

Projet Web

Objectifs :

- Sur la base du Blog conçu lors des différents exercices du cours, ajouter des fonctionnalités
- Réaliser la conception d'un site web selon le besoin exposé
- Être autonome sur la création de site web sans Framework PHP
- Être capable de se débrouiller seul (*recherche / compréhension des erreurs*).
- Organiser correctement son code et le commenter
- Avoir de l'ambition et découvrir des fonctionnalités non-abordées en cours.

1

Contexte

Vous êtes tous (*chaque apprenant*) issus d'une même entreprise qui réalise de la prestation web sur mesure. Ainsi, vous n'êtes absolument pas en compétition. L'idée est de répondre à un besoin de création d'un blog pour une société qui souhaite dynamiser son secteur d'activité.

Les attendus

En partant du blog que vous avez conçu lors des exercices du cours vous devez réaliser les fonctionnalités suivantes :

Catégorie

Faire un CRUD des catégories

Puis faire une liaison entre les Articles (post) et les catégories. Un article appartient à une catégorie et une catégorie peut appartenir à plusieurs articles

Sur le moteur de recherche de la page d'accueil ajouter une liste déroulante des catégories. Cette liste participe au filtre de recherche. Il doit y avoir le filtre « -- tous les catégories » puis ensuite la liste des catégories

Article

Chaque article doit être approuvé par un administrateur pour être visible sur le site Internet (y compris dans la recherche)

Concevoir donc une vue de tableau de bord des articles :

- Article Valider
- Article en ligne
- Article refusés

Thème

L'administrateur doit avoir la possibilité de modifier le ou les fichier CSS depuis son espace d'administration. Le champ qui édite le ou les fichiers CSS doit être en coloration syntaxique

Compte en base de données

Les comptes (nom, mail, mot de passe) doivent être stockés en base de données avec le mot de passe hashé.

Permettre à n'importe quel utilisateur de pouvoir s'inscrire sur le site Internet pour ajouter du contenu. Chaque compte doit être approuvé par un administrateur (si non l'authentification ne validera pas la connexion)

Créer une page « Mon espace » qui affiche les données de l'utilisateur connecté (son mail, son nom)

Api sécurisés

Créer une API permettant de lister les 5 derniers articles (Titre, auteur et description).

Cet API doit être sécurisée (nécessite une authentification) :

- Dans l'espace utilisateur, prévoir un bouton qui génère un Token propre à l'utilisateur (token Stocké en Base de données). Quand le token est généré il est toujours affiché dans la page « Mon espace »
- Quand on interroge la liste des 5 derniers articles, il faut envoyer le Token (dans un paramètre GET)
 - o Le token est comparé à celui trouvé en base de donnée, s'il est valide l'API retourne le résultat sinon un message d'erreur « token Invalide »

Rôles

Les rôles de l'application seront les suivants :

- **Administrateur** : pourra tout faire
- **Rédacteur** : Peut ajouter un article, modifier un article

Formulaire de Contact

Dans le Front, en bas de chaque article prévoir un champ de contact, qui lorsqu'il est validé envoie un mail à l'administrateur :

- Sujet = Nom de l'article
- Description = Lien de l'article + la saisie du champ contact par le visiteur

Cadre du projet

- **Vous travaillez en autonomie**
- Aucune limite ne vous est imposée, laissez libre cours à votre imagination
- Respect de certaines contraintes définies ci-après dans la partie « *Prérequis du projet* »
- La **présentation de votre projet** durera 10 minutes, elle devra comprendre la démonstration des attendus en live (préparez votre scénario de test)

ATTENTION

Le site doit être avant tout un espace de consultation, mais il doit permettre également une **interaction** entre les **visiteurs** et le **contenu** (*site dynamique*). Enfin, il sera possible d'administrer les contenus du site, tout en prenant le soin d'implémenter une gestion de droits (*rôles / privilèges*).

Prérequis du projet et contraintes techniques

- Vous devez utiliser l'**HTML5** et le **CSS3**
- Vous devez utiliser **PHP7**
- Respecter la structure **MVC**
- Utiliser le moteur de Template **TWIG**
- Le site web devra reposer sur une base de données.
 - o Vous utiliserez l'extension **PDO** fournit avec **PHP7** pour accéder à vos données
- Le cœur du site devra implémenter un système de **CRUD** pour les données (*Create, Read, Update, Delete* / *créer, lire, mettre à jour, supprimer* → requête SQL classique).
- Vous devrez réaliser un espace d'administration et ainsi implémenter une **gestion de droits / privilèges** complète.
- Vous devrez manipuler les dates avec précaution afin que celles-ci soient toujours correctement insérées dans la BDD et affichées à l'écran en français.
- **L'envoi d'email** sera possible et fonctionnel (*au moins en test* – [Mailtrap](#))
- Vous devrez utiliser des fonctions, services et librairies non-abordées en cours, par exemple :
 - o Enregistrer en PDF, ...
- Vos fichier JS et CSS doivent être dans le **/public/assets**
- Votre site doit être sécurisé (contrôle des champs formulaire, CSRF, Hash mot de passe, faille XSS, faille upload FILE)

Prérequis du projet et contraintes fonctionnelles

- Le site devra forcément proposer une page d'accueil
- Le visiteur aura accès à un menu pour naviguer sur les différentes pages du site
- Un espace utilisateur et donc une gestion de compte seront présents.
- Les pages du backoffice devront être limités à certains rôles.
- Il sera possible d'administrer les contenus.
- Certaines fonctionnalités sont obligatoires (**en rouge dans le tableau d'évaluation**)

Base de données

- Implémentation d'une gestion de contenu (*à votre guise : des textes, articles, recettes, fiches, entités spécifiques, produits, messages, etc.*)
- Implémentation d'un formulaire de contact
 - o Insérer les messages dans la base de données
 - o Envoi de mail (*en local*)
- Implémentation d'une gestion utilisateur
 - o Inscription, avec validation par email peut être un plus
 - o Connexion
 - o Session PHP
 - o Actions / Interactions avec le contenu

Serveur

- Apache devra être installé (*Sur Windows avec XAMPP par exemple*)
- Un **vhost** sera créé pour l'accès à votre site
- URL Rewrite

Espace administration

- Espace dédié pour administrer les contenus et les utilisateurs
 - o *Les articles, utilisateurs, messages, commentaires et autres entités du site.*
- Sécuriser l'espace administrateur en accès
- Gestion de droits et de permissions avancées
 - o Interdire en accès les pages admin

5

Récapitulatif de l'évaluation

En rouge les taches obligatoires :

| Champ d'action | Tâche | Note |
|------------------------|--|------|
| Visiteurs | <input type="checkbox"/> Consulter du contenu | |
| | <input type="checkbox"/> Envoyer un mail de contact pour chaque article | |
| | <input type="checkbox"/> BONUS : Implémenter un Bundle CAPTCHA | |
| | <input type="checkbox"/> Moteur de recherche | |
| Administrateurs | <input type="checkbox"/> Gérer le contenu (éditer, ajouter, supprimer) | |
| | <input type="checkbox"/> Approuver un article | |
| | <input type="checkbox"/> Consulter la liste des utilisateurs inscrits | |
| | <input type="checkbox"/> Valider une inscription | |
| | <input type="checkbox"/> CRUD des catégorie | |
| Rédacteur | <input type="checkbox"/> Ajouter un article | |
| | <input type="checkbox"/> Modifier un article | |
| | <input type="checkbox"/> Liaison de l'article à une catégorie | |
| Champ TEXT | <input type="checkbox"/> BONUS : Editeur WYSIWYG pour les articles | |
| | <input type="checkbox"/> Coloration syntaxique pour la modification du CSS | |
| CSS | <input type="checkbox"/> Modification du CSS en live | |
| Techno | <input type="checkbox"/> Structure MVC | |
| | <input type="checkbox"/> L'URL rewriting sera implémenté avec le routeur php | |
| | <input type="checkbox"/> Moteur de vue TWIG | |
| Code | <input type="checkbox"/> Le code sera organisé correctement | |
| | <input type="checkbox"/> Le code sera commenté et facilement maintenable | |
| Inscription | <input type="checkbox"/> Formulaire d'inscription d'un nouvel utilisateur | |
| | <input type="checkbox"/> Génération TOKEN API | |

| | | |
|-----------------|---|--|
| API | <input type="checkbox"/> Sécurisation de l'API | |
| | <input type="checkbox"/> Affichage des 5 derniers articles | |
| SECURITE | <input type="checkbox"/> Contrôle / Validation des données | |
| | <input type="checkbox"/> Faille Upload File | |
| | <input type="checkbox"/> Hash Mot de passe | |
| | <input type="checkbox"/> Tous les Formulaires ont le token CSRF | |
| TOTAL : | | |

6

Bon courage !