

Projet Web 4A

ESIEA 2019-2020 S1

Equipe classe SI1:

MAZIARSKI Charles

BLEUZE Hugo

GEILLON Martin

The Quizz

Url site : <https://thebootfinder-projet4as1-1.glitch.me/>

Github : <https://github.com/Thebootfinder/Projet4AS1.git>

Table des matières

Présentation	3
Intro.....	3
Url du site, Github, identifiant	3
Que fait le site ?	3
Réalisation technique.....	3
Cahier des charges :	3
Analyse des tâches / GANTT :	4
Projet et fonctionnalité	4
Etape de conception, réalisation	4
Déploiement.....	5
Difficultés rencontrées.....	5
Conclusion :	6

Présentation

Intro

Dans le cadre du projet Web du 1^{er} semestre de 4eme année. Nous devons réaliser un site web utilisant des technologies précises. Le projet avait plusieurs contraintes que nous vous détaillons par la suite.

Url du site, Github, identifiant

Url site : <https://thebootfinder-projet4as1-1.glitch.me/>

Github : <https://github.com/Thebootfinder/Projet4AS1.git>

Identifiant : admin

Password : admin

Cependant vous avez la possibilité de vous enregistrer en quelques secondes seulement , cet identifiant ne devrait pas vous servir.

Que fait le site ?

Notre site web est un site offrant la possibilité à une personne de créer un compte, s'authentifier. La personne peut faire les deux quizz que nous proposons. Chaque quizz est présenté de la façon suivante : Une question avec pour réponse, oui ou non. A chaque réponse l'utilisateur gagne plus ou moins de points. Une fois le quizz terminé, l'utilisateur peut décider d'enregistrer son score dans le tableau des derniers scores. Ce tableau affichant le nom de l'utilisateur, le quizz qu'il a fait, le nombre de point qu'il a eu et à quelle date.

Sur le site l'utilisateur peut également retrouver les règles et une manière de joindre un support en cas de problème.

Réalisation technique

Cahier des charges :

- 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](#)

Analyse des tâches / GANTT :

Une fois le cahier des charges établis, compris et analysé, nous avons décidé de réaliser un site web proposant :

- Réalisation d'au minimum un Quizz
- Système d'authentification
- Possibilité d'enregistrer ou non le score du quizz
- Interface ergonomique et pratique

GANTT :

Tâche à accomplir	État	Septembre					Octobre				
		1				30	1				31
Lancement											
Définition du projet à réaliser	Fait										
Définition des étapes de réalisation	Fait										
Définition du GANTT	Fait										
Etape du projet / BackEnd											
Système d'authentification	Fait										
Réalisation du Quizz	Fait										
Réalisation des scores Quizz	Fait										
Correction bug et vérification	Fait										
Etape du projet / FrontEnd											
Création page spécifique Quizz	Fait										
Création design page d'accueil	Fait										
Création de tout les autres pages	Fait										
Correction bug et vérification											
Formation											
Prise en main de GitHub	Fait										
Prise en main de Vue JS	Fait										
Jalons											
Date de rendu : 30/10	Fait										
Rédaction du rapport	Fait										
Déploiement du site	Fait										

Projet et fonctionnalité

Etape de conception, réalisation

Les étapes de conception et réalisation ont été réalisé à partir du GANTT prévisionnel.

Formation : En premier lieu nous avons commencé par prendre en main Vue.js et Git Hub. Nous avons donc fait de multiples tests sur les différentes balises et les composants. Après un mois d'entraînement, nous avons commencé, en parallèle, le système de Login/Register ainsi que le design et la structuration de notre page d'accueil.

Login : Pour le Login/Register, nous avons donc dû créer les échanges entre le client et le server afin de sécuriser les identifiants des utilisateurs.

Accueil : Pour la page d'accueil, nous en avons créé plusieurs différents. Chacune sous un modèle différent. Une première à l'aide de Listes, permettant de classer chaque catégorie de manière claire et ergonomique. Une seconde à l'aide de <V-card> nous permettant de faire beaucoup plus de chose et de personnalisation (Entre autres mettre des listes si besoins) puis une troisième sur le même principe avec un design différent. Nous avons finalement choisi la 2eme version qui était plus adaptée à notre besoin, plus clair et plus « jolie » pour l'expérience utilisateur.

Quizz : Une fois que l'utilisateur peut s'enregistrer, se login et se trouve sur la page d'Accueil, il faut qu'il puisse lancer les différents Quizz mit à sa disposition. Nous avons donc réalisé un système de Quizz Oui/Non avec des boutons cliquables, qui, en fonction des questions, donnent plus ou moins de points.

Score : Pour terminer, nous avons réalisé la partie Affichage du Score qui nous a pris le plus de temps que ce soit le JavaScript ou le Template. Nous avons décidé d'afficher les Scores dans une Liste avec les informations utiles comme le Username du joueur, à quel Quizz il a joué, la date à laquelle son score est enregistré... Pour le stockage du score et de différentes variables, nous avons utilisé le localStorage/sessionStorage. Une fois que le joueur a enregistré son Score il peut revenir à l'écran d'Accueil.

Les plus : Nous avons aussi ajouté quelques informations supplémentaires comme « Comment nous contacter » ou encore « La team de conception du site », qui sont des composants uniquement visuels sans interaction utilisateur mais qui ont le mérite d'exister pour la cohérence de notre site web.

Pour les routes entre les différents composants nous avons utilisé le module « router » et pour la communication avec le server le module « axios ».

Déploiement

Pour le déploiement nous avons utilisé Glitch. Le déploiement à mis un certain temps en raison d'une très mauvaise connexion mais à bien fonctionné. Cependant une fois déployer nous avons des problèmes sur certaine pages. En effet il était impossible d'arriver sur la page score où nous avons l'erreur Cannot GET. Nous réfléchissons à des solutions efficaces.

Difficultés rencontrées

Durant ce projet nous avons rencontré plusieurs difficultés :

- Nous avons eu du mal à bien comprendre où étaient nos erreurs au début car ESLint est très sévère et la moindre « , » à la place d'un « ; » peut engendrer une erreur. Résolu avec l'expérience.
- Difficulté à utiliser le module « axios » au début car c'était la première fois que nous programmions coté server. Malgré ça, c'est une très belle découverte. Résolu avec l'expérience.

- Problème de Merge sur Git Hub, résolu avec des modifications à la main dans notre répertoire.
- Impossible de sauvegarder la Liste des différents Scores utilisateurs après rafraichissement de la page.
- Problème de requête sur la page sur le site déployé. Nous n'avons pas trouvé pourquoi nous arrivons pas à y accéder , supposition : utilisation de LocalStorage qui n'est pas bien traité en dehors du 'local'. Il faudrait donc revoir l'intégralité de la page.

Conclusion :

Ce projet était une très bonne expérience pour les membres de l'équipe. Nous avons pu travailler avec des outils que nous ne connaissions pas ou d'autre pas assez pour réellement les maîtriser. Ceci nous a permis d'apprendre et de nous améliorer avec un projet très pratique et visuel. Même si à ce jour le site peut encore être amélioré, nous apprécions d'avoir déjà pu apprendre à faire cela. Nous espérons dans un futur proche pour faire un site plus accès avec une interaction base de données afin de donner une nouvelle dimension à notre projet.