Get Started'. A note at the bottom says 'Scroll to see more sections'. On the right, there's a color palette with several colored squares and a color hex code table. The table is as follows:

Accent	#C2241E
Secondary	#F4BAB8
Primary	#771613
Background	#FEFBFB
Text	#0D0202

At the bottom, there's a toolbar with various icons for color selection and tools.

Avec l'implémentation du mode sombre (dark mode) :



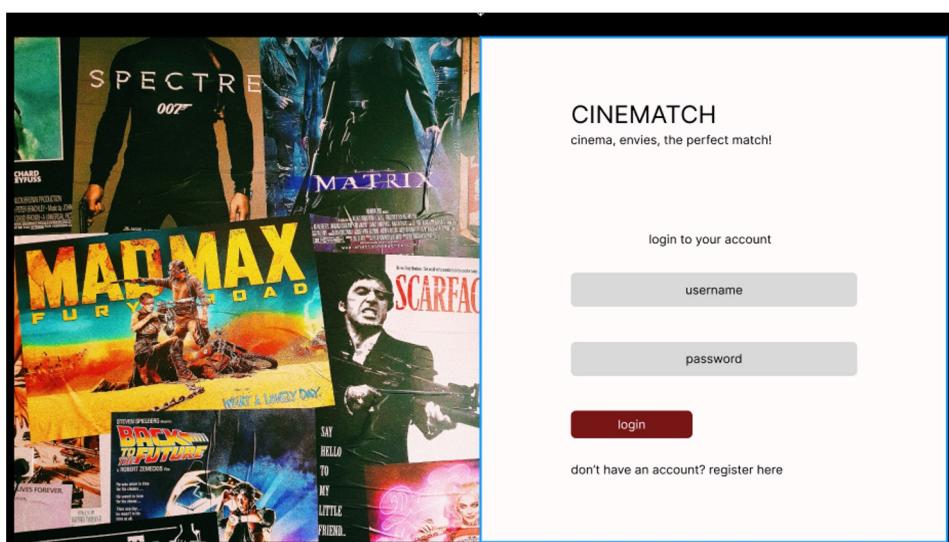
--accent-dark:	#E1423D
--secondary-dark:	#470D0B
--primary-dark:	#EC8C89
--background-dark:	#040101
--text-dark:	#FDF2F2
--black:	#000000
--white:	#e5e7eb

4. Maquettes

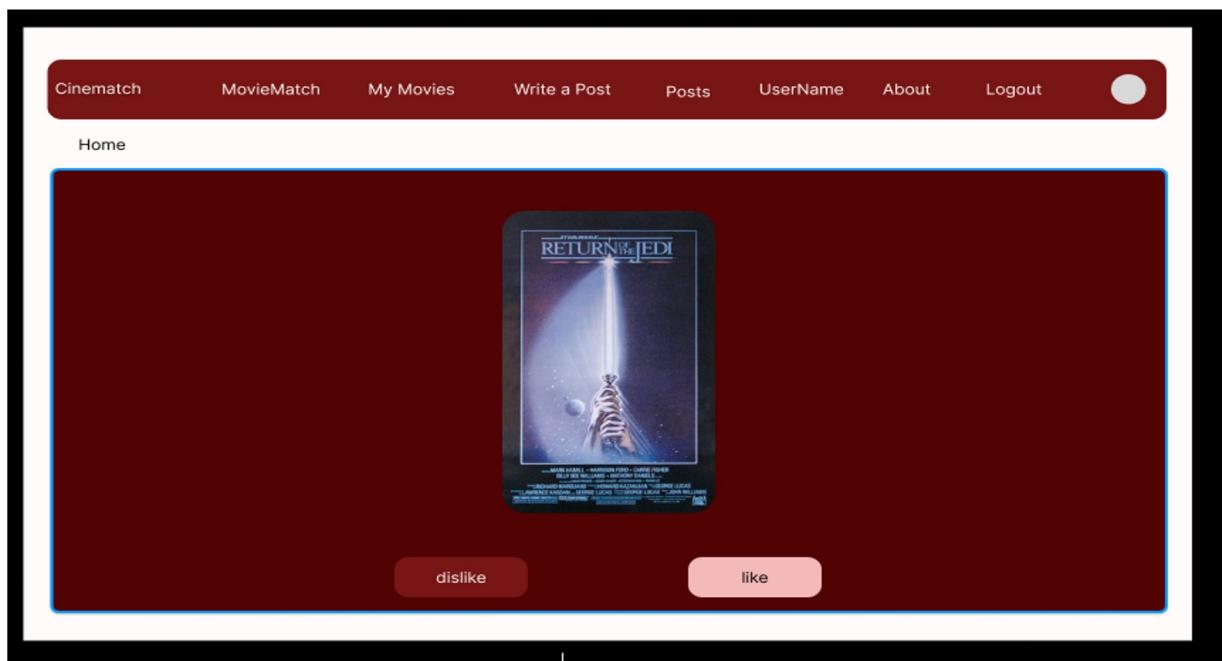
Une fois la charte graphique déterminée, j'ai pu commencer à élaborer des mock-up grâce à Figma.

Un mock-up est une représentation visuelle statique d'une interface utilisateur, une version non fonctionnelle et non-interactive du produit final, créée en amont du développement. Les avantages de son utilisation sont multiples : elle permet de réduire les erreurs de conception, l'amélioration de la communication entre les membres de l'équipe, et la possibilité de tester les idées de conception avant de se lancer dans un processus complet et coûteux.

La page de connexion, accessible une fois que l'utilisateur a créé son compte, offre l'accès aux fonctionnalités essentielles de l'application. Ici, l'utilisateur peut saisir ses identifiants pour se connecter à son compte personnel, lui permettant ainsi de profiter pleinement de toutes les fonctionnalités proposées. Cette étape garantit la sécurité des données et assure une expérience personnalisée. Une fois connecté, l'utilisateur peut explorer et utiliser l'application en toute simplicité, accédant ainsi aux services et aux contenus qui lui sont spécifiques.

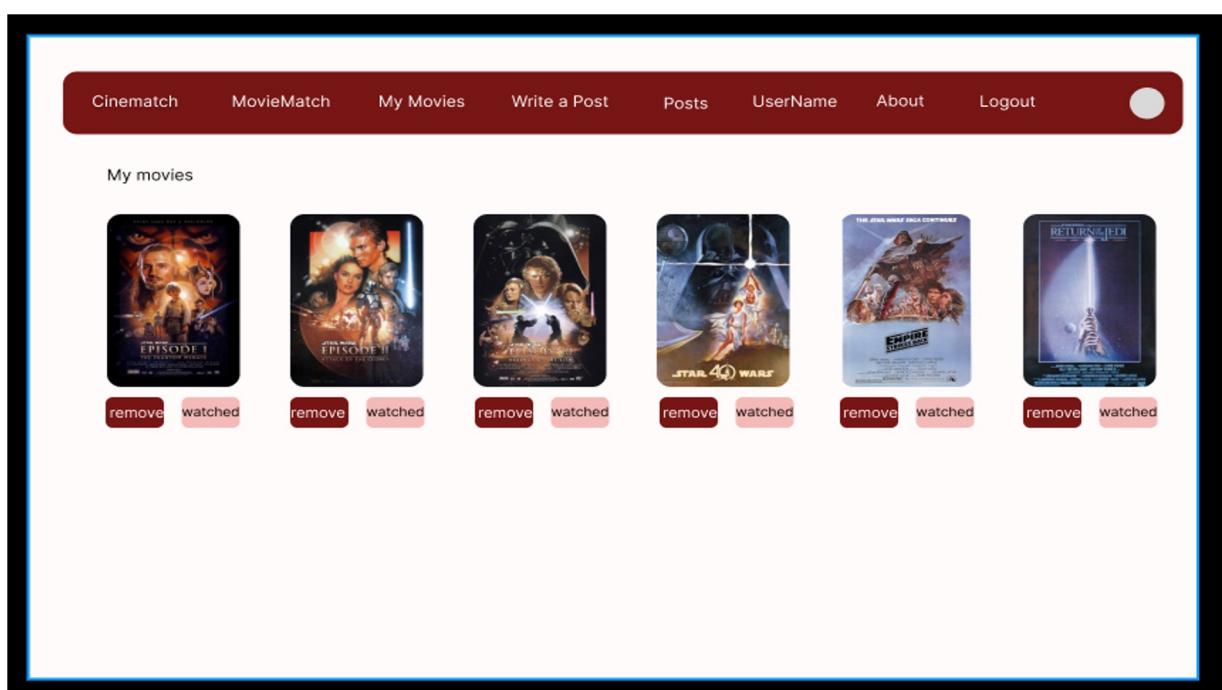


Lorsqu'un utilisateur se connecte à notre plateforme, il arrive sur la page d'accueil où il découvre une sélection de films. Seules les affiches des films sont visibles à ce stade, incitant l'utilisateur à se forger une première impression visuelle. Il peut exprimer son intérêt en cliquant sur "J'aime" ou "Je n'aime pas". Cette approche encourage une découverte cinématographique spontanée et instinctive. La page d'accueil favorise ainsi l'engagement des utilisateurs en les invitant à explorer notre catalogue de manière aléatoire.



Une fois que l'utilisateur a aimé un film, celui-ci se retrouvera dans la page "Mes films". L'utilisateur pourra alors obtenir davantage de détails sur le film, tels que le titre, le synopsis, l'année de sortie, la bande-annonce et une note attribuée par les spectateurs.

Sous chaque affiche, on trouvera un bouton "Supprimer" permettant de retirer le film de la liste des favoris, ainsi qu'un bouton "Vu" pour indiquer que l'on a regardé le film, ce dernier ayant une légère opacité une fois le bouton activé.

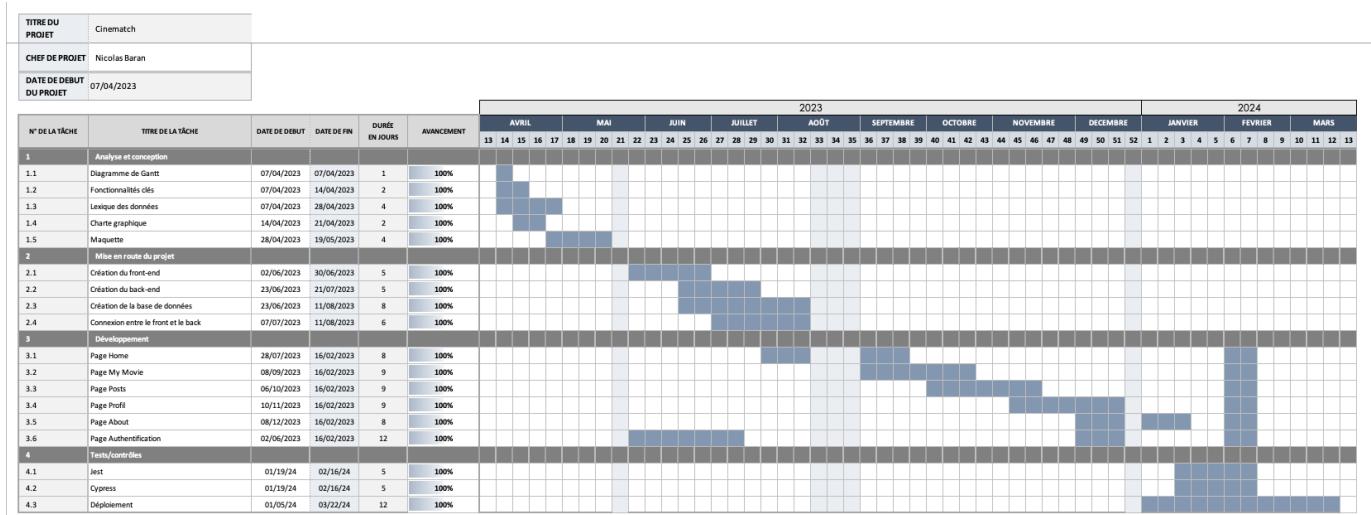


Gestion de projet

Dans le cadre du développement de Cinematch, bien que le projet ait été réalisé individuellement sans la collaboration directe d'autres développeurs, la gestion de projet a été un élément central de mon processus de travail. Convaincu de l'importance de planifier et d'organiser efficacement les différentes étapes du projet, j'ai adopté une approche méthodique pour garantir sa réussite dans les délais impartis.

Utilisation d'un diagramme de Gantt

Pour la réalisation de CineMatch, j'ai donc choisi de formaliser assez rapidement un diagramme de Gantt afin notamment de lister, de planifier et de visualiser toutes les étapes nécessaires à la réalisation du projet et les différentes tâches à accomplir. Ce diagramme m'a également permis d'établir des échéances pour chaque étape du développement de CineMatch et de pouvoir mesurer in itinere mon avancement, ce qui a été source de motivation.



J'ai formalisé mon diagramme de Gantt autour de 4 grandes phases :

- Analyse et conception
- Mise en route du projet
- Développement
- Tests/contrôles.

Chacune de ces phases a été découpée en tâches à réaliser, y associant des objectifs calendaires de réalisation ainsi que des taux d'avancement in itinere.

La version du diagramme présentée ci-dessous est la version finale de ce dernier, une fois le projet terminé.

Utilisation de Git et Github

En parallèle, j'ai mis en place une gestion efficace des versions en utilisant Git et Github pour sauvegarder régulièrement le code source de l'application. Cela m'a permis de suivre l'évolution du projet, de gérer les différentes versions du code et de revenir à des versions antérieures si nécessaire, assurant ainsi la stabilité et la fiabilité de l'application tout au long du processus de développement.

The screenshot shows a GitHub repository page for 'ncls-brn / movie-app-v4'. The main content area displays a list of commits from 'ncls-brn' with the message 'update files front'. The commits are ordered by date, with the most recent at the top. The commit history includes:

File	Message	Date
client	update files front	last week
coverage	update test jest	2 months ago
cypress	test back ok	last month
node_modules	update files	3 weeks ago
server	update movie match	2 weeks ago
.env	update files	3 weeks ago
.gitignore	update files	2 weeks ago
README.md	Initial commit	6 months ago
cypress.config.js	install cypress	2 months ago
package-lock.json	update files	3 weeks ago
package.json	update files	3 weeks ago

On the right side, there are sections for 'About', 'Releases', 'Packages', and 'Deployments'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and has a link to 'Create a new release'. The 'Packages' section says 'No packages published' and has a link to 'Publish your first package'. The 'Deployments' section shows 4 entries, with one entry for 'cinematch' 2 weeks ago.

The screenshot shows a GitHub 'Branches' page for the same repository. It lists three branches: 'main', 'nicolas', and 'nicolas'. The 'main' branch is the default branch. The 'Your branches' section shows the 'nicolas' branch was updated 2 weeks ago. The 'Active branches' section also shows the 'nicolas' branch was updated 2 weeks ago. A 'New branch' button is visible in the top right corner.

Les branches sur Git représentent un outil essentiel pour le développement, facilitant le travail en parallèle sur différentes fonctionnalités sans interférer avec le code en production.

Spécifications fonctionnelles du projet

Les spécifications fonctionnelles de CineMatch décrivent en détail les fonctionnalités, les comportements et les services que l'application fournit pour répondre aux besoins des utilisateurs. Cela comprend la manière dont les utilisateurs interagissent avec l'application, les processus que l'application doit exécuter, et les réponses attendues.

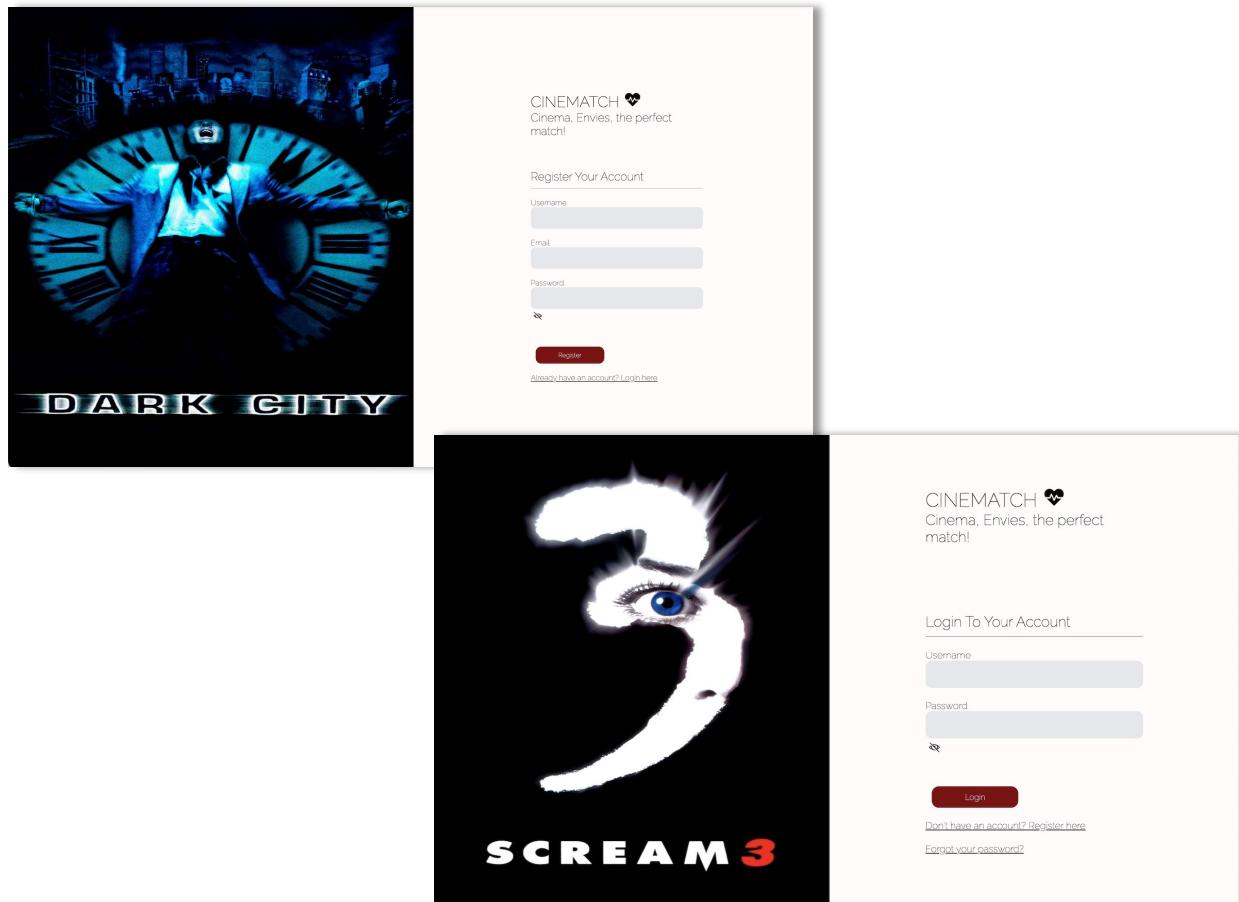
1. Inscription et authentification

Objectif

L'objectif de l'inscription et de l'authentification dans CineMatch est de créer un environnement sécurisé où les utilisateurs peuvent accéder à des fonctionnalités personnalisées, partager et interagir autour de leur passion pour le cinéma. Ce processus garantit que chaque utilisateur dispose d'un espace privé pour gérer ses préférences, ses films favoris, ses matchs de films, et participer activement à la communauté par des posts, des commentaires et des likes.

Fonctionnalités

- **Inscription** : Cette fonctionnalité permet aux utilisateurs de créer un compte en fournissant un e-mail, un nom d'utilisateur et un mot de passe. L'application doit vérifier que l'email n'est pas déjà utilisé.
- **Authentification** : Cette fonctionnalité permet aux utilisateurs de se connecter à leur compte en utilisant leur e-mail et mot de passe.



2. Découverte de films

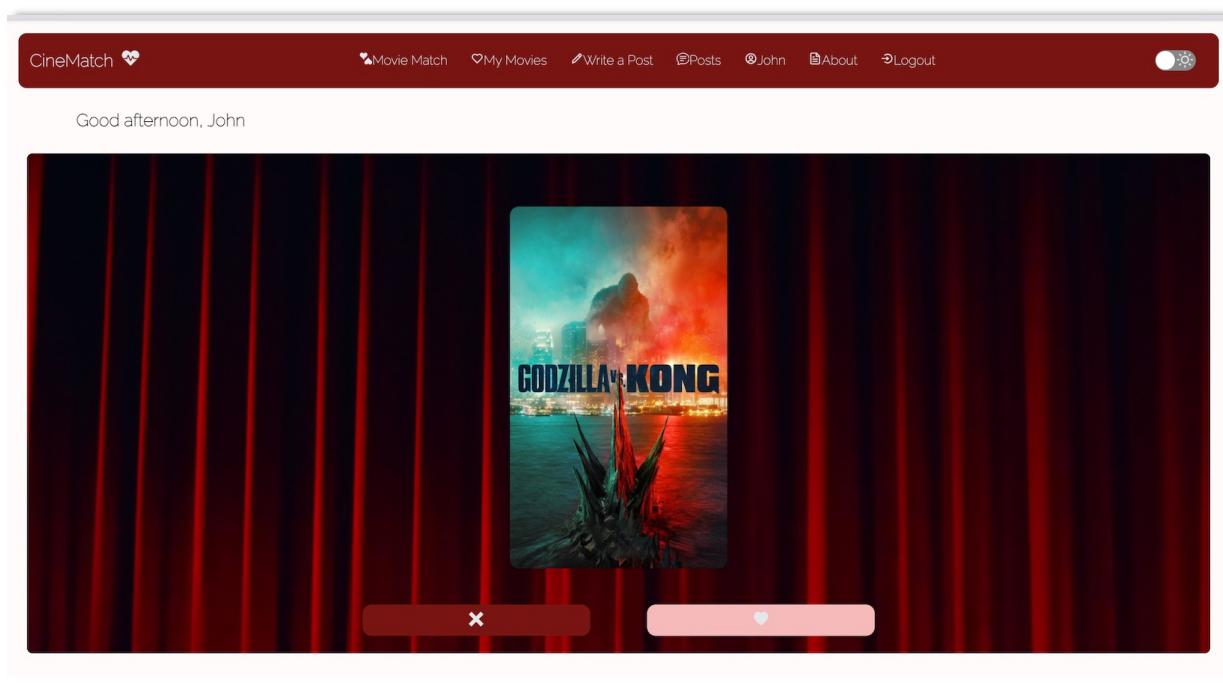
Objectif

Cinematch doit offrir aux utilisateurs une méthode intuitive et rapide pour exprimer leur intérêt pour les films qui leur sont proposés de manière aléatoire, en se basant sur un système de swiping similaire à celui de nombreuses applications de rencontres.

Fonctionnalités

- **Swiping** : Cette fonctionnalité permet à l'utilisateur lorsqu'il accède à la page de découverte de films, de lui présenter des films un à un, sous forme de cartes contenant uniquement l'affiche du film. L'utilisateur peut interagir avec la carte de deux manières principales :
- **Swiping à gauche** : Cette fonctionnalité permet à l'utilisateur de rejeter le film et d'indiquer qu'il n'est pas intéressé par ce dernier. Le film rejeté est alors retiré de la file d'attente de découverte et ne sera, de fait, pas stocké dans les favoris de l'utilisateur.
- **Swiping à droite** : Cette fonctionnalité permet à l'utilisateur d'indiquer qu'il a un intérêt pour le film. Le film aimé est automatiquement ajouté à la liste des favoris de l'utilisateur pour un accès facile ultérieurement.

Après le swiping (à gauche ou à droite), une nouvelle carte de film apparaît, permettant à l'utilisateur de continuer le processus de sélection.



Cette spécification fonctionnelle pour le swiping de films dans CineMatch met l'accent sur une interaction utilisateur fluide et engageante, tout en fournissant une méthode simple et directe pour filtrer les films selon les préférences personnelles, favorisant ainsi une expérience utilisateur personnalisée.

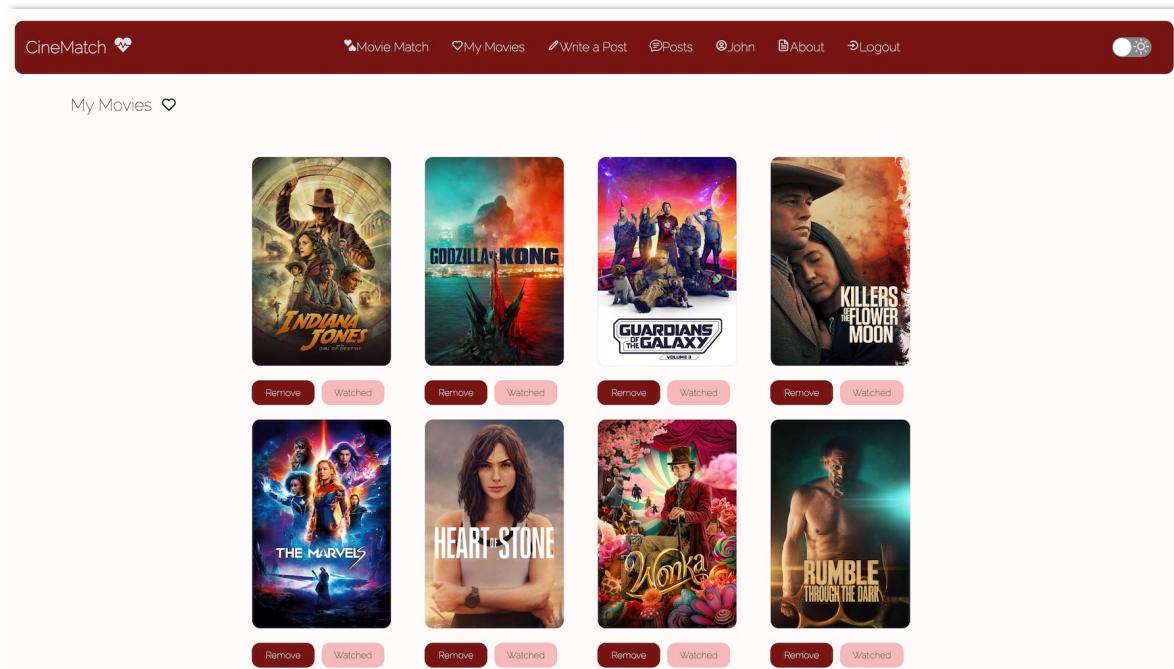
3. Page des Favoris

Objectif

L'objectif est de fournir aux utilisateurs un espace dédié où ils peuvent gérer leur sélection de films préférés, accéder à des informations détaillées sur chaque film, et marquer les films comme vus.

Fonctionnalités

- **Vue des Favoris** : La page des favoris liste tous les films que l'utilisateur a aimé en swipant à droite.
- **Suppression d'un film** : Sous chaque film dans la liste des favoris, un bouton « Supprimer » permet à l'utilisateur de retirer un film de ses favoris. Cette action est immédiate.
- **Marque un film comme vu** : Un bouton « Marqué comme vu » est disponible. En cliquant sur ce bouton, le film est marqué comme vu par l'utilisateur. L'affiche du film change alors d'opacité pour montrer le changement de statut.
- **Accès aux informations complémentaires et bande-annonce** : En sélectionnant un film, l'utilisateur peut accéder à une page de détails offrant des informations plus complètes, comme une synopsis, l'année de sortie, et une note sur 10. Un bouton pour visionner la bande-annonce permet à l'utilisateur de regarder celle-ci directement dans l'application.
- **Interactions et expérience utilisateur** : La page des favoris est conçue pour être intuitive et facile à naviguer, avec des boutons clairement indiqués pour chaque action. Les animations et transitions douces améliorent l'expérience lors de la suppression des films ou du marquage comme vus. La possibilité de regarder les bandes-annonces directement depuis l'application enrichit l'expérience utilisateur et aide à la décision de visionnage.



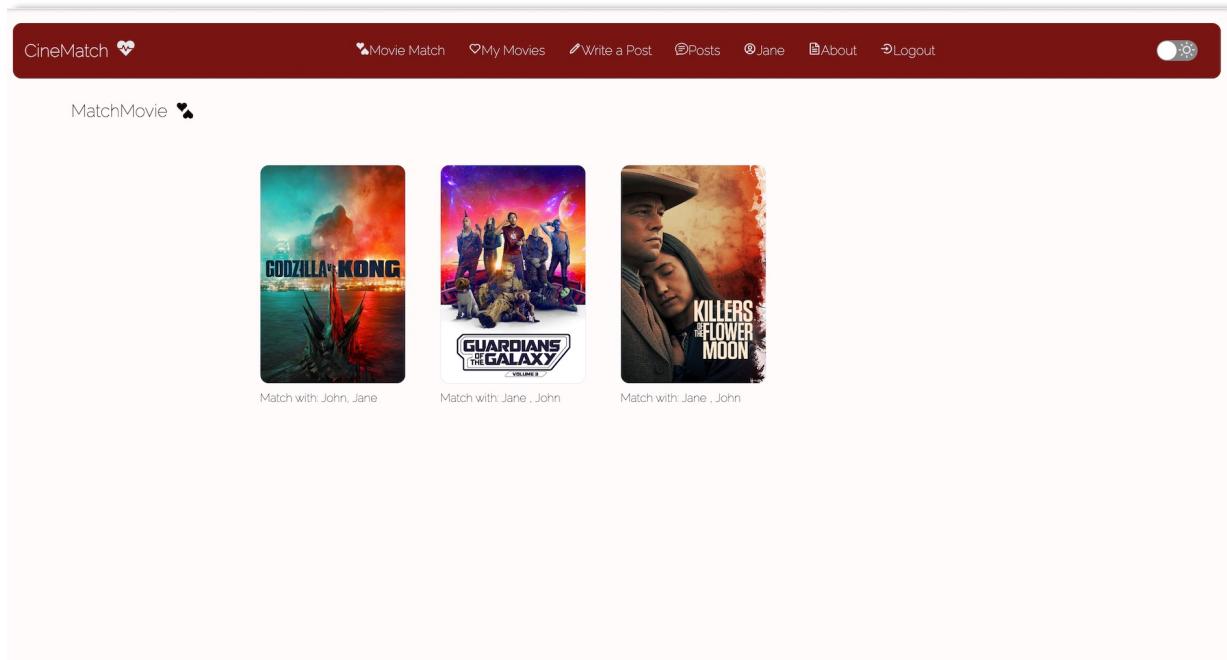
4. Match des films

Objectif

L'objectif de cette fonctionnalité est de renforcer l'aspect social de CineMatch en permettant aux utilisateurs de découvrir qu'ils partagent des intérêts avec d'autres, à travers le concept de « match » de films.

Fonctionnalités

- **Affichage des matchs** : Une section dédiée sur l'application affiche la liste des films sur lesquels deux utilisateurs ou plus ont swipé à droite.
- **Gestion des films matchés** : Les films matchés bénéficient des mêmes fonctionnalités que ceux de la page des favoris, c'est-à-dire d'accéder à des informations détaillées avec la possibilité de visionner la bande-annonce.
- **Identification des films matchés** : Sous l'affiche de chaque film matché, les noms des utilisateurs ayant également aimé le film sont affichés. Cela permet de mettre en évidence les utilisateurs partageant des intérêts similaires et peut faciliter les discussions et les échanges autour des films.



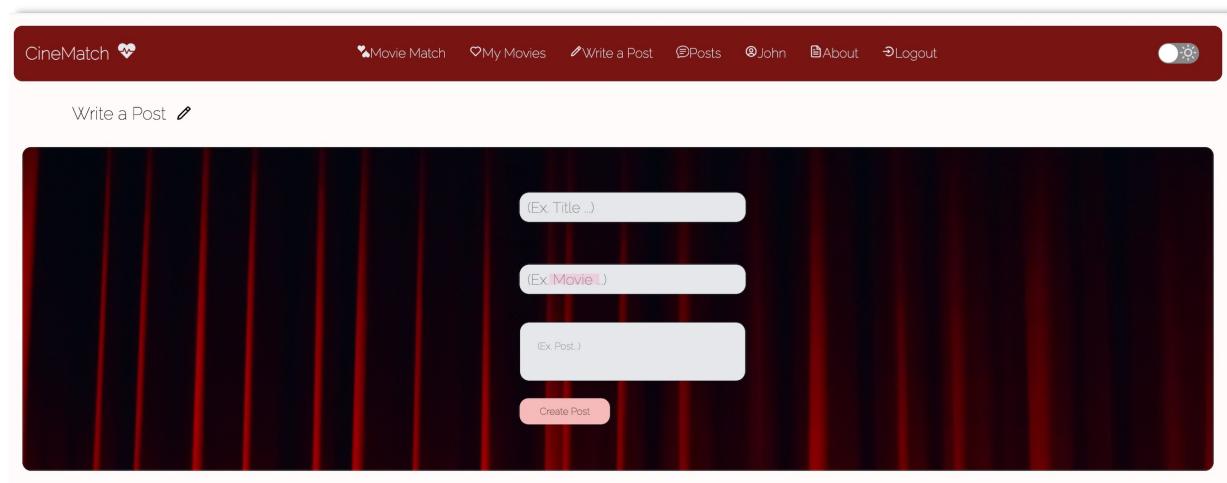
5. Page des Posts

Objectif

L'objectif est d'offrir une plateforme interactive au sein de CineMatch sur laquelle les utilisateurs peuvent partager leurs opinions et commentaires sur les films, encourager les discussions et renforcer la communauté.

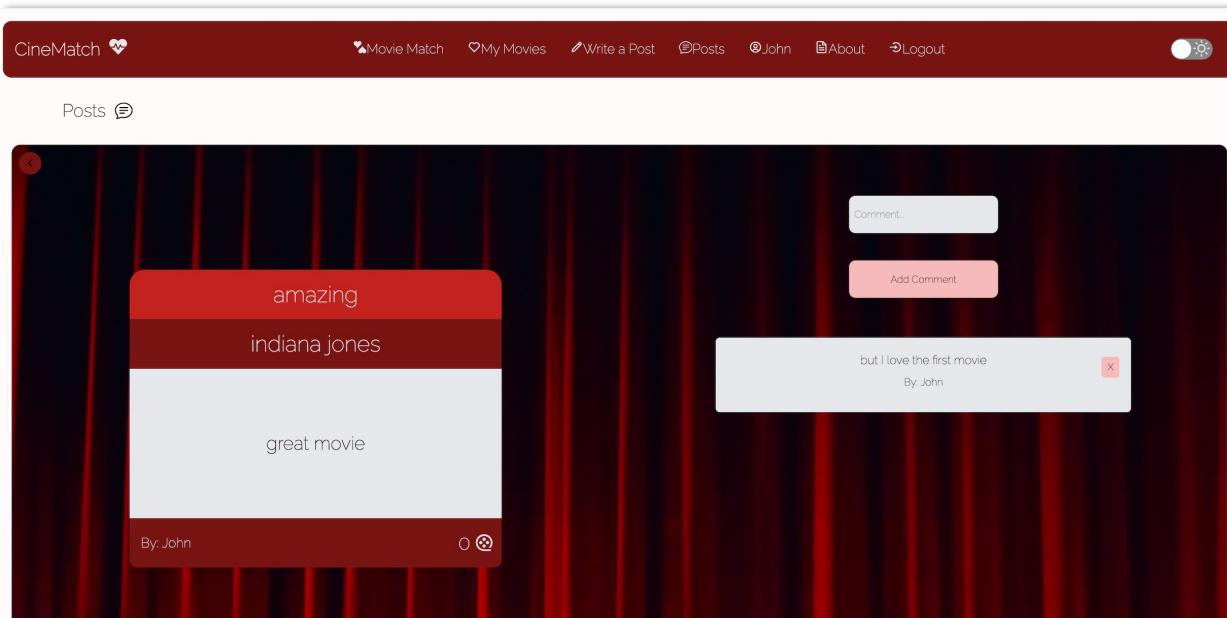
Fonctionnalités

- Création et édition de posts** : Les utilisateurs peuvent créer des posts pour exprimer leurs avis ou partager des informations sur les films. Chaque auteur a la possibilité de modifier ou supprimer son post, offrant ainsi une gestion flexible de son contenu.



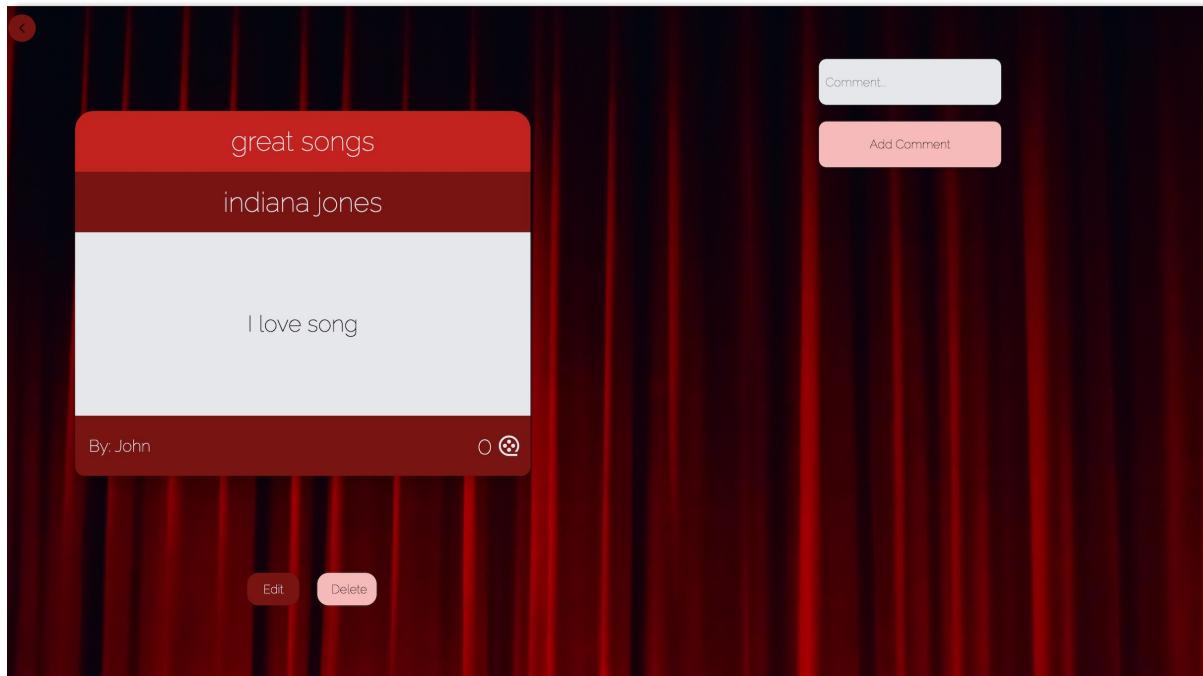
- Interaction avec les posts** : 2 fonctionnalités d'interactions avec les posts existent :

- Liker les posts : Les utilisateurs peuvent liker les posts pour exprimer leur accord ou leur appréciation du contenu partagé. Un compteur de likes permet de mesurer la popularité des posts.
- Commentaires : Sous chaque post, un espace de commentaire permet aux autres utilisateurs d'engager des discussions, de partager leur point de vue, ou de poser des questions à l'auteur du post.



- **Gestion des posts par l'auteur** : 2 fonctionnalités de gestion des posts existent :

- Suppression des posts : L'auteur d'un post a le droit de le supprimer à tout moment. Cette action est définitive.



- Modification des posts : L'auteur peut également modifier le contenu de son post, permettant ainsi de corriger des erreurs, d'ajouter des informations supplémentaires ou de mettre à jour son avis.

A screenshot of the CineMatch application's "Edit a Post" screen. The top navigation bar includes links for "Movie Match", "My Movies", "Write a Post", "Posts", "John", "About", "Logout", and a light/dark mode switch. The main content area is titled "Edit a Post" with a pencil icon. It contains four input fields: "Edit Title ...", "Edit Movie ...", and "Edit Post...". Below these is a large, prominent "Edit Post" button.

6. Page « À propos »

Objectif

La page « À propos » de CineMatch sert de point central pour accéder à des informations institutionnelles, légales et de contact. Elle vise à informer les utilisateurs sur les principes de fonctionnement de l'application, ses règles, et offre des moyens de communication avec les développeurs ou le support.

The screenshot shows the 'About' section of the CineMatch application. At the top, there is a navigation bar with links for 'Movie Match', 'My Movies', 'Write a Post', 'Posts', 'John', 'About', and 'Logout'. Below the navigation bar, the 'About' section is titled 'About'. It contains three main boxes: 'Terms and Conditions' (with a preview and 'Read More' button), 'Privacy Policy' (with a preview and 'Read More' button), and 'Code of Conduct' (with a preview and 'Read More' button). At the bottom of the section, there is a 'Contact Us' area with a preview and a 'Contact us' link, followed by a 'Connect with Us:' section featuring social media icons for YouTube, Facebook, and Twitter.

Fonctionnalités

La page « À propos » permet la navigation des utilisateurs vers des informations clés :

- Termes et Conditions** : Un lien dirige les utilisateurs vers une page détaillant les termes d'utilisation de CineMatch, incluant les droits et obligations des utilisateurs.

The screenshot shows the 'Terms and Conditions' page of the CineMatch application. At the top, there is a navigation bar with links for 'Movie Match', 'My Movies', 'Write a Post', 'Posts', 'John', 'About', and 'Logout'. Below the navigation bar, the page title is 'Terms and Conditions'. It includes a 'Last update [26 September 2023]' note and a welcome message: 'Welcome to Cinematch (the "Application")! Please read these Terms and Conditions of Use ("Terms") carefully before using the Application. By using Cinematch, you agree to these Terms and all policies mentioned herein. If you do not agree to these Terms, please do not use the Application.' The page then lists several numbered points:

- 1. How to use the Application
 - 1.1 You must be at least [18] years old to use the Application. the Application.
 - 1.2 You agree not to use the Application for any purpose that is prohibited by law.
 - 1.3 You agree not to use the Application to disseminate, store or transmit any content that is illegal, defamatory offensive, obscene, or harmful to others.
- 2. User account
 - 2.1 You may be required to create a user account to access certain features of the Application. You are responsible for the confidentiality of your login information.
 - 2.2 You agree to provide accurate and complete information when creating your user account and to keep this information up to date. keep this information up to date.
 - 2.3 You are responsible for all activities that occur under your user account.
- 3. Data protection
 - 3.1 We collect, store and use your data in accordance with our Privacy Policy. By using the Application, you consent to such collection and use.
- 4. Customer content
 - 4.1 You are solely responsible for any content you post or share on the Application. You warrant that you have the necessary rights necessary to share such content.
 - 4.2 We reserve the right to remove any content that violates these Terms or is otherwise deemed inappropriate.
- 5. Intellectual Property
 - 5.1 Cinematch holds all intellectual property rights related to the Application. You may not reproduce, distribute or use these elements without authorization.
- 6. Application and Terms Modification
 - 6.1 We reserve the right to modify or update the Application and these Terms at any time without notice.
- 7. Cancellation
 - 7.1 We may terminate or suspend your user account at our sole discretion our sole discretion, with or without notice, for any reason, including breach of these Terms.

- **Code de conduite :** Ce lien mène à une page qui décrit les comportements attendus et inacceptables au sein de la communauté CineMatch, afin de maintenir un environnement respectueux et sûr pour tous.

The screenshot shows the CineMatch mobile application interface. At the top, there is a dark header bar with the CineMatch logo and a heart icon. Below the header, the title "Code of Conduct" is displayed next to a small heart icon. The main content area contains the "CineMatch Code of Conduct" section. It starts with a welcome message: "Welcome to CineMatch! Our community is dedicated to creating an enjoyable, respectful, and inclusive cinematic experience for all our users. We encourage you to carefully read and follow our Code of Conduct to ensure a positive experience for all." Following this, there are seven numbered rules:

1. Respect Diversity: We celebrate diversity in all its forms, including race, gender, sexual orientation, religion, nationality, and culture. We expect our users to respect and treat every member of our community with respect, dignity, and understanding.
2. No Harassment: We do not tolerate harassment, discrimination, or bullying. Harassment may include, but is not limited to, offensive comments, insults, threats, discrimination, and any form of harmful behavior towards other users.
3. Privacy Policy: Respect the privacy of other users. Do not share sensitive personal information without their consent. Don't post content that violates the privacy of others.
4. Appropriate Content: Make sure the content you share is appropriate and respectful. Avoid posting images, videos, comments, or any other content that could be considered offensive, vulgar, or inappropriate.
5. Help New Users: If you're an experienced user, don't hesitate to help new users by answering their questions and guiding them through the application. Let's make sure everyone feels welcome.
6. Report Problems: If you encounter behavior that violates this Code of Conduct or that you deem inappropriate, please report it to our moderation team. We will take the necessary steps to resolve the problem.
7. Respect the Rules of the Application: Also, observe the specific rules of the application, including the terms of use and community guidelines. Any behavior in violation of these rules may result in disciplinary action, including suspension or deletion of your account. We reserve the right to take appropriate action for violations of this Code of Conduct, including the removal of content, suspension, or deletion of user accounts at our sole discretion.

 Below the rules, a note states: "By using CineMatch, you agree to abide by this Code of Conduct and help create a positive and respectful film community."

- **Politique de confidentialité :** Cette dernière fournit des détails sur la manière dont les données des utilisateurs sont collectées, utilisées et protégées par CineMatch.

The screenshot shows the CineMatch mobile application interface. At the top, there is a dark header bar with the CineMatch logo and a heart icon. Below the header, the title "Privacy Policy" is displayed next to a small info icon. The main content area contains the "Privacy Policy" section. It starts with a note: "Last update: 25 september 2023". The text explains that CineMatch ("we", "us" or "our") operates the CineMatch application (hereinafter referred to as "the Application"). This page informs you of our policy regarding the collection, use and disclosure of personal information personal data we receive from users of the Application. By using the Application, you consent to the collection and use of information in accordance with this policy.

Information Collection and Use

We may collect personally identifiable information when you use the Application. Personally identifiable information may include, but is not limited to, your name, e-mail address, telephone number and other information that identifies you as an individual. We use this information to improve your user experience, provide personalized features and respond to your requests. We may also use this information to send you updates, newsletters, special offers and other information relating to the Application. If you do not wish communications from us, you can unsubscribe by following the instructions provided in these messages.

Cookies and Usage Data

We may use cookies and other tracking technologies to collect information about your use of the Application and to improve our services. Cookies are small text files stored on your device that contain information about your browsing habits. We may also collect usage data, such as your IP address, the type of browser you are using, the pages you have visited and other diagnostic data.

Information disclosure

We will not sell, rent or share your personal information with third parties without your consent, except as described in this privacy policy or where required by law. We may disclose your personal information to the following third parties:

- To our third party service providers who assist us in operating the Application and provide our services.
- To law enforcement agencies or other governmental authorities necessary to comply with legal obligations or to protect our legal obligations or to protect our rights, privacy, security or property of us, our users or the general public.

Security

The security of your personal information is important to us, but please remember that no method of transmission over the Internet, nor no method of electronic storage is 100% secure. Please we strive to use commercially acceptable means to protect your personal information, we cannot absolute security.

Links to third-party websites

The Application may contain links to third-party websites. If you click on a third-party link, you will be redirected to that website. Please note that these third-party websites have their own privacy policies, and we accept no responsibility or legal liability for their policies. We encourage you to consult the privacy policies of these third-party websites before submitting any personal information.

Changes to Privacy Policy

We reserve the right to update or modify our privacy policy at any time. We will notify you of by posting the new privacy policy on this page. We recommend that you consult this page.

- **Contact** : Une section spécifique est dédiée au contact par mail, permettant aux utilisateurs de signaler des problèmes, de poser des questions ou de fournir des retours directement au support de CineMatch.

The screenshot shows the 'Contact Us' page of the CineMatch application. At the top, there's a dark header bar with the CineMatch logo, navigation links like 'Movie Match', 'My Movies', 'Write a Post', 'Posts', 'John', 'About', and 'Logout', and a user profile icon. Below the header is a large input form with three text fields: 'Your Full Name', 'Email Address', and 'Your Message'. A red error message 'Veuillez renseigner ce champ.' is displayed at the bottom right of the form area.

- **Réseaux sociaux** : Des liens vers les comptes officiels de CineMatch sur GitHub, LinkedIn et Twitter permettent aux utilisateurs de suivre l'actualité du projet, de contribuer ou de se connecter avec la communauté et les développeurs.

7. Page « Profil »

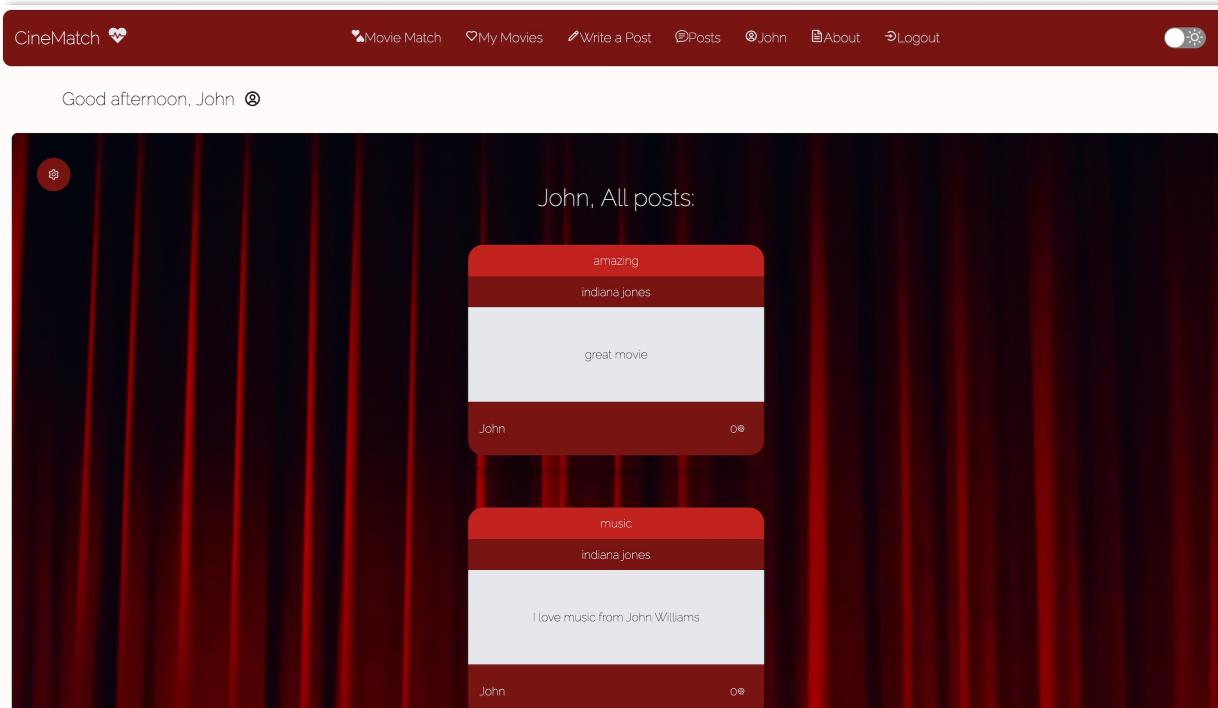
Objectif

L'objectif de la page de profil dans l'application CineMatch est de fournir un espace personnalisé où les utilisateurs peuvent visualiser et gérer leurs contributions à la communauté, ainsi que personnaliser leurs paramètres de compte. Cette page vise à centraliser les informations et activités de l'utilisateur, offrant une vue d'ensemble de ses interactions au sein de l'application, et permettant une gestion simplifiée de son compte.

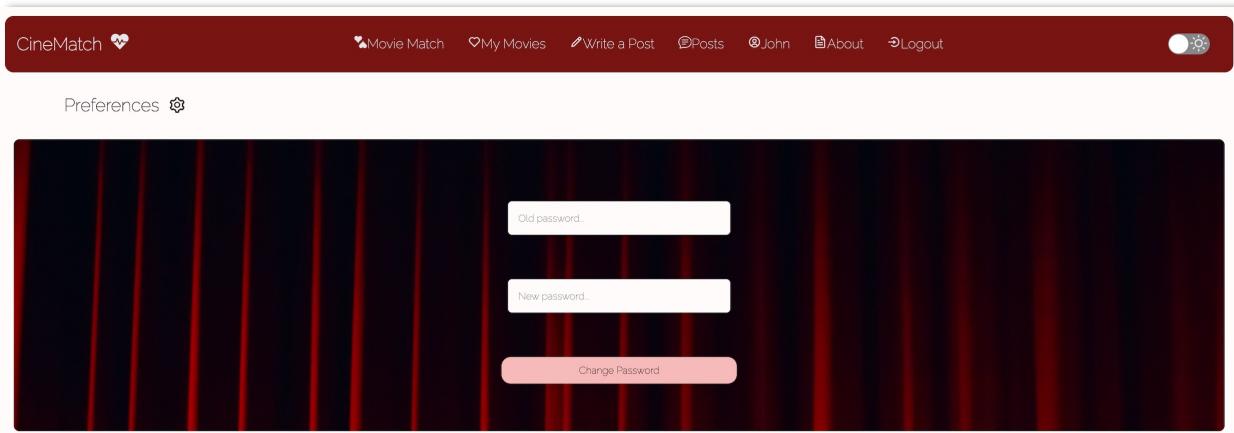
Fonctionnalités

La page « Profil » permet l'affichage des posts de l'utilisateur, l'interaction avec les posts ainsi que la gestion des préférences.

- **Affichage des posts de l'utilisateur** : La page de profil regroupe tous les posts créés par l'utilisateur, lui permettant de revoir facilement ses contributions passées, comme des avis sur des films, des discussions initiées, ou des questions posées à la communauté. Cette fonctionnalité encourage les utilisateurs à rester actifs et engagés avec le contenu de l'application.

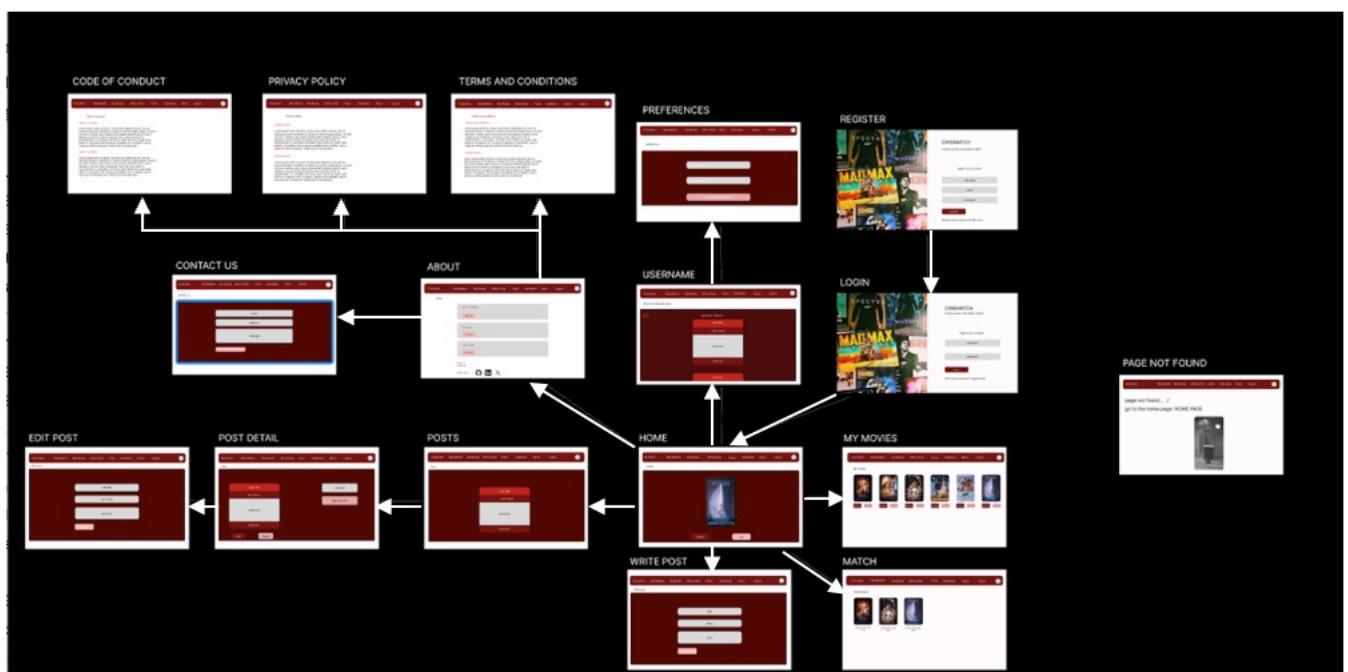
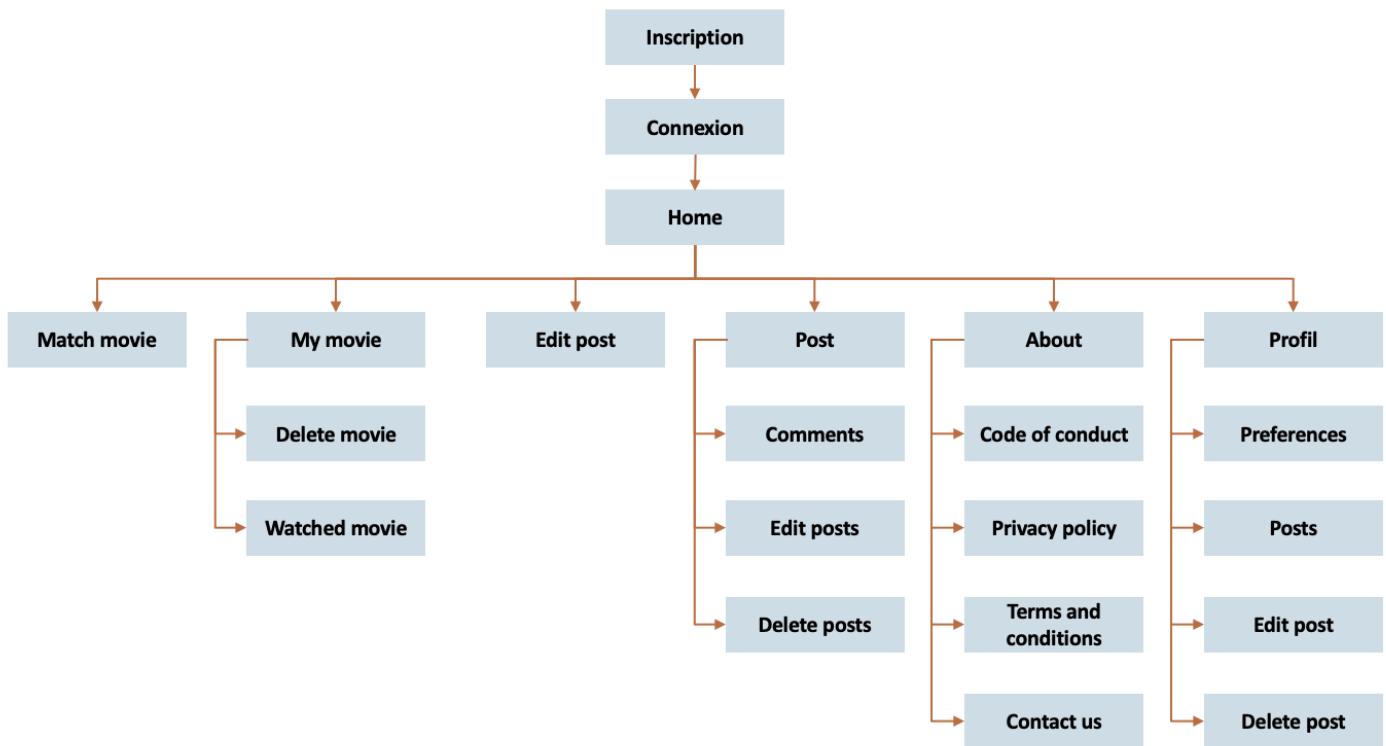


- **Interaction avec les posts** : La page permet non seulement de visualiser les posts de l'utilisateur, mais offre également la possibilité d'interagir avec eux, par exemple en les éditant ou en les supprimant. Cela donne aux utilisateurs un contrôle total sur le contenu qu'ils partagent sur la plateforme.
- **Gestion des préférences** : Un bouton ou une section de préférences est disponible pour permettre à l'utilisateur de modifier ses paramètres de compte, tel que le changement de mot de passe. Cette fonctionnalité assure que les utilisateurs peuvent maintenir la sécurité de leur compte et personnaliser leur expérience selon leurs préférences.



8. Arborescence du site

Concernant l'arborescence du site, j'ai tout d'abord formalisé un schéma visant à mettre en avant les fonctionnalités des différentes pages de l'application et à en comprendre les imbrications, avant de la formaliser à nouveau avec les maquettes sur Figma.



Spécifications techniques du projet

La stack technique, composée de React pour le front-end, Node.js pour le back-end, et MySQL pour la base de données, offre un ensemble robuste et polyvalent pour le développement d'applications web modernes. Nous allons donc détailler chacune de ces technologies ainsi que leurs avantages.

1. React

Description

React est une bibliothèque JavaScript open-source conçue pour créer des interfaces utilisateur. Développée et maintenue par Facebook, elle permet de construire des applications web et mobiles rapides et interactives en utilisant des composants réutilisables.

Avantages

- **Performance élevée** : React utilise un DOM virtuel pour minimiser les manipulations directes du DOM réel, ce qui améliore les performances.
- **Composants réutilisables** : React permet de construire des interfaces utilisateurs avec des composants réutilisables, rendant le code plus propre et facile à maintenir .
- **Large écosystème** : La technologie React donne accès à un vaste écosystème de bibliothèques et outils, facilitant l'ajout de fonctionnalités.
- **Flexibilité** : React est utilisé pour le développement de single-application (SPA) ou intégré dans des applications multi-pages.
- **Communauté active** : React bénéficie d'une large communauté de développeurs et d'un soutien important, ce qui facilite la résolution de problèmes et l'apprentissage.

2. Node.JS

Description

Node.JS est un environnement d'exécution JavaScript côté serveur basé sur le moteur V8 de Chrome. Il permet de développer des applications back-end en utilisant JavaScript, un langage traditionnellement utilisé pour le développement front-end, favorisant ainsi une expérience de développement plus uniforme entre le client et le serveur.

Avantages

- **Unicité du langage** : Node.JS utilise le même langage de programmation côté serveur et client, simplifiant le développement et réduisant le temps d'apprentissage.
- **Asynchrone et non bloquant** : L'architecture de Node.JS est basée sur des opérations d'entrée/sortie non bloquantes, ce qui rend les applications rapides et efficaces.
- **Écosystème riche** : Npm, le gestionnaire de paquets de Node.JS, est l'un des plus grands écosystèmes de bibliothèques au monde.
- **Scalabilité** : Note.JS est conçu pour être léger et efficace, idéal pour des applications en temps réel traitant un grand nombre de connexions simultanées.
- **Polyvalence** : Node.JS peut être utilisé pour une variété de projets des applications web et mobiles, en passant par des applications en temps réel.

3. MySQL

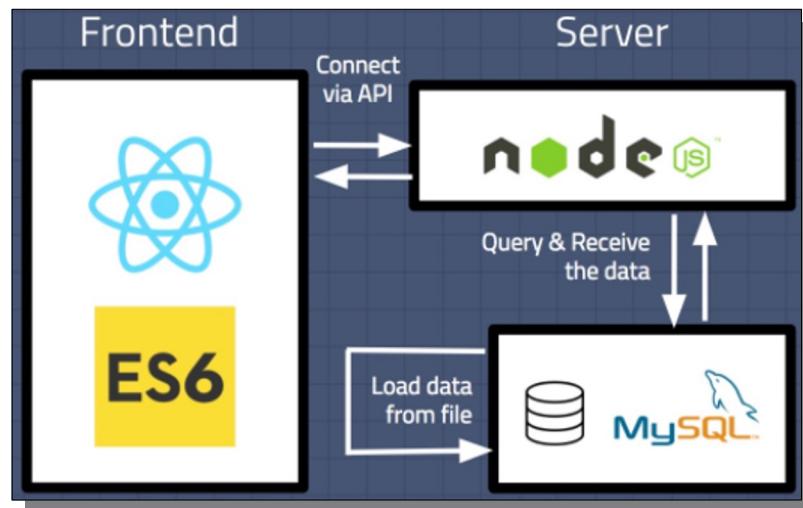
Description

MySQL est un système de gestion de base de données relationnelles (SGBDR) open-source. Il est largement utilisé pour stocker, récupérer, et gérer des données dans des applications web.

Avantages

- **Fiabilité** : MySQL est réputé pour la stabilité, la fiabilité et l'efficacité dans la gestion des bases de données.
- **Facilité d'utilisation** : La syntaxe SQL est relativement simple à comprendre et à utiliser pour la gestion des données.
- **Sécurité** : MySQL est utilisé par des entreprises du monde entier pour des applications critiques, prouvant sa fiabilité. Il offre également un ensemble robuste de fonctionnalités de sécurité pour protéger les données sensibles.
- **Compatibilité** : MySQL est pris en charge par la plupart des plateformes d'hébergement et facile à intégrer avec divers langages de programmation, notamment Node.js
- **Large adoption** : MySQL est l'une des bases de données les plus utilisées au monde, bénéficiant d'une grande communauté et de nombreuses ressources.

En combinant React, Node.js et MySQL, la stack technique de CineMatch offre une solution complète et moderne pour le développement d'applications web. Cette combinaison tire parti des avantages spécifiques de chaque technologie, de la construction d'interfaces utilisateur interactives à la gestion efficace des données côté serveur, tout en assurant la performance, la scalabilité et la maintenabilité de l'application.



Le client React envoie des requêtes HTTP et récupère les réponses HTTP à l'aide d'Axios, consomme des données sur les composants. React Router est utilisé pour naviguer vers les pages. Node.js Express exporte les API REST et interagit avec la base de données MySQL.

Réalisations personnelles

Concernant mes réalisations de développement, j'ai choisi de structurer leur valorisation en 3 temps : celles qui relèvent du front-end, celles qui relèvent du back-end et enfin les travaux relatifs à la base de données.

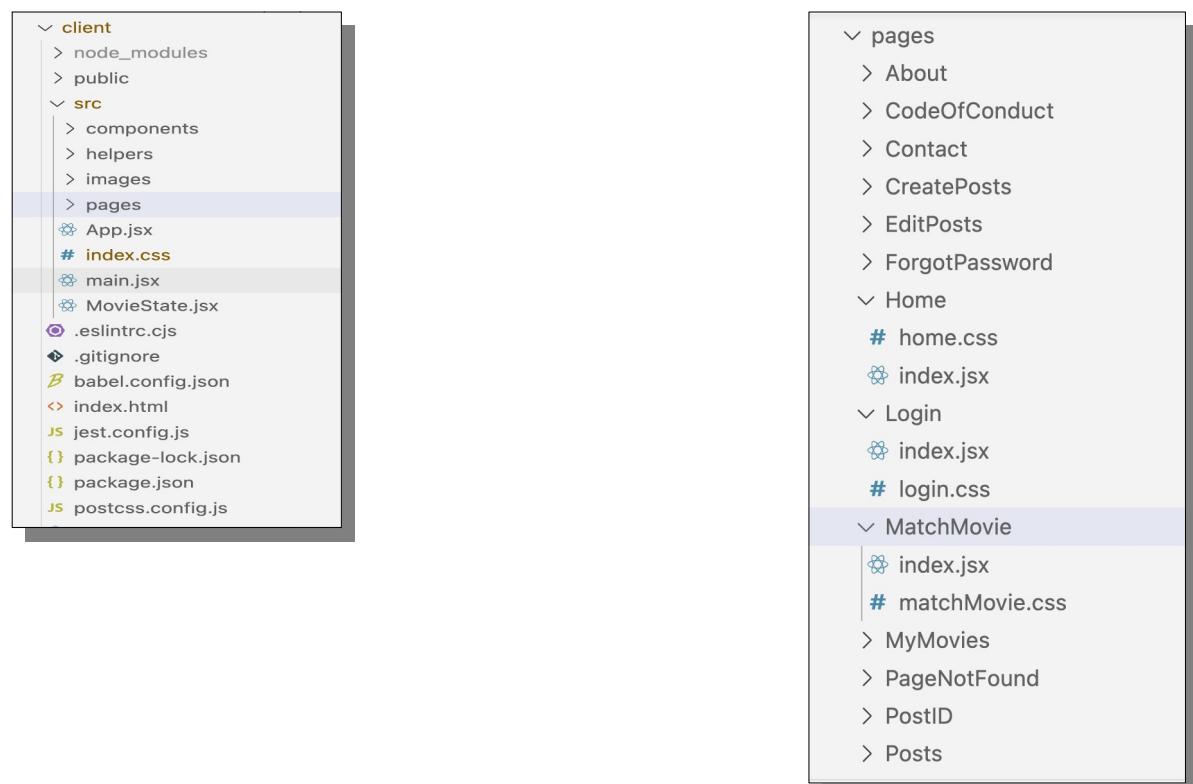
1. Front-end

Le front-end de l'application est développé en React.js. Il s'agit d'une bibliothèque JavaScript open source développée par Facebook qui permet d'élaborer des interfaces utilisateur interactives et dynamiques. Elle utilise une approche basée sur les composants pour la construction de l'interface, ce qui offre alors un gain de temps en évitant la répétition de code et en facilitant la maintenance de l'application. Son utilisation d'un Virtual DOM (Document Object Model) minimise aussi les mises à jour du DOM réel, réduisant ainsi les temps de chargement et améliorant la fluidité de l'application.

1.1 Arborescence du projet

J'ai donc suivi les bonnes pratiques en matière d'architecture des fichiers pour une application React. J'ai notamment séparé les fichiers en fonction de leur fonctionnalité : par exemple, un dossier pour les composants, un autre pour les pages, etc.

Pour des questions de confort de lecture, chaque fichier CSS se trouve dans le même dossier que la page correspondante.



J'ai également mis en place des variables CSS : elles permettent de centraliser les valeurs répétitives telles que les couleurs du site, afin que l'on puisse les modifier rapidement en un seul endroit, tout en évitant les erreurs de saisie, en facilitant la maintenance et en améliorant la lisibilité du code.

```
:root{
  /*light mode color*/
  --accent:#C2241E;
  --secondary:#F4BAB8;
  --primary:#771613;
  --background:#FEFBFB;
  --text:#0D0202;
  --black:#000000;
  --white:#e5e7eb;
  --green1-color:#84cc16;
  --green-color:#65a30d;
  --yellow-color:#ca8a04;
  --red-color:#881337;
  --red2-color:#dc2626;

  /*dark mode color*/
  --accent-dark:#E1423D;
  --secondary-dark:#470D0B;
  --primary-dark:#EC8C89;
  --background-dark:#040101;
  --text-dark:#FDF2F2;
  --black:#000000;
  --white:#e5e7eb;
}
```

1.2 Routage

Le routage permet à l'application de charger une nouvelle vue en fonction de l'URL demandée, sans actualiser la page entière.

Le gestionnaire de routage choisi, React Router, est une bibliothèque JavaScript qui gère les routes définissant les chemins URL et les vues associées. Les routes peuvent également inclure des paramètres dynamiques pour des vues plus personnalisées, comme par exemple : "/profile/:id".

```
function App() {
  return (
    <ThemeProvider>
      <div className="app-background">
        <BrowserRouter>
          <AuthContext.Provider value={{ authState, setAuthState }}>
            <Routes>
              {routes.map((route, index) => (
                <Route key={index} path={route.path} element={route.element} />
              ))}
            </Routes>
          </AuthContext.Provider>
        </BrowserRouter>
      </div>
    </ThemeProvider>
  );
}
```

ROUTE	PAGE	USER
CONNEXION		
`/auth`	Page d'inscription	Non-connecté
`/auth/login`	Page de connexion	Non-connecté
MENU		
`/`	Page home	Connecté
`/matchmovie`	Page MatchMovie	Connecté
`/mymovies`	Page des favoris	Connecté
`/createposts`	Page de création de posts	Connecté
`/post`	Page des posts	Connecté
/post/:id	Page des posts avec les commentaires	Connecté
`/about`	Page à propos	Connecté
`/profile/:id`	Page de profil de l'utilisateur	Connecté
`/contact`	Page de contact	Connecté
`/privacypolicy`	Page de politique de confidentialité	Connecté
`/codeofconduct`	Page du code de conduite	Connecté
`/termsandconditions`	Page des termes et des conditions	Connecté
`/editposts/:id`	Page permettant de mettre à jour un post	Connecté
`/preferences/:id`	Page pour mettre à jour le mot de passe de l'utilisateur	Connecté

1.3 Extraits de code

La page HOME est un composant React qui sert de page d'accueil pour l'application web "CineMatch", où les utilisateurs peuvent swiper à gauche ou à droite pour aimer ou ne pas aimer des films. Voici une explication des fonctions et des éléments utilisés dans le code :

- **useState** : Hook de React est utilisé pour déclarer des états locaux dans le composant, ici pour stocker les films (movies), l'index du film actuel (currentIndex) et les films aimés (likedMovies).
- **useEffect** : Hook est exécuté au montage du composant pour charger les films populaires via une requête API en utilisant axios.
- **useContext** : Permet d'accéder au contexte d'authentification (AuthContext) et au thème de l'application (ThemeContext), pour gérer les informations de l'utilisateur connecté et le thème actuel.
- **axios** : Est une bibliothèque de requêtes HTTP pour effectuer des appels API, ici utilisée pour récupérer les films populaires et poster les "likes" des films.

```
export default function Home() {
  const [movies, setMovies] = useState([]);
  const [currentIndex, setCurrentIndex] = useState(0);
  const [likedMovies, setLikedMovies] = useState([]);
  const { authState } = useContext(AuthContext);
  const { toggleTheme, theme } = useContext(ThemeContext);

  useEffect(() => {
    fetchMovies();
  }, []);

  const fetchMovies = async () => {
    try {
      const response = await axios.get(
        "https://api.themoviedb.org/3/movie/popular",
        {
          params: {
            page: Math.floor(Math.random() * 10),
            api_key: "d4450c97d6097f870aa6ce5d66f8963d",
          },
        }
      );
      setMovies(response.data.results);
    } catch (error) {
      console.error("Erreur lors de la récupération des films :", error);
    }
  };

  const handleSwipeLeft = () => {
    setCurrentIndex(currentIndex + 1);
    console.log("Swipe left"); 'Swipe left'
  };

  const handleSwipeRight = async () => {
    const likedMovie = movies[currentIndex];

    try {
      await axios.post("http://localhost:3001/movies/like", {
        userId: authState.id,
        movieId: likedMovie.id,
      });

      setLikedMovies([...likedMovies, likedMovie]);

      toast.success("Movie liked successfully!");
      console.log("Movie liked successfully"); 'Movie liked successfully'
    } catch (error) {
      toast.error("Error liking the movie..."); 
      console.error("Erreur lors de l'enregistrement du like :", error);
    }
  };

  setCurrentIndex(currentIndex + 1);
};

const renderMovies = () => {
  return movies.slice(currentIndex, currentIndex + 1).map((movie) => (
    <TinderCard
      key={movie.id}
      onSwipeLeft={handleSwipeLeft}
      onSwipeRight={handleSwipeRight}
    >
      {movie.poster_path && (
        <img
          className="movie"
          src={`https://image.tmdb.org/t/p/original${movie.poster_path}`}
        />
      )}
      <MovieCard movie={movie} />
    </TinderCard>
  ));
};
```

- **TinderCard** : Est un composant de la bibliothèque react-tinder-card qui implémente le comportement de swipe (glissement) pour aimer ou ne pas aimer les films.
 - **MovieCard** : Est un composant personnalisé qui affiche les détails d'un film.
 - **React Icons (ImCross, BsHeartFill)** : Sont utilisés pour afficher des icônes stylisées pour les boutons de swipe
 - **toast de react-hot-toast** : Est utilisé pour afficher des notifications à l'utilisateur après une action (par exemple, un film aimé avec succès).
 - **fetchMovies** : Est une fonction asynchrone qui récupère une liste de films populaires depuis l'API TMDB et met à jour l'état movies.
 - **handleSwipeLeft** et **handleSwipeRight** : Sont des fonctions déclenchées lorsqu'un utilisateur swipe sur un film.
- HandleSwipeRight enregistre également le film aimé dans la base de données via une requête POST et met à jour la liste des films aimés.
- **renderMovies** : Est une fonction qui rend le film actuel à swiper, en utilisant TinderCard pour activer les gestes de swipe.

Ce composant illustre une intégration complexe de différentes technologies et pratiques dans le développement d'applications React comme la gestion d'états, les effets latéraux, la consommation d'APIs, et l'interaction utilisateur avancée avec le swipe.

```
return (
  <>
    <Navbar />
    <div className="background-home" id={theme}>
      <h2 className="header" id={theme}>
        <WelcomeUsers />
      </h2>
      <div className="home-screen" id={theme}>
        <div className="movie-home">{renderMovies()}</div>
        <div className="swipe-buttons-home">
          <button
            className="wrong-button-home"
            onClick={handleSwipeLeft}
            id={theme}>
            <ImCross className="wrong-button-home-icon" />
          </button>
          <button
            className="valide-button-home"
            onClick={handleSwipeRight}
            id={theme}>
            <BsHeartFill className="valide-button-home-icon" />
          </button>
        </div>
        <div>
          <Toaster />
        </div>
      </div>
    </div>
  );
}
```

1.4 Conception responsive

J'ai conçu l'application de manière à ce qu'elle soit responsive. Cet aspect-là d'un site web est lié à sa capacité à s'adapter à différents appareils et tailles d'écran, comme les ordinateurs de bureau, les tablettes et les smartphones, en réorganisant et en redimensionnant les éléments du site pour optimiser l'expérience utilisateur.

Pour cela, j'ai mis en place des media queries, qui permettent de modifier le style et la mise en page d'un site en fonction de la taille de l'écran. Les pages web ont été conçues avec plusieurs points de rupture, où la mise en page, la taille de la police et les images peuvent être ajustées pour mieux s'adapter à chaque taille d'écran.



1.5 Bibliothèques externes

J'ai choisi plusieurs bibliothèques externes pour la partie front de mon projet. J'ai tâché de les limiter afin d'éviter des problèmes de fonctionnement de l'application si jamais les bibliothèques n'étaient pas maintenues.



Parmi elles il y a Formik, une bibliothèque open source pour React qui facilite la gestion des formulaires dans les applications web. Elle fournit des composants prêts à l'emploi, tels que **Form**, **Field**, **ErrorMessage** pour créer des formulaires. Elle gère l'état du formulaire, y compris les valeurs des champs, les erreurs et le statut de soumission. Elle offre également une intégration avec Yup, une bibliothèque de validation de schémas JavaScript, pour faciliter la validation des données saisies dans le formulaire.

Autre bibliothèque : jwt-decode, qui permet de décoder les JSON Web Tokens (JWT) en JavaScript. Les JWT peuvent être utilisés pour authentifier les utilisateurs, échanger des informations entre différents services ou pour stocker des informations d'état de session.

1.6 Sécurité

Côté front, des mesures de sécurité ont été mises en place notamment lors de l'authentification. Les mots de passe autorisés doivent contenir au moins 8 caractères dont 1 majuscule, 1 caractère spécial et 1 chiffre. S'il y a une erreur dans le mot de passe ou le mail de l'utilisateur, on ne précise pas sur lequel des deux l'erreur porte, afin d'éviter de donner des indices lors de possibles attaques.

1.7 Connexion du front et du back avec Axios

Axios est une bibliothèque cliente HTTP basée sur les promesses pour le navigateur et Node.js, utilisée pour effectuer des requêtes HTTP. Elle offre une interface facile à utiliser pour envoyer des requêtes asynchrones et gérer les réponses, avec une prise en charge automatique de la transformation des données JSON. Axios facilite également l'interception de requêtes et de réponses, le traitement d'erreurs global et la protection contre les attaques CSRF/XSRF.

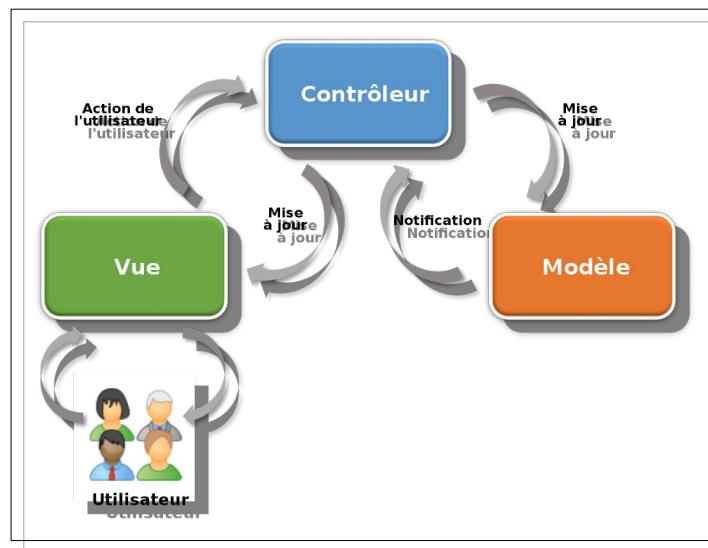
2. Back End

Le back-end de l'application s'appuie sur Node.js et le framework Express.

Node.js est une plateforme open source qui permet l'exécution de JavaScript côté serveur, unifiant ainsi le développement web autour d'un seul langage. Caractérisé par un modèle asynchrone et orienté événement, Node.js excelle dans la gestion d'applications en temps réel et de services web, offrant des performances élevées grâce à son exécution non bloquante. Il intègre npm, le plus grand écosystème de bibliothèques open source, enrichissant ainsi les projets avec une variété d'outils et de modules facilement accessibles. Multiplateforme, Node.js supporte divers systèmes d'exploitation, facilitant le déploiement et la maintenance des applications. Malgré certains défis, tels que la gestion des tâches CPU-intensives, Node.js reste une solution de choix pour de nombreux développeurs, grâce à sa communauté dynamique et à son approche flexible du développement web.

Express.js est un framework web rapide et minimaliste pour Node.js, facilitant la création d'applications web et d'APIs grâce à des fonctionnalités simplifiées de routage et de middleware. Lancé en 2010, il se distingue par sa flexibilité et sa rapidité de développement, permettant une personnalisation aisée selon les besoins du projet. Express.js supporte une vaste gamme de bases de données et outils de template, rendant le développement d'applications modernes accessible. Sa popularité repose aussi sur une large communauté et un riche écosystème de plugins, consolidant sa position comme le framework préféré pour les projets web en Node.js.

Ces fonctions intermédiaires s'exécutent entre la réception d'une requête HTTP d'un client et l'envoi d'une réponse par le serveur. Elles sont utilisées généralement pour effectuer des tâches telles que la validation des données, l'authentification des erreurs, l'autorisation d'accès, le stockage de sessions, la compression de fichiers, etc.



2.1 Architecture

Le projet s'appuie sur le modèle MVC (Modèle-Vue-Contrôleur), un modèle d'architecture logicielle. Il s'appuie sur trois composants principaux :

Le modèle : Celui-ci représente les données (MySQL) et la logique métier de l'application (dossier models).

La vue : Elle représente l'interface utilisateur de l'application. Elle est responsable de l'affichage des données à l'utilisateur et de la réception des entrées de l'utilisateur. React occupe ce rôle.

Le contrôleur : Il agit comme un médiateur entre la vue et le modèle (dossier Routes). Il reçoit les entrées de l'utilisateur à partir de la vue, traite ces entrées et effectue des modifications appropriées sur le modèle. Il est également responsable de la mise à jour de la vue pour refléter les modifications apportées au modèle.

Ici, le back-end est associé au modèle et au contrôleur car il est responsable de la gestion des données et de la logique métier de l'application.

✓ middlewares
JS AuthMiddleware.js
✓ models
JS Comments.js
JS index.js
JS LikedMovies.js
JS Likes.js
JS Movies.js
JS Posts.js
JS Users.js
> node_modules
✓ routes
JS Comments.js
JS ForgotPassword.js
JS Likes.js
JS Movies.js
JS Posts.js
JS Users.js

2.2 Routage

Les routes du back-end servent de pont entre l'interface utilisateur et la base de données, gérant les requêtes HTTP pour fetcher, créer, mettre à jour ou supprimer des données. Elles sont essentielles pour assurer la fluidité des interactions de l'utilisateur avec l'application, permettant une expérience utilisateur cohérente et dynamique à travers les différentes fonctionnalités offertes.

MÉTHODE	ENDPOINT	OUTPUT	AUTH
CONNEXION			
POST	`/auth`	Permet de s'inscrire	Non-connecté
POST	`/login`	Permet de se connecter	Non-connecté
GET	`/valid`	Permet d'être authentifier avec le token	Connecté
USER			
GET	`/profileinfo/:id`	Retourne le profil d'un utilisateur	Connecté
PUT	`/changepassword`	Retourne la mise à jour du mot de passe	Connecté
POST			
POST	`/posts`	Création de posts	Connecté
GET	`/posts`	Retourne la page de tous les posts	Connecté
PUT	`/posts/byId/:id`	Mise à jour des posts	Connecté
DELETE	`/posts/:postId`	Supprime le post par son ID	Connecté
MOVIES			
POST	`/like`	Création d'un film en favori	Connecté
GET	`/movies`	Récupère les informations des films	Connecté
DELETE	`/remove/:id`	Supprime le film des favoris	Connecté
PATCH	`/watched/:id`	Boolean pour mettre le film en vu	Connecté
GET	`/match-liked-movies`	Requête pour trouver les films aimés par plusieurs utilisateurs	Connecté
COMMENTAIRES			
GET	`/:postId`	Permet d'obtenir le post lié au commentaire	Connecté
POST	`/`	Ajoute un nouveau commentaire	Connecté
DELETE	`/:commentId`	Supprime un commentaire par son ID	Connecté
LIKE			
POST	`/`	Ajoute ou supprime un like sur un post	Connecté
GET	`/count/:postId`	Compte le nombre de like par post	Connecté

2.3 Composants d'accès aux données

Parmi les composants d'accès aux données, nous avons d'abord la base de données qui stocke les données (développée grâce à MySQL), avec possibilité de les manipuler via des requêtes.

Il y a aussi l'API (Interface de programmation d'application) qui permet aux apps d'accéder aux données à travers des protocoles standardisés. La nôtre est donc RESTful et est basée sur l'idée que chaque ressource (par exemple, un utilisateur, un commentaire, une photo, etc.) doit être accessible via une URL unique.

Les actions possibles sur chaque ressource (telles que la lecture, la création, la mise à jour ou la suppression) sont représentées par des méthodes HTTP, comme GET, POST, PUT et DELETE.

```
// Register a new user
router.post("/", async (req, res) => { ... });

// Login endpoint
router.post("/login", async (req, res) => { ... });

// Auth endpoint
router.get("/valid", validateToken, (req, res) => { ... });

//profile endpoint
router.get("/profileinfo/:id", validateToken, async (req, res) => { ... });

//change password endpoint
router.put("/changepassword", validateToken, async (req, res) => { ... });

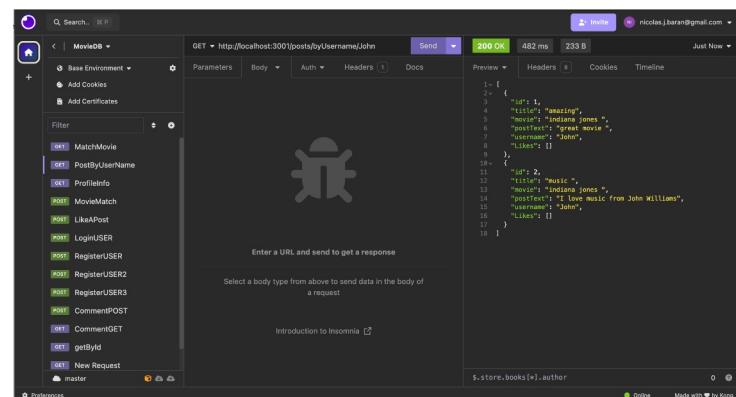
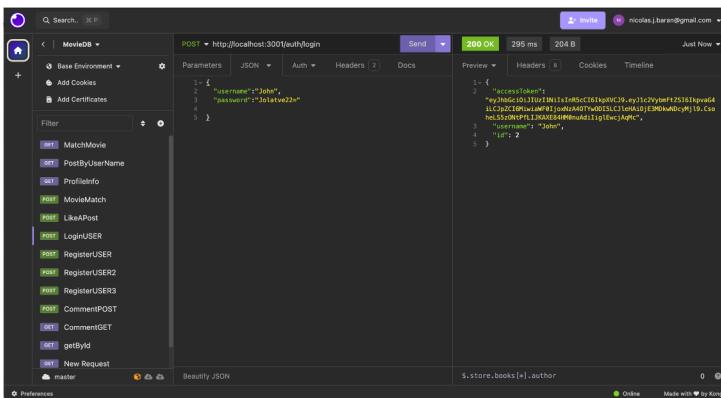
module.exports = router;
```

J'ai aussi un ORM (Object-relational mapping) qui est Sequelize. Il fournit une abstraction entre les objets du code et la base de données, facilitant l'accès et la manipulation de celles-ci. Cette dernière s'effectue dans les controllers.

Sequelize est un ORM (Object-Relational Mapping) pour Node.js, qui permet aux développeurs de travailler avec des bases de données relationnelles de manière plus intuitive et orientée objet. Au lieu d'écrire des requêtes SQL brutes, Sequelize permet d'utiliser des méthodes et des fonctions JavaScript pour interagir avec la base de données, ce qui rend le code plus lisible, plus facile à maintenir et moins sujet aux erreurs.

En tant qu'ORM, Sequelize prend en charge des fonctionnalités clés comme la modélisation de données, la migration, la relations/associations, sécurité et les opérations CRUD

De plus j'ai utilisé un outil de test API (Insomnia) qui facilite la conception, l'exécution et le débogage de requêtes REST. Il permet l'envoi de différents types de requêtes HTTP comme GET, POST, PUT, DELETE, etc., à des endpoints d'API, la gestion d'environnements et de variables, ainsi que l'authentification intégrée pour tester des endpoints sécurisés. Il m'a permis d'organiser les requêtes en collections, visualiser les réponses de manière flexible, et même valider automatiquement les réponses via des tests. Insomnia supporte également l'extension de ses fonctionnalités via des plugins.



2.4 Bibliothèques externes

Dans le développement de l'application CineMatch, j'ai utilisé plusieurs bibliothèques back-end cruciales pour créer une expérience serveur optimisée et sécurisée.

Ces bibliothèques, listées dans le fichier package.json de notre projet Node.js, fournissent un ensemble d'outils variés qui facilitent le développement et améliorent la performance de notre application.

Ces bibliothèques sont au cœur de notre back-end et jouent un rôle vital dans le développement d'une application sécurisée, fiable et facile à maintenir.

```
"dependencies": {
  "@cfaester/enzyme-adapter-react-18": "^0.7.1",
  "@emailjs/browser": "^3.11.0",
  "cors": "^2.8.5",
  "crypto-js": "^4.2.0",
  "dotenv": "^16.4.5",
  "express": "^4.18.2",
  "formik": "^2.4.4",
  "nodemailer": "^6.9.7",
  "nodemon": "^3.0.1",
  "react-router-dom": "^6.21.3",
  "react-switch": "^7.0.0",
  "sequelize": "^6.33.0",
  "sqlite3": "^5.1.7",
  "validator": "^13.11.0"
},
```

2.5 Tests unitaires

Pour tester le back-end de notre projet, la bibliothèque externe Jest a été utilisée.

Jest est un framework de test JavaScript populaire et complet, maintenu par Facebook. Il est conçu pour être exécuté avec des applications Node.js, React, Angular, Vue et autres frameworks JavaScript modernes. Jest est apprécié pour sa simplicité d'utilisation et nécessite généralement peu ou pas de configuration pour démarrer.

```
const app = express();
app.use(bodyParser.json()); // pour parser les requêtes JSON
app.use("/", usersRouter); // Attachez votre routeur ici

describe("POST /", () => {
  it("devrait créer un nouvel utilisateur", async () => {
    const newUser = {
      username: "testUser",
      email: "test@example.com",
      password: "password123",
    };

    Users.create.mockResolvedValue(newUser); // Simule une réponse réussie de Users.create

    const response = await request(app).post("/").send(newUser);

    expect(response.statusCode).toBe(201);
    expect(response.body).toEqual(expect.any(Object));
  });
});
```

2.6 Sécurité

Pour sécuriser l'application, je me suis aidé de plusieurs bibliothèques externes.

D'abord, bcrypt, une fonction de hashage de mot de passe. Elle utilise un algorithme de chiffrement symétrique, appelé Blowfish, qui emploie une technique de salage rendant chaque hashage unique. Le salage consiste à ajouter une chaîne aléatoire au mot de passe avant de le hasher, de sorte que même si deux utilisateurs ont le même mot de passe, leur hashage sera distinct en raison de la différence de sel.

Afin de fournir une couche de sécurité supplémentaire, j'ai associé bcrypt à JWT (JSON Web Token). Il s'agit d'un standard ouvert pour la création de jetons d'authentification visant à partager des informations d'utilisateur de manière sécurisée. Pour créer un JWT, le serveur crée un payload (charge utile), signe le jeton à l'aide d'une clé secrète ou privée, puis renvoie le jeton au client. Le client stocke le jeton dans un cookie ou un local storage et l'envoie avec chaque requête au serveur. Celui-ci peut alors vérifier la validité du jeton, via la signature à l'aide de la clé publique correspondante, et que la charge utile contient les informations nécessaires pour autoriser la requête.

Par exemple le code ci-dessous illustre un endpoint de connexion dans une application Node.js utilisant Express.js. Il extrait le nom d'utilisateur et le mot de passe de la requête, vérifie l'existence de l'utilisateur dans la base de données, et utilise Bcrypt pour comparer le mot de passe fourni avec celui stocké. Si l'authentification réussit, il génère un token d'accès JWT (JSON Web Token) contenant le nom d'utilisateur et l'identifiant de l'utilisateur, qu'il envoie ensuite au client. Si l'utilisateur n'est pas trouvé ou si la combinaison nom d'utilisateur/mot de passe est incorrecte, il renvoie une erreur 401 et le token a une durée de validité de 24 heures.

```
// Login endpoint
router.post("/login", async (req, res) => {
  const { username, password } = req.body;
  const user = await Users.findOne({ where: { username: username } });

  if (!user) {
    res.status(401).json({ error: "User not found" });
    return;
  }

  const match = await bcrypt.compare(password, user.password);

  if (!match) {
    res.status(401).json({ error: "Wrong Username and password combination" });
    return;
  }

  // Création d'un token d'accès avec jsonwebtoken
  const accessToken = sign({ username: user.username, id: user.id }, "secrettoken", {
    expiresIn: "24h",
  });
  res.json({ accessToken: accessToken, username: user.username, id: user.id });
});
```

J'ai également créé un middleware pour vérifier si les requêtes à l'API contiennent un jeton JWT valide dans l'en-tête d'autorisation. Si c'est le cas, le middleware permet à la requête de continuer. Sinon, il renvoie une réponse d'erreur.

```
const { verify } = require("jsonwebtoken");

const validateToken = (req, res, next) => {
  const accessToken = req.header("Authorization");

  if (!accessToken) {
    return res.status(401).json({ error: "Unauthorized: User not logged in!" });
  }

  try {
    const decodedToken = verify(accessToken, "secrettoken");
    req.user = decodedToken; // Stocke les informations du jeton décodé dans l'objet
    next(); // Passe au middleware suivant ou à la route.
  } catch (error) {
    return res.status(403).json({ error: "Forbidden: Invalid token" });
  }
};

module.exports = { validateToken };
```

3. Base de données

3.1 Modèle conceptuel de données

USER	MOVIE	POST	COMMENT
ID	ID	ID	ID
UserName	Title	Title	CommentBody
Email	Original Language	Movie	UserName
Password	Original title	PostText	
	Overview	UserName	
	Popularity		
	Poster path		
	Backdrop path		
	Release date		
	Vote		
	Watch		

LIKE	LIKED MOVIE
ID	ID
UserID	Movield
IsLike	UserId
PostID	

Relations :

- Un User peut avoir 0 ou plusieurs Posts.
- Un Post appartient à 1 User.
- Un User peut avoir 0 ou plusieurs Comments.
- Un Comment appartient à 1 User et 1 Post.
- Un Movie peut avoir 0 ou plusieurs Likes.
- Un Post peut avoir 0 ou plusieurs Likes.
- Un User peut aimer 0 ou plusieurs Movies et Posts.

3.2 Modèle logique de données

Ce modèle détaille les structures spécifiques telles que les tables, les clés primaires et étrangères, et les attributs exacts de la base de données.

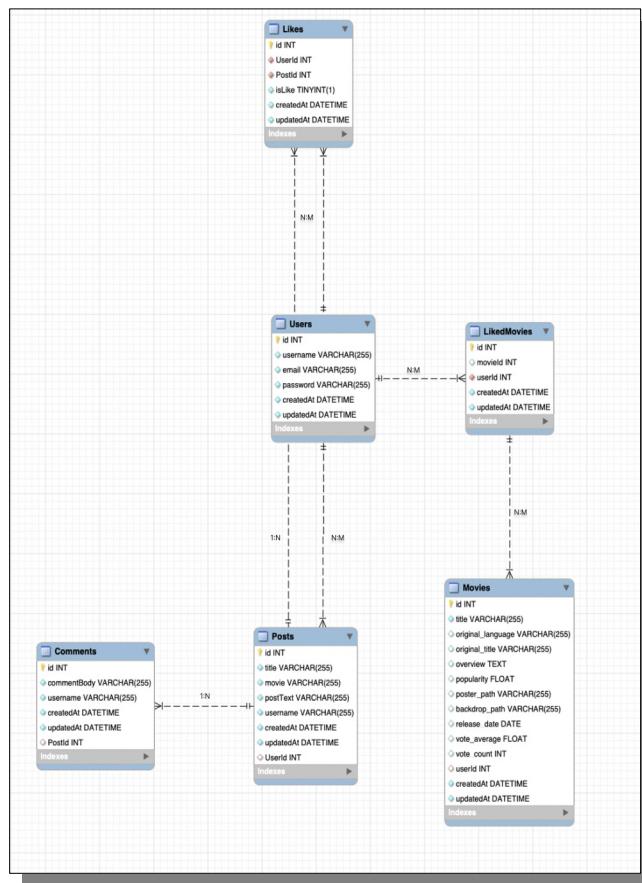
USER		MOVIE	
ID	int AI PK	ID	int PK
UserName	varchar(255)	Title	varchar(255)
Email	varchar(255)	Original Language	varchar(255)
Password	varchar(255)	Original title	varchar(255)
		Overview	text
		Popularity	float
		Poster path	varchar(255)
		Backdrop path	varchar(255)
		Release date	date
		Vote	float
		Watch	tinyint(1)

POST		COMMENT	
ID	Int AI PK	ID	Int AI PK
Title	varchar(255)	CommentBody	varchar(255)
Movie	varchar(255)	UserName	varchar(255)
PostText	varchar(255)	PostId	int
UserName	varchar(255)		
UserId	int		

LIKE		LIKED MOVIE	
ID	Int AI PK	ID	Int AI PK
UserID	int	Movield	int
IsLike	tinyint(1)	UserId	int
PostID	int		

3.3 Diagramme UML

Le diagramme UML illustre les différentes entités et leurs relations.



Voici une description de ce diagramme :

- **User** : Cette entité représente les utilisateurs de l'application avec des attributs tels que l'identifiant, le nom d'utilisateur, l'e-mail et les dates de création et de mise à jour. Chaque utilisateur peut avoir des interactions multiples avec des films et des posts, représentées par les relations N-N avec les entités "LikedMovies" et "Posts".

- **LikedMovies** : Cette entité capture les films que les utilisateurs ont aimés. Il existe une relation N-N entre "Users" et "Movies", car un utilisateur peut aimer plusieurs films et un film peut être aimé par plusieurs utilisateurs.
- **Movies** : Elle représente les films disponibles dans l'application. Chaque film possède des détails tels que le titre, la langue originale, la popularité, et d'autres attributs pertinents. Les films sont liés aux "LikedMovies" et peuvent être associés à des posts.
- **Posts** : Les utilisateurs de CineMatch peuvent créer des posts qui sont liés à des films spécifiques. Cette entité a des relations N-1 avec "Users", indiquant que chaque post est créé par un seul utilisateur, et N-N avec "Movies", car un post peut être associé à plusieurs films.
- **Comments** : Ils sont associés à des posts spécifiques, avec une relation N-1 indiquant qu'un commentaire est associé à un seul post, mais un post peut avoir plusieurs commentaires.
- **Likes** : Cette entité est liée aux posts, avec une relation N-M montrant qu'un utilisateur peut aimer plusieurs posts et qu'un post peut être aimé par plusieurs utilisateurs.

3.4 Sécurité

En termes de sécurité et en alignement avec les exigences du Règlement Général sur la Protection des Données (RGPD), il est crucial d'intégrer une Politique de Confidentialité, des Termes et Conditions, un Code de Conduite et un moyen de Contact par E-mail pour CineMatch. Ces documents jouent un rôle essentiel pour garantir la transparence, la sécurité et le respect de la vie privée des utilisateurs, tout en définissant le cadre juridique et éthique de l'utilisation de l'application.

Politique de Confidentialité

Ce document détaille les méthodes de collecte, de traitement, et de sécurisation des données personnelles des utilisateurs. Elle indique clairement les droits des utilisateurs en vertu du RGPD, tels que le droit à l'information, à l'accès, à la rectification et à l'effacement des données. La politique met en lumière les mesures prises pour protéger les données contre les accès non autorisés et les fuites.

[Privacy Policy](#)

Last update: 26 september 2023

CineMatch ('we', 'us' or 'our') operates the CineMatch application (hereinafter referred to as 'the Application'). This page informs you of our policy regarding the collection, use and disclosure of personal information personal data we receive from users of the Application. By using the Application, you consent to the collection and use of information in accordance with this policy.

[Information Collection and Use](#)

We may collect personally identifiable information when you use the Application. Personally identifiable information may include, but is not limited to, your name, e-mail address, telephone number and other information that information that identifies you as an individual. We use this information this information to improve your user experience, provide personalized personalized features and respond to your requests. We may also use this information to send you updates, newsletters updates, newsletters, special offers and other information other information relating to the Application. If you do not wish communications from us, you can unsubscribe by following the instructions provided in these messages.

[Cookies and Usage Data](#)

We may use cookies and other tracking technologies to collect information about your use of the Application and to improve our services. Cookies are small text files stored on your device that contain information about your browsing your browsing habits. We may also collect usage data, such as your IP address, the type of browser browser you are using, the pages you have visited and other other diagnostic data.

[Information disclosure](#)

We will not sell, rent or share your personal information with third with third parties without your consent, except as described in this described in this privacy policy or where required by law. We may disclose your personal information to the following third parties:

- To our third party service providers who assist us in operating the the Application and provide our services.
- To law enforcement agencies or other governmental authorities necessary to comply with legal obligations or to protect our legal obligations or to protect our rights, privacy privacy, security or property of us, our users or the general of our users or the general public.

[Security](#)

The security of your personal information is important to us, but please remember that no method of transmission over the Internet, nor no method of electronic storage is 100% secure. Please we strive to use commercially acceptable means to protect your commercially acceptable means to protect your personal information, we cannot absolute security.

[Links to third-party websites](#)

The Application may contain links to third-party websites. If you click on a third-party link, you will be redirected to that website. Please note that these third-party websites have their own privacy policies, and we accept no responsibility or legal liability for their policies. We encourage you to consult the privacy policies of these third-party websites before submitting any personal information.

[Changes to Privacy Policy](#)

We reserve the right to update or modify our privacy policy at any time. privacy policy at any time. We will notify you of by posting the new privacy policy on this page. We recommend that you consult this page

Termes et Conditions

Ils constituent le contrat entre l'application et ses utilisateurs, énumérant les droits et obligations de chaque partie. Cela inclut les modalités d'utilisation, les restrictions, ainsi que les mesures disciplinaires en cas de violation des termes. Ils servent également à limiter la responsabilité de l'application en cas d'utilisation abusive par les utilisateurs.

The screenshot shows the CineMatch application interface with a dark header bar. The header includes the logo "CineMatch" with a heart icon, navigation links like "Movie Match", "My Movies", "Write a Post", "Posts", "John", "About", and "Logout", and a user profile icon. The main content area is titled "Terms and Conditions" with a lock icon. It displays the last update date as "25 september 2023". A note states: "Welcome to Cinematch (the 'Application')! Please read these Terms and Conditions of Use ('Terms') carefully before using the Application. By using Cinematch, you agree to these Terms and all policies mentioned herein. If you do not agree to these Terms, please do not use the Application." Below this, there are several sections with numbered terms:

- 1 How to use the Application
 - 1.1 You must be at least [18] years old to use the Application; the Application
 - 1.2 You agree not to use the Application for any purpose that is prohibited by law
 - 1.3 You agree not to use the Application to disseminate, store or transmit any content that is illegal, defamatory offensive, obscene, or harmful to others
- 2 User account
 - 2.1 You may be required to create a user account to access certain certain features of the Application. You are responsible for the confidentiality of your login information
 - 2.2 You agree to provide accurate and complete information when creating your user account and to keep this information up to date
 - 2.3 You are responsible for all activities that occur under your user account
- 3 Data protection
 - 3.1 We collect, store and use your data in accordance with our Privacy Policy. By using the Application, you consent to such collection and use
- 4 Customer content
 - 4.1 You are solely responsible for any content you post or share on the share on the Application. You warrant that you have the necessary rights necessary to share such content
 - 4.2 We reserve the right to remove any content that violates these Terms or is otherwise deemed inappropriate
- 5 Intellectual Property
 - 5.1 Cinematch holds all intellectual property rights related to the Application. You may not reproduce, distribute or use these elements without authorization
- 6 Application and Terms Modification
 - 6.1 We reserve the right to modify or update the Application and these the Application and these Terms at any time without notice
- 7 Cancellation
 - 7.1 We may terminate or suspend your user account at our sole discretion our sole discretion, with or without notice, for any reason, including breach of these Terms

Code de Conduite

Il s'agit d'une charte qui définit les comportements attendus des utilisateurs pour maintenir une communauté respectueuse et sécurisée. Le Code de Conduite énonce clairement ce qui est acceptable et ce qui ne l'est pas, encourageant ainsi un environnement positif et inclusif.

The screenshot shows the CineMatch application interface with a dark header bar. The header includes the logo "CineMatch" with a heart icon, navigation links like "Movie Match", "My Movies", "Write a Post", "Posts", "John", "About", and "Logout", and a user profile icon. The main content area is titled "Code of Conduct" with a heart icon. It displays the welcome message: "Welcome to CineMatch! Our community is dedicated to creating an enjoyable, respectful, and inclusive cinematic experience for all our users. We encourage you to carefully read and follow our Code of Conduct to ensure a positive experience for all." Below this, there are seven numbered rules:

- 1 Respect Diversity:
We celebrate diversity in all its forms, including race, gender, sexual orientation, religion, nationality, and culture. We expect our users to respect and treat every member of our community with respect, dignity, and understanding.
- 2 No Harassment:
We do not tolerate harassment, discrimination, or bullying. Harassment may include, but is not limited to, offensive comments, insults, threats, discrimination, and any form of harmful behavior towards other users.
- 3 Privacy Policy:
Respect the privacy of other users. Do not share sensitive personal information without their consent. Don't post content that violates the privacy of others.
- 4 Appropriate Content:
Make sure the content you share is appropriate and respectful. Avoid posting images, videos, comments, or any other content that could be considered offensive, vulgar, or inappropriate.
- 5 Help New Users:
If you're an experienced user, don't hesitate to help new users by answering their questions and guiding them through the application. Let's make sure everyone feels welcome.
- 6 Report Problems:
If you encounter behavior that violates this Code of Conduct or that you deem inappropriate, please report it to our moderation team. We will take the necessary steps to resolve the problem.
- 7 Respect the Rules of the Application:
Also observe the specific rules of the application, including the terms of use and community guidelines. Any behavior in violation of these rules may result in disciplinary action, including suspension or deletion of your account. We reserve the right to take appropriate action for violations of this Code of Conduct, including the removal of content, suspension, or deletion of user accounts at our sole discretion.

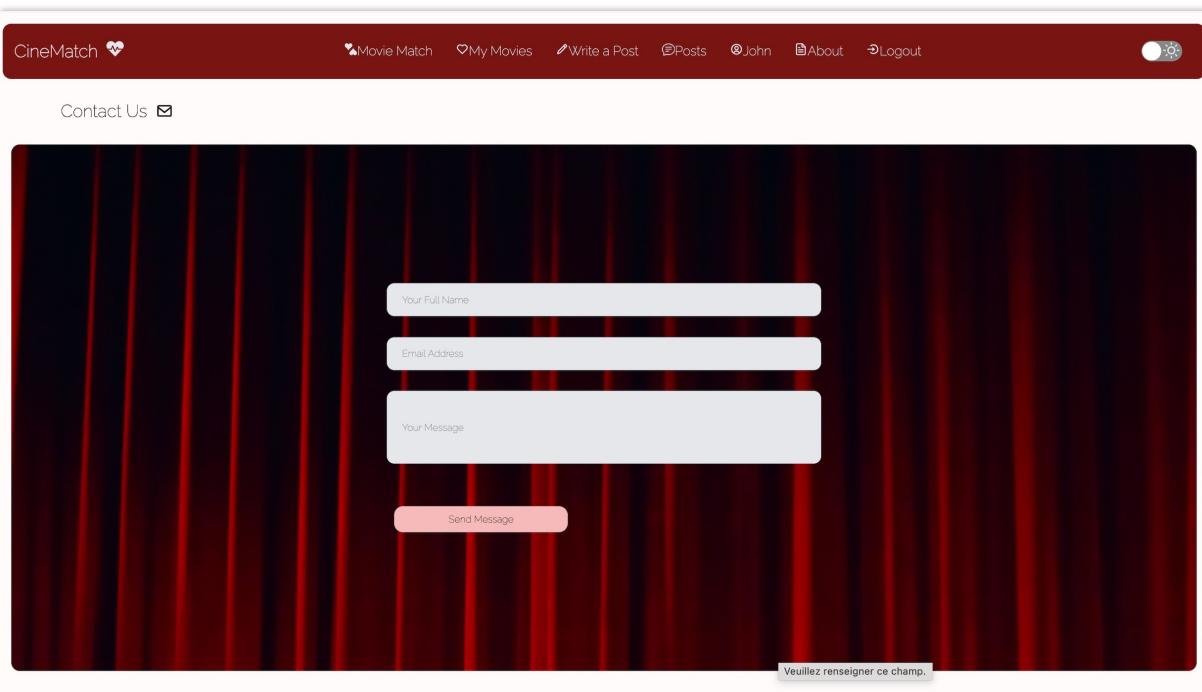
By using CineMatch, you agree to abide by this Code of Conduct and help create a positive and respectful film community.

Contact par E-mail

Cette fonction permet aux utilisateurs de communiquer directement avec les opérateurs de CineMatch pour toute question ou préoccupation concernant leur vie privée ou leur expérience utilisateur.

Un tel moyen de communication démontre l'engagement de l'application à rester accessible et réactive aux besoins de sa base d'utilisateurs. Ces éléments ne se contentent pas de répondre aux exigences légales du RGPD, mais ils renforcent également la confiance des utilisateurs envers CineMatch, en leur assurant que leurs données sont traitées avec soin et respect.

La mise en place et le maintien de ces composants reflètent l'engagement de l'application envers la protection des données et l'éthique, des aspects de plus en plus valorisés dans le paysage numérique contemporain.



The screenshot shows the 'Contact Us' page of the CineMatch application. At the top, there is a dark red header bar with the CineMatch logo on the left and navigation links for 'Movie Match', 'My Movies', 'Write a Post', 'Posts', 'John', 'About', and 'Logout' on the right. Below the header, the main content area has a black background with red vertical stripes. It contains three input fields: 'Your Full Name', 'Email Address', and 'Your Message'. A pink 'Send Message' button is located below the message input field. In the bottom right corner of the form area, there is a small text box containing the placeholder 'Veuillez renseigner ce champ.'

Présentation du jeu d'essai

J'ai procédé à l'élaboration d'un jeu d'essai pour tester les fonctionnalités clés de l'application en utilisant le logiciel **Insomnia**.

Ce logiciel, en tant qu'outil de test d'API REST, m'a permis de simuler les interactions entre le front-end et le back-end de l'application. Les étapes suivies pour la réalisation de ces tests sont détaillées ci-après.

Pour réaliser ce jeu d'essai je vais me servir du logiciel Insomnia qui permet de tester, et de documenter les requêtes vers une API.

Dans un premier temps, j'observe l'état de ma base de données avant le match de plusieurs films entre deux utilisateurs. La table sera mise à jour lors d'un match de un ou plusieurs films.

The screenshot shows the Insomnia application interface. On the left, the 'Administration' sidebar displays the 'SCHEMAS' section for 'MovieDB', showing tables like Comments, LikedMovies, Likes, Movies, Posts, and Users. The 'Tables' section is expanded, showing sub-tables for each. The main workspace shows a query editor with the SQL command 'SELECT * FROM MovieDB.Users;'. Below the query is a 'Result Grid' table with columns: id, username, email, password, createdAt, and updatedAt. The table has one row with all fields set to 'NULL'. At the bottom of the grid, it says 'Users 1'. On the right side, there are various toolbars and a sidebar with icons for Form Editor, Field Types, and Query Stats. At the bottom, there's an 'Object Info' panel for the 'LikedMovies' table, showing its columns: id (int AI PK), movieId (int), userId (int), createdAt (datetime), and updatedAt (datetime). Below that is an 'Action Output' table with two rows of data, showing the execution time and response for two SELECT queries against the 'Users' and 'LikedMovies' tables.

Time	Action	Response	Duration / Fetch Time
17:52:31	SELECT * FROM MovieDB.Users LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000028...
21:19:05	SELECT * FROM MovieDB.LikedMovies LIMIT 0, 1000	37 row(s) returned	0.011 sec / 0.000045...

Création des utilisateurs

J'ai commencé par créer trois utilisateurs fictifs : Jane, John, et Bob, en utilisant l'endpoint `/auth` de l'API. Pour chaque utilisateur, j'ai fourni des informations basiques telles que le nom, l'email et le mot de passe. Cela a permis de simuler l'inscription de nouveaux utilisateurs sur la plateforme CineMatch.

Création de l'utilisateur Jane

The screenshot shows the Insomnia REST client interface. On the left, there's a sidebar with a list of API endpoints, including `ProfileInfoBob`, `MovieLiked`, `LikeAPost`, `LoginUSER`, `LoginJanes`, `LoginJohn`, `Loginbob`, `RegisterUSER`, `RegisterJanes`, `RegisterJohn`, `Registerbob`, `RegisterUSER2`, and `RegisterUSER3`. The main panel shows a POST request to `http://localhost:3001/auth` with the following JSON payload:

```
1 {  
2   "username": "Janes",  
3   "email": "janes@gmail.com",  
4   "password": "Qwerty22"  
5 }
```

The response is a 201 Created status with a response time of 715 ms and 54 B. The response body is:

```
1 {  
2   "success": true,  
3   "message": "User created successfully"  
4 }
```

Création de l'utilisateur John

This screenshot is identical to the previous one, showing the creation of user John. The sidebar and the main request details are the same, including the endpoint `http://localhost:3001/auth` and the JSON payload for creating a user named John.

Création de l'utilisateur Bob

The screenshot shows the Insomnia REST client interface. On the left, a sidebar lists various API endpoints such as ProfileInfobob, MovieLiked, LikeAPost, LoginUSER, LoginJanes, LoginJohn, Loginbob, RegisterUSER, RegisterJanes, RegisterJohn, Registerbob, RegisterUSER2, and RegisterUSER3. The main panel shows a POST request to `http://localhost:3001/auth`. The request body is a JSON object:

```

1: {
2:   "username": "bob",
3:   "email": "bob@gmail.com",
4:   "password": "Qwerty22="
5: }

```

The response status is **201 Created**, with a response time of 99.8 ms and a response size of 54 B. The response body is:

```

1: {
2:   "success": true,
3:   "message": "User created successfully"
4: }

```

Login des utilisateurs

Après la création des comptes, j'ai procédé au login de chaque utilisateur via l'endpoint `'/auth/login'`. Cette étape était cruciale pour authentifier les utilisateurs et obtenir les tokens d'accès nécessaires pour les requêtes authentifiées ultérieures.

Connexion de Jane

The screenshot shows the Insomnia REST client interface. The sidebar lists the same set of API endpoints as the previous screenshot. The main panel shows a POST request to `http://localhost:3001/auth/login`. The request body is a JSON object:

```

1: {
2:   "username": "Janes",
3:   "password": "Qwerty22="
4: }

```

The response status is **200 OK**, with a response time of 175 ms and a response size of 207 B. The response body is:

```

1: {
2:   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJc2VybmtSISI6IkphbmVzIiwiaWQ1OjEsImlhCIGMTxMDYzM0N8Sw1ZXhvIjoxNzEwNzE2NDQ1fQ.ØSy7c1DEicVqRdu5amAXRP1PlxRxZLmgVopBTSvLejA",
3:   "username": "Janes",
4:   "id": 1
5: }

```

Connexion de John

The screenshot shows the MovieDB API documentation in Postman. A POST request is made to `http://localhost:3001/auth/login` with the following JSON parameters:

```
1: {  
2:   "username": "John",  
3:   "password": "Qwerty22"  
4:  
5: }
```

The response is a **200 OK** status with a response time of 100 ms and a body size of 204 B. The response header includes `Content-Type: application/json`. The response body is:

```
1: {  
2:   "accessToken": "",  
3:   "username": "John",  
4:   "id": 2  
5: }
```

The left sidebar lists various API endpoints such as ProfileInfo, MovieLiked, LikeAPost, LoginUSER, LoginJanes, LoginJohn, LoginBob, RegisterUSER, RegisterJanes, RegisterJohn, RegisterBob, RegisterUSER2, and RegisterUSER3.

Connexion de Bob

The screenshot shows the MovieDB API documentation in Postman. A POST request is made to `http://localhost:3001/auth/login` with the following JSON parameters:

```
1: {  
2:   "username": "bob",  
3:   "password": "Qwerty22"  
4:  
5: }
```

The response is a **200 OK** status with a response time of 105 ms and a body size of 202 B. The response header includes `Content-Type: application/json`. The response body is:

```
1: {  
2:   "accessToken": "",  
3:   "username": "bob",  
4:   "id": 3  
5: }
```

The left sidebar lists various API endpoints such as ProfileInfo, MovieLiked, LikeAPost, LoginUSER, LoginJanes, LoginJohn, LoginBob, RegisterUSER, RegisterJanes, RegisterJohn, RegisterBob, RegisterUSER2, and RegisterUSER3.

Vérification des informations de profil

Avec les tokens d'accès obtenus, j'ai ensuite fait des requêtes à l'endpoint `/profile/\${userId}` pour chaque utilisateur. Cela m'a permis de vérifier que les informations de profil étaient correctement enregistrées et accessibles après l'authentification.

Vérifications des informations de Jane

The screenshot shows the Insomnia REST client interface. On the left, a sidebar lists various API endpoints. In the center, a request is being made to `http://localhost:3001/auth/profileinfo/1`. The response status is **200 OK**, with a response time of 215 ms and a body size of 131 B. The response was received 3 Hours Ago. The response content is a JSON object:

```
1 {  
2   "id": 1,  
3   "username": "Janes",  
4   "email": "janes@gmail.com",  
5   "createdAt": "2024-03-15T16:51:21.000Z",  
6   "updatedAt": "2024-03-15T16:51:21.000Z"  
7 }
```

The right panel shows the Headers, Cookies, and Timeline sections. At the bottom, there are buttons for Online and Made with ❤ by Kong.

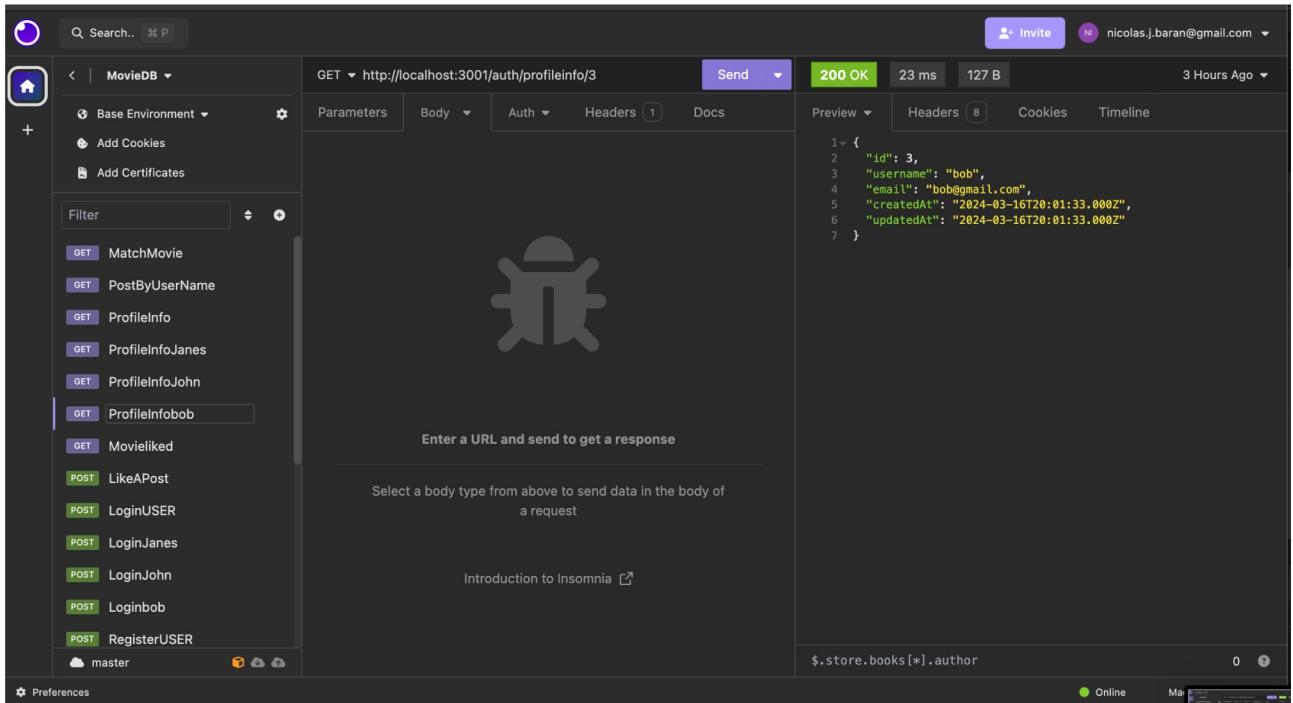
Vérifications des informations de John

This screenshot is identical to the previous one, showing the Insomnia REST client interface. It displays a successful GET request to `http://localhost:3001/auth/profileinfo/2` for user John. The response status is **200 OK**, with a response time of 28.5 ms and a body size of 129 B. The response was received 3 Hours Ago. The response content is a JSON object:

```
1 {  
2   "id": 2,  
3   "username": "John",  
4   "email": "john@gmail.com",  
5   "createdAt": "2024-03-16T19:59:41.000Z",  
6   "updatedAt": "2024-03-16T19:59:41.000Z"  
7 }
```

The right panel shows the Headers, Cookies, and Timeline sections. At the bottom, there are buttons for Online and Made with ❤ by Kong.

Vérifications des informations de Bob



The screenshot shows the Insomnia REST client interface. On the left, there's a sidebar with various API endpoints listed under the 'MovieDB' project. The main area shows a successful GET request to `http://localhost:3001/auth/profileinfo/3`. The response status is **200 OK**, with a response time of 23 ms and a body size of 127 B. The response was received 3 Hours Ago. The response body is a JSON object:

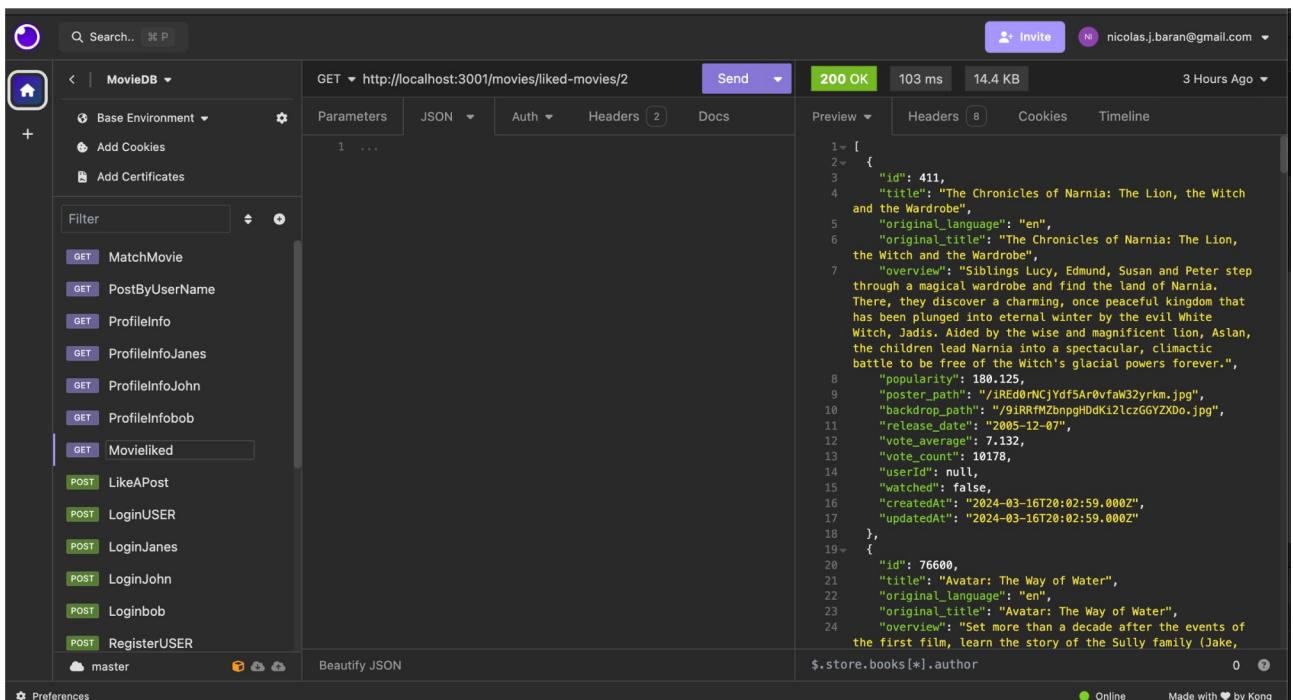
```

1- {
2-   "id": 3,
3-   "username": "bob",
4-   "email": "bob@gmail.com",
5-   "createdAt": "2024-03-16T20:01:33.000Z",
6-   "updatedAt": "2024-03-16T20:01:33.000Z"
7- }

```

Like des films

En utilisant l'endpoint `/movies/liked-movies/${userId}`, j'ai simulé le processus de "swiping à droite" pour aimer différents films pour chacun des utilisateurs. Cette action reflète l'une des fonctionnalités clés de CineMatch, où les utilisateurs expriment leur intérêt pour des films en les ajoutant à leur liste de favoris.



The screenshot shows the Insomnia REST client interface. On the left, there's a sidebar with various API endpoints listed under the 'MovieDB' project. The main area shows a successful GET request to `http://localhost:3001/movies/liked-movies/2`. The response status is **200 OK**, with a response time of 103 ms and a body size of 14.4 KB. The response was received 3 Hours Ago. The response body is a JSON array:

```

1- [
2-   {
3-     "id": 411,
4-     "title": "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
5-     "original_language": "en",
6-     "original_title": "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
7-     "overview": "Siblings Lucy, Edmund, Susan and Peter step through a magical wardrobe and find the land of Narnia. There, they discover a charming, once peaceful kingdom that has been plunged into eternal winter by the evil White Witch, Jadis. Aided by the wise and magnificent lion, Aslan, the children lead Narnia into a spectacular, climactic battle to be free of the Witch's glacial powers forever.",
8-     "popularity": 180.125,
9-     "poster_path": "/1REd6rNCjYdf5Ar0vfaW32yrkm.jpg",
10-    "backdrop_path": "/9iRRM2bnpgHDDk12lczGGYZXDo.jpg",
11-    "release_date": "2005-12-07",
12-    "vote_average": 7.132,
13-    "vote_count": 10178,
14-    "userId": null,
15-    "watched": false,
16-    "createdAt": "2024-03-16T20:02:59.000Z",
17-    "updatedAt": "2024-03-16T20:02:59.000Z"
18-   },
19-   {
20-     "id": 76600,
21-     "title": "Avatar: The Way of Water",
22-     "original_language": "en",
23-     "original_title": "Avatar: The Way of Water",
24-     "overview": "Set more than a decade after the events of the first film, learn the story of the Sully family (Jake, "

```

Vérification de match entre utilisateurs

Enfin, j'ai utilisé l'endpoint `/movies/match-liked-movies` pour vérifier les matchs de films entre les utilisateurs. Cet endpoint m'a permis de détecter quels films étaient aimés par plusieurs utilisateurs, illustrant ainsi le potentiel de CineMatch à créer des connexions entre les utilisateurs à travers leurs goûts cinématographiques communs.

The screenshot shows the Insomnia API client interface. On the left, a sidebar lists various API endpoints under the 'MovieDB' environment. The main area displays a successful GET request to 'http://localhost:3001/movies/match-liked-movies'. The response body is a JSON object containing two movie records. The first movie is 'The Zone of Interest' (id: 467244) and the second is 'Transformers: Rise of the Beasts' (id: 667538). The status bar at the bottom indicates the response was received 306 ms ago, is 3.4 KB in size, and was made 3 hours ago.

```
1 {  
2   "success": true,  
3   "matchedMovies": [  
4     {  
5       "id": 467244,  
6       "title": "The Zone of Interest",  
7       "original_language": "en",  
8       "original_title": "The Zone of Interest",  
9       "overview": "The commandant of Auschwitz, Rudolf Höss, and his wife Hedwig, strive to build a dream life for their family in a house and garden next to the camp.",  
10      "popularity": 207.711,  
11      "poster_path": "/hu9uzyZmdd8VZegKi1iK1VkJRYS.jpg",  
12      "backdrop_path": "/aP0sJ3yLDUOhJYnSqkhKrkJTAw.jpg",  
13      "release_date": "2023-12-15",  
14      "vote_average": 7.31,  
15      "vote_count": 820,  
16      "userId": null,  
17      "watched": false,  
18      "createdAt": "2024-03-16T20:02:17.000Z",  
19      "updatedAt": "2024-03-16T20:02:17.000Z",  
20    },  
21    {  
22      "id": 667538,  
23      "title": "Transformers: Rise of the Beasts",  
24      "original_language": "en",  
25      "original_title": "Transformers: Rise of the Beasts",  
26      "overview": "When a new threat capable of destroying  
the entire planet emerges, Optimus Prime and the Autobots",  
27    }  
28  ],  
29  "users": [  
30    {  
31      "id": 3,  
32      "name": "Nicolas J. Baran",  
33      "email": "nicolas.j.baran@gmail.com",  
34      "password": "password123",  
35      "role": "admin",  
36      "createdAt": "2024-03-16T20:02:17.000Z",  
37      "updatedAt": "2024-03-16T20:02:17.000Z",  
38    },  
39    {  
40      "id": 2,  
41      "name": "John Doe",  
42      "email": "john.doe@example.com",  
43      "password": "password123",  
44      "role": "user",  
45      "createdAt": "2024-03-16T20:02:17.000Z",  
46      "updatedAt": "2024-03-16T20:02:17.000Z",  
47    }  
48  ]  
49}  
$.store.books[*].author
```

The screenshot shows the Insomnia API testing tool interface. The left sidebar lists various API endpoints for a "MovieDB" environment, including GET requests for MatchMovie, PostByUserName, ProfileInfo, ProfileInfoJanes, ProfileInfoJohn, ProfileInfoBob, Movieliked, LikeAPost, LoginUSER, LoginJanes, LoginJohn, LoginBob, and RegisterUSER. The main workspace displays a successful GET request to "http://localhost:3001/movies/match-liked-movies" with a status of 200 OK, response time of 306 ms, and size of 3.4 KB. The response body is a JSON object representing a movie, with the "id" field set to 1011985 and the title "Kung Fu Panda 4". The response also includes a detailed overview of the movie's plot and cast information.

Q Search.. % P

Invite nicolas.j.baran@gmail.com

MovieDB

Base Environment Add Cookies Add Certificates

Filter

GET MatchMovie

GET PostByUserName

GET ProfileInfo

GET ProfileInfoJanes

GET ProfileInfoJohn

GET ProfileInfoBob

GET Movieliked

POST LikeAPost

POST LoginUSER

POST LoginJanes

POST LoginJohn

POST LoginBob

POST RegisterUSER

master

Parameters Body Headers 1 Docs

Send 200 OK 306 ms 3.4 KB 3 Hours Ago

Preview Headers 8 Cookies Timeline

```
86 ]
87 },
88{
89   "id": 1011985,
90   "title": "Kung Fu Panda 4",
91   "original_language": "en",
92   "original_title": "Kung Fu Panda 4",
93   "overview": "Po is gearing up to become the spiritual leader of his Valley of Peace, but also needs someone to take his place as Dragon Warrior. As such, he will train a new kung fu practitioner for the spot and will encounter a villain called the Chameleon who conjures villains from the past.",
94   "popularity": 1728.91,
95   "poster_path": "/kbDp1vBnMpe8ak4rjg13CLELqjU.jpg",
96   "backdrop_path": "/gjL5kp5MopB2sN4WZYnNT5u0u.jpg",
97   "release_date": "2024-03-02",
98   "vote_average": 6.976,
99   "vote_count": 105,
100   "userId": null,
101   "watched": false,
102   "createdAt": "2024-03-15T16:53:04.000Z",
103   "updatedAt": "2024-03-15T16:53:04.000Z",
104{
105   "users": [
106     1,
107     3
108   ]
109 }
110 }
```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia

\$.store.books[*.author

Ces tests, effectués à l'aide d'Insomnia, ont non seulement démontré la fonctionnalité et la fiabilité de l'API de CineMatch, mais ont également souligné l'importance de la création d'un jeu d'essai complet pour valider chaque aspect de l'application avant son déploiement. Ces étapes de validation s'avèrent essentielles pour garantir une expérience utilisateur fluide et sans faille.

Description de la veille sur les vulnérabilités de sécurité

Concernant la veille sur les vulnérabilités de sécurité de l'application, cette dernière s'est notamment concentrée sur le cryptage du mot de passe utilisateur via bcrypt et sur l'ORM Sequelize.

Cryptage du mot de passe utilisateur et JWT

Pour établir le système de connexion des utilisateurs, j'ai opté pour l'utilisation de Json Web Tokens (JWT). Parallèlement, je me suis interrogé sur la sécurisation du stockage des mots de passe des utilisateurs dans la base de données.

Après des recherches sur les méthodes de stockage des mots de passe, j'ai choisi de les hasher en utilisant la bibliothèque **bcrypt**, comme préconisé dans les ressources suivantes :

- <https://blog.logrocket.com/password-hashing-node-js-bcrypt/>
- https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#hashing-vs-encryption

J'ai ensuite effectué des recherches sur les bonnes pratiques à adopter en matière de sécurisation des JWT :

- <https://curity.io/resources/learn/jwt-best-practices/>
- https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html#jwt
- <https://stackoverflow.com/questions/27301557/if-you-can-decode-jwt-how-are-they-secure>

Enfin, la ressource suivante a permis d'éclairer mes interrogations sur la transmission des JWT entre le front-end et le back-end :

- <https://levelup.gitconnected.com/bearer-token-authentication-and-authorization-6e4d16890833>

ORM Sequelize

Pour permettre la prise en charge des systèmes de gestion de base de données (SGBD), j'ai utilisé un ORM (Object-Relational-Mapping) : Sequelize. L'utilisation de Sequelize comme ORM offre plusieurs avantages en termes de sécurité pour les applications web : prévention des injections SQL, validation des entrées, gestion sécurisée des transactions et migrations sécurisées.

Pour mettre en place l'ORM Sequelize je me suis notamment appuyé sur la documentation officielle :

- <https://sequelize.org/docs/v6/>
- <https://www.npmjs.com/package/sequelize>

Description d'une situation de travail ayant nécessité une recherche

Concernant la description d'une situation de travail ayant nécessité une recherche, j'ai choisi de revenir plus en détails sur la **fonctionnalité dark mode** que j'ai cherché à mettre en place pour la première fois dans le cadre de ce projet.

Pour commencer, j'ai effectué une recherche sur internet avec les mots clés suivants : « dark mode React ». Globalement pour effectuer mes recherches, j'effectue mes requêtes en anglais plutôt qu'en français avec des termes simples et évalue la pertinence des résultats notamment en fonction des dates de mise en ligne/mise à jour, ou encore de la réputation des sites.

Le résultat a été probant pour cette recherche, puisque j'ai trouvé plusieurs solutions sur les sites suivants :

- <https://blog.logrocket.com/dark-mode-react-in-depth-guide/>
- <https://www.npmjs.com/package/react-switch>
- <https://dev.to/abbeyperini/toggle-dark-mode-in-react-28c9>

Pour la mise en œuvre du dark mode, plusieurs étapes ont été nécessaires :

- **Création d'un composant ThemeProvider.jsx**

Le ThemeProvider est un composant qui met en œuvre une gestion globale du thème via l'API Context de React, permettant ainsi aux utilisateurs de personnaliser l'interface en alternant entre un mode clair et sombre.

Il encapsule l'état du thème actuel et offre une fonction toggleTheme pour basculer entre les modes, rendant cette fonctionnalité accessible à l'ensemble de l'arborescence des composants sans propager manuellement les props. Ce composant augmente l'expérience utilisateur en facilitant l'adaptabilité visuelle de l'application selon les préférences de l'utilisateur, tout en préservant une cohérence de style à travers les différents éléments de l'interface.

```
// ThemeProvider.jsx
import React, { createContext, useState } from "react";

export const ThemeContext = createContext(null);

export const ThemeProvider = ({ children }) => {
  const [currentTheme, setCurrentTheme] = useState("light");

  const toggleTheme = () => {
    setCurrentTheme((prevTheme) => (prevTheme === "light" ? "dark" : "light"));
  };

  return (
    <ThemeContext.Provider value={{ theme: currentTheme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
};
```

- **Importation du composant ThemeProvider**

Dans mon fichier App.jsx, j'ai importé mon composant ThemeProvider afin de permettre à toute l'application d'accéder et de manipuler le thème dark mode. En insérant au plus haut niveau de l'application, il assure que chaque composant enfant a la capacité de réagir dynamiquement au changement de thème.

```

    return (
      <ThemeProvider>
        <div className="app-background">
          <BrowserRouter>
            <AuthContext.Provider value={{ authState, setAuthState }}>
              <Routes>
                {routes.map((route, index) => (
                  <Route key={index} path={route.path} element={route.element} />
                ))}
              </Routes>
            </AuthContext.Provider>
          </BrowserRouter>
        </div>
      </ThemeProvider>
    );
  }

```

- **Mise en œuvre du dark mode dans les différents composants/pages**

La logique suivante a été appliquée à l'ensemble des composants/pages de l'application :

- ✓ Modification de l'état initial du thème défini sur Light en Dark en utilisant la fonction `toggleTheme`
- ✓ Intégration d'un commutateur (ReactSwitch) dans le composant Navbar.jsx

```

import ReactSwitch from "react-switch";
import { ThemeContext } from "./ThemeProvider";
import { FiSun } from "react-icons/fi";
import { FaMoon } from "react-icons/fa";
import { LuFileText } from "react-icons/lu";
import { GiOppositeHearts } from "react-icons/gi";

export default function Navbar() {
  const navigate = useNavigate();
  const { authState, setAuthState } = useContext(AuthContext)
  const [isMenuActive, setIsMenuActive] = useState(false);
  const { toggleTheme, theme } = useContext(ThemeContext);

  const handleNavigate = (path) => {
    navigate(path);
    setIsMenuActive(false);
  };
}

```

```

<ReactSwitch
  className="reactswitch"
  onChange={toggleTheme}
  checked={theme === "dark"}
  checkedIcon={
    <div className="switch-icon">
      <FaMoon />
    </div>
  }
  uncheckedIcon={
    <div className="switch-icon">
      <FiSun className="sun" />
    </div>
  }
/>

```

- ✓ Chaque élément de l'interface utilisateur qui doit réagir au changement de thème reçoit « `id={theme}` ». Cette propriété est utilisée pour assigner dynamiquement un identifiant CSS basé sur le thème actuel light ou dark.

```

<>
<Navbar />
<div className="background-home" id={theme}>
  <h2 className="header" id={theme}>
    <WelcomeUsers />
  </h2>
  <div className="home-screen" id={theme}>
    <div className="movie-home">{renderMovies()}</div>
    <div className="swipe-buttons-home">
      <button
        className="wrong-button-home"
        onClick={handleSwipeLeft}
        id={theme}
      >
        <ImCross className="wrong-button-home-icon" />
      </button>
    </div>
  </div>
</div>

```

```

#light .background-home{
  background-color: var(--background);
}
#dark .background-home{
  background-color: var(--background-dark);
}

#dark .home-screen{
  background-color: var(--background-dark);
}
#light .wrong-button-home{
  background-color: var(--primary);
}
#light .wrong-button-home:hover{
  background-color: var(--primary);
}

```

Intégration et déploiement continus

L'intégration et le déploiement continu (CI/CD) visent à automatiser le processus de construction et de déploiement des applications. Le but est de détecter rapidement les erreurs de code et d'incompatibilités entre les parties de l'application afin de les résoudre rapidement et de réduire les risques.

Ces tâches sont exécutées automatiquement chaque fois que l'on pousse du code sur Github.

Pour la partie front-end, j'ai utilisé **Vercel**, une plateforme de cloud qui se distingue par sa facilité de déploiement et ses performances optimales en termes de chargement de site. Elle est particulièrement adaptée pour des projets utilisant des frameworks comme React.

The screenshot shows the Vercel dashboard interface. At the top, there's a navigation bar with links for Overview, Integrations, Activity, Domains, Usage, Monitoring, Storage, AI, and Settings. Below the navigation is a search bar labeled "Search Repositories and Projects..." and a button to "Add New...". The main area is divided into two columns: "Recent Previews" and "Projects". In the "Recent Previews" column, there's a card for "portfolio-ncls-brn" which has a preview link, source link, and a "portfolio" tag. In the "Projects" column, there are cards for "portfolio-ncls-brn", "movie-app", "netflix-app", "meals-app", "travel-app", and "travel". Each card displays the project name, URL, last deployment status, and time. For example, "portfolio-ncls-brn" shows "No Production Deployment 1h ago". The URL listed is <https://vercel.com/ncls-brn/portfolio-ncls-brn>.

Pour la partie back-end, j'ai utilisé Heroku, une plateforme reconnue pour sa simplicité d'utilisation. Heroku offre un environnement d'exécution parfait pour des applications Node.js comme le back-end de CineMatch, permettant une gestion fluide des dépendances et une intégration continue depuis Github.

The screenshot shows the Heroku dashboard for the "cinematch" application. At the top, there's a header with the GitHub icon, the app name "cinematch", and a "main" branch indicator. Below the header are tabs for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. The "Metrics (last 24hrs)" section shows three charts: Response Time (91ms), Throughput (< 1 rps), and Memory (14%). The "Installed add-ons" section lists "ClearDB MySQL" and "Punch MySQL 8 cleardb-trapezoidal-55520". The "Dyno formation" section shows a cost of "\$0.010/hour". The "Latest activity" section lists recent deployments by "nicolas.baran@proton.me" with timestamps and links to compare diff and view build log. The bottom of the dashboard indicates "This app is using basic dynos".

Evolutions potentielles

Les opportunités de continuer à faire évoluer CineMatch pour enrichir davantage l'expérience des utilisateurs sont nombreuses. Parmi les axes à explorer, plusieurs pistes se dégagent :

- Développement de la fonctionnalité de **confirmation d'inscription** (mail, capcha)
- Développement de la fonctionnalité **mot de passe oublié**
- **Amélioration de l'algorithme de match** entre utilisateurs (création de sous-groupes d'utilisateurs notamment)

