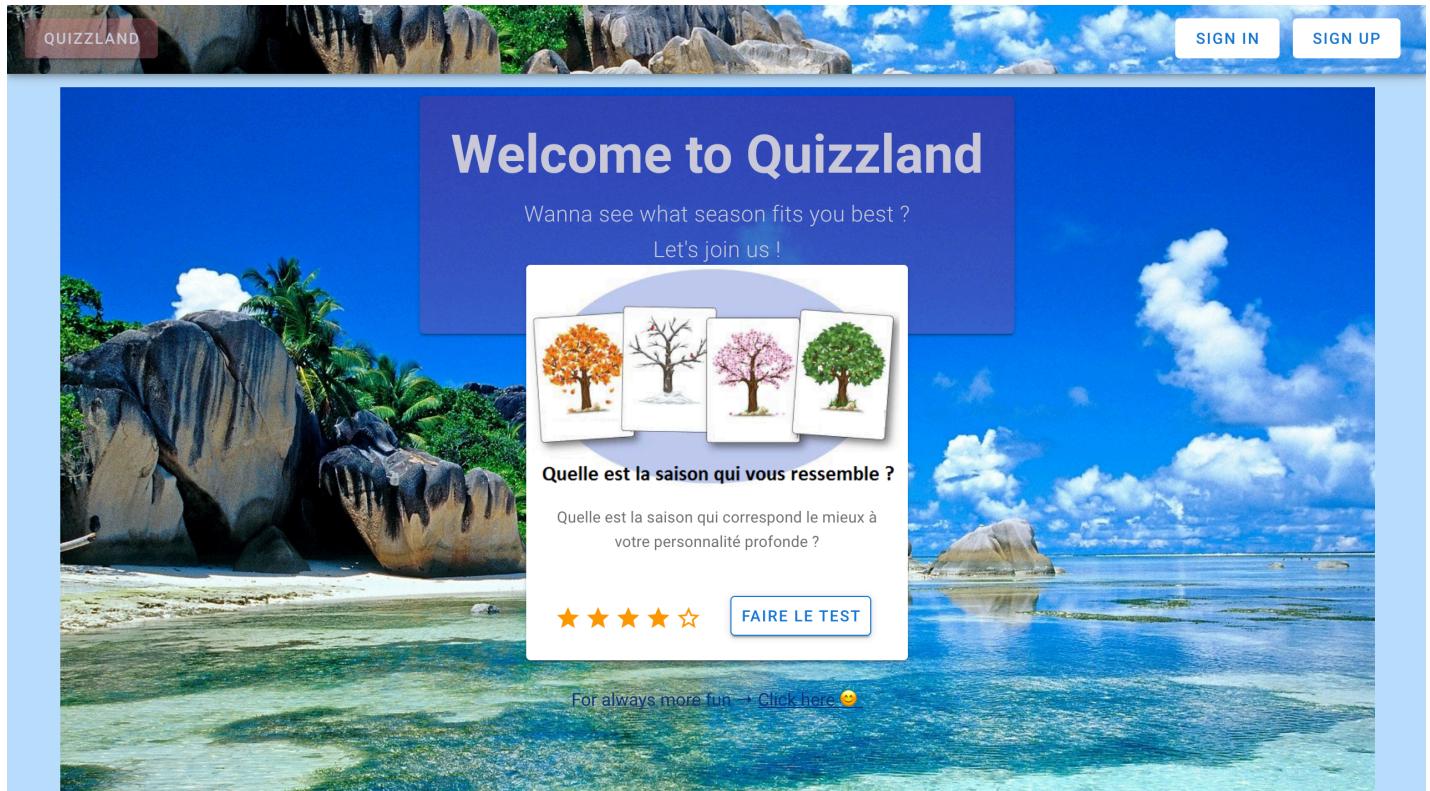

PROJET WEB : TEST DE PERSONNALITÉ



Rapport Projet Web

Préparé pour : CHEREL Louis

Préparés par : DAHO Lenny, OUTEMZABET Nesrine, BLANC-TALON Théodore

28 October 2019

Objective

L'Objectif du projet est de réaliser un site web moderne codé en Vue.js, et l'apport de Material Design avec Vuetify.

Le projet doit comporter plusieurs contraintes :

La création d'un formulaire de registre composé des éléments suivants :

- Le nom, l'email et un mot de passe.

Puis lorsque l'on clique sur le bouton Submit, on puisse enregistrer les données dans un tableau du coté serveur.

Ensuite, on ajoute les différents éléments du une todolist et nous passons à la page du test après la souscription du membre. Après celui ci, nous devons être capable d'utiliser un bouton delete sur la liste pour supprimer un élément.

Nous avons décidés de travaillé sur un test de personnalité.

L'url de votre site :

Description du projet, en quoi il répond aux contraintes demandées :

Pour ce projet, il nous a été demandé de réaliser une application web avec une authentification et une connexion à un serveur.

Notre projet n'a pas rempli toute les contraintes du projet mais nous avons essayé de faire au mieux.

D'une part, les étapes de Front-end ont toutes été plus ou moins respecté. C'est la première partie que nous avons travailler. En y ajoutant des règles et des éléments de web design.

L'architecture que l'on a pensé(e) est la suivante :

Étapes de conception, réalisation, déploiement :

Nous voulions créer une page d'accueil, puis s'authentifier pour pouvoir réaliser le test de personnalité du projet.

Pour la partie serveur, tout n'est pas fonctionnel. Nous arrivons à obtenir des réponses de requêtes POST effectué par la fonction "register ()" dans le register.vue, et la fonction "async login ()" dans le login.vue, toutes deux définis dans la partie méthode effectuée dans script, écrit en javascript. mais on peut faire le test sans créer de comptes. Nous n'avons pas réussi à créer un lien entre la page register et la page login.

Pour le reste des contraintes comme créer une Todolist dynamique dès l'inscription du client, nous avons fait en sorte d'être obligé de remplir tout les critères des règles que nous avons imposé dans le register pour valider la todolist avec une commande :disabled="!valid" sur les boutons de validation de list, que se soit pour le login ou celui du register.

Pour la réalisation du test de personnalité, nous nous sommes basé(e)s sur la méthodes suivante :

On récupère les données des questions dans une matrice, puis on mélange l'ordre des réponses par le mélange de l'attribut des id de la carte sur laquelle on clique. Pour l'attribution de la réponse au test de personnalité, le résultat est attribué en fonction de la somme des id des cartes cliquées.

Par exemple : Si on obtient un résultat inférieur à 1/4 du résultat maximal possible, la réponse sera l'hiver.

Si on obtient un résultat en 1/4 er 1/2, la réponse sera Automne, etc

Pour le déploiement sur le site, Nous avons suivis les étapes suivantes :

1. Modifier server.js et ajouter les deux lignes suivantes après app.use(cors()):

```
const path = require('path')
app.use(express.static(path.join(__dirname, 'dist')))
```

2. Nous avons enlever toute référence à http://localhost:4000 dans vos fichiers .vue => http://localhost:4000/api/login devient simplement /api/login

3. dans le fichier .gitignore, retirez la ligne /dist, puis

```
git add .gitignore
git commit -m "remove /dist"
```

3. dans le dossier du projet à chaque nouvelle version, lancez la commande

```
npm run build
git add ./dist
git commit -m "add new version"
=> cette commande doit être tapée à chaque nouvelle version de votre code
```

4. Dans votre fichier package.json, après la ligne **"lint": "vue-cli-service lint"**, ajoutez la ligne suivante:

```
"start": "node server.js"
n'oubliez pas de mettre une virgule à la fin de la ligne précédente
```

4. Allez sur [glitch.com](#)

5. New Project > Clone from git repo. Nous avons utilisé l'url Github : <https://github.com/Doucina/Personality-Test>

6. Cliquez sur "show" en haut à gauche

7. Attendez jusqu'à ce que ça fonctionne Pour débugger les problèmes, vous pouvez visualiser les logs dans "Tools" > Logs

Quelles difficultés ont été rencontrées :

Nous avons eu énormément de difficulté à essayer de mettre en place le serveur du projet. Nous avons pas compris comment le réaliser sans la base de données en sql. Cette partie n'est toujours pas fonctionnel.

De plus, la compréhension de l'utilisation du Github nous a aussi pris énormément de temps, comme par exemple le fait de pouvoir commit sur le git de quelqu'un d'autre, mais aussi de bien comprendre les différentes étapes pour bien effectuer des commit dans le Github.

Un des moments où on a voulu merger nos projets dans une des branches, nous avons eu des conflits, et nous avions du tout reprendre sur Github de 0.

Puis nous avons eu des difficultés pour le déploiement du site sur Glitch.