Projet ISIWeb

Clément RENIERS Simon PRIBYLSKI

Projet ISIWeb

Organisation du binôme

L'organisation s'est développé selon les compétences de chacun. Simon, plus adepte du back-end, a pris en charge la partie serveur et la base de données (PHP, MySQL). Clément, plus orienté front-end, a pris en charge la partie client (Twig). Les interactions entre les deux parties ont dû en revanche être nombreuses pour que le projet soit cohérent.

Difficultés rencontrées

Côté PHP, un temps important à été passé à écrire le code des classes du modèle avec leur documentation. Le fichier `Test.php` a permis de réperer un nombre important d'erreurs dans le modèle et des intéractions avec la base de données. Les requêtes SQL sont générées afin d'éviter les répétitions (et donc les eventuelles erreurs), ce qui a également permis de gagner du temps. Toutes les opérations CRUD ont donc été implémenté assez facilement. De plus, les mots de passe dans la base de données sont hashés afin de garantir une meilleur sécurité du site. Par ailleurs, les fonctions qui dépendent de la session en cours comme `setNextToast` ou encore l'authentification ont été un peu pénible à déboguer/tester car il faut démarrer une nouvelle session à chaque fois. Pour finir, la documentation a pris pas mal de temps à être rédigé à la fin (même si théoeriquement elle devrait l'être avant l'implémentation même des fonctions). Côté front-end, la gestion des `extend` de Twig a été un peu compliquée à mettre en place au début, mais après quelques recherches, tout s'est bien passé. L'utilisation de certaines des classes de Bootstrap5 n'était pas non plus évidente, mais la documentation a bien aidé. Enfin, la gestion des types en base (notamment les dates) a été un peu compliquée, donc la base a été modifiée pour simplifier l'interaction entre PHP et MySQL.

Architecture de l'application

Le fonctionnement de l'application est détaillé dans la documentation, générée par (à lancer à la racine de l'application) :

```
doxygen && firefox doc/html/index.html
```

(firefox peut être remplacé par le navigateur de votre choix (sous Linux, google-chrome, chromium, etc...))

La seule modification apportée au projet initial est la modification de la recherche dans les différents onglets, performée par la bibliothèque DataTables.

Fonctionnement de l'application

Le fichier `assets/config.ini` peut (et doit) être modifié selon les besoins et l'architecture sur laquelle va être déployé le site réel. Les logs de toutes les requêtes SQL est disponible dans le fichier `assets/logs.txt` par défaut. Cela peut être modifié dans le fichier de configuration.

L'application peut être déployée à l'aide de docker. Plus d'informations dans `docker/README.md`.

> [!WARNING] Pour utiliser l'application, il est inutile d'utiliser un serveur php tel que xampp. En effet, pour augmenter la portabilité de l'application, des containers Docker ont été mis en place.

Prérequis

Pour utiliser l'application, il est nécessaire d'avoir installé Docker et Docker-compose.

Lancement de l'application

Pour lancer l'application, il suffit de se placer à la racine du projet et de lancer la commande suivante :

```
cd docker
docker-compose up
```

L'application utilisera la base de données du projet (disponible dans le dossier `docker/images`).