

# Programmation Web

## Projet

### 4A- FSI2

LUCAS Antoine

BIAU Alexandre



**URL de notre site :** <https://antoineantix-warzazat-project.glitch.me/>

**Identifiants par défauts :** Username : admin ; Password : admin

**Description du projet :** Le but de ce projet est de créer une plateforme qui permet aux utilisateurs de réaliser un test et d'analyser leurs résultats de manière interactive et ergonomique.

**Étapes de conception :**

Pour la réalisation de ce projet nous avons deux fichiers «.vue », « App.vue » et « warzazat.vue ».

Pour le changement de page nous changeons de « v- container » à l'aide de variables de type booléenne.

1. Création de la page d'accueil

Nous avons décidé de dédier notre page d'accueil pour la connexion au serveur. Pour cela nous avons créé un cadre (« v-card ») contenant la requête pour le nom d'utilisateur (« v-text-field »), le mot de passe, bouton d'inscription et de connexion. Lorsque l'on clique sur le bouton d'inscription un formulaire apparaît afin de finaliser l'inscription. Concernant l'en-tête de la page nous avons mis une barre avec le titre du projet et un bouton de déconnexion qui apparaît au moment de la connexion. Sans s'être inscrit auparavant, l'utilisateur ne peut pas se connecter et un message apparaît. De même pour l'inscription (si l'utilisateur est déjà enregistré ou non).

Enfin en bas de page nous avons écrit nos nom et prénom et nous avons affiché le logo de l'école.

2. Création de la page utilisateur :

Lorsque l'utilisateur entre ses identifiants et qu'il est inscrit alors cela nous permet d'afficher le contenu de notre « v-container » affilier à l'écran d'accueil de l'utilisateur. Nous affichons un message de bienvenue avec le nom de l'utilisateur et en dessous nous affichons son meilleur score (qui est enregistré dans le serveur). À partir de l'instant où l'utilisateur réalise deux tests une courbe de la progression apparaît et s'actualise à chaque nouveau test via la balise « v-sparkline ». Les scores sont enregistrés sur le serveur ce qui permet leur affichage à chaque connexion de l'utilisateur.

De plus à partir du moment où l'utilisateur est connecté nous avons la possibilité d'avoir accès à une « navigation bar » qui apparaît au moment du clique sur le bouton affilié à gauche de l'écran.

Cette dernière nous permet d'accéder à la page d'information sur le joueur, l'historique des parties, au classement des utilisateurs et enfin un bouton permettant la suppression du compte.

3. Création du questionnaire :

Le questionnaire est entré en « dur » dans les données du script cependant les réponses sont enregistrés dans le serveur ce qui permet à l'utilisateur de ne pas y avoir accès, et ainsi de ne pas tricher. Le lien se fait via une variable index, qui est la même pour le tableau des questions et celui des réponses, celle-ci est directement envoyé au serveur pour avoir accès aux solutions et ainsi vérifier les réponses de l'utilisateur.

4. Historique des parties :

Chaque réponse de l'utilisateur est enregistrée dans un tableau qui est envoyé au serveur, ce tableau est mis à jour après chaque test où les réponses s'ajoutent les unes à la suite des autres.

---

L'index lui va correspondre au test que l'on veut observer lors de la diffusion de l'historique. Ainsi on évalue chaque réponse enregistrée avec les réponses aux questions et on indique à l'utilisateur quand il s'est trompé ou non.

#### 5. Classement des utilisateurs :

Pour la réalisation du classement de chaque utilisateur nous trions dans le serveur les utilisateurs en fonction de leur meilleur score et nous créons un second tableau avec les données non personnelles de l'utilisateur (username, rank et high score). Ce tableau est envoyé depuis le serveur et il servira de base de données pour le classement dans la partie front-end.

#### 6. Suppression du compte :

Nous avons mis à disposition de l'utilisateur un bouton qui lui permet de supprimer son compte. Afin de le supprimer dans le serveur, nous cherchons dans le tableau des utilisateurs l'index correspondant aux username et password du compte en question (en effet, ces derniers sont uniques pour chaque compte). Une fois que nous avons l'index correspondant, nous faisons un `array.splice` qui nous permet de supprimer l'utilisateur correspondant.

#### **Les difficultés rencontrées :**

Le premier problème auquel nous avons dû remédier était la coordination du travail. En effet ayant dans notre binôme un seul possesseur d'ordinateur portable et étant donné qu'une grande partie du projet ait été effectué à l'école nous avons dû travailler sur une seule machine. Puis lorsque nous voulions travailler en simultané mais ne pouvions pas nous réunir, l'utilisation du logiciel TeamViewer était nécessaire.

La seconde difficulté que nous avons eu fut la synchronisation avec le serveur pour chaque fonctionnalité. Pour y remédier, nous étions contraints de créer des variables ou tableaux spécifiques pour la liaison entre le `server.js` et notre application.

La plus grosse difficulté que nous avons eu concerne la partie front-end. En effet, nous devons accorder les données qui nous viennent du serveur avec les différents composants que nous avons voulu afficher à l'écran (boutons, graphique, etc.). De ce fait, nous devons travailler sur un seul fichier «`.vue`». Pour cela nous avons créé plusieurs variables booléennes qui nous permettent de changer d'écran ou de réaliser certaines fonctions.

---