PLANE SHOP

Programmation Web

Kevin Berdah
Younes Ghenai
4SI2 – ESIEA 2018

Table des matières

I. Introduction

Pour ce projet web, nous avions reçu plusieurs requêtes et instructions de la part de notre quatrième client, nous devions développer un site viable de vente en ligne en HTML5, CSS3 et JavaScript. L'utilisation d'outils modernes tels que Node.js et de Vue.js a été une des contraintes du projet.

Une fois le site créé, le client devait avoir accès à une liste de produits, disposant chacun d'un stock prédéterminé et d'un panier où il aurait accès à ses articles. Une connexion lui serait requise dans le cas où il voudrait procéder au paiement.

Nous nous sommes donc orientés vers site de commercialisation d'avions de différentes marques et de différents domaines, allant du simple voyage à la guerre.

I. L'idée

Qu'est-ce que Plane Shop ?

Plane Shop est un site d'e-commerce dédié au secteur de l'aviation. Il est même possible, pour les plus passionnés d'entre vous, d'acheter divers avions selon vos envies. Sur notre site de vente d'avions en ligne, un choix divers et varié d'avions de marques s'offrent à tous, parmi eux :

- ✓ Airbus
- ✓ Boeing
- ✓ Dassault Aviation

Son design épuré et intuitif sera synonyme de confort, et passer du temps pour faire son shopping sur Plane Shop sera pour le client un moment convivial et agréable.

II. Le projet

A. La structure

Dès le commencement, nous avons réalisé une maquette sur ce à quoi notre site pourrait ressembler, nous nous sommes tout de suite mis d'accord pour cette forme :

- Une barre de navigation : fixée en haut de notre page.
- Un bloc unique central : affichage des produits.
- Un bas de page : centré en bas du site contenant quelques informations.

Nous avons rapidement avancé sur la partie minimale du contenu HTML, c'est-à-dire une barre de navigation, un bas de page ainsi qu'un bloc pour les articles, nous ne voulions pas nous attarder là-dessus sachant l'ampleur du travail que nous avions à fournir. De même pour le CSS nous l'avons terminé pratiquement en un jour, et en rajoutant quelques éléments au fur et à mesure que nous ajoutions du contenu HTML. Nous nous sommes ensuite renseignés d'une part sur le framework Vue.js puis sur Node.js lorsque le site était viable avant de mettre en place le serveur.

Concernant le Vue.js, il nous était alors inconnu au départ. Mais en lisant la documentation et en l'implémentant petit à petit, nous avons saisi la grande utilité de ce framework. En effet, une fois le Vue.js implémenté, nous évitions ainsi d'avoir à réaliser plusieurs fichiers HTML correspondant au nombre de pages du site. Toutes ces pages étaient donc factorisées dans un seul et même fichier HTML. Enfin, une fois l'environnement Node.js implémenté, notre serveur étant ainsi fonctionnel, nous avons pu intégrer une plateforme de connexion pour le client.

B. Le code

Pour ce qui est du détail de la partie technique, elle se décompose de la manière suivante. Nous disposons de deux tableaux d'objets, présents côté client et côté serveur (quatre au total). L'un concerne les produits mis en vente, avec son nom, son prix, son stock, son image ainsi qu'une description. L'autre sert de référence au panier, contenant également des objets parallèlement à la liste des produits.

Le bloc central présent sur la page de chaque marque d'avion est constitué de cadres. Ces cadres font référence à chacun des items de la liste des produits. Il y a autant de cadres qu'il n'y a de produits. Nous indiquons également au site de différencier et classer chaque produit dans une page différente en fonction de sa marque. Ainsi, les Airbus se retrouvent sur une même page, les Boeing sur une autre, et les Dassault sur une autre encore.

Pour chaque produit, le client peut incrémenter sa quantité dans le panier, en cliquant sur le bouton « Commander ». Le stock est alors mis à jour côté client et côté serveur. A chaque clic sur le bouton, une requête est envoyée au serveur pour vérifier que le stock est toujours positif. Lorsque le stock de l'article est épuisé, la page envoie une alerte au client lui indiquant que tel ou tel produit est en rupture.

Une fois ses achats terminés, le client souhaite valider son panier pour payer. Il est alors impératif pour lui de se connecter s'il a déjà un espace personnel, ou de s'inscrire en tant que nouveau client. Ainsi, à chaque tentative de connexion, le serveur vérifie dans sa liste d'utilisateurs (dans un fichier JSON) si le mail ou le mot de passe correspondent à un utilisateur connu, auquel cas il aura accès à un bouton « Valider et payer » dans le panier pour finaliser sa commande.

S'il s'agit de son premier achat sur Plane Shop, le client entre ses informations pour pouvoir s'inscrire. Le serveur va alors les récupérer et les écrire dans le fichier JSON, pour que le client soit reconnu à sa prochaine tentative de connexion. Une fois enregistré, le client est redirigé vers son panier pour valider son achat, puis une fois cela fait, il est encore une fois redirigé vers une page confirmant que son paiement a été accepté.

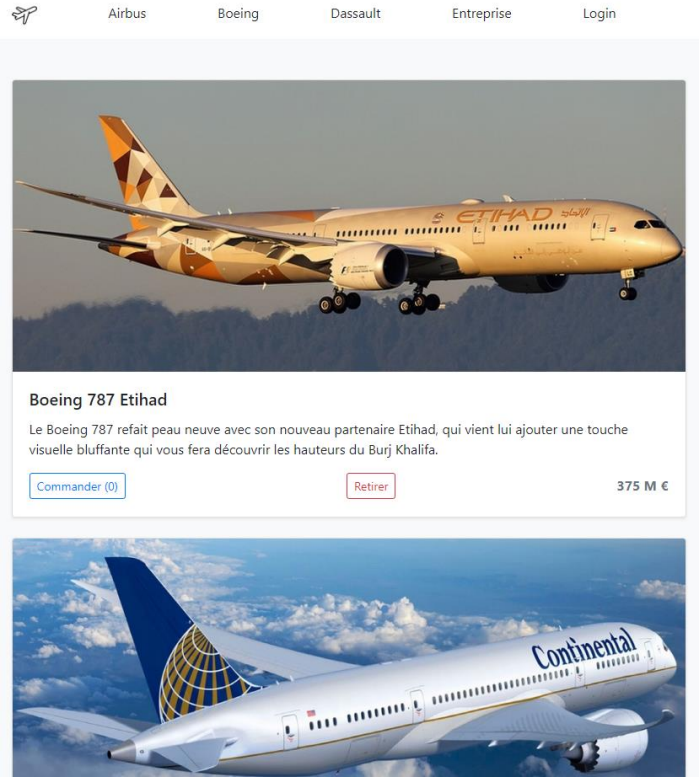
L'avantage d'écrire les utilisateurs dans un fichier JSON réside dans la sauvegarde des données dans le cas où notre serveur est éteint, chaque modification sera donc enregistrée.

C. Le déploiement

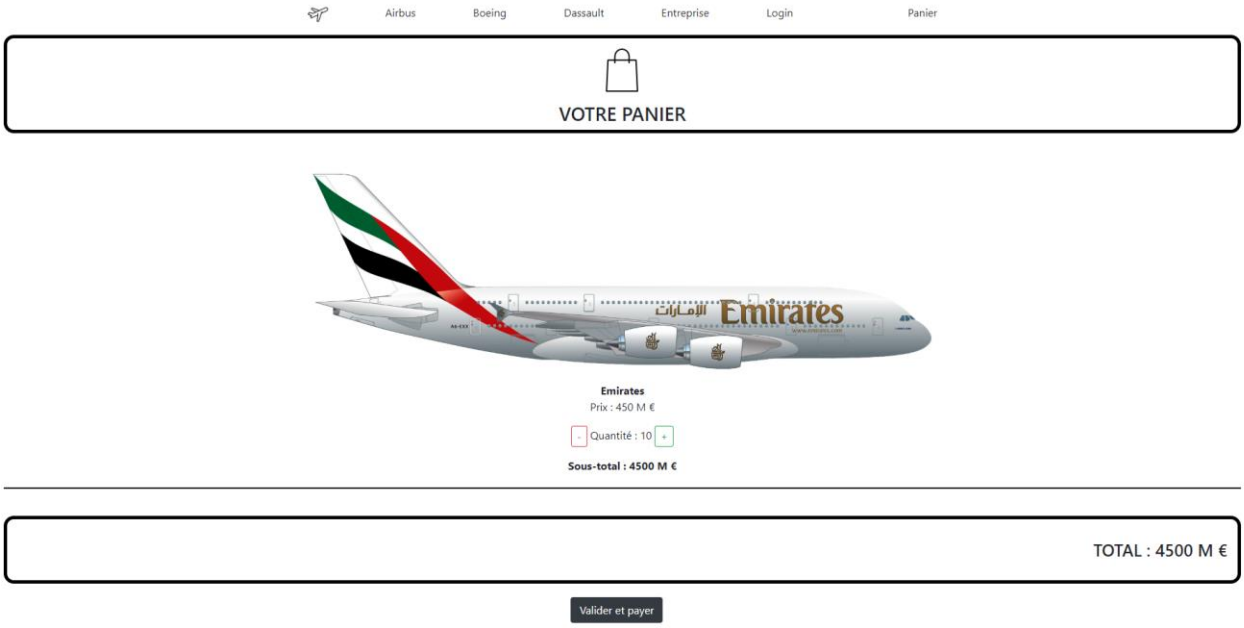
Au fur et à mesure que le projet se dessinait et se complexifiait, nous mettions régulièrement à jour notre projet sur Github à fin d'être synchronisés. Cette façon de faire nous était utile, puisque nous pouvions faire une copie pour la modifier, sans pour autant écraser la version antérieure. Ainsi, si un problème majeur devait survenir, telle qu'une perte des fichiers ou une modification trop conséquente (et donc pratiquement irréversible) du code, nous aurions toujours une version sûre.

Enfin, la publication de notre site sur l'hébergeur a été rapide et intuitive, et nous pouvons en voir le résultat final :

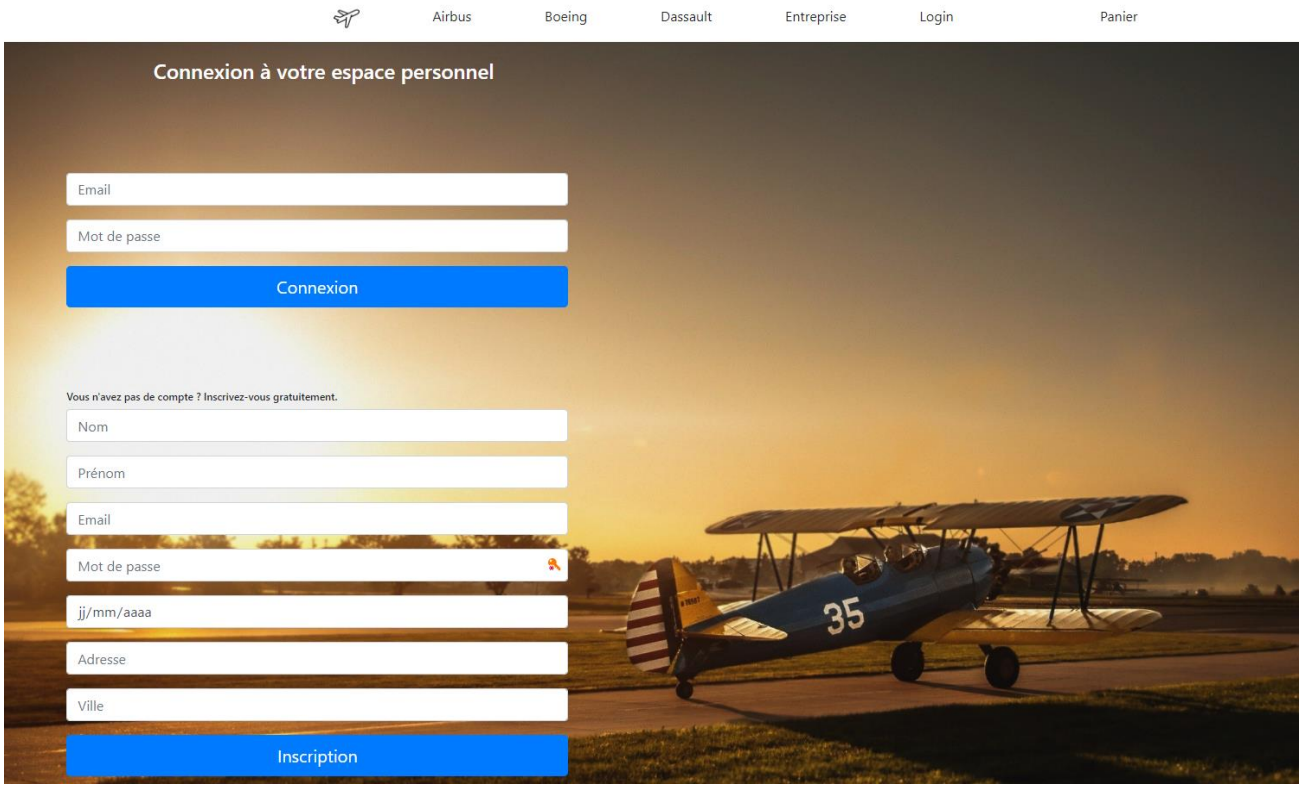
Page d'accueil Plane Shop



Page des produits



Panier de commandes



Connexion à l'espace personnel

III. Les difficultés

A. Vue.Js

Nous avions un peu plus d'une dizaine de pages html. Toutes les pages html étaient reliées entre elles via le menu de notre site. Mais la contrainte étant d'utiliser Vue.js, nous en avons profité pour découvrir ce framework qui nous a permis de réduire ce nombre de pages à une seule unité. En effet grâce à Vue.js, une seule page index.html réside.

En comparaison, sans l'utilisation de Vue.js, chaque page est chargée lorsque l'on clique dessus, alors qu'avec l'utilisation de Vue.js, lors du chargement de la page d'accueil, cela prend un peu plus de temps, mais il n'y a plus de chargement de pages durant toute la durée de notre navigation.

L'apprentissage de Vue.js a été assez long et compliqué au départ. En effet nous ne savions pas par où commencer, nous avons suivi de nombreux tutoriels, qui souvent ne menaient pas à la fonctionnalité souhaitée, mais à force de persévérance nous avons réussi. Avec un peu de recul ce n'est pas la partie la plus compliquée, notamment avec le Node.Js que nous devions assimiler.

B. Node.Js

Concernant le Node.js, nous appréhendions beaucoup son implémentation, ce qui fut le cas. Nous avons rencontré pas mal de difficultés quant au panier, comme lorsque nous ajoutions des produits au panier à partir de la page produit, le panier s'incrémentait bien, le stock décrémentait bien, mais dès lors qu'on modifiait la quantité voulue à partir du panier, le stock revenait à son origine.
Mais après plusieurs jours d'apprentissage, de lecture, de tutoriels, de vidéos sur Youtube, et l'aide de l'encadrant, nous sommes parvenus à mettre notre projet sur un serveur local.

Enfin, nous avons rencontré une ultime difficulté quant au login. Lorsque le client s'inscrit, il est bien connecté automatiquement et il peut valider son panier. Mais lorsqu'il tente de se reconnecter, la fonction login ne lit pas le fichier JSON. Nous avons vérifié l'écriture des fonctions ainsi que le lien entre le côté client et le côté serveur, mais nous n'avons pas réussi à déceler l'erreur.

IV. Conclusion

Finalement, ce projet nous a permis de faire une faire une réelle remise en question de ce que nous pensions savoir de la programmation Web, et des innombrables possibilités d'applications que nous pouvions aborder.

Nous avions eu une certaine liberté quant thème de notre projet, nous avons donc pu mettre en place une idée qui nous plaisait bien, ce qui a littéralement exacerbé notre motivation. Nous nous sommes démenés pour parvenir à un site viable et confortable tant visuellement que techniquement, nous n'en sommes pas peu fiers.

V. Annexes

Informations complémentaires :

Un client a déjà passé commande, il a accepté de nous laisser ses identifiants pour faciliter la connexion :

- ✓ Email : younes@gmail.com
- ✓ Mot de passe : younes

Bibliographie / Webographie :

L'énoncé du projet : https://github.com/Strift/esiea-web-programming
Notre projet : https://github.com/Younes-ghenai/Younes_Kevin_PlaneShop
URL de notre site : https://plane-shop.glitch.me

- ✓ URL du site hébergeur : https://glitch.com
- ✓ Liens utiles pour la programmation : https://etherpad.net/p/esiea-4a-web-fsi2
- ✓ Guide VueJS : https://vuejs.org/v2/guide
- ✓ Guide NodeJS : https://nodejs.org/en/docs
- ✓ Exemples Bootstraps : https://getbootstrap.com/docs/4.1/examples/