

Rapport Programmation WEB

La bonne montre



Joffrey Franceschini

Nizar Zidouh

ESIEA 2018 - 4SI2

Table des matières

I. Introduction.....	p. 3
II. Description du projet.....	p. 4
III.Mise en œuvre du projet.....	p. 4 à 8
1. Conception	
2. Réalisation	
3. Déploiement	
IV . Difficultés rencontrées.....	p. 9
1. Le framework VueJS	
2. Le NodeJS	
V. Conclusion.....	p. 10
VI. Annexe.....	p. 11

I. Introduction

À l'ère du numérique, les sites internet sont les premiers point d'entrée vers ce monde. Nous pouvons trouver tout ce que nous voulons sur le web, aussi bien pour se renseigner, se cultiver ou faire ses propres achats. Pour parfaire notre cursus en école d'ingénieurs plus particulièrement pour la section système d'information, nous devons réaliser un site web mettant en œuvre les dernières technologies du web, comme le HTML5 le CSS3 et aussi le JS avec le framework VUE.JS ainsi que l'environnement Node.JS .

Ce rapport expliquera donc notre projet et retracera toutes les étapes de la conception au déploiement en passant par la réalisation.

II. Description du projet

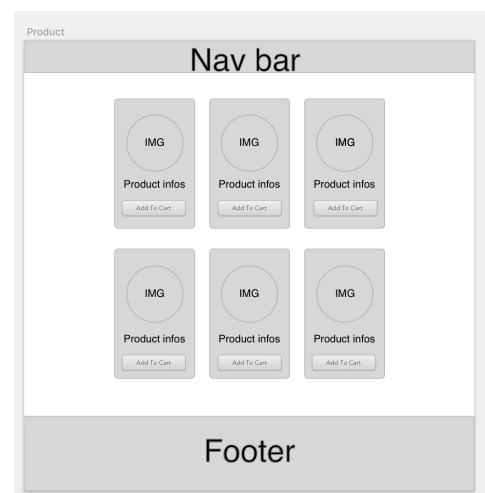
Le but du projet était de nous apprendre à utiliser les outils modernes du côté front-end et back-end. Nous avons donc décidé de réaliser un site marchand, proposant plusieurs modèles de montre. Le site est décomposé de la manière suivante une page principale décrivant le site, une page ou les produits sont afficher, une page panier récapitulant les produits que l'utilisateur à sélectionnés et une parti login qui permet de créer un identifiant ou de se connecter l'utilisateur est déjà créé. Pour pouvoir procéder au paiement, il faudra obligatoirement être enregistré.

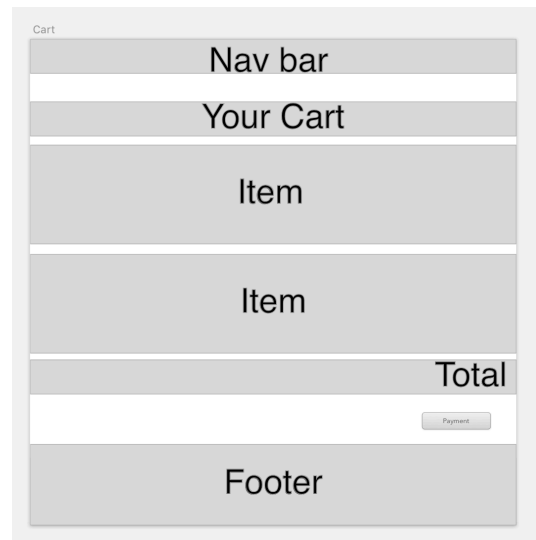
III. Mise en oeuvre du projet

A . Conception

Tout d'abord, nous avons réalisé des maquettes de conception de notre site en respectant les principes de développement d'un simple site web sur le MDN. C'est-à-dire que nous avons déjà en tête de créer une barre de navigation sur le haut du site, puis afficher le contenu de la page et ensuite finir la page avec une partie de pied page contenant des liens utile pour se rediriger sur le site.

Voici les designs que nous avons établie sur "sketch" avant de commencer le projet :





B. Réalisation

Nous avons commencé par la partie HTML et CSS en ne s'occupant pas dès le début de la partie Vue.js, car nous ne connaissions pas bien le framework jusqu'à lors. Mais en se renseignant sur comment fonctionnait ce dernier nous avons revue directement notre réalisation nous nous sommes plus penché sur VueJS et nous avons compris qu'il allait nous simplifier la vie plus rapidement. Nous avons donc commencé à adapter notre code en utilisant les sections. En utilisant des boucles pour créer des "div" tant que nous avions des items à afficher.

Notre code fonctionne comme ceci :

Nous avons 4 listes, une pour les stocks et une pour le panier, deux côté serveur et deux côté client. Contenant chacune d'elle les informations sur les produits, telles que le nom, une description, le prix, la quantité en stock, un lien vers l'image. Nous créons donc des cartes pour chaque objet dans la liste et en utilisant ses informations dedans. Nous avons un bouton par article qui permet de le mettre dans le panier tant que nous l'avons en stock. Une fois l'article plus en stock nous ne pouvons plus cliquer sur le bouton car il se "disable". Quand nous ajoutant un article au panier nous envoyons une requête http de type post au serveur pour vérifier que la disponibilité n'a pas change entre temps. Si elle n'a pas changé, nous mettons donc à jour les 2 listes coté client et les deux listes coté serveur.

Pour la partie login, nous avons une liste de "users" coté serveur qui nous permet de vérifier si l'utilisateur entré par le client est connu dans la liste. Pour récupérer les valeurs entrées par l'utilisateur nous utilisons la partie v-model de Vue.JS qui permet de stocker les valeurs dans une variable temporaire pour que nous les envoyions au serveur avec une requête post, pour qu'il puisse vérifier l'utilisateur. Nous avons aussi une méthode qui permet d'envoyer un nouvel utilisateur au serveur avec une requête post encore et qui "push" toutes les informations envoyées dans la liste des users. Le code nous permet aussi de vérifier les entrées de l'utilisateur.

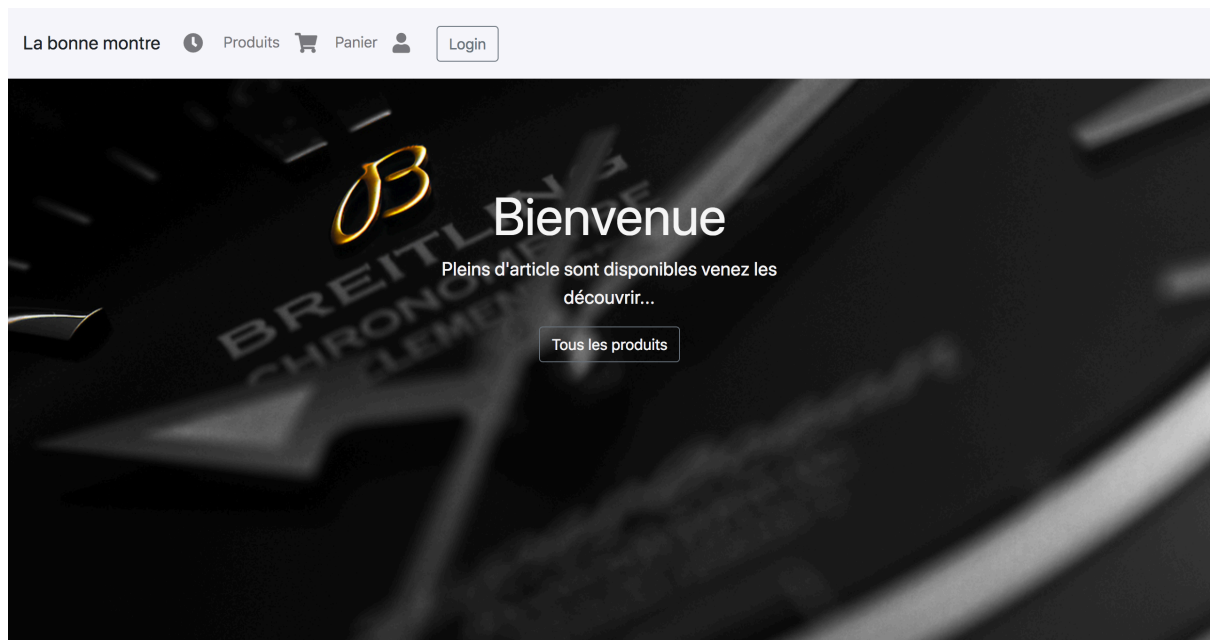
Pour afficher le panier nous utilisons une boucle v-for qui permet d'afficher tout les objets contenu dans le panier, de plus nous avons utilisé une condition qui permet d'afficher l'objet qui si sa quantité est supérieure à 1. Nous avons aussi deux boutons qui permettent d'ajouter ou de supprimer un objet rapidement. Nous affichons aussi le sous-total de chaque article et un total global. Nous avons un bouton de paiement fictif qui permet de finaliser la commande, il se débloquent uniquement si nous sommes connectés avec un compte.

Pour récupérer toutes les listes du serveur côté client nous effectuons des requêtes HTTP de type "GET". Le plus de notre projet est que nous stockons les listes sur un format "JSON", ce qui permet de sauvegarder toutes les informations des stocks et des utilisateurs si le serveur s'éteint, c'est-à-dire que nous enregistrons dans le fichier dès que nous avons une modification.

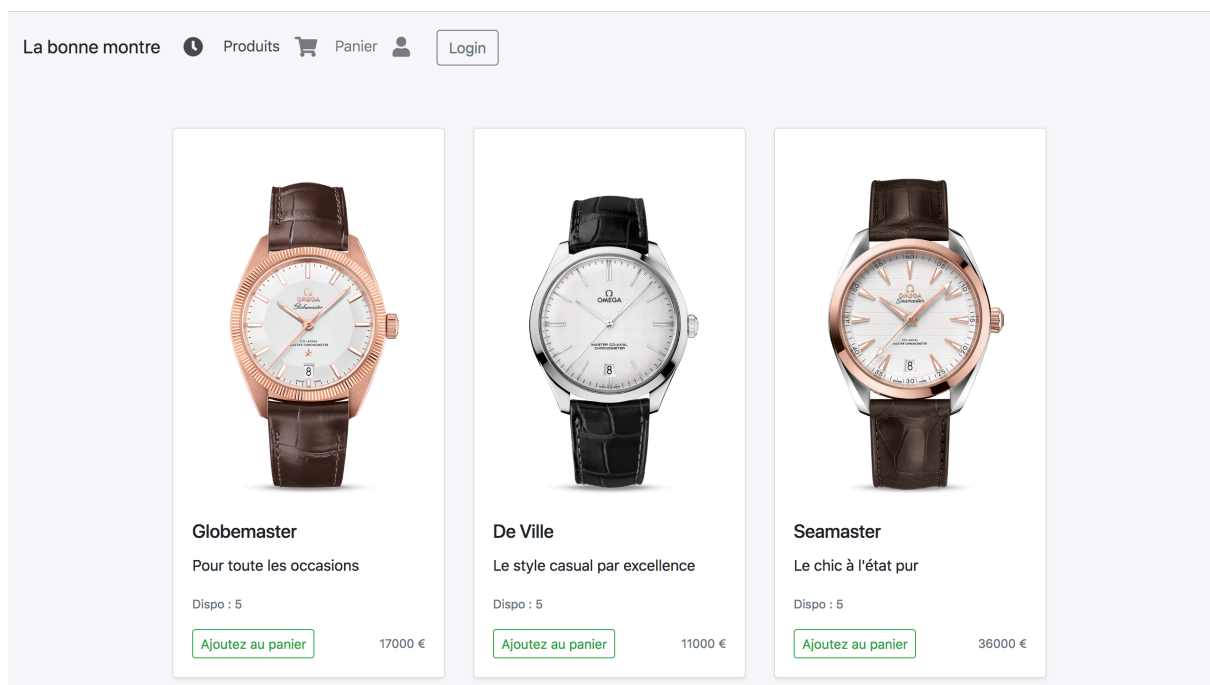
C . Déploiement

Tout au long du projet, nous avons utilisé git pour avoir plusieurs versions de notre code et pour avoir toujours un back-up au cas où nous avons un souci avec notre code sur notre machine. Pour le déploiement, nous avons utilisé Glitch en deux clics en important notre repo GitHub le site était en ligne.

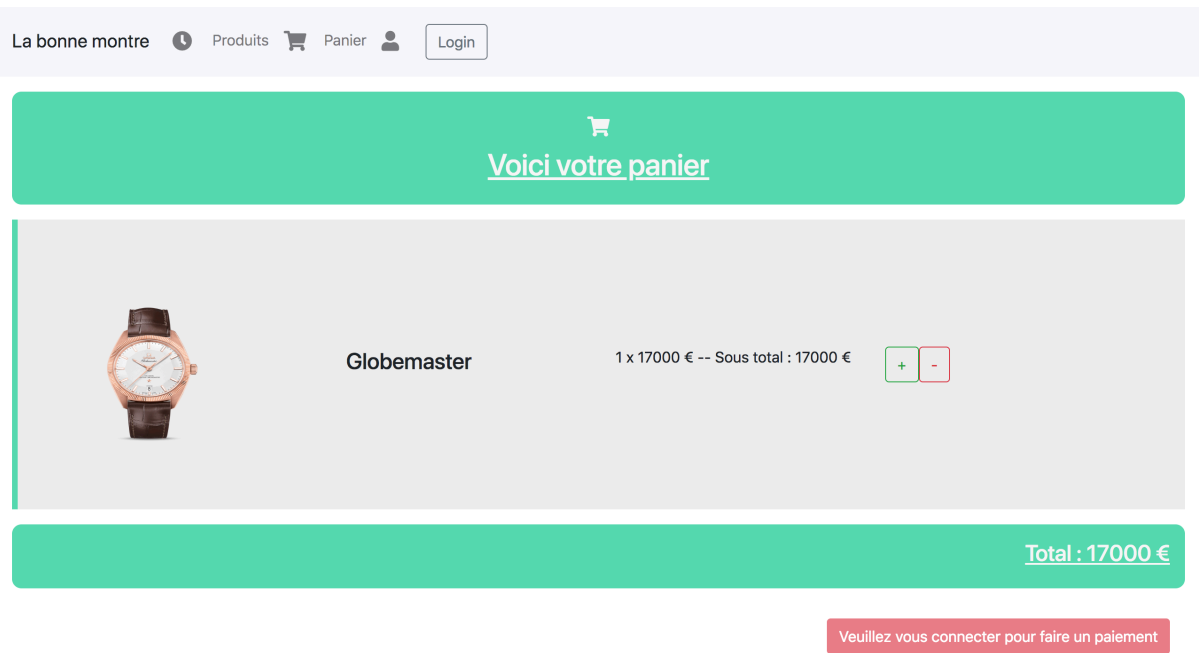
Voici donc quelques screens de notre application finale :



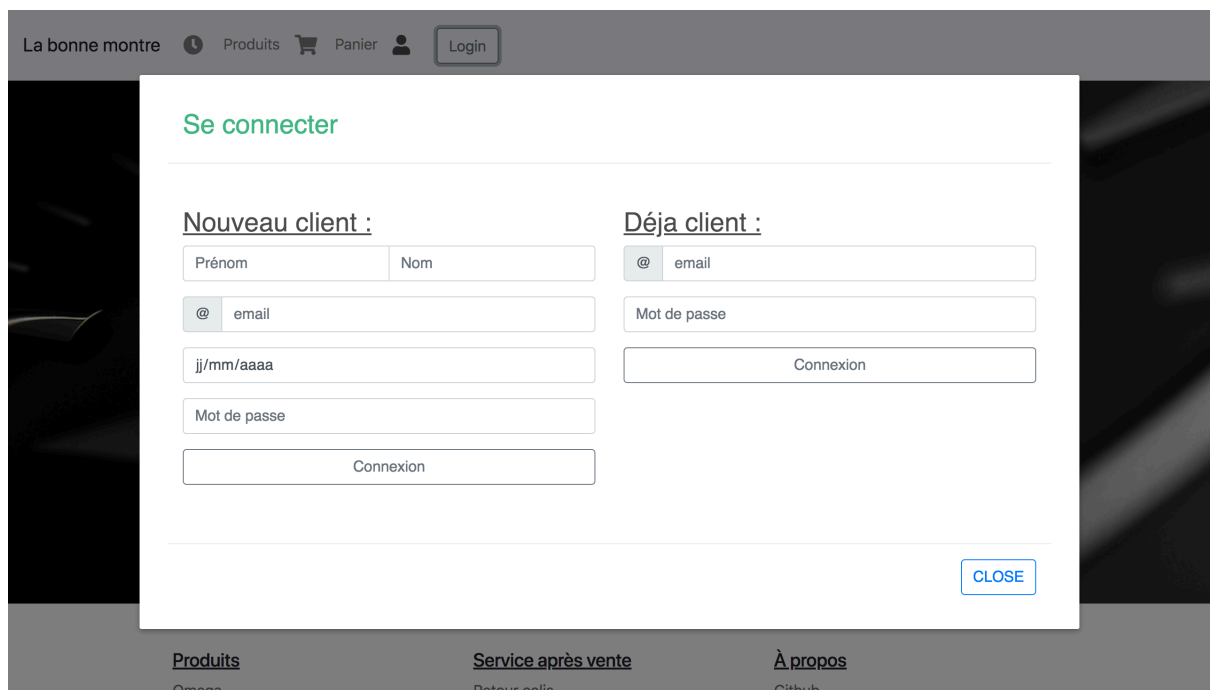
Notre page principale



Notre page de produits



Notre panier



Notre pop-up de connexion

IV. Difficultés rencontrées

A. Le framework

La première difficulté rencontrée est le framework VueJS, en effet ce framework était encore inconnu pour tout les individus du groupe. Nous nous sommes donc référés tout au long du projet à la documentation et en s'y reprenant à plusieurs fois nous sommes arrivés à comprendre le fonctionnement de ce framework. Au début du projet, nous pensions que ce framework était plus une contrainte qu'autre chose, mais avec le recul le projet maintenant terminé nous sommes bien contents de l'avoir utilisé, car il nous a simplifié la vie à plusieurs reprises.

B. Le NodeJS

Le NodeJS n'était pas pour nous inconnu nous avons déjà réalisé un PST avec l'an dernier, mais nous nous sommes rendu compte que nous l'avions mal utilisé et que les requêtes HTTP étaient plus faciles à utiliser que ce que nous pensions, le cours étant bien expliqué nous avons vite assimilé ce qu'il fallait faire et comment le faire.

V. Conclusion

Tout au long de ce projet, nous avons pu apprendre et mettre en place des nouvelles compétences. Nous avons pu explorer la partie serveur qui était jusqu'à lors inconnu pour nous. Nous avons pris conscience que le monde du web est vaste et que nous plusieurs moyens pour faire une chose et que c'est a nous de le choisir. Nous sommes donc très satisfait de ce projet et d'avoir mené a bien cette réalisation de A à Z.

VI. Annexes

Bibliographie :

- Github de l'énoncé : <https://github.com/Strift/esiea-web-programming>
- Github du projet : <https://github.com/JoffreyFrancesch/>

PROJET WEB 4A ESIEA

- Glitch du projet : <https://labonnemontre.glitch.me/>
- Etherpad : <https://etherpad.net/p/esiea-4a-web-fsi2>
- Guide VueJS : <https://vuejs.org/v2/guide/>
- Guide NodeJS : <https://nodejs.org/en/docs/>
- Livre sur le JS : <https://eloquentjavascript.net/>

Informations concernant le projet :

Un utilisateur est déjà enregistré dans l'application :

- email : franceschini@et.esiea.fr
- mot de passe : Joffrey1

Pour créer un utilisateur, voici comment remplir les champs :

- Prénom : 3 lettre minimum et sans caractères spéciaux ni chiffre
- Nom : 3 lettre minimum et sans caractères spéciaux ni chiffre
- Email : Doit ressembler à une adresse email
- Date : sélectionner la date directement
- Mot de passe : 8 caractères une majuscule et un chiffre