
Rapport

-

Projet WEB

Membres:
Charles Cohen

Introduction

Dans le cadre de notre 4ème année à l'ESIEA, nous avons l'objectif de réaliser en moins de deux semaines une interface web en utilisant des langages de programmation HTML, CSS, Vue.js et Node.js.

Nous devons intégrer un serveur en utilisant la plate-forme Node.js et le framework Express. Chaque page HTML devra être statique et les actions sauvegardées, même en rechargeant la page seront enregistrées dans le serveur.

L'application web comportera également une page login : l'utilisateur pourra se connecter et s'enregistrer à l'aide de ses informations personnelles.

Le design, les fonctionnalités particulières et le choix du thème sont à définir nous-même.

Table des matières

Introduction

1. Présentation du projet.....	4
2. Réalisation et architecture du projet.....	5
3. Fonctionnalités.....	6
4. Difficultés rencontrés.....	7
5. Conclusion.....	8

1. Présentation du projet

Nous avons décidé de faire un site de e-commerce sur le thème du sport. Les clients devront à leur arrivée se connecter ou s'inscrire avant d'accéder au contenu de notre site. Une page de présentation les accueillera.

Les utilisateurs auront accès à une liste de programmes de sport qu'ils pourront acheter. Ils auront à leur disposition une rapide description du produit et pourront l'ajouter à leur panier. Le prix des articles sera automatiquement calculé.

La conception des pages sera réalisée en html, le design en CSS et Vue.js permettra de rendre dynamique. Enfin, le serveur Node.js permettra d'enregistrer les données des utilisateurs et de garder en mémoire notre panier tout en consultant le catalogue de notre site MyWorkout.

2. Réalisation et architecture du projet

Nous avons débuté par la réalisation d'une interface html et CSS pour avoir un visuel des différentes fonctionnalités que nous allons ajouter. Cette partie nous semble la plus simple car nous en avons déjà fait l'année dernière.

Intéressons-nous à Node js, express et Vue.js.

A) Node.js

Node est un serveur d'application qui comporte un serveur http ainsi qu'un framework, Express. Ici, l'utilisateur envoie une requête http au serveur, le serveur http demande les informations au framework et renvoie la réponse par un url au serveur. Il finit par renvoyer la requête du serv

eur http à l'utilisateur. La partie Node.js se code en JavaScript. Elle nous a permis de garder en mémoire les loggers et les articles de notre panier.

B) Express

Après l'installation du serveur, nous avons configuré nos paramètres dans la partie app.js. Nous avons enregistré les différents URL à laquelle l'application doit répondre puis nous avons instancié l'application express et paramétrer son serveur en lui attribuant son port.

On a également indiqué a vue JS l'emplacement des fichiers statiques afin de diriger notre page html. On l'a récupéré en utilisant get.

C) Vue.js

Vue.js est très simple d'utilisation. Lorsque que l'on passe d'une page à une autre normalement, on crée une nouvelle page html ; mais grâce à Vue.js, on a inséré des sections, permettant de garder une page unique et de rester sur le même lien html.

Les fonctions d'ajout, de suppression de panier et la partie login se fait en Vue.js.

Chaque application est initialisée en créant une instance de Vue et chaque propriétés utilisés sont ajoutées dans l'objet data. Quand une valeur de ces propriétés change, la Vue, se met à jour pour concorder avec les nouvelles valeurs.

3. Fonctionnalités

Nous avons commencé par la partie Front-end afin d'avoir un visuel, et de partir sur une bonne base. Les premiers jours ont été consacrés à la recherche et à l'apprentissage des nouveaux langages.

L'utilisateur arrive sur une page d'accueil et peut commencer par se connecter en cliquant sur l'onglet « Sign Up ». Nouveau utilisateur, il pourra accéder au formulaire d'inscription. Il devra y rentrer son pseudo et son mot de passe. Un pop-up apparaîtra afin de confirmer son inscription. Il ne lui reste plus qu'à se connecter avec ses informations. Vous avez maintenant accès au Eshop depuis votre compte. Vous y trouverez les différents programmes de musculation suivis de leur prix et de ce qu'ils contiennent dans l'onglet « Basket ». Vous pouvez les ajouter à votre panier en cliquant sur le bouton « Buy ». Dès lors, un pop-up apparaît, indiquant l'article que vous avez ajouté. Ils seront disponibles dans l'onglet « basket » de la nav bar. Un total de votre shopping est calculé.

Vous désirez passer au paiement ? C'est possible. Il vous suffit de rentrer les informations relatives à l'envoi de votre colis et les données de votre carte de crédit.

Ces dernières n'ont pas été configurées faute de temps.

En tant que fondateur du site, vous pouvez ajouter et supprimer. Il vous suffit de vous connecter en tant que « Admin » avec le mot de passe « 123 ». Dans la partie shop, vous pouvez y ajouter une photo de votre article, lui ajouter une description, un prix et le nombre d'exercice qu'il contient.

Vous pouvez également, supprimer des articles. Un bouton delete se trouve au niveau de chacun d'eux.

Nous avons ajouté un onglet pour nous contacter. Il vous permet d'envoyer des requêtes ou de poser des questions. Faute de temps, celui-ci renvoie simplement à l'accueil.

4. Difficultés rencontrés

La première difficulté qu'on a rencontrée est l'apprentissage accéléré de Vue.js mais après 2 jours d'étude, les fonctionnalités utilisées nous semblait faisable.

La partie où l'on a rencontré le plus de difficultés et le fait de garder en mémoire, sans utiliser de base de données, le panier de chacun des utilisateurs, de garder en mémoire les informations des loggers et garder les informations de la page même en effectuant une recharge. On a commencé par tout faire en Vue.js en se rendant compte rapidement que ça n'allait pas fonctionner. Puis après quelques jours, nous avons trouvé une technique permettant de garder en mémoire dans le cache de notre navigateur. Les données sont enregistrées dans le localStorage et garde en mémoire les objets enregistrés. Celui-ci se vide des lors que le navigateur se ferme.

Mais avec persistance, on a décidé de reprendre en JavaScript afin d'utiliser le serveur Node.js.

4. Conclusion

Contrairement à ce que l'on pensait utiliser, Node js apporte un gain de temps. On utilise du JavaScript dans le front-end et le back-end sans passer par d'autres pages html.

Ce projet nous a permis de développer nos capacités et d'apprendre de nouveaux langages de programmation. Il nous a permis de garder la main sur le html et consolider nos acquis.

Ce projet nous a également permis de travailler en équipe et d'utiliser au mieux les compétences de chacun afin d'être le plus efficace possible. Nous en sortons avec plus de compétences et d'aisance dans ces langages mais également dans la programmation en elle-même. Nous comprenons de plus en plus la logique et améliorons nos performances.