Contexte et Présentation du Projet

Dans le cadre du développement d'une plateforme web de gestion de formations et de playlists “Mediatekformation”, le projet a pour but de créer une interface de gestion permettant d'ajouter modifier lister et supprimer des formations playlists et catégories. La plateforme va donc devoir posséder un système d'authentification pour sécuriser l'accès à la partie back office et la mise en place d'un système de sauvegardes automatisées de la base de données.

L'Existant

Le projet est sur un système de gestion des données via une base de données SQL et des contrôleurs en PHP qui gèrent les actions de l'utilisateur. Le système de gestion des formations des playlists et des catégories est construit avec un back office sécurisé permettant aux administrateurs de manipuler les contenus à travers un tableau de bord. Des tests sont réalisés pour garantir le fonctionnement de l'application et un déploiement continu est mis en place via GitHub.

Technologies et Langages Utilisés

- PHP
- Twig
- MySQL
- JavaScript (Node.js)
- CSS et HTML
- SonarLint
- Git et GitHub
- Composer
- WampServer
- Apache NetBeans

Environnement de Développement

Le projet est développé en utilisant les outils suivants :

- Visual Studio Code
- WampServer
- GitHub

Tâche 1 : Nettoyer le code (2h) :

J'ai nettoyé le code en me concentrant sur les fichiers créés par le développeur en suivant les recommandations de SonarLint. Les actions suivantes ont été réalisées :

- Élimination des chaînes en dur pour éviter string literals duplicated

- Nomination des constantes en majuscules pour respecter les bonnes pratiques de codage.

- Fusion des tests imbriqués inutiles pour simplifier et améliorer le visuel du code.

- Ajout des attributs alt à toutes les images pour améliorer l'accessibilité.

- Ajout de l'attribut "description" à toutes les tables pour une meilleure documentation.
  Le seul problème restant est l'absence d'un header dans une table ce qui a été demandé explicitement et SonarLint n'a pas relevé d'autres problèmes. L'application est stable et fonctionne correctement.

Temps estimé : 2h
Temps réel : 0.5h

Tâche 2 : Ajouter une colonne pour le nombre de formations par playlist (50m)

J'ai ajouté une colonne "Nombre de formations" à la page des playlists pour afficher le nombre de formations associées à chaque playlist. :

- Modification de l'entité Playlist : ajout d'une méthode getFormationCount() pour calculer le nombre de formations associées.

- Modification du contrôleur PlaylistsController : calcul du nombre de formations avant de passer les données à la vue.

- Modification de la vue Twig playlists.html.twig : ajout d'une colonne pour afficher le nombre de formations avec des liens pour trier cette colonne.

- Affichage sur la page d'une playlist individuelle : dans playlist.html.twig ajout d'une ligne pour afficher le nombre de formations.
  Le nombre de formations peut désormais être trié et est visible à la fois sur la page principale et sur la page d'une playlist individuelle.

Temps estimé : 50m
Temps réel : 30m

Tâche 3 : Gérer les formations (5h):

J'ai créé une page pour lister les formations avec des boutons de suppression (après confirmation). La suppression d'une formation enlève celle-ci de la playlist associée. J'ai intégré les tris et filtres du front office au back office et j'ai mis en place un formulaire d'ajout et de modification avec des contrôles spécifiques :

- Champs optionnels pour la description et les catégories.

- Validation de la date : elle doit être inférieure ou égale à la date du jour.
  La gestion des formations fonctionne correctement et la base de données est bien mise à jour.

Temps estimé : 5h
Temps réel : 2.5h

Tâche 4 : Gérer les playlists (5h):

J'ai créé une page listant les playlists avec des boutons pour supprimer et modifier chaque playlist. La suppression d'une playlist est possible uniquement si aucune formation n'est rattachée à celle-ci. J'ai également intégré les tris et filtres du front office dans le back office. Un bouton permet d'ajouter une playlist avec un contrôle de saisie où seul le champ "name" est obligatoire. Le formulaire de modification affiche la liste des formations mais sans possibilité d'ajout ou de suppression.

Temps estimé : 5h
Temps réel : 1.5h

Tâche 5 : Gérer les catégories (3h) :

J'ai créé une page permettant de lister les catégories avec un bouton de suppression disponible pour chaque catégorie mais uniquement si elle n'est pas rattachée à une formation. Un mini-formulaire permet d'ajouter une nouvelle catégorie à condition que son nom soit unique. La gestion des catégories est opérationnelle et la base de données est correctement mise à jour.

Temps estimé : 3h
Temps réel : 2h

Tâche 6 : Ajouter l'accès avec authentification (4h):

J'ai mis en place un système d'authentification pour sécuriser l'accès au back office. Seul un profil administrateur peut y accéder. Un lien de déconnexion a également été ajouté permettant à l'utilisateur de se déconnecter facilement. L'accès est maintenant sécurisé et la fonctionnalité de déconnexion fonctionne comme prévu.

Temps estimé : 4h
Temps réel : 2.4h

Tâche 7 : Gérer les tests (7h):

J'ai effectué plusieurs tests pour garantir la stabilité du site et la bonne implémentation des fonctionnalités :

- Tests unitaires sur la méthode retournant la date de parution. Aucun problème n'a été relevé.
- Tests d'intégration pour vérifier les règles de validation des dates et des données saisies. Tout fonctionne correctement.
- Tests d'intégration des Repository : j'ai testé toutes les méthodes ajoutées dans les classes Repository. Aucune anomalie détectée.
- Tests fonctionnels sur l'accessibilité des pages et la gestion des tris et filtres. Tout fonctionne comme prévu.
- Tests de compatibilité avec plusieurs navigateurs. Aucune anomalie n'a été rencontrée.

Temps estimé : 7h
Temps réel : 3h

Tâche 8 : Créer la documentation technique (1h):

J'ai vérifié que tous les commentaires normalisés étaient présents dans le code. Ensuite j'ai généré la documentation technique en excluant le code généré automatiquement par Symfony. La documentation inclut uniquement le code ajouté par les développeurs excluant ainsi tout le contenu du dossier vendor.

Temps estimé : 1h
Temps réel : 0.2h

Tâche 9 : Déploiement (2h):

J'ai déployé le site la base de données et la documentation technique chez l'hébergeur https://cron-job.org/en/. La page des CGU a été mise à jour avec la bonne adresse et l'accès sécurisé à la partie admin a été configuré. Le déploiement s'est bien passé bien qu'il y ait eu quelques complexités dues à la configuration de l'hébergeur. Le site est désormais opérationnel en ligne.

Temps estimé : 2h
Temps réel : 2.1h

Tâche 10 : Sauvegarde et gestion de la tâche cron (1h):

J'ai programmé une sauvegarde journalière automatisée de la base de données selon les bonnes pratiques. La restauuration peut se faire manuellement via un script de sauvegarde. La gestion de la tâche cron a été compliquée en raison des restrictions de l'hébergeur en version gratuite mais cela a été résolu.

Temps estimé : 1h
Temps réel : 1h

Tâche 11 : Déploiement continu avec GitHub (1h):

J'ai configuré le dépôt GitHub pour que le site se mette à jour automatiquement à chaque push. Le déploiemeent continu fonctionne maintenant parfaitement après quelques ajustements pour garantir une mise à jour fluide du code à chaque modification.

Temps estimé : 1h
Temps réel : 0.3h

Bilan:

Le projet a atteint les objectifs suivants :

- Gestion des formations playlists et catégories :
    - sustème complet de gestion des formations playlists et catégories via un back office sécurisé.
    - creation des pages permettant d'ajouter modifier lister et supprimer des formations et playlists avec un contrôle strict des saisies.
    - ajouts des filtres et tris pour améliorer l'expérience utilisateur sur les pages de gestion.
    - mise ne en place des catégories avec des contraintes d'unicité des noms et la possibilité de les lier à des formations.
    - authentification et sécurisation du Back Office avec l'ajout d'un système d'authentification permettant d'assurer que seul un utilisateur avec un profil administrateur puisse accéder à la partie back office.
    - ajout d'une fonctionnalité de déconnexion accessible sur toutes les pages
    - test et validation du code pour garantir un bon fonctionnement des services de gestion des formations playlists et catégories.
    - test fonctionnels pour vérifier l'accessibilité des pages et la bonne gestion des tris et filtres.
    - test de compatibilité sur plusieurs navigateurs pour assurer que la plateforme fonctionnesur les navigateurs les plus utilisés.

- Déploiement et mise en Production
    - déploiement du site, de la base de données et de la documentation technique sur un hébergeur externe.
    - ajout d'une sauvegarde automatisée de la base de données via un système de tâches cron avec une restauration manuelle possible.
    - configuration du dépôt github pour un déploiement continue pour que chaque push sur le dépôt mette automatiquement à jour le site.

- Amélioration continue :
    - correction de la qualité du code avec l'utilisation de SonarLint permettant d'éliminer les erreurs et les mauvaises pratiques de codage

Problèmes rencontrées :

- configuration de l'hébergeur :
    - le déploiement sur l'hébergeur externe a été complexe à cause de certaines restrictions liées à l'hébergement gratuit notamment pour la configuration de la tâche cron de sauvegarde de la base de données.

- Problème de gestion des images et tables :
    - alors bien quee l'attribut "alt" ait été ajouté à toutes les images et l'attribut "description" à toutes les tables, le header manquant dans une table n'a pas pu être corrigé sans affecter l'affichage des données.

- Navigateur :
    - alors que que les tests de compatibilité aient été réalisés sur plusieurs navigateurs des petits soucis d'affichage ont été détectés sur certains navigateurs.

Conclusion : Le projet a été mené à bien en respectant la plupart des objectifs. Les principales fonctionnalités attendues telles que la gestion des formations des playlists et des catégories ainsi que la sécurisation de l'accès au back office ont été implémentées avec succès. Les testes ont montré que la plateforme est stable et compatible avec les navigateurs principaux. Malgré les problèmes liés à la configuration de l'hébergeur et à la gestion des données, les problèmes ont été traités. Le déploiement continu et la gestion des sauvegardes automatisées permettent de garantir un maintien du site.