



PROGRAMMATION WEB

Boutique de Chaussure

ROLLIER David / LO CASALE Lorenzo

Cahier des charges

Learning outcomes

- Building a website with modern tooling and understanding of front-end and back-end concepts
- Front-end technologies: HTML5, CSS3 (Bootstrap - optional), Javascript (Vue.js)
- Back-end technologies: Node.js (Express)

Project requirements

- Web interface built with HTML, CSS and Vue.js
- Web server built with Node.js and Express
- User authentication (and registration if your application requires it)
- [CRUD](#) interactions with the server
- Data persistence is optional: in-memory is good enough, but you can try to write to the file system (no database!)
- Deployed on [Google Cloud Platform](#) or [Glitch](#)

You will not need and should not use any other libraries. This means no jQuery! jQuery is a Bootstrap dependency. So, if you are using Bootstrap you are free to include jQuery to make it work, but you should not call it by yourself.

Mr. CHEREL groups (FSI1, FSI2, CFA)

A minima (d'autres dossiers peuvent être présents) ==> mais pas node_modules, se renseigner sur .gitignore

Report requirements

Le rapport nous permet à nous enseignants de comprendre votre logique de pensée lors de la réalisation de votre projet. Il nous permet également de vérifier que vous avez pris un certain recul sur la tâche technique et que vous comprenez les tenants et les aboutissants d'une gestion de projet réussie.

Les éléments essentiels (non exhaustifs) du rapport sont :

- Description du projet, en quoi il répond aux contraintes demandées
- Étapes de conception, réalisation, déploiement
- Quelles difficultés ont été rencontrées

En aucun cas vous ne devez nous copier/coller du code dans votre rapport, ce n'est pas l'objet d'un tel support.

Submission process

- Le rendu doit être effectué sur un dépôt github *public*
- Le nom des membres du projet doit être écrit dans un fichier MEMBERS.txt à la racine où chaque ligne respecte le format NOM Prénom (exemple ci-dessous)

```
DUPONT Bob
```

```
DUPOND Léo
```

- Le rapport au **format PDF** doit être à la racine de votre dépôt avec le nom REPORT.pdf
- Vous devez envoyer l'url de votre projet (sous la forme <https://github.com/username/repository>, pas de .git ni de gist !) ainsi que l'url de votre application déployée à louis.cherel@platypus.academy, avec l'objet suivant: ESIEA WEB 4A FSI1/FSI2/INT/CFA
 - Ex : l'objet pour un étudiant du groupe FSI1: "ESIEA WEB 4A FSI1"
- L'email contenant l'URL du dépôt doit être envoyé *au plus tard le vendredi 7 septembre à 20h* pour FSI1 et INT, *le vendredi 14 septembre à 20h* pour FSI2, et *le lundi 19 novembre à 23h59* pour les 4A-CFA

NB : le contenu des dépôts va être téléchargé et analysé automatiquement à l'heure exacte de la deadline + 1 minute. Ce téléchargement n'aura lieu qu'une seule fois et un oubli de votre part serait donc sanctionné par un 0.

NB2 : tout manquement, même "léger" à l'une des règles opère un malus de 3 points sur la note finale, de manière cumulative.

A propos du projet

Le projet a été réalisé avec le framework JavaScript Vue js et fonctionne sur un serveur node js.

HTML, CSS et Vue.js

Ce projet dispose d'une seule page HTML (index.html) dont le style est géré dans les portions de style (CSS) situées dans nos différents composants Vue.js.

Routage et contenu variable

Nous avons utilisé le module « **vue-router** » pour effectuer le routage de nos « pseudo pages ». En effet, le site web ne dispose que d'une seule page HTML avec un contenu variable en fonction de l'url. Le paramétrage du routeur se fait dans un tableau où chaque item du tableau correspond à une route.

User Authentication

Pour l'authentification du site nous avons utilisé **Firebase**. L'utilisation de Firebase pour ce projet n'est pas forcément pertinente mais l'occasion s'est présentée de découvrir l'outil et nous pensons que ces connaissances ne sont pas négligeables pour la suite de notre cursus.

Pour notre application Web nous avons fait le choix arbitraire d'imposer une authentification obligatoire pour les utilisateurs.

CRUD application

Notre application de vente de chaussure dispose d'une boutique et d'une zone panier, afin de permettre à nos utilisateurs d'ajouter des éléments dans leur panier et par la suite leur laisser la possibilité de supprimer des articles de leur panier.

Puisque nous imposons une authentification nous avons aussi ajouté un formulaire d'inscription. Il est donc possible de créer des utilisateurs, ajouter/supprimer des articles et donc modifier le contenu de notre panier.

Persistence et gestion des données

Pour la persistance (non obligatoire) des données nous utilisons Realtime Database pour la « Boutique » et Firestore Cloud pour les données « User » et « Panier ». L'application accède à toutes ces données grâce à notre fichier firebaseConfig.js configuré sur notre projet Firebase qui met à disposition nos variables de référence sur les données. Afin de partager les données entre les différents composants de notre application, nous avons utilisé le module « **Vuex** » dans un fichier store.js qui s'occupe de mettre à jour nos variables globales (userProfile, currentUser, userPanier) lorsque des actions sont effectuées dessus (ajout, suppression et modification) et de rendre ces données globales facilement accessibles dans l'application.

Conception

Les difficultés rencontrées :

La plus grande partie du travail fut de s'auto-former sur le framework vue.js afin de découvrir les diverses fonctionnalités qu'il met à disposition (même s'il en reste encore beaucoup à découvrir). Grâce à ces recherches approfondies nous savons désormais faire une pagination (rubrique La Boutique), configurer un store (fichier store.js) et partager/exploiter ses données, faire du routage (module vue-router), faire de l'affichage conditionnel, importer des icons fontawesome, etc.

Pour le développement du projet nous avons :

- Analysé les éléments static de notre site web (Header, sections, etc.)
- Réfléchi à une manière dynamique d'afficher nos contenus afin d'éviter la redondance de code HTML dans le site.
- Compartimenté notre code en différents composants avec des fonctionnalités propre à eux.
- Développé en « first-destkop » car selon nous c'est le média où nous appréhendons le plus de travail à fournir en Javascript mais toujours en pensant responsive pour que derrière nous ayons le moins de travail possible sur cette partie.

Nous avons aussi rencontré des problèmes avec le déploiement de l'application sur Glitch. Visiblement le déploiement d'un projet avec WebPack est plus difficile que nous le pensions.

Déploiement

Vous trouverez donc sur GitHub deux repository :

- Le repository standard avec l'architecture du projet Web, le MEMBERS.TXT et le REPORT.PDF : <https://github.com/DouuZzer/MaBoutiqueChaussure>
- Le repository utilisé pour le déploiement sur Glitch, il contient le contenu du dossier « dist » généré après un **`npm run build`**. : <https://github.com/DouuZzer/MaBoutiqueDistDirectory>

Vous pourrez accéder au site hébergé sur Glitch via l'url suivant : <https://douuzzer-maboutiquedistdirectory.glitch.me/>

Nous avons uniquement développé en compatibilité Google Chrome, nous vous remercions de bien vouloir ouvrir le site web sur Chrome.

Prérequis pour vous connecter

Pour effectuer les tests nous vous avons créé au préalable un compte :

Login : musinux@admin.fr

Password : 123456

Où si vous préférez, vous pouvez directement créer votre propre compte.