Rapport du projet Programmation Web 2

Gaydamakha Mikhail

Dans ce projet j'ai mis en oeuvre de framework - un pour backend et un pour frontend - *Laravel/5.7* et *React.js* respectivement. L'architecture du projet est basée évidemment sur le MVC pattern.

En ce qui concerne du backend, il utilise la base de données *MySql* et le serveur *Apache2*. Tous les deux sont installe localement sur la machine de Microsoft louée pour les mêmes buts.

Pour remplir la base de données (pas assez bien vu de manque du temps) j'ai utilisé les migrations et seeding d'artisan, qui vient avec *Laravel.* Création des modèles est confié à *Eloquent*, qui est fourni aussi par Laravel. Donc tout ce que j'avais besoin pour créer model Article par exemple, c'est lancer

```
$ php artisan make:model Article
```

et après mettre une variable locale

```
$table='articles'
```

pour dire à *Eloquent,* avec quel table est liée notre model. Donc pour les requêtes *SQL* j'ai utilisé des méthodes réalisé pars *Eloquent* comme

```
$article = Article::select('id','name')->where('name','like','SG-868')->get();
```

pour obtenir les articles correspondants. Du coup, toutes les modèles se trouve dans le répertoire ./app/ .

Les contrôleurs se trouvent dans le répertoire ./app/Http/Controllers et ce sont eux qui font toute les requêtes qu'on a besoin.

Le seul view "index.php" se trouve dans le répertoire ./public qui est seul site disponible pour l'utilisateur du site.

Il est créé par la commande “npm run build” du script ./build.sh . L’avantage principale du *React.js* c’est ce qu’il est très dynamique - il ne fait pas mis-à-jour des composants qui n'étaient pas modifiés, donc on a pas besoin de recharger la page pour obtenir un component. Pour cela *React.js* utilise le stockage *Redux.*

Tous les fichiers source se trouve dans le répertoire ./front/src/components, et la-dedans vous pouvez trouver les fichiers .jsx correspondants à chaque components de la page.

Par ailleurs, le format des données échange entre serveur et client est JSON. Pour ca j’ai créé un middleware qui traite les données envoyées par le client, et ca permet de faire la réception des messages plus sécurisée.

En ce qui concerne des styles, j’ai utilisé les solutions prête créés par google - Material Design.

Par contre, j’ai rencontré des problèmes avec session de *Laravel* (la session se réinitialise après chaque requête), donc j’ai pas réussi d’ajouter les articles dans le panier, mais toutes les méthodes qu’on a besoin pour réaliser l’ajout dans le panier et pour les montrer dans sa page se trouve dans ./app/Http/CartController.php .

Pour montrer que toute est pret pour ca et il me manque que la fonctionnalité de **session**, j’ai effectué deux requêtes sur la page de la panier pour les visualiser.