

# BAWE - Anime Quiz

---

## Présentation

---

Ce projet est un Quiz sur des personnages d'animés. Lors d'une partie le joueur aura 10 images de personnages d'animes, pour chaque image il aura 4 réponses possibles avec une seule de correcte. Le but étant donc d'avoir 10 réponses correctes.

## Les différentes pages

---

### Bare de navigation

---

On commence par là car, ce n'est pas une page mais elle est présentes sur toutes les pages du site.



On a donc à gauche les différentes pages disponibles que l'on exploreras plus tard.

A droite on a le bouton pour chnanger de thème, il suffit simplement de cliquer dessus et encore à droite on a la liste des langues disponibles. Actuellement il y a Français et Anglais mais il est possible d'en ajouter simplement en modifiant le fichier `text.json`.

### "Jouer"

---

Lors-ce qu'on arrive sur le site, on arrive par défaut sur la page "Jouer", qui nous permet pour l'instant de jouer en tant qu'invité en cliquant sur le bouton "Commencer le Quiz".

## Jouer en tant que : guest ?

[Commencer le quizz](#)

Une fois le quiz lancé, les choix arrivent au fur et à mesure et le résultat est donné à la fin.

## Jouer en tant que : guest ?

[Commencer le quizz](#)

[Gintama. Shirogane no Tamashii-hen - Kouhansen](#)[Koukaku Kidoutai: Stand Alone Complex 2nd GIG](#)[Odd Taxi](#)[Haikyuu!! To the Top Part 2](#)[Gintama. Porori-hen](#)[Yojouhan Shinwa Taisei](#)[Perfect Blue](#)[Shinseiki Evangelion Movie: Air/Magokoro wo, Kimi ni](#)

## Paramètre

Cette page sert à se connecter / à créer un compte sur l'application. Pour ça il suffit d'entrer son nom et son mot de passe dans les champs nécessaires (on remarquera que le nom par défaut est guest pour les personnes sans comptes), et de cliquer sur Connexion ou S'inscrire selon ce que l'on veut faire.

## Profile

Fuyuki

.....

[Connexion](#)[S'inscrire](#)

## Profile

Fuyuki

.....

[Connexion](#)[S'inscrire](#)

## Leaderboard

Finalement la page de leaderboard montre le score de toutes les personnes inscrites, le score correspond au score maximum fait en une partie.

## Leaderboard

#	Pseudo	Score	Date
1	test	10	2025-11-16 16:35:50
1	Carpe Argentee	10	2025-11-17 19:43:26
1	Grochaton8	8	2025-11-17 21:39:35
1	Saucisse	6	2025-11-18 19:25:59
1	Fuyuki	4	2025-11-17 20:42:47

## Leaderboard

#	Pseudo	Score	Date
1	test	10	2025-11-16 16:35:50
1	Carpe Argentee	10	2025-11-17 19:43:26
1	Grochaton8	8	2025-11-17 21:39:35
1	Saucisse	6	2025-11-18 19:25:59
1	Fuyuki	4	2025-11-17 20:42:47

# Techologies

## Front-end

Pour le front-end, ayant déjà fait des projets web avant avec plusieurs framework comme NextJs / React ou Svelte, J'ai cette fois-ci décidé d'utiliser un nouveau framework, et en me baladant dans mes recommandations je suis tombé sur `ripple.js`, un nouveau framework encore en développement actif qui

a une approche particulière :

Il ne renvoie pas le code HTML dans des retour de fonction (comme Next et React), ni n'inclue du JS dans le formatage de l'HTML (comme svelte) mais à l'inverse, permet de mettre du HTML directement au sein du JS.

### Note

Au début du développement le lsp n'était pas disponible via npm donc je ne n'ai pas pu tester sur neovim avec la dernière version, cependant l'extension VSCode, qui était la seul disponible avant, utilise une ancienne version de JS pour détecter certaines erreurs, ce qui provoque certains faux positifs sur les erreurs.

## Back-end

Même résonnement pour le backend, actuellement ripple ne permet pas d'en créer un j'ai donc choisis un framework que je n'avais utilisé avant, partant donc sur `Hono`, qui est un framework simple pour créer des endpoints.

## Détails techniques

---

### Gestion des pages

Habituellement dans les framework JS un routeur est fournis pour gérer les différentes pages, ici aucun, j'ai donc utilisé un rendu conditionnel visible dans le `App.ripple` pour afficher les pages en fonction de laquelle est sélectionnée dans la navbar.

### Database

Ne voulant pas compliquer l'application avec une base de donnée externe soit disponible depuis mes différents poste de travail soit avec des données différentes j'ai choisis d'utiliser SQLite en tant que base de donnée. Le fichier de base de donnée est donc situé à la source et s'appelle `database.db`

### Style globaux

Au début du développement les styles globaux n'existaient pas, il fallait donc modifier le `index.html` en ajoutant un style. Finalement ils ont ajoutés une balise css `:global()` qui permet d'appliquer le même style à travers les différents modules.

On peut aussi aborder ici le sujet des styles, qui ne sont pas écrits dans des fichiers CSS mais écrits directement dans les modules. Dans les autres framework il est possible d'importer des fichiers de style dans la partie JS cependant en commandant le projet je n'ai trouvé aucune informations là-dessus sur la documentation de ripple. Il est possible que cette fonctionnalité ait été ajouté depuis mais je n'ai pas pris le temps de regarder la documentation régulièrement, le style est donc actuellement inclus dans les balises styles de chaque modules.