



Rapport programmation WEB

HYPER SITE CONSULTING

Url du site :

<https://hugomatt-progweb-7.glitch.me/>

Cédric DAUMUR - Stéphane
JOANANY- Hugo MATTIODA

Classe FS12



Table des matières

Description du projet.....	2
Étapes de conception, réalisation, déploiement.	3
Conception :.....	3
Réalisation :.....	3
Déploiement :	4
Difficultés rencontrées.....	5
Conclusion.	6

Description du projet

Notre projet consiste à créer un blog dont les utilisateurs connecté(e)s peuvent consulter des articles. L'administrateur peut à tout moment en créer de nouveaux ou en supprimer.

Pour ce projet nous utilisons Vuetify, qui nous permet d'implémenter très facilement un design très soigné tout en ayant de la légèreté et une certaine ergonomie dans le code. Il nous permet de nous affranchir du HTML et du CSS. Avec la documentation très complète nous pouvons inclure dans notre projet plusieurs composants.

Pour les échanges avec le serveur nous utilisons JSON qui nous permet de transiter des informations avec le serveur via les requêtes GET, PUT, POST, DELETE. (Nous utilisons également des fonctionnalités de « vue-session » pour gérer les connexions des utilisateurs mais nous détaillerons cette fonctionnalité dans la partie développement)

Avant d'être connecté sur la page web nous devons nous identifier.
Soit nous nous connectons en tant qu'administrateur avec les identifiants suivants :

User : Admin

Password : changethispassword,

Soit nous nous créons notre propre compte via l'onglet « Register » pour pouvoir se connecter en tant qu'utilisateur grâce aux informations renseigné (Ici l'utilisateur ne pourra que consulter les articles, en aucun cas en créer ou en supprimer).

Sur notre site web nous pouvons créer des articles via l'onglet « article » uniquement si nous sommes connectés en tant qu'administrateur. Nous affichons des données depuis le serveur tel que l'utilisateur ou bien la description de l'article grâce à la requête GET. Nous n'avons pas réussi dans le temps imparti à faire une modification des articles mais nous pouvons tout de même supprimer des articles via l'onglet « Article » lorsque nous sommes connectés en tant qu'administrateur également.

ATTENTION : Après de la suppression d'un article il devient impossible d'en créer un nouveau sans actualiser la page pour une raison inconnue. Si l'actualisation ne fonctionne pas, aller sur l'onglet « Home » puis revenez sur l'onglet « Article ». Il vous sera de nouveau possible de créer un nouvel article (Si vous êtes toujours connecté en tant qu'administrateur bien entendu).

Étapes de conception, réalisation, déploiement.

Conception :

Pour la conception du projet, nous voulions que le l'idée soit simple et qu'elle puisse se résumer en une phrase : la création blog qui publie des articles. À ce stade-là nous avions une idée très vague du rendu final sur le point de vue du design du site. Nous avons d'abord commencé par dessiner sur papier les différentes idées de présentation du site avant de nous lancer dans la conception. Une fois que le design et les fonctionnalités ont été accepté par l'équipe nous avons pu entamer la réalisation.

Réalisation :

Pour la partie réalisation du projet, nous avons séparé le projet en deux parties :

- Une partie back-end qui s'occupe de tout ce qui est communication entre le site web et le serveur. Il s'agit notamment de la conception du fichier server.js qui va servir en quelque sorte de « Base de données » à notre site web. Lors de la création d'un utilisateur ou la création d'un article, la partie back-end va être sollicité pour envoyer les informations au serveur, les vérifier et les sauvegarder. Pour les informations de connexion (User et Password) nous avons utilisé le système « vue-session » et « vue-sessionstorage », lorsque l'utilisateur et le mot de passe sont correct nous ouvrons une session grâce au commande session qui vont nous permettre de récupérer plus facilement les articles sur le serveur. Pour que ces commandes fonctionnent il est nécessaire de les installer grâce aux commandes suivantes :
 - Vue-session : npm i vue-session
 - Vue-sessiostorage : npm i vue-sessionstorage

Le rôle du back-end dans ce site web où les informations doivent être manipuler avec précautions est donc très importants. Les commandes permettant de lier les informations saisis par les utilisateurs sur le site avec le serveur sont présentes dans quasiment chacune des vues (saisis des identifiant, ajout de nouveaux articles, suppression d'article, affichage des articles, création d'un nouvel utilisateur...)

- En ce qui concerne la partie front-end, cette partie va nous servir du côté client à récupérer les informations saisies par l'utilisateur. Ces informations sont primordiales car ce sont elles qui seront traitées dans le back-end. De plus, tous les éléments de la partie front sont directement aperçus sur le navigateur de l'utilisateur, c'est le contact direct avec l'utilisateur. C'est pour cela que le site doit être simple à utiliser avec un design accrocheur.

Les vues suivantes sont celles liées à la partie front-end :

- About.vue
- Menu.vue (vue principale présente sur toutes les pages de notre site (toolbar))
- Login.vue
- Register.vue
- Article.vue
- Home.vue

Déploiement :

Une fois notre site web fonctionnel nous sommes passé à la partie déploiement. Nous avons suivi les différentes étapes données dans la description du projet ce qui nous a permis de déployer le site sur glitch.com.

Le déploiement c'est déroulé sans encombre voici l'url de notre site web :

<https://hugomatt-progweb-7.glitch.me/>

Néanmoins le déploiement du site pose un problème majeur. Afin d'être correctement déployé nous avons dû corriger les commandes « axios » qui permettes l'envoi des données au serveur ou la récupération des données du serveur en supprimant la partie suivante dans chacune des commandes : `http://localhost:4000`.

Dorénavant le site fonctionne correctement sur glitch.com mais si quelqu'un tente de récupérer notre projet (sur GitHub) et le teste en local sur son ordinateur (avec `npm run serve` et `nodemon server.js`) les commande axios ne fonctionnerons plus, il faudra alors rajouter la partie `http://localhost:4000` pour refaire fonctionner la communication avec le serveur.

Difficultés rencontrées.

Les principales difficultés rencontrées durant ce projet sont majoritairement liées aux liaisons entre le serveur et le client.

En effet, dans un premier temps nous avons eu un problème sur la récupération des données clients afin de l'envoyer côté serveurs grâce aux différentes requêtes GET et POST. (Requête GET pour récupérer les informations envoyer côté serveur, requête POST pour envoyer les informations côté client.)

Une fois ce problème résolu nous voulions récupérer les informations stockées sur le serveur afin de les afficher du côté client. Ce fut le plus gros problème durant ce projet car nous avions du mal à interpréter la transition des données du serveur au client. Nous parvenions à envoyer les données mais nous ne parvenions pas à les afficher. Nous sommes finalement parvenus à corriger ce problème grâce aux commandes « session ».

Le problème résolu, nous nous sommes penchés sur la suppression d'un article. C'est dans cette partie que nous nous sommes rendu compte qu'il allait falloir rajouter un ID unique afin d'identifier chaque article. Cependant après plusieurs échecs de suppression nous nous sommes rendu compte que l'id unique était inutile et qu'il fallait juste renseigner la ligne du tableau qu'il fallait supprimer dans une fonction « splice ».

La dernière difficulté de ce projet a été le design. En effet ce fut une étape longue à réaliser car il fallait vraiment effectuer des placements aux pixels près afin d'avoir un bon rendu. De plus, la partie back-end était encore en cours de réalisation, il fallait donc réajuster le site afin de prendre en compte les dernières fonctionnalités.

Conclusion.

Nous avons finalement réussi à produire un site web fonctionnel correspondant aux attentes exigé. Nous sommes particulièrement fières du résultat car aucun des membres de notre groupe n'avait manipulé le Vue.js avant cette année.

Nous avons rencontré différents problèmes qui nous ont permis d'apprendre assez rapidement ce langage. L'une des principales difficultés fut également l'utilisation de GitHub que nous ne sommes pas parvenu à maîtriser vraiment parfaitement.