

Rapport Projet Web

RALAMBOSON Mathieu
ARMAND Shenyi

Description du projet :

Le projet que nous avons réalisé répond à toutes les contraintes et nécessiter proposer par le projet Web de l'année 2019.

En effet du coté front, l'interface du site à été coder en HTML5, CSS3, Javascript (Vue.js) et du coté back, nous avons utilisé Express (Node.js) et Axios.

Notre site internet permet à n'importe quelle personne se connectant en tant que « Utilisateur », de consulté nos articles, décrivant les Superhéros issus des comics et dessin moderne.

Si l'utilisateur se connecte en tant qu'« Administrateur », il aura accès à la modification , et à la suppression des articles.

Une fonctionnalité supplémentaire fournit par notre site est l'utilisation de URL pour imager les articles dans le site.

Nous avons pris soin d'utiliser chaque requête CRUD avec Axios soit le get, le post, le put et le delete.

Afin d'apporter une expérience utilisateur agréable nous avons décidé de nous orienter vers un design épuré. La librairie utilisée est Vuetify.

Le GitHub du projet : <https://github.com/MathieuRalamboson/BalanceTonArticle.git>

L'URL du site : <https://mathieuralamboson-balancetonarticle.glitch.me>

Contraintes :

Authentification réaliser :

Les identifiants nécessaires à la connexion au site internet :

Pour être administrateur :

- ✓ Identifiant : *admin*
- ✓ Mot de passe : *admin*

Pour être utilisateur :

- ✓ Identifiant : *user*
- ✓ Mot de passe : *user*

Ou

- ✓ Identifiant : *lol*
- ✓ Mot de passe : *lol*

CRUD réaliser :

- ✓ Create : Bouton « Ajouter un article »
- ✓ Read : Récupération et Affichage des derniers articles
- ✓ Update : Bouton « Modification de l'article choisie »
- ✓ Destroy : Bouton « Suppression de l'article choisie »

Étapes de conception :

La principale difficulté à été de choisir le thème du projet.

Tous les deux fans de superhéros nous nous sommes facilement accordés sur le fait de choisir ce sujet.

Ensuite est venue la partie programmation, nous avons créé une branche séparée par personne sur GitHub pour que chacun apporte ses modifications.

Quelles difficultés ont été rencontrées :

Les difficultés rencontrées ont surtout été techniques.

Par exemple :

- Pour les requêtes : `this.$router.push`, afin de passer en paramètres une donnée, il faut impérativement écrire le nom de la prochaine page (name), alors que ce nom n'est pas indispensable.
Ceci n'est mentionné nulle part dans la doc vue.js et ce problème nous a pris du temps à résoudre.
Cette résolution a permis la différenciation User, Admin et l'affichage de certains boutons en fonction du statut.

```
this.$router.push(
  { path: "/page-1" ,
    name: 'Page View',
    params: { admin: response.data.admin}
  });
```

- Pour l'update des URL des images, il fallait trouver la bonne syntaxe pour utiliser les URL des images fournies.

```
<img v-bind:src=imgUrl>
```

- Dans l'affichage des articles, tout est écrit ligne par ligne. Ainsi il fallait trouver une solution afin de ne pas afficher toute une description de 600 caractères. Cela nous a pris une semaine de recherche.