Dossier de conception

Projet web

# Présentation du projet

## Objectifs :

|  |  |
| --- | --- |
| **Contexte :** | Le bureau des élèves du CESI nous demandent de leur réaliser un site web pour gérer la vente de produits dérivés ainsi que la gestion des évènements organisés par ce dernier. |
| **L’équipe :** | * *Chef de projet* : Tristan BOULESTEIX * *Développeurs web*: Charlotte MADROUX, Maxime MAITRE et Tristan BOULESTEIX |
| **Utilisateurs cibles :** | Les membres du BDE, du CESI ainsi que tous les étudiants affiliés. |
| **Type de projet :** | Site web & API |
| **Date de mise en production :** | 30/01/2019 |

## Planification :

Pour planifier notre travail, nous avons de réaliser un diagramme de Gantt ainsi qu’un WBS. Vous les trouverez joins à ce dossier sous le nom « wbs.pdf » et « MsProject.pdf ».

Nous nous sommes réparti les tâches en prenant en compte les points forts de chacun. La création du site était organisée ainsi :

* **Maxime Maitre :** Réalisation des différentes vues en Responsive Design et de tous les éléments Front-End dont le code JavaScript des pages dynamiques.
* **Charlotte Madroux :** Réalisation de l’API RESTful sous Node.js et des procédures stockées des bases de données. Réalisation du template en Blade et de la mise en place d’Ajax.
* **Tristan BOULESTEIX :** Réalisation du contrôleur et du modèle (Back-end) sous Laravel ainsi que la mise en place des sessions et des migrations.

# Analyse et conception

## Bases de données

Nous devons stocker les données du site sur des bases de données. Il fallait absolument que les données relatives aux utilisateurs soient disponibles sur un serveur national tandis que les autres données devaient être spécifiques à un centre donné.

Nous avons donc choisi de faire deux bases de données nommées respectivement « projet-web » et « projet-web-users ».

Nous avons réfléchi ensemble à la structure de nos bases de données. Vous pouvez retrouver plus d’informations dans les fichiers « MCD.pdf » et « MLD.pdf » ci-joint.

## Wireframe

Pour l’interface utilisateurs, nous devons nous inspirer de la chartre graphique de l’école d’ingénieurs CESI. Nous avons donc tenté de faire une interface graphique similaire. Pour visualiser le site dans les différentes situations, nous avons choisi de réaliser un wireframe (voir « wireframe.pdf »).

## Chartre graphique

Nous avons réutilisé la chartre graphique du site de l’école d’ingénieurs CESI comme demandé. Nous avons repris à peu près les mêmes éléments (footer, header, menu latéral…) avec les mêmes couleurs. Ainsi, nous avons des menus sur fond noir et un contenu sur fond blanc. Le texte des menus au survol devient jaunâtre. Nous avons aussi repris le même style pour les boutons. Nous avons récupéré la police d’écriture du site pour une meilleure cohérence.

Pour conserver une cohérence visuelle entre les différentes pages, nous avons utilisé une feuille de style pour tous les éléments présents globalement sur le site.

## Choix techniques

**Laravel Framework :**

Nous avons choisi d’utiliser le Framework Laravel car, comme tous les Framework, il permet tout d’abord de mieux organiser son code. De plus, il suit l’architecture MVC (Model, View, Controller) qui est très répandue. Il intègre des bibliothèques telles que des ORM, des outils de gestion de sessions… On peut en installer d’autres facilement grâce à un gestionnaire de package nommé Composer.

**MySQL :**

Nous avons choisi MySQL car il est demandé dans le cahier des charges du BDE.

**Apache Server :**

Nous avons choisi ce serveur car il est répandu, stable et simple à configurer. De plus, nous savons bien le manipuler.

### Node.js & Express

Nous avons décidé d’utiliser Node.js car nous devions réaliser une API sur cet environnement d’exécution JavaScript selon le cahier des charges du BDE. Nous avons choisi d’y implémenter un Framework Express car c’est rapide c’est facile à mettre en place et c’est modulable ce qui permet d’implémenter des API rapidement. Les API ont été testé avec Postman.

**GitHub**

Nous avons décidé d’utiliser GitHub car il s’agit d’un gestionnaire de version efficace qui nous permet de travailler à plusieurs sur le code tout en restant efficace. Nous avons respecté le Git Flow qui améliore notre organisation et la gestion du code.

**HTML5, CSS3, JavaScript, PHP, SQL**

Les deux premiers sont indispensables pour réaliser un site en Responsive Design et optimisé.

Nous avons aussi utilisé JavaScript qui nous permet de rendre le site dynamique, d’utiliser Ajax facilement côté client et d’avoir un serveur d’API sous Node.js.

Le langage PHP a été utilisé pour le Back-End sous Laravel. C’est un langage stable et dont fiabilité est reconnue.

SQL est le langage utilisé par MySQL qui est facile à faire communiquer avec PHP.

**Nous avons décidé de ne pas valider les formulaires en AJAX :**

Nous avons utilisé du PHP pour valider les formulaires car JavaScript est facilement manipulable et donc peu sécurisé contrairement à PHP qui les gère coté serveur.

**Utilisation des Media Queries :**

Afin de créer des views responsive nous avons utilisé des média Queries pour adapter notre site à plusieurs plateformes. Ainsi il est parfaitement fonctionnel sur téléphone, tablette et laptop.

**Menu Hamburger :**

Ces menus ont été utilisé lors de la mise en place des média Queries pour téléphone. Cela a permis un meilleur confort de navigation et cela a empêché la surcharge d’informations à l’écran.

**Utilisation de BLADE :**

Il permet d’avoir un site sécurisé et évolutif et manipuler le PHP plus facilement.

**Pourquoi notre API n’est pas full REST :**

La consigne nous demandait d’utiliser MySQL, or pour une API REST full il faudrait utiliser Mangoose-DB qui est une base de données orientée ficher. Cependant l’API renvoie les JSON.

**Nous avons utilisé un Template BLADE :**

Afin de ne pas à avoir à réécrire le ccs principal du cite web à chaque fois.

**Nous avons utilisé Eloquent :**

Pour éviter de réécrire du code SQL ou des procédure stockées et de sécuriser les requêtes. Il permet de manipuler données comme des objets.

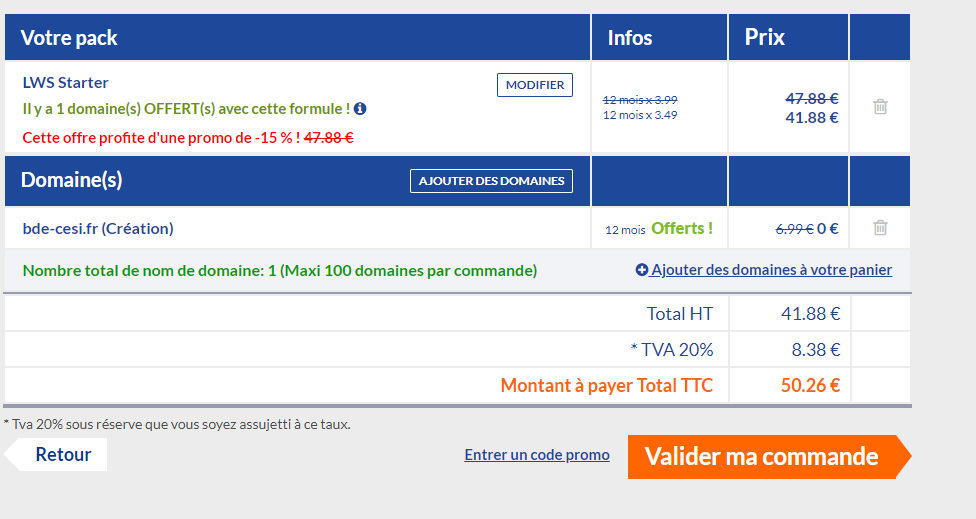
**Nous avons utilisé plusieurs navigateurs :**

Afin de s’assurer du bon fonctionnement du site web sur des navigateurs différents et du rendu de celui-ci.

**Nous avons utilisé GitFlow :**

Pour avoir un contrôle des versions et permettre à chacun de travailler sur sa branche et avoir une branche finale complète.

# Hébergement

Nous avons trouvé une offre chez LWS comprenant une promotion -15% ainsi qu’un nom de domaine offert. L’hébergement coûte ainsi 41.88€/ans hors taxes.

Sinon, nous avons aussi cherché le coût d’un hébergement chez OVH. Nous avons obtenu le prix de 35.88€. Cela comprend aussi le nom de domaine pendant un an ainsi que l’hébergement. OVH propose aussi de manière optionnelle un certificat SSL sur notre site pour 99.99€/ans. L’avantage d’OVH est qu’il offre un contrat modulable.

