

INF4041 - Programmation

WEB Avancée

4A FSI2



IGNITE

Lien Glitch : <https://sumptuous-elk.glitch.me/>

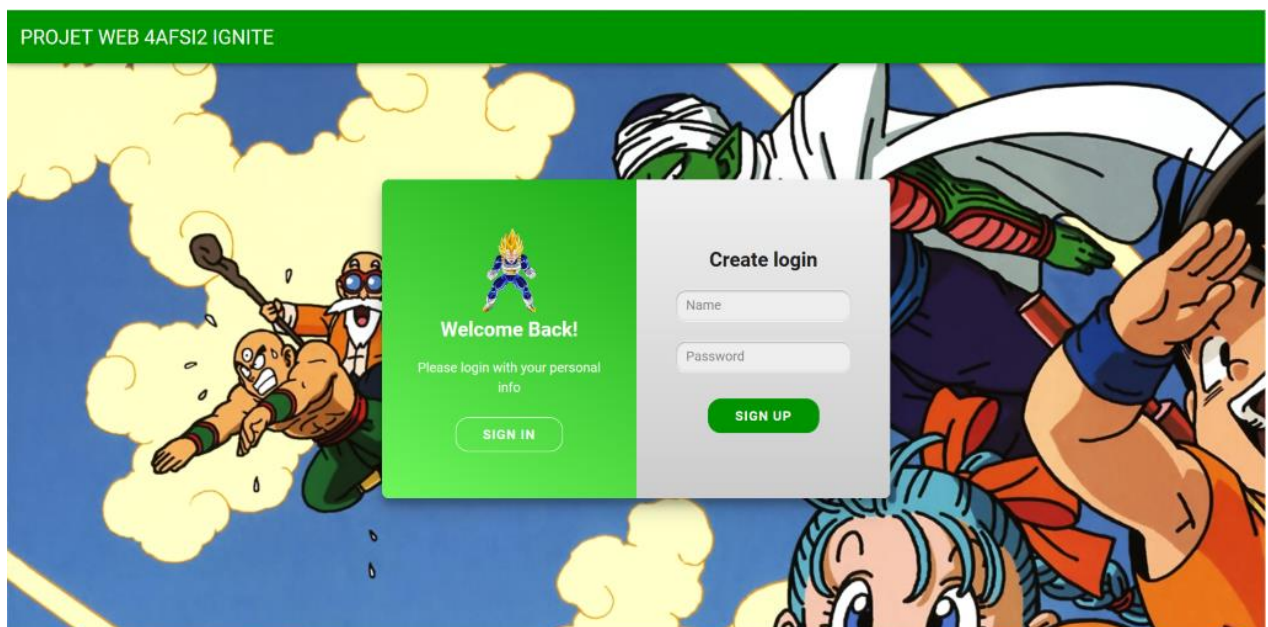
Les identifiants par défaut de notre site son admin en nom et admin en mot de passe mais il est facile de créer un nouvel utilisateur (notamment en créant un identifiant et un mot de passe vide, ce qui permet une plus grande rapidité de test)

Les consignes à respecter sont les suivantes :

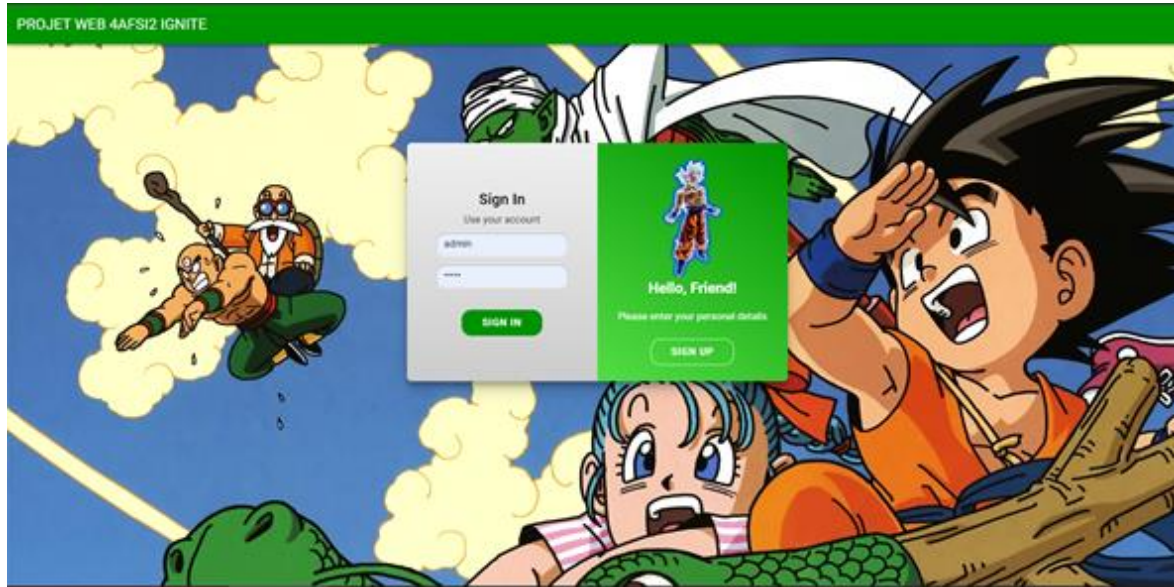
- Web interface built with HTML, CSS and Vue.js
- Web server built with Node.js and Express
- User authentication (and registration if your application requires it)
- CRUD interactions with the server
- Data persistence is optional: in-memory is good enough, but you can try to write to the file system (no database!)
- Deployed on Google Cloud Platform or Glitch

Lorsqu'on arrive sur la page, deux choix s'offrent à nous :

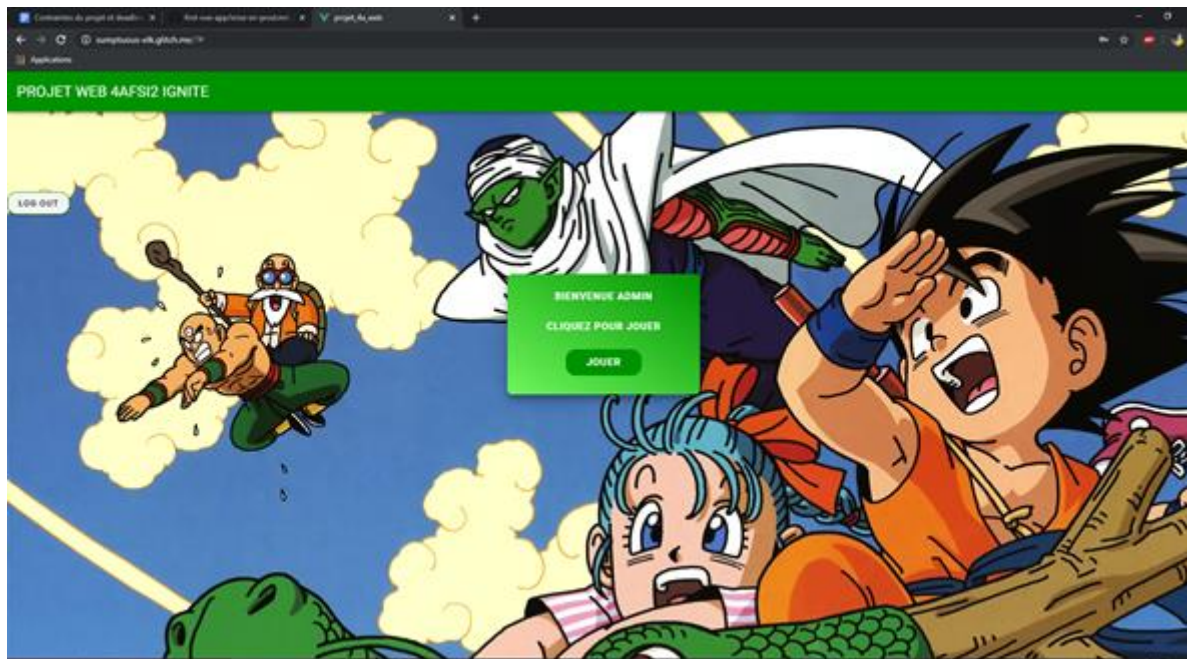
1. S'inscrire



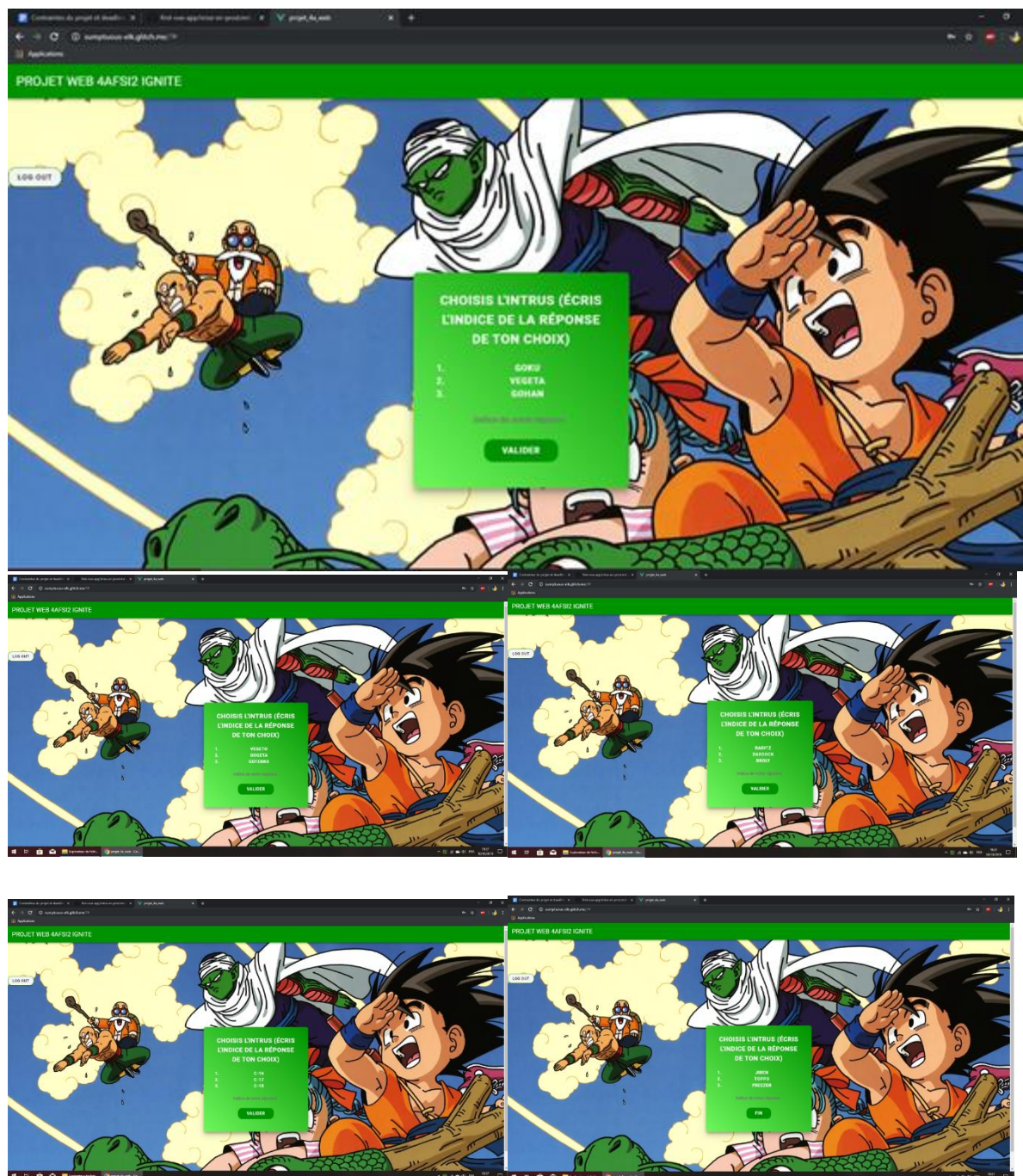
2. Se connecter



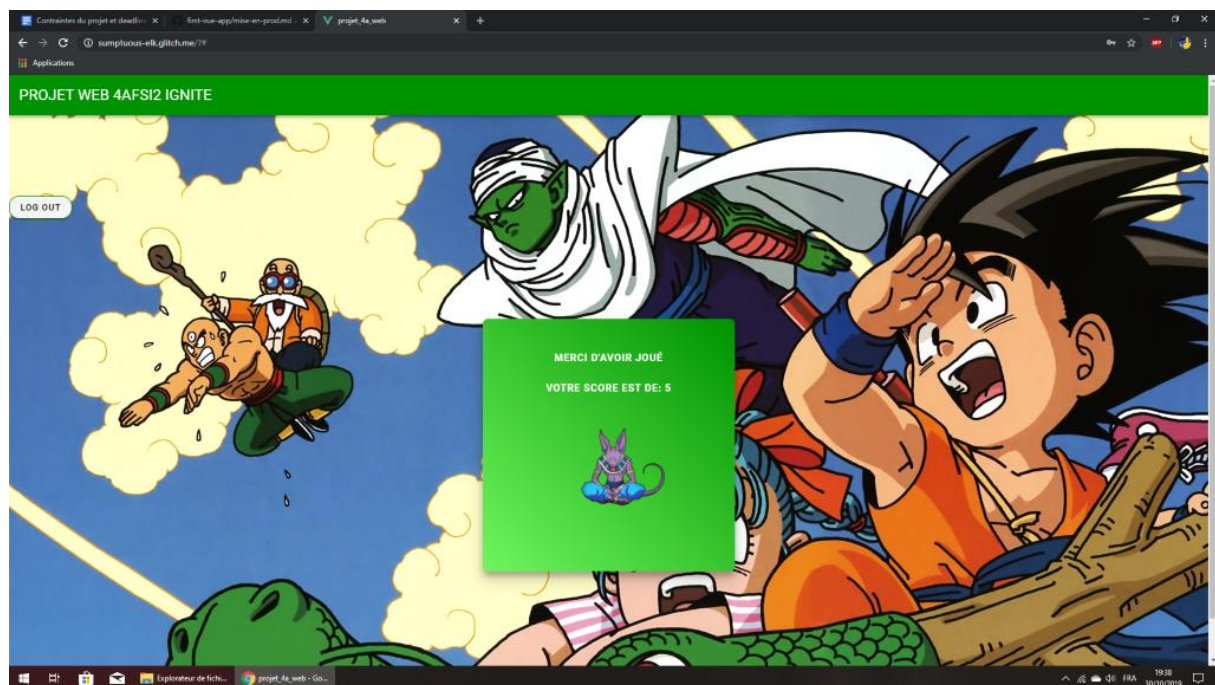
Une fois connecté, on arrive sur un menu qui nous propose de commencer à jouer ou de nous déconnecter (log out).



Ensuite, si on appuie sur jouer, un mini quizz sur le manga Dragon Ball Z se lance. Le quizz comporte 5 questions où il faut trouver l'intrus. Par exemple, la première question nous demande de déterminer qui est l'intrus entre Goku, Vegeta et Gohan. L'intrus est ici : Gohan car il est terrien-saiyan contrairement à Goku et Vegeta qui sont totalement saïyan.



Une fois que vous avez répondu ou passé toutes les questions, votre score apparaît.



Il y a aussi possibilité de se déconnecter pendant le jeu ou après avoir joué.

Pour résumer avec les consignes à respecter :

On a une interface Web **qui fonctionne avec HTML, CSS et Vue.js**.

La connexion est une **authentification de l'utilisateur** auprès du serveur **qui fonctionnent avec Node.js**

CRUD (Create Read Update Delete) :

On peut créer un identifiant et un mot de passe, pour commencer une partie, les champs identifiants et mot de passe sont lus et analysés par le serveur. A chaque question le score est mis à jour en fonction de la réponse (bonne ou mauvaise réponse), à la fin de la partie le score est retourné et en se déconnectant la session de jeu est détruite. On peut recommencer une partie avec le même utilisateur ou changer d'utilisateur, le message de bienvenue change en fonction du nom de l'utilisateur.

Les **données ne sont pas enregistrées**, seul l'identifiant est sauvegardé. Lorsqu'on se déconnecte, on doit recommencer le QCM depuis le début.

Nous avons déployé notre interface Web **sur Glitch**. Elle est accessible à partir du lien précédemment donné.

Etapes de conception, de réalisation et de déploiement :

1. Réflexion sur ce qu'on souhaitait mettre dans notre site
2. Création du bloc pour "slider" entre connexion et inscription
3. Lier les boutons à des fonctions qui communiquent avec le serveur (ex : pour se connecter il faut que les identifiants existent et soit validés par le serveur)
4. Création des questions/réponses du quizz
5. Création d'un bloc qui s'affiche après être connecté pour commencer à jouer
6. Création du premier bloc du jeu qui affiche question, réponses et un champ pour écrire la réponse choisie et un bouton pour valider
7. Création du bouton log out pour se déconnecter
8. Fonction qui incrémente l'index pour switcher de question dès qu'on appuie sur le bouton valider
9. Fonction qui compare les réponses de l'utilisateur et la solution pour incrémenter le score
10. Création du bloc final qui affiche un message de fin et le score de l'utilisateur
11. Ajout d'images au début à l'écran d'inscription/connexion, à la fin du quizz et sur le fond du site
12. Différents tests sur le site
13. Déploiement du site sur Glitch.com
14. Tests sur le site déployé sur Glitch.com

Difficultés rencontrées :

1. Insérer des images
2. Coder à deux sur Github (problèmes de conflits entre les deux codes)
3. Réussir à lier le frontend et le backend pour l'inscription et la connexion
4. Problème de l'image de fond sur glitch
5. Bouton log out
6. Utilisation de Glitch