

Conception Logicielle 2023

Rapport Vortex Race +

1. Contexte

Le rapport du projet Streamlit pour Vortex Race + s'inscrit dans le cadre du cours de développement logiciel donné par Monsieur Picca qui s'est déroulé de février 2023 à mai 2023. L'objectif était de créer un logiciel en parallèle du site web de la Vortex Race, une course populaire créée il y a deux ans par un comité d'étudiants en Science du Sports à l'université de Lausanne. Vortex Race + est une application web qui permet aux utilisateurs de rassembler toutes les vidéos qui les intéressent pour leurs entraînements, de consulter rapidement leurs dernières performances et d'accéder en un clic au site web officiel de la Vortex Race.

Pour réaliser cette application, nous avons utilisé le framework Streamlit qui permet de créer rapidement des applications web en Python. Streamlit nous a permis de développer une interface utilisateur conviviale et facile à utiliser pour les utilisateurs de l'application. Nous avons également utilisé des bibliothèques Python telles que : webbrowser, os, PIL, streamlit_card, streamlit_login_auth_ui.widgets, streamlit_extras.colored_header. Toutes ces importations nous ont permis d'améliorer notre site pour une meilleure expérience utilisateur.

Nous avons rencontré plusieurs défis tout au long du développement de Vortex Race +, tout d'abord, la prise en main de Streamlit a été une tâche complexe et chronophage. Il nous a fallu un certain temps pour comprendre comment fonctionne cette bibliothèque et comment l'utiliser efficacement pour créer une interface utilisateur. Ensuite, la coordination entre les membres de l'équipe pour relier nos avancées personnelles a été un autre défi. Nous avons travaillé sur des fonctionnalités différentes, ce qui a nécessité une communication et une planification étroites pour nous assurer que tout fonctionne bien ensemble. Enfin, nous avons dû nous adapter aux fonctionnements de Streamlit et accepter que certaines choses ne sont pas toujours possibles à réaliser. Cela a parfois été frustrant, mais nous avons travaillé dur pour trouver des solutions alternatives et atteindre nos objectifs dans les délais impartis. Malgré ces difficultés, nous sommes fiers de ce que nous avons accompli et sommes convaincus que cette expérience nous a permis de devenir de meilleurs développeurs logiciels.

Dans l'ensemble, le projet Streamlit pour Vortex Race + a été une expérience enrichissante pour nous en tant qu'étudiants en développement logiciel. Nous avons pu appliquer nos compétences en Python et découvrir de nouvelles technologies telles que Streamlit pour créer une application web utile pour les utilisateurs.

2. Organisation, problématique, résolution

Pour reprendre les termes que nous avons utilisés dans notre premier rendu, voici les objectifs initialement déclarés :

“Nous aimerions développer un logiciel permettant au coureur (user) d’accéder, via un login, à leur informations concernant la course ; le temps, leur départ, les photos, distance parcourue, heure de départ, ce serait un dashboard complet pour les coureurs (users).

Il serait également possible de créer une équipe avec d’autres coureurs (user), via un onglet correspondant. Possibilité de compléter les équipes déjà existantes, ou de chercher des coureurs pour compléter sa propre équipe. Une fois l’équipe créée, un nouveau dashboard équipe apparaît avec les membres et autres stats.

Nous mettrons également en place une médiathèque communautaire, qui permet aux utilisateurs de partager des vidéos youtube d’entraînement qu’ils trouvent utiles. On peut like/dislike les vidéos pour les faire ressortir ou non. Possibilité d’ajouter des vidéos à son dashboard (favoris).”

Nous avons réussi à mettre en place un système de login, ainsi qu’une page profil pour chaque coureur contenant ses informations personnelles et ses temps de course. De plus, nous avons réussi à créer une médiathèque de vidéos YouTube, avec la possibilité pour les utilisateurs de liker et d’ajouter des vidéos à leurs favoris. Les vidéos les plus aimées de la médiathèque sont celles qui seront mises en avant.

Cependant, nous n’avons pas pu créer la page collaborative pour les équipes, ni développer la page complète statistiques, par manque de temps et de connaissances. Nous avons tout de même ajouté une page d’accueil pour présenter les différentes fonctionnalités de notre web app, ainsi qu’une page Vortex Race renvoyant au site officiel de l’événement.

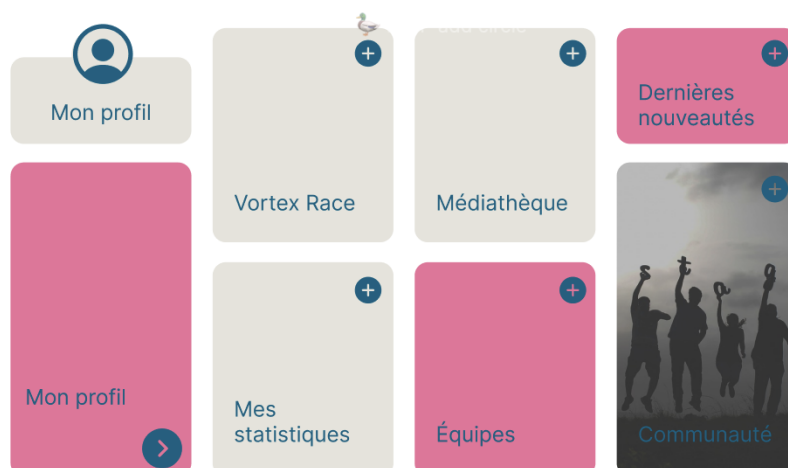


Figure 1 : Page “Accueil” de la maquette.

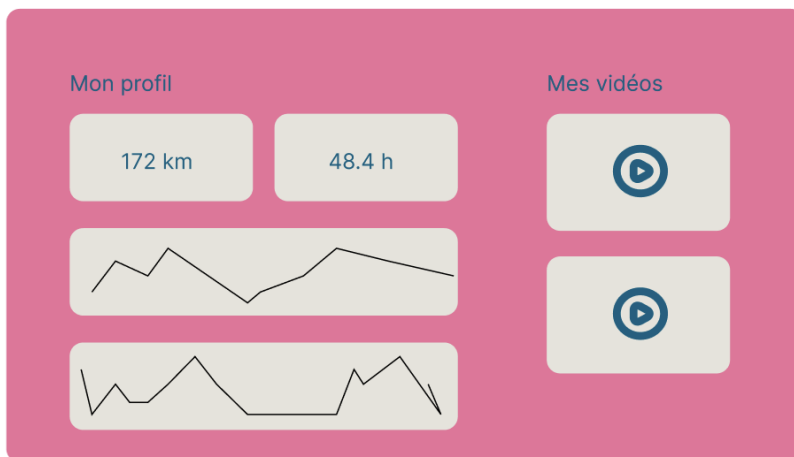


Figure 2 : Page "Mon Profile" de la maquette.

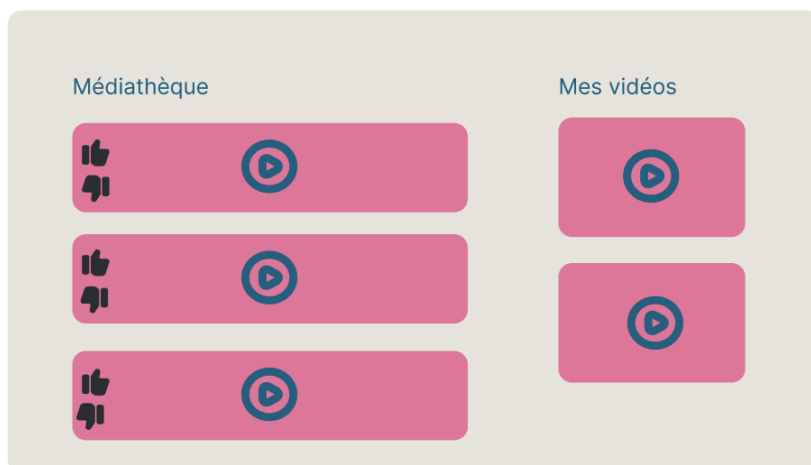


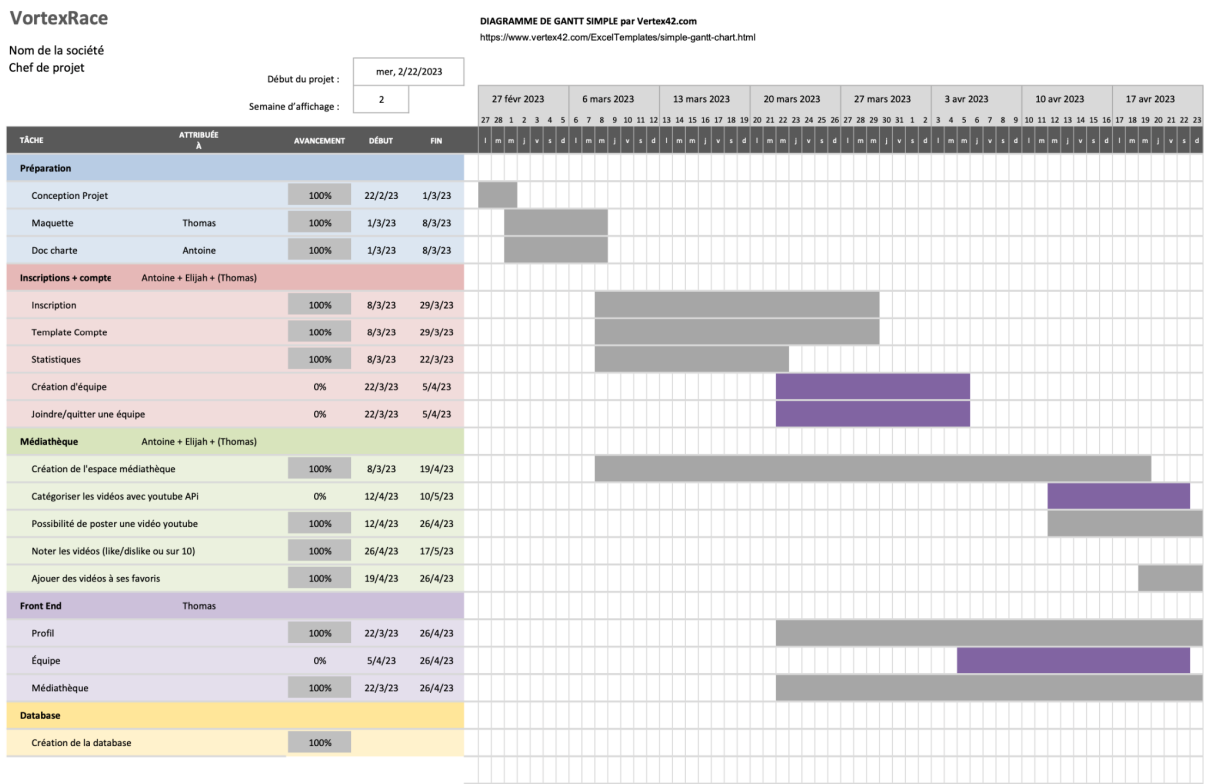
Figure 3 : Page "Médiathèque" de la maquette.

Au départ, nous avons créé une maquette de notre site web sur Figma en utilisant exclusivement du HTML et du CSS. Cependant, une fois que nous avons commencé à travailler sur le projet, nous avons rapidement réalisé que l'implémentation de cette maquette serait très fastidieuse et compliquée. Nous avons donc décidé de repenser complètement le frontend de notre projet et d'utiliser les fonctionnalités offertes par Streamlit nativement ou via des packages externes que nous avons importés.

Nonobstant ce changement de direction, nous avons tout de même cherché à nous rapprocher autant que possible de notre vision initiale en utilisant les outils disponibles sur Streamlit. Cela nous a permis de créer une application web fonctionnelle et efficace pour notre projet de logiciel de course Vortex Race +.

Malgré toutes ces limites, nous sommes satisfaits de notre avancée car nous avons fait le choix de nous concentrer sur les fonctionnalités les plus importantes et avons tenté de les réaliser dans les temps. Nous espérons que notre travail sera utile pour les coureurs de la Vortex Race, et qu'il pourra éventuellement être amélioré et développé davantage à l'avenir.

3. Gantt chart et calendrier



4. Code

Dans notre projet Streamlit, nous avons adopté une approche organisée pour structurer notre code. Nous l'avons divisé en plusieurs fichiers logiques pour faciliter la gestion et la compréhension. Notre fichier principal main.py contient la logique principale de notre application. Nous avons également créé un fichier mediatheque.py qui regroupe toutes les fonctionnalités liées à la médiathèque.

Chaque fichier a été organisé en utilisant des fonctions pour regrouper des blocs de code liés. Par exemple, dans main.py, nous avons des fonctions pour la configuration initiale de l'application, la création de la mise en page et la gestion des interactions avec l'utilisateur. Dans mediatheque.py, nous avons des fonctions pour l'affichage des vidéos, la gestion des likes et des favoris, etc.

Nous avons divisé le travail en 3 axes principaux :

- Le login a été réalisé principalement par Elijah Green
- La médiathèque a été réalisé principalement par Antoine Vigand
- Le frontend a été réalisé principalement par Thomas Rywalski

Des réunions hebdomadaires nous ont permis de mettre en commun nos différentes avancées afin de se coordonner, mais également de nous entre-aider afin de résoudre les problèmes techniques rencontrés.

Nous avons également utilisé des commentaires pour expliquer le but et le fonctionnement des différentes parties de notre code. Cela facilite la compréhension pour les autres développeurs et nous permet de nous rappeler rapidement de la logique derrière chaque bloc de code.

En adoptant cette approche modulaire, nous avons favorisé la réutilisabilité du code. Par exemple, nous avons extrait des fonctions communes dans des fichiers séparés et les avons importées lorsque nécessaire. Cela nous a permis d'avoir un code plus concis et évolutif, en évitant la duplication de code et en facilitant les modifications ultérieures.

En dernier lieu, nous avons suivi une convention de nommage cohérente pour nos variables et fonctions, ce qui a rendu notre code plus lisible et compréhensible.