DOSSIER TECHNIQUE

File rouge - The Movie DB
Front End application de CoMick SARL

Réalisé par Mickael et Corentin

Sommaire

Introduction..... 3
Travail de chaque membre de l'équipe..... 3
Gestion de projet.....3
Outils et technologies utilisées..... 4
Charte graphique..... 4
User Stories.....5
Arborescence du projet.....5
Plan de l'application Web.....6
Maquette et rendu avec description.....6
Jeux d'essais.....11
- **Corentin : movie.html**.....11
- **Mickael : index.html**..... 11

GitHub..... 11

Introduction

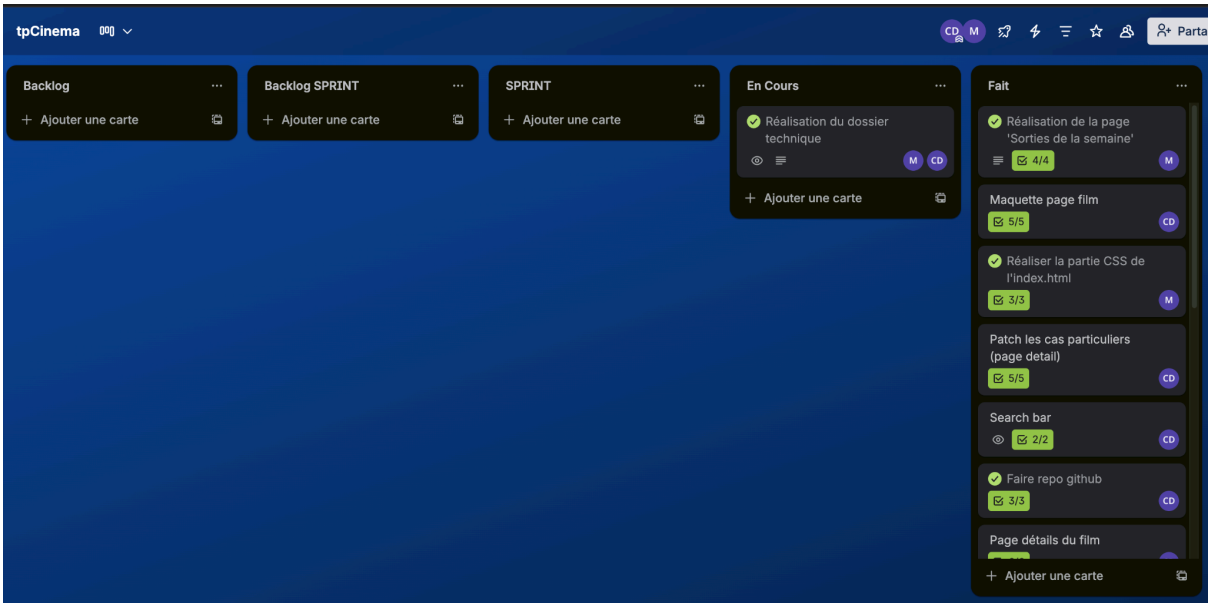
L'objectif de ce fil rouge était de créer une application web dynamique référençant des films en faisant appel aux données de l'api TMDB. L'application devait être responsible c'est-à-dire présenter une expérience utilisateur équivalente sur toutes les tailles d'écran. Nous aurons dans cette application au moins deux pages différentes, la première étant la page d'accueil sur laquelle on aurait entre autres la liste des films à l'affiche cette semaine. La seconde page présenterait les détails du film sur lequel on aurait cliqué. Notre application doit autant que possible gérer les erreurs qui pourraient résulter de données incomplètes dans l'api ou d'une mauvaise utilisation.

Travail de chaque membre de l'équipe

Le fil rouge étant découpé en différents TP (Travaux pratique), nous avons décidé de nous les répartir au sein de l'équipe, Mickael fut chargé des TP 1 et 4 (Page d'accueil) et Corentin fut chargé des TP 2 et 3 (Page de détails des films).

Gestion de projet

Afin de gérer au mieux le projet nous avons fait un Trello en utilisant la méthode agile SCRUM. Nous avons dans un premier temps lister toutes les tâches que nous avions à faire pour répondre à l'ensemble des user stories du fil rouge que nous avons mis dans le BACKLOG puis à chaque Sprint nous réalisons un ensemble de fonctionnalités qu'il nous semble cohérent de développer. Voici à quoi ressemblait notre trello a la fin de notre projet:



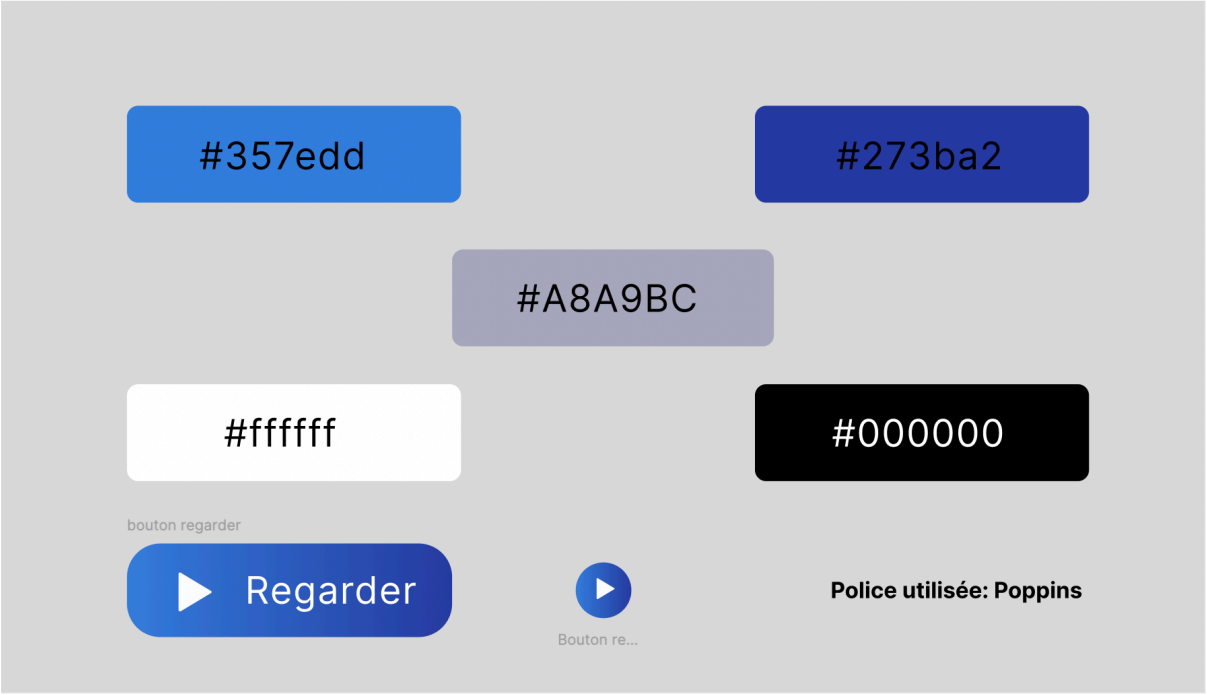
Capture d'écran de notre Trello en fin de projet

Nous avons également convenu d'un workflow: Chaque matin nous mergeons à tour de rôle notre travail avec la branche develop du repo github ce qui nous permet d'actualiser notre code et de repartir avec la version fonctionnelle la plus récente du code. Nous profitons également de ce moment pour faire un point et définir le prochain sprint sur lequel nous allons travailler pendant la journée.

Outils et technologies utilisées

- Trello
- Bootstrap
- HTML
- CSS
- JavaScript
- SplideJS
- API

Charte graphique



Chartres graphique utilisée pour notre projet

User Stories

1 - Voir les films à l'affiche

En tant qu'utilisateur je veux pouvoir voir les films à l'affiche via le carrousel des affiches des films de la semaine (titre, date de sortie, note) et faire défiler le carrousel de la droite vers la gauche et vice-versa.

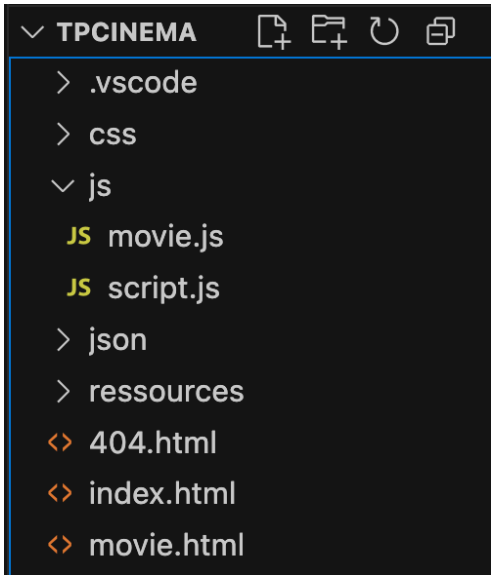
2 - Consulter les détails d'un films

En tant qu'utilisateur je veux pouvoir consulter les détails d'un film (Titre, Genre, Date de sortie, Note, Duree, Réalisateur, Acteur, Synopsis).

Exception: le film n'est pas référencé dans l'api TMDB

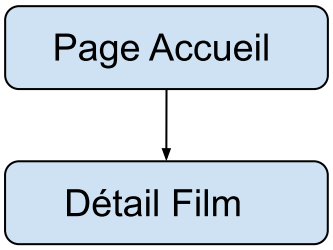
Arborescence du projet

Nous avons opté pour une arborescence classique pour ce projet à savoir toutes les pages html à la racine, le css dans un dossier css et le js dans un dossier js. Nous avons séparé la logique de chaque page html dans un fichier javaScript pour limiter les conflits lors des "merge" et éviter certaines erreurs qui avaient lieu au chargement des pages. Cette logique à été appliquée pour le CSS.



Screenshot arborescence projet

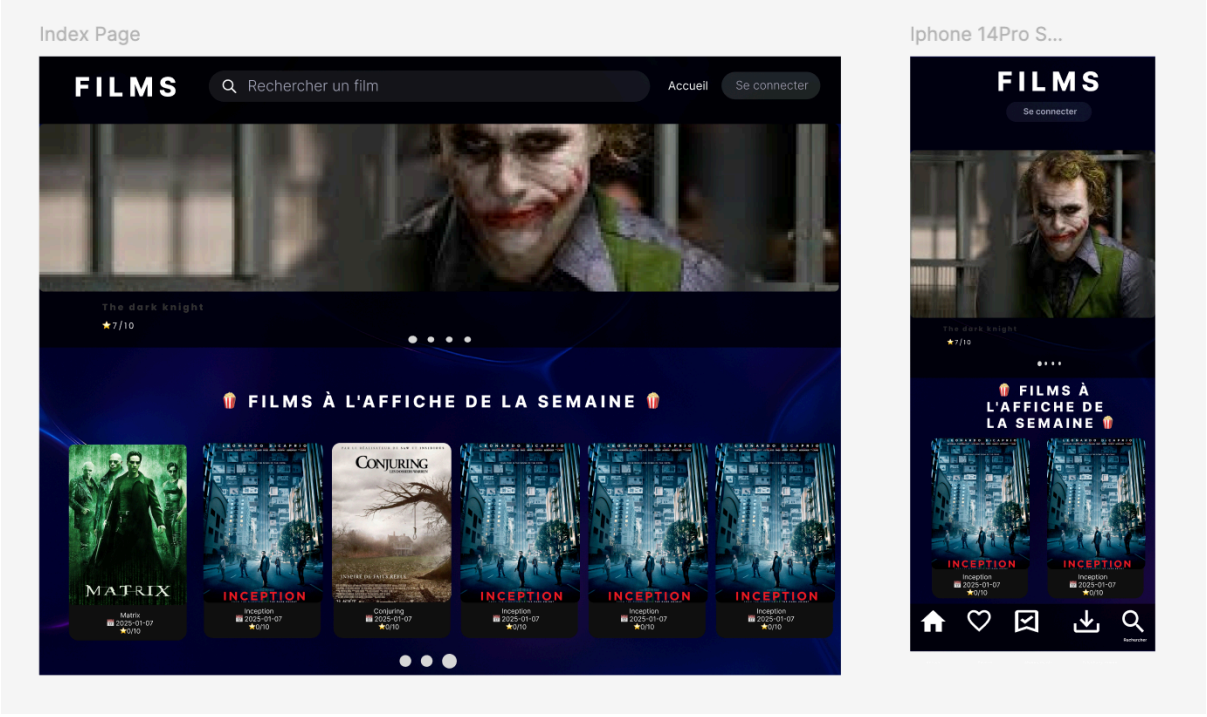
Plan de l’application Web



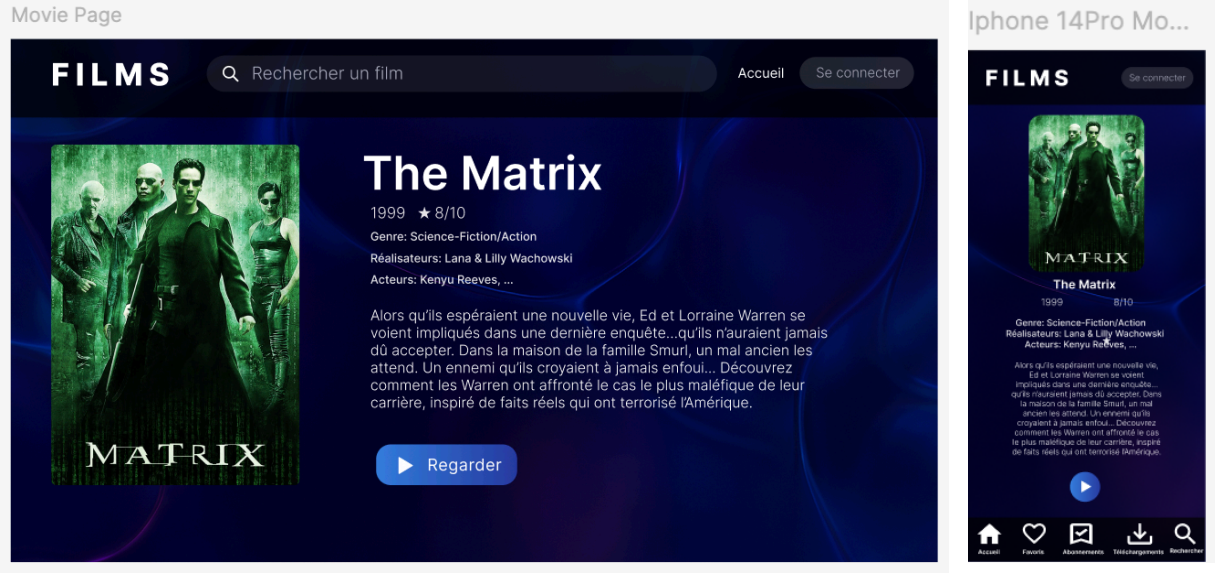
Schema de navigation de notre application web

Maquette et rendu avec description

Avant de nous lancer dans le code brut nous avons travaillé sur une maquette afin de visualiser davantage comment faire notre application web et nous assurer d’avoir une cohérence graphique entre les pages. Nous avons réalisé cette maquette sur Figma, elle comprend les deux pages: accueil et détails de film. Pour chacune nous avons fait une version Desktop et une version Mobile.

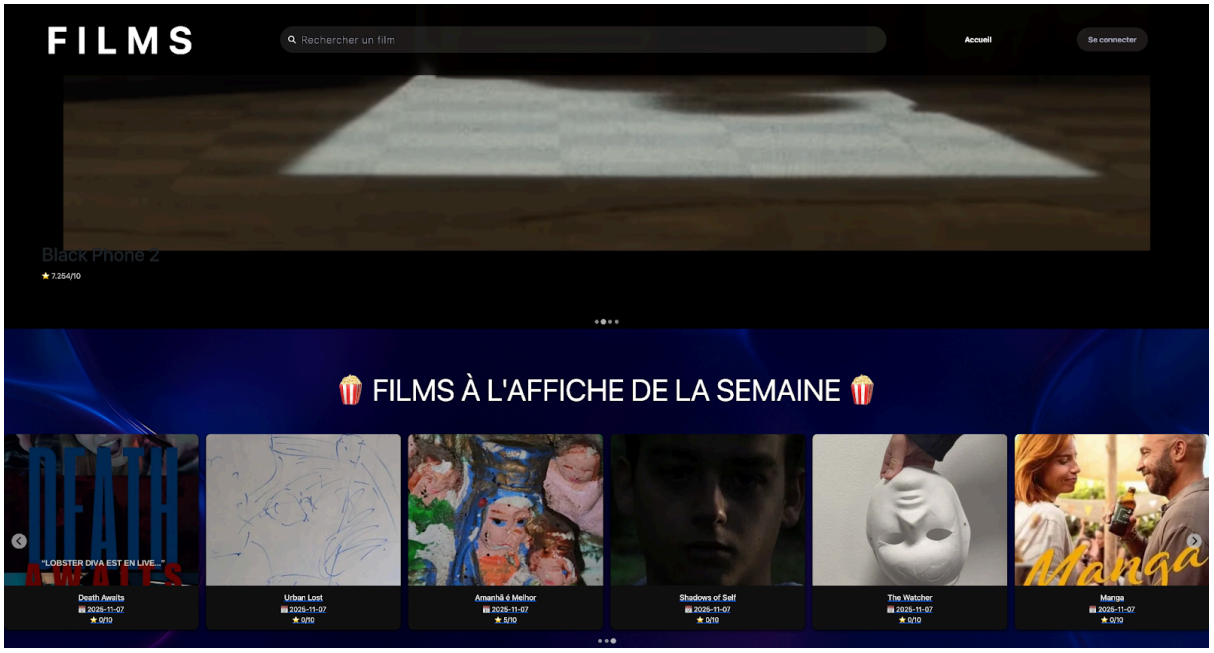


Maquettes de la page d’accueil (index.html) : Desktop et mobile ↑

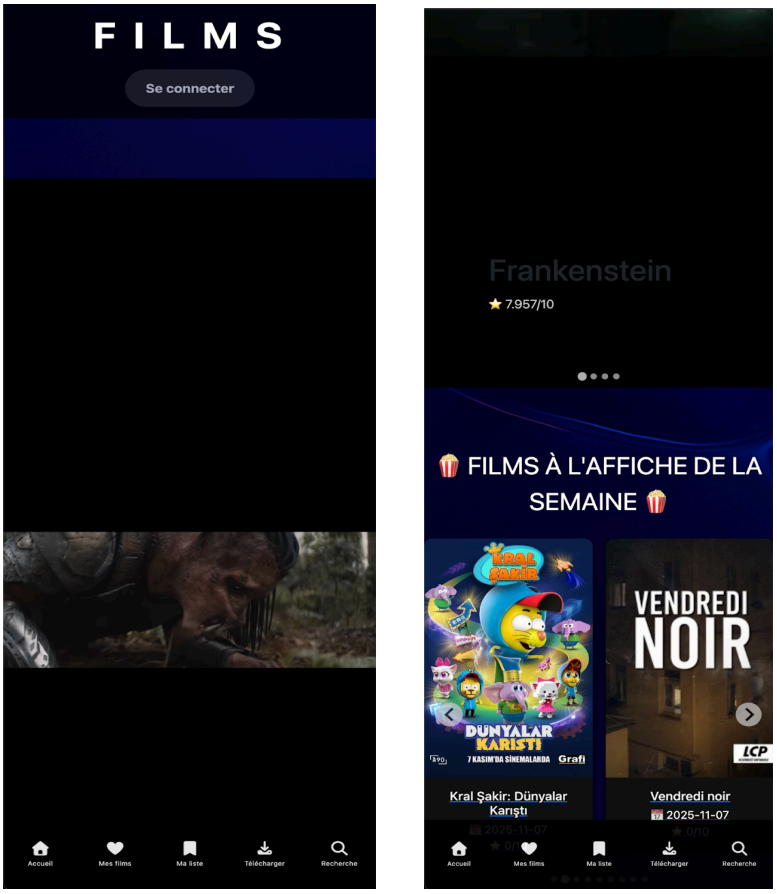


Maquettes de la page détail film: Desktop et mobile ↑

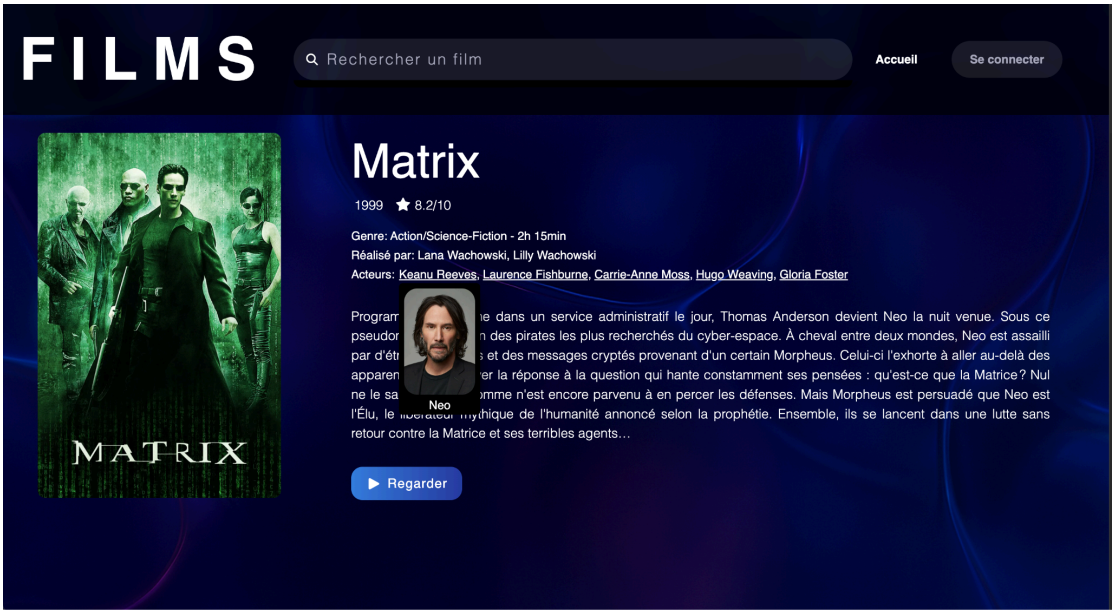
Nous avons ensuite réalisé ces pages le plus fidèlement possible comme ci-dessous:



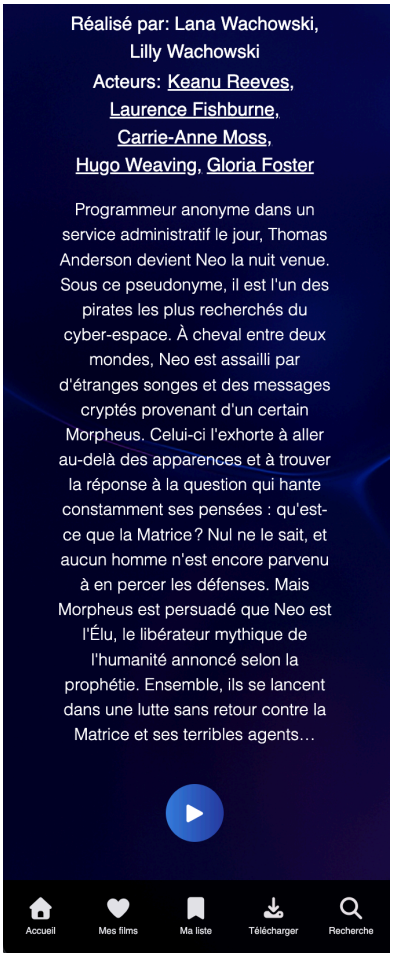
Capture de la page d'accueil (index.html) Desktop



Capture de la page d'accueil (index.html) Mobile

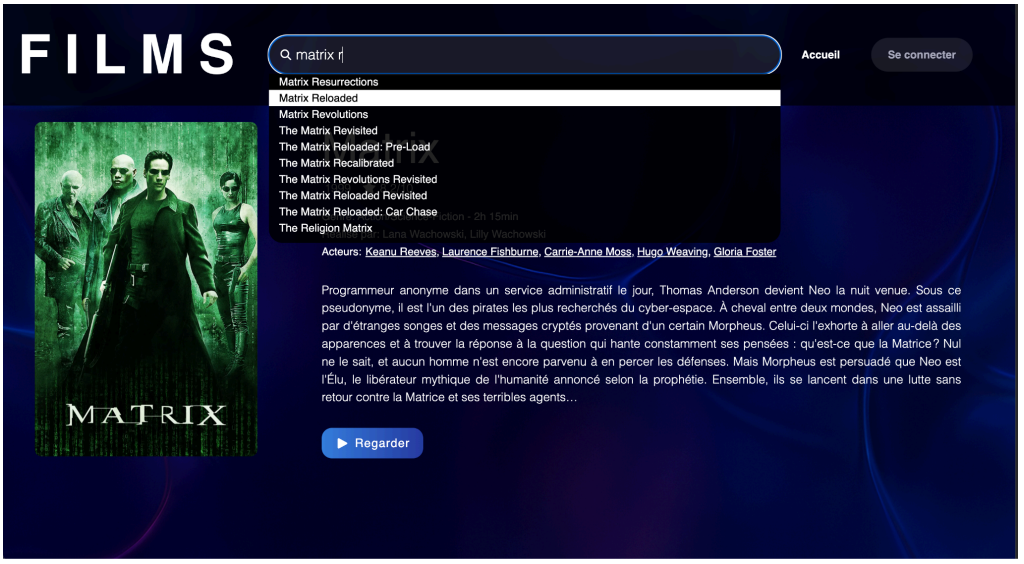


Capture de la page Détail Film Desktop



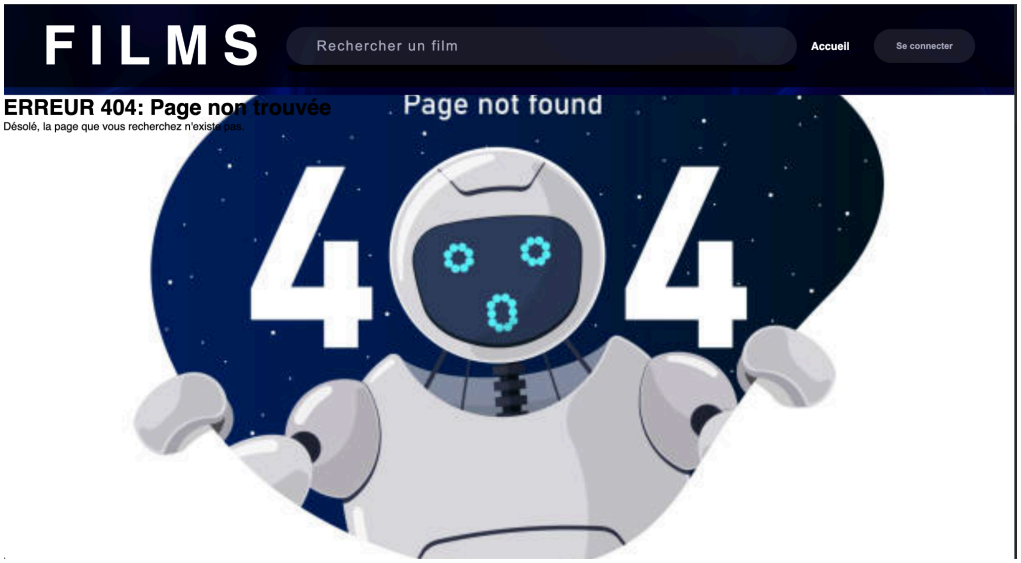
Capture de la page détail film en version mobile

Nous avons également développé des fonctionnalités additionnel comme la barre de recherche, voir ci dessous:



Capture d'écran de la barre de recherche

De la même manière si l’url contient un id invalide nous avons une page qui traite cette exception comme ci-dessous.



Capture d'écran erreur 404

Jeux d’essais

Corentin : movie.html

Cas	url
L’utilisateur clique sur le film matrix	https://c-desevaux.github.io/tpCinema/movie.html?id=603
La page d’accueil envoie un id qui n’existe pas dans la l’api	https://c-desevaux.github.io/tpCinema/movie.html?id=99999999
L’utilisateur écrit à la main un id qui n’existe pas dans l’url	https://c-desevaux.github.io/tpCinema/movie.html?id=qwerty

Mickael : index.html

Cas	url
L'utilisateur charge la page d’accueil (index.html). les carrousels se chargent normalement	https://c-desevaux.github.io/tpCinema/index.html
L'utilisateur clique sur l'affiche du film dans le carrousel des affiches, il est redirigé vers la page détails de ce film (movie.html)	https://c-desevaux.github.io/tpCinema/movie.html?id=???

GitHub

https://c-desevaux.github.io/tpCinema