

# Journal de développeur :

## Lundi 6 mars :

Nous avons commencé le projet dans l'après-midi après la présentation d'Ethan vers 14h15. J'ai par la suite commencé à chercher quelle API j'allais utiliser parmi les choix proposés de la consigne. J'ai tout d'abord pensé à l'API de Spotify, mais après plusieurs investigations, celle-ci s'avérait trop compliquée avec une clé changeante toutes les heures. J'ai donc cherché et j'ai eu deux choix l'API de recherche d'iTunes ou bien l'API de la Nasa. J'ai finalement tranché pour l'API d'iTunes qui a été validé par Ethan vers les 15h30. J'ai ensuite dû mettre fin à ma journée pour des raisons techniques car mon ordinateur crashait toutes les 10 minutes m'empêchant d'être productif, j'ai donc décidé de rentrer le réparer.

## Lundi 13 mars :

### Mise en place de l'espace de travail :

Après une semaine bien remplie entre maladie et problème technique je reprends le projet, le lundi en classe. J'attaque par la mise en place du GitHub ainsi que sa publication sur le Moodle. Une fois fait, je mets un moment à réinstaller les outils manquants comme go et git. Je commence par faire du front en attaquant le header qui sera commun à toutes les pages.

### Création des éléments de base :

Vers les 11h je termine tous les assets et base du site (header, footer, et css pour le contenu, mise en place du serveur go) et je le push sur le GitHub sous le nom « Création des éléments essentiels du projet »

### Création de la page d'accueil :

Je m'attèle ensuite à la création de la page d'accueil qui conformément à la consigne doit contenir une présentation de l'API ainsi que de ces différents paramètres pour les recherches. Cette partie du travail est encore une fois du front car il s'agit encore une fois d'ajouter tout simplement le contenu de la page d'accueil ce qui permet au passage d'apprendre les détails de chacun des arguments de l'API. Cette page a été finie vers les 15h00 et a été push sous le nom « Finition contenu page d'accueil »

## Mardi 14 mars :

### Création de la page de recherche :

Je repris le projet groupé tracker le lendemain dans l'après-midi, je me suis dit que quitte à avoir commencé par la autant finir avec le front, je m'attaque donc à la page de recherche où il m'a fallu créer la barre de recherche ainsi que les cartes. Une fois fait j'essayerais d'afficher les résultats de l'API. Le front fut fini vers les 15h30 et pushé sous le nom de « Finition front page recherche ».

### Récupération formulaire utilisateur et traitement :

Dans cette partie je m'occupe de la récupération et du traitement des données de la barre de recherche de la page recherche. Cette partie a été rapidement réglée dans les 20 minutes car elle consistait en une fonction du package « net/url ».

### Affichage des données :

J'ai par la suite commencé le traitement des données ainsi que son affichage dans la page de recherche vers les 16h. N'ayant pas eu le temps de terminer j'ai push sous « avancement affichage des données » à 17h30.

### Jeudi 16 mars :

#### Affichage des données :

Après une journée de Yboost bien chargé je reprends dans la matinée le projet groupie tracker. J'en étais au point où la requête fonctionnait tout juste, mais plusieurs problèmes restaient. Il fallait maintenant afficher les bons noms dans les cards. Il fallait changer l'image qui était en qualité trop basse puis il fallait ajouter un filtre pour chercher dans un type seulement à la fois. J'ai donc commencé vers les 10h par les filtres car une fois fait le reste sera simplifié (notamment l'affichage des cards qui n'a pas les mêmes informations à afficher) J'ai pu sur le reste de la matinée finir les filtres ainsi que régler le bug de la qualité d'image, il me restait toutefois à faire du tri sur les données que j'affiche sur le Template, il faut aussi encore que je rajoute certains filtres et que je leur applique leur CSS. J'ai cependant dû arrêter de bosser en fin de matinée, étant indisponible l'après midi pour des raisons médicales. J'ai toute fois push de retour sur mon ordinateur vers les 19h mon avancement de la journée, sous le nom de « Avancement des filtres utilisateur + restructuration et traitement de la data »

### Vendredi 17 mars :

#### Affichage des données :

Je reprends donc le projet en fin de matinée vers les 11h dû à encore une fois des problèmes techniques. Je reprends donc l'affichage des données entamé la veille. Tout en réglant plein de petites choses à droites. Notamment j'ai ajouté deux fonctions l'une qui consiste à formater la date afin de la rendre lisible à l'utilisateur et une qui consiste à vérifier si un contenu est marqué comme explicit et si tel est le cas à rajouter à la fin de son titre un petit « © ». De plus j'ai ajouté dans le tri de mes données une catégorie music-video car jusqu'à présent je ne l'avais pas vu mais une requête de type musique pouvait renvoyer un clip et ce type étant catégorisé comme default avait pour conséquence d'afficher une card vide. A ce point j'ai push mon avancement sous le nom « Correction de plusieurs bugs et optimisation » vers les 12h. Une fois fais-je commence le vrai nom de cette catégorie le paramétrage de l'affichage des données. Une grande partie du traitement se fait en Golang directement avec ma nouvelle structure, mais il est toujours nécessaire de faire des ajustements avec des if sur le Template, comme pour par exemple choisir entre afficher une description ou un album en fonction d'un livre ou d'une musique. Vers les 12h30 j'ai fini pour le pop-up mais if pour afficher la bonne chose, cependant deux problèmes persistent, tout d'abord je constate que de temps en temps le prix peut être de 0 ou négatif, ce problème peut être très rapidement réglé avec un if, puis second problème plus important. Je me rends compte que mon pop-up est beaucoup trop petit pour les informations à afficher. La solution est connue et bien que je l'aie évité, je m'attaque désormais au single page. Je push donc vers les 12h30 mon avancement décrits avec le problème 1 résolu sous le nom de « Adaptation du Template au contenu + correction bug affichage prix »

#### Système de page unique :

J'ai donc repris mon travail vers les 14h après ma pause repas pour intégrer ce système, pour ce faire je charge dans ma fonction handler deux templates et si l'URI correspond au format demandé par regex (merci Marianne pour ça) il exécute le Template de single page avec l'identifiant, sinon celui de recherche classique. Une fois fini vers les 15h30 avec comme à chaque fois avec le réglage de quelques

bugs comme celui des recherches ne trouvant pas de résultat (Eh oui c'est là que je l'ai ajouté). Je push sur le GitHub sous le nom « Ajout système de page unique + correction de bugs »

#### Design de page unique :

Une fois ce système de page fonctionnel, reste à designer ce que l'on mettra dedans, je m'inspire du design d'Apple music mais en plus moche (la mocheté n'est pas volontaire) puis je commence à répartir l'affichage des informations conformément à mes besoins. Une fois fait pour la musique je push ma progression sous le nom « Création page résultat » vers les 18h15, avant d'adapter mon code pour les autres types de données. Après une journée bien rempli j'ai fini d'adapter le Template à gros coup de « if » pour les 5 types supporté à l'heure sur mon site. Comme à chaque fois je vais donc push, actuellement il est 19h33 et le push sera sous le nom de « Adaptation de la page résultat en fonction du type de contenu ».

### Samedi 18 mars :

#### Design de page unique :

J'ai repris le lendemain logiquement dans la matinée vers les 9h et ceux pour avancer le css, je le pensait certes finis mais j'avais en réalité complètement oublié la description présente dans les films et série télé. J'ai donc passer beaucoup trop de temps sur son implémentation à cause d'un bug css que je n'arrivais pas à régler. Je push donc cette implémentation vers les 11h30 sous le nom de « Ajout de la description pour les contenu vidéos »

#### Optimisation et réglage de bugs :

Une fois cette partie finis, j'ai techniquement mon projet de fait, toute les fonctionnalités présentes sont fonctionnelles, reste à optimiser le tout, finir le css et notamment le responsive, régler les deux trois bugs mineurs, puis ce projet groupie tracker sera derrière moi, il restera aussi bien sur le readme et les bonus de la consigne. Je commence donc par optimiser le Golang à tous les étages, jusqu'à présent fonctionnel il était loin d'être optimal. Je push donc son optimisation vers les 12h. Une fois fais-je me suis mis à optimiser les templates. Une fois fini vers 13h30 après une pause midi je push sous le nom « Optimisation Template et correction de bug mineurs »

#### Finition du front :

Bon la on attaque la partie sérieuse, une partie que je fais jamais au bon moment mais qui est nécessaire, le responsive. Maintenant que nous avons fait les optimisations principales et finis les templates, quoi de plus logique que de faire le responsive. C'est donc ce que j'ai commencé à faire cet après-midi. Heureusement je n'ai pas eu trop de difficulté majeure, c'est donc après 215 lignes 4 media queries et trois bonne heures que j'ai finis le responsive en pushant donc vers les 16h « Finition du responsive »

#### Gestion d'erreur :

Nous commençons à rentrer dans les détails de ce projet et c'est peu dire, dans cette partie j'ai ajouté la page d'erreur 404. Ça m'aura quand même pris 45min parce que j'ai galéré à trouver comment le faire (encore une fois merci Marianne). Je push donc sous le nom « Ajout gestion erreur 404 » vers les 17h.

## Dimanche 19 mars :

### Correction page 404 :

Le dimanche jour de repos ce journal ne sera pas très rempli toutefois en allant me coucher hier et en réfléchissant au bug je me suis rappelé que ma chère image de la page d'erreur n'était pas adaptée au téléphone. Chose désormais faite.

## Lundi 20 mars :

### Améliorations diverses :

Nous voilà dans la dernière ligne droite du projet, et je connais les tâches qu'il me reste, entre détails de dernière minute et partie manquante la matinée devrait largement suffire à tout terminer, voici la liste de mes tâches :

- Restructuration de fichier
- Création du README.MD
- Ajout de la persistance des champs du formulaire après une recherche
- Correction bug scroll tableau

J'ai mis les tâches dans l'ordre de réalisation. Et donc tout en écoutant la présentation d'Ethan je commence à restructurer mes fichiers. Cette tâche a été aussi rapide que la présentation d'Ethan en somme. Je push donc directement sous le nom « restructuration de fichier ».

J'ai ensuite commencé ce que j'ai longtemps procrastiner à faire, le readme. En le rédigeant j'écris que je l'ai testé sur chrome brave et Firefox, ne l'ayant pas encore fait au moment où j'écris les mots, je m'attèle à la tâche, le site fonctionne parfaitement sur les trois navigateurs, cependant j'ai remarqué grâce à la lenteur de Firefox que mon site a besoin d'une image de chargement, on verra si j'ai le temps d'implémenter cette fonctionnalité. Une fois finis vers les 9h30 je push le readme accompagné d'une correction mineure css.

Je commence donc ma troisième tâche qui devrait rapidement être réglé, il suffit de passer les données du formulaire dans la structure de renvoi et d'adapter le Template. Une fois réglé je le push sous « Ajout persistance recherche » vers les 9h45.

N'ayant pas envie de régler ce bug du tableau qui m'a déjà fait perdre pas mal de temps je vais essayer de voir si je peux mettre des images de chargement. Changement de programme je suis allé voir Erika qui m'a gentiment réglé mon problème de tableau, j'attaque donc ma dernière tâche l'image de chargement. Au premier abord mettre une image de remplacement paraissait compliqué, il suffit en réalité de rajouter juste une image background qui viens se faire recouvrir une fois notre image chargée.

C'est donc à 10h11 que je finis ce beau projet, honnêtement il était cool et svp fais nous bosser seul plus souvent. Puis là le module rien à redire le projet est ultra intéressant et instructif on sent que ce que l'on a fait est vraiment utile. Honnêtement GG pour le module.