****

**Développeur multimédia**

**Projet Back – « ConfPHP »**

Durée : 6h

**Contexte**

Il vous est demande de réaliser un site en PHP/MySQL administrable. Le site propose une liste des conférences passées ou futures autour de PHP. Il est porté par l'AFUP (l'association française des utilisateurs de PHP).

Vous avez à réaliser les pages suivantes cote « front »: une page d'accueil, listant les conférences à venir, une page à « propos » de présentation de l'association AFUP ainsi qu'une page contact et mentions légales.

**Contraintes techniques**

• Vous utiliserez **Laravel5** pour réaliser ce projet.

• Vous utiliserez les controllers RESTFul du framework.

• Vous utiliserez **MySQL** comme SGDB.

• Vous créerez un « service provider » MyHtml, cette classe sera votre Helper pour les vues.

◦ Vous n'implémenterez que les méthodes suivantes dans ce service : **radio**, **link** et **thumb :**

▪ MyHtml::radio($name, []) ← input radio

▪ MyHtml::link( $title, $href) ← a.href

▪ MyHtlml::thumb($class, $href) ← div.thmb img.class

• Pour la partie Front vous partirez de la css et du modèle HTML proposé dans le brief (voir le dossier doc). Vous pouvez cependant utiliser vos propres CSS et HTML, dans ce cas vous devez vous inspirer du modèle proposé dans le brief.

• Vous utiliserez boilerplate pour rendre le site responsive.

• Vous utiliserez **Gulp** et **Sass** pour la partie intégration, tous les fichiers css et JS seront mimifiés.

• Vous organiserez les ressources de l'application comme propose **à la fin du brief.**

• Vous ferez un script **install.sh** bash de déploiement de l'application.

• L'application sera protégée des attaques XSS et CSRF.

• Vous mettrez en place des « slugs » administrables pour le référencement.

**Fichiers transmis**

• Le brief « confPHP » au format pdf

• Un dossier doc dans lequel se trouve les dossiers et fichiers suivants :

◦ data.txt (les données d'exemple a formater pour la base de données)

◦ index.html

◦ assets avec des images et les css style.css

**Livrables**

Vous devez rendre une application, sans les vendors, avec un script « install.sh » à la racine du projet permettant :

• De créer la base de données avec un utilisateur spécifique pour cette base.

• D'installer le framework Laravel5.

• De créer les migrations avec les seeders.

• De lancer automatiquement un server a l'adresse suivante : <http://localhost:8000>

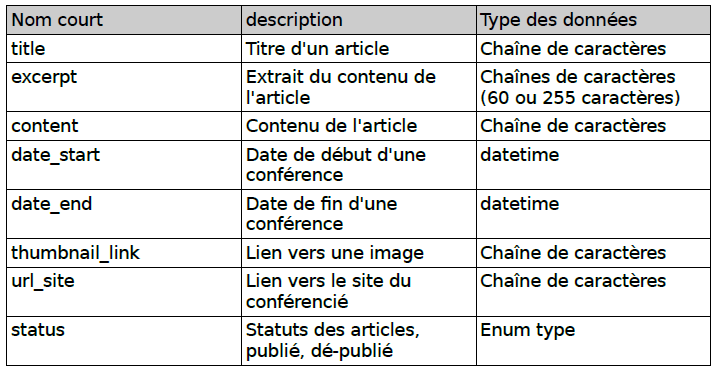
**Directives**

**1. Schéma des données**

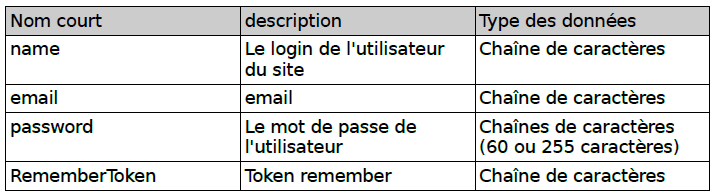
> **Créez une base de données « conference », puis les tables suivantes dans cette base de données :**

**Table posts ~ aux conférences dans le site**

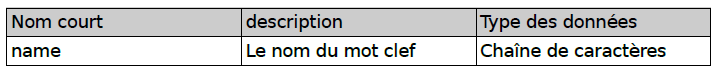
*Pour la suppression des posts vous utiliserez une suppression douce.*



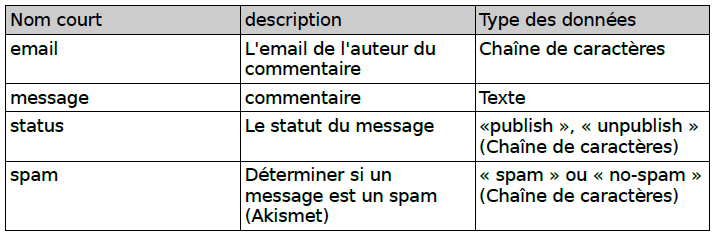
**Table users**



**Table tags**



**Table comments**



**Les relations entres les tables sont les suivantes :**

• Un user peut avoir de 0 a N post(s).

• Un post peut etre associee a 0 ou N tag(s).

• Un tag peut etre associe a 0 ou N post(s).

• Un post peut avoir de 0 a N comment(s).

**2. Page d'accueil**

Le menu principal possède les liens cliquables : accueil, a propos et la page contact.

On affichera toutes les conférences dont le statut est « publish » en base.

Les conférences sont ordonnées par ordre décroissant de date de début de conférence.

Chaque conférence sera composée de :

• titre de la conférence cliquable (affichage du contenu de l'article clique)

• un texte court

• le lien vers le site du conférencier si il existe

• une image de la conférence cliquable (affichage du contenu de l'article clique)

• les mots clefs de la conférence

• la date de début et de fin de la conférence

• le nombre de commentaire lie a chaque

(voir ci-dessous)

**Page d'accueil**

****

**3. Page de présentation d'une conférence**

Une fois que l'on a cliqué sur une conférence en page d'accueil, on affiche les détails suivants :

• titre de la conférence

• texte long de la conférence

• lien vers le site web de la conférence

• mot clefs de la conférence

• un formulaire pour laisser un commentaire

• suivi des commentaires, si il(s) existe(nt)



**4. Page contact**

Une fois que l'on a clique sur contact, on affiche le formulaire suivant :

• un champ email

• un champ pour calculer une somme aleatoire (anti-bot…)

• un commentaire

• un bouton submit



Remarque : vous devez configurer Laravel5 pour tester l'envoi de mail. Une solution consiste a configurer Laravel5 en « **Local Development »**, [documentation en ligne](https://laravel.com/docs/5.1/mail#sending-mail).

**5. Accès à l'administration de l'application**

• Page de login

Mettre en place une page de connexion à l'espace prive du site (administration). Il faudra gérer les erreurs de saisies dans le formulaire cote client et serveur (FormRequest). Si la personne est correctement identifiée avec son login et mot de passe, elle sera redirigée vers le « dashboard » de l'administration.

• **Dashboard de l'administration**

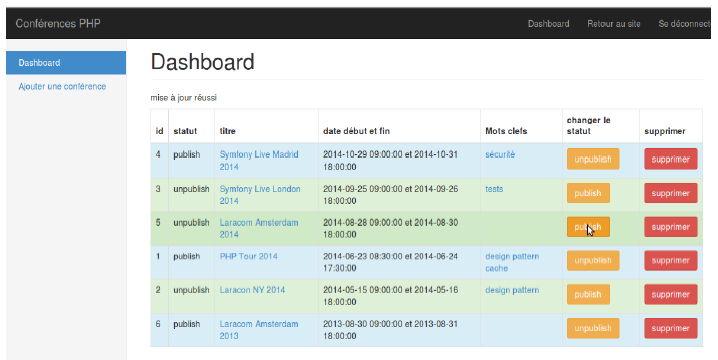
*On vous conseille d'utiliser Twitter Bootstrap pour cette partie.*

Une fois dans l'administration des conférences, on doit pouvoir modifier, a partir du tableau de bord, le statut d'une conférence pour la de-publier/publier et la supprimer.

Toutes les actions d'administration sur les conférences doivent s'accompagner d'un message de succès ou d'échec.

Pour la suppression, afficher un pop-in permettant de valider ou non la suppression d'une conférence.

On rappelle que la suppression est une suppression douce.



• CRUD conférences

Vous devez mettre en place le CRUD de la ressource « post » qui correspond aux conférences dans le site.

La partie suppression et changement de statut se fera directement a la racine du dashboard (voir image ci-dessus).

Les autres parties seront faites de manière classique en suivant la logique des contrôleurs de ressource REST de Laravel5.

Vous devez mettre en place une gestion des mots clefs associe aux posts.

**Partie facultative**

La partie CRUD des commentaires n'est pas demandée. Si vous avez le temps vous pouvez l'implémenter.

Vous pouvez également tester une partie de l'application de manière unitaire (phpunit).

*Organisation des ressources dans le framework*

**auth/**

**front/**

**includes/**

**header.blade.php**

**sidebar.blade.php**

**partials/**

**menu.blade.php**

**index.blade.php**

**…**

**dashboard/**

**partials/**

**index.blade.php**

**errors/**

**layouts/**

**post/**

**partials/**

**index.blade.php**

**...**