

Compte Rendu Appli Web MediaTek86

Compte rendu pour le devoir 1 de la 2^{ème} année de l'atelier de professionnalisation du BTS SIO au CNED par Julien Kerboeuf.

Sommaire

Sommaire	1
Contexte	2
Mission	2
Objectif	2
Périmètre	2
Description Graphique	2
Description Fonctionnelle	3
Contraintes	3
Attentes	3
Réalisation	4
Étape 1 - Préparations	4
Objectifs	4
Démarche	4
Résultats	4
Étape 2 - Nettoyage	4
Objectifs	4
Démarche	4
Résultats	5
Étape 3 - Nombre de playlists	5
Objectifs	5
Démarche	5
Résultats	5
Étape 4 - Gestion des formations	5
Objectifs	5
Démarche	6
Résultats	6
Étape 5 - Gestion des playlists	6
Objectifs	6
Démarche	6
Résultats	7
Bilan	7

Contexte

Vous travaillez en tant que technicien développeur junior pour l'ESN InfoTech Services 86 qui vient de remporter le marché pour différentes interventions au sein du réseau MediaTek86, dont certaines dans le domaine du développement d'application. Un contrat de développement a ainsi été rédigé entre MediaTek86 et InfoTech afin de réaliser 3 applications. Ce compte rendu porte sur la production de la 1ère application : une application web Symfony.

Mission

Corriger et faire évoluer une application web (Symfony) exploitant une base de données relationnelle MySQL et qui met à disposition des vidéos d'auto-formation en ligne proposées par MediaTek86.

Objectif

Permettre l'accès aux vidéos des formations en ligne du pôle "formation" de "Mediatek86".

Périmètre

- Langue : français.
- Support d'utilisation : ordinateur uniquement (une application spécifique est prévue pour les supports mobiles).
- Référencement : uniquement utilisation du référencement naturel (à optimiser).

Description Graphique

- Le maître d'œuvre doit faire des propositions de maquettes à valider.
- Seul le logo est imposé et une demande de prise en compte des codes couleur.

Description Fonctionnelle

Le front office de l'application devra proposer les fonctionnalités suivantes :

- Page de présentation avec le logo de MediaTek86, un menu d'accès aux différentes fonctionnalités, un texte de présentation et la présentation des 2 dernières vidéos de formation.
- Page "Formations" avec la liste des formations (titre, playlist, catégories, date et une photo miniature). Possibilité de trier (dans les 2 sens) sur le titre, le nom de la playlist et la date. Possibilité de filtrer sur le titre, le nom de la playlist (filtrage partiel) et sur la catégorie. La sélection d'une formation doit permettre d'accéder à la page de la formation.
- Page "Formation" avec la vidéo de la formation (qui peut être lancée), la date, le titre, la playlist, les catégories, la description.
- Page "Playlists" avec la liste des playlists (nom, catégories, nombre de formations). Possibilité de trier (dans les 2 sens) sur le nom et le nombre de formations. Possibilité de filtrer sur le nom (filtrage partiel) et sur la catégorie. Possibilité de sélectionner une playlist pour accéder à la page de la playlist.
- Page "Playlist" avec le nom, les catégories, le nombre de formations, la description et la liste des formations concernées (miniature et titre). La sélection d'une formation doit permettre d'accéder à la page de la formation.
- Page "CGU" accessible à partir de toutes les pages.

Le back office de l'application devra proposer les fonctionnalités suivantes :

- Possibilité de gérer les formations, les playlists (ajout, modification et suppression) et les catégories (ajout, suppression).
- L'accès au back office doit être sécurisé.

Contraintes

- Compatibilité avec les navigateurs : Chrome et Firefox.
- Langages et SGBD : site développé avec le Framework Symfony et le SGBDR MySQL.
- Serveur d'authentification : Keycloak.

Attentes

- Développement du site complet.
- Déploiement (serveur Keycloak, site, BDD)
- Documentation technique en ligne.
- Documentation d'utilisation en vidéo.

Réalisation

Étape 1 - Préparations

Objectifs

- Préparer l'environnement de travail
- Préparer la base de données
- Préparer le dépôt et projet GitHub

Démarche

1. Tous les outils nécessaires demandés sont installés (WampServer, Composer, Git, NetBeans)
2. Le plugin SonarLint est installé pour l'IDE NetBeans
3. Le début d'application est récupéré via le lien GitHub donné (<https://github.com/CNED-SLAM/mediatekformation>)
4. L'application est installée en local et la base de données est initialisée grâce au fichier SQL fourni et les instructions d'installation
5. Le dépôt distant et le projet kanban est initialisé et configuré (<https://github.com/JKerboeuf/mediatekformation>)
6. Configuration du plugin GitHub Issues Support pour NetBeans

Résultats

- Les outils nécessaires sont correctement installés et configuré
- La BDD est prête à l'emploi
- Le début d'application est récupéré
- L'application est lié au dépôt distant sur GitHub
- Le Kanban est configuré et prêt à l'emploi

Étape 2 - Nettoyage

Objectifs

- Nettoyer le code en utilisant SonarLint

Démarche

1. Un scan de SonarLint est lancé dans les dossier src et templates
2. Une erreur SonarLint indique que le tableau sur la page d'accueil servant à l'affichage des 2 dernières formations n'a pas d'entêtes, il est alors changé pour des balises div utilisant les classes row et col de Bootstrap pour le même résultat avec des balises plus approprié
3. De nombreux espaces en fin de lignes ou lignes vides sont supprimés
4. L'indentation est changé pour des espaces par 4 au lieu de tabs

5. Des balises
 étaient mal écrites (en majuscules ou avec un / derrière), elles sont maintenant corrigées
6. Des attributs alt sont ajoutés aux images n'en n'ayant pas
7. Des constantes sont créées pour remplacer des textes en brut répétés
8. Ajout de la mention public sur certains constructeur de classes qui de donnait par leur visibilité
9. Les switch ne contenant qu'un seul cas sont changé en if
10. Des ifs imbriqués sont fusionnés avec des opérateurs &&
11. Des uses inutilisés sont supprimés
12. Les balises <i> sont changés pour des balises

Résultats

- Le code est nettoyé et plus lisible
- Le code suit les bonne conduites de code
- Le site fonctionne toujours

Étape 3 - Nombre de playlists

Objectifs

- Afficher le nombre de formations par playlist
- Permettre le tri sur le nombre de formations des playlists
- Afficher le nombre de formations sur la page d'une playlist

Démarche

1. En utilisant le filtre permettant d'obtenir la taille d'un tableau, le nombre de formation est ajouté sur la page des playlists
2. Ajout de la colonne du nombre de formations dans la page listant les playlists
3. Ajout d'une méthode triant la liste des playlists selon leur nombre de formations
4. Ajout d'un cas pour le tri sur le nombre de formations dans le controleur

Résultats

- La page d'une playlist affiche bien le nombre de formations contenu
- La page listant les playlist affiche bien le nombre de formations pour chaque playlist
- Le tri sur le nombre de formations par playlist est possible en croissant ou décroissant

Étape 4 - Gestion des formations

Objectifs

- Les formulaires doivent être sécurisé par un jeton CSRF
- Le traitement des formulaires doit prévenir les injections SQL
- Ajouter une page pour lister et gérer les formations avec des boutons menant aux formulaires adéquats (ajout, modification, suppression)

- La page doit avoir les mêmes possibilités de tri que la page des formations
- Un bouton doit permettre d'ajouter une formation
- Les boutons de modification et d'ajout doivent afficher le même formulaire

Démarche

1. Pour ne pas avoir à le faire pour chaque formulaires, l'option 'csrf_protection' est mise à true dans la configuration de l'application pour ajouter automatiquement un jeton CSRF aux formulaires
2. Toujours dans la configuration, on définit le thème des formulaires en Bootstrap 5 car c'est ce qu'utilise déjà le site pour le style
3. Ajout de la page de gestion des formations, très similaire à la page normales des formations mais avec une colonne en plus contenant les boutons pour ajouter, modifier et supprimer les formations
4. Ajout du fichier html.twig du formulaire de base qui sera exploité pour la gestion des formations
5. Ajout des méthodes et routes de gestion des formations reprenant le même principe que les pages de base, le nom des routes est préfixé par 'admin.'
6. Ajout des méthodes affichant les formulaires d'ajout, de modification et de suppression des formulaires

Résultats

- Les formulaires possèdent bien un jeton CSRF caché
- La page de gestion présente bien la même liste des formations et permet le tri et la recherche comme demandé
- La page possède bien les boutons pour ajouter, modifier et supprimer une formation
- Les boutons ajouter et modifier donnent bien le même formulaire, celui pour modifier est prérempli des données de la formation sélectionnée
- Les catégories manquent dans le formulaire d'ajout et de modification
- Les bouton d'envoi des formulaires ne mènent à rien

Étape 5 - Gestion des playlists

Objectifs

- Ajouter une page pour lister et gérer les playlists avec des boutons menant aux formulaires adéquats (ajout, modification, suppression)
- La page doit avoir les mêmes possibilités de tri que la page des playlists
- Un bouton doit permettre d'ajouter une playlist
- Les boutons de modification et d'ajout doivent afficher le même formulaire

Démarche

1. Ajout de la page de gestion des playlists, très similaire à la page normales des playlists mais avec une colonne en plus contenant les boutons pour ajouter, modifier et supprimer les playlists

2. On utilise toujours le même fichier html.twig du formulaire de base qui est suffisant
3. Ajout des méthodes et routes de gestion des playlists reprenant le même principe que les pages de base, le nom des routes est préfixé par 'admin.'
4. Ajout des méthodes affichant les formulaires d'ajout, de modification et de suppression des playlists

Résultats

- La page de gestion présente bien la même liste des playlists et permet le tri et la recherche comme demandé
- La page possède bien les boutons pour ajouter, modifier et supprimer une playlists
- Les boutons ajouter et modifier donnent bien le même formulaire, celui pour modifier est prérempli des données de la playlists sélectionnée
- Les bouton d'envoi des formulaires ne mènent à rien

Bilan

Le code est propre, suit les codes de bonne conduite. Malheureusement, un manque de temps ne m'a pas permis de mener le projet à bout.