

Application de suivi de l'activité

Vous devez programmer les interfaces liées une application Web de suivi de l'activité. Cette application permettra à un utilisateur de saisir des informations sur son activité physique et de les consulter.

Votre application doit accéder à une base de données MySQL pour stocker les informations. Un fichier de départ est fourni. Vous devez l'importer dans votre base de données.

Vous développer une application mono-page (SPA). Donc, la page du navigateur ne doit pas être rechargée lors de la navigation entre les différentes 'pages'.

Le HTML, CSS et le PHP sont déjà fournis. Vous devez ajouter le JavaScript pour rendre l'application fonctionnelle.

Liste des exercices

La page principale de l'application sert à afficher la liste des exercices. Chaque exercice est affiché sous forme de carte. Chaque carte contient les informations suivantes:

- date de l'exercice
- type d'exercice. Ex: Course à pied, Vélo, Marche, etc.

Vous récupérerez les données avec une requête FETCH GET.

Détail d'un exercice

Lorsqu'on clique sur une carte, on affiche les détails de l'exercice. La section du détail affiche les informations suivantes:

- date de l'exercice
- type d'exercice. Ex: Course à pied, Vélo, Marche, etc.
- durée de l'exercice
- description de l'exercice
- difficulté de l'exercice. 1 à 5

Vous devez récupérer les données avec une requête FETCH GET contenant un paramètre d'URL pour identifier l'exercice à afficher. EX: "?id=1"

Formulaire d'ajout d'un exercice

Il est possible d'ajouter un exercice via un formulaire. Vous devez envoyer les infos au serveur php avec une requête POST JavaScript.

Le formulaire est gérée par une classe qui sert à bloquer la soumission du formulaire si les champs ne sont pas valides. Vous devez ajouter les règles de validation pour chaque champ du formulaire. Si un champ n'est pas valide, vous devez afficher un message d'erreur sous le champ en question et désactiver le bouton d'envoi du formulaire. Des classes CSS sont fournies pour vous aider à afficher les messages d'erreur.

Une fois les informations envoyées, vous devez afficher un message de succès ou d'erreur et retourner à la liste des exercices.

Vous n'envoyez pas les données par le formulaire traditionnel car cela rechargerait la page. Vous devez envoyer les données par une requête FETCH.

Suppression d'un exercice

Il est possible de supprimer un exercice. Une fois l'exercice supprimé, vous devez afficher un message de succès ou d'erreur et retourner à la liste des exercices et réafficher la liste des exercices.

Utilisez une requête FETCH GET pour supprimer un exercice en passant l'id de l'exercice à supprimer.

Routeur

Vous devez avoir un système de routage côté-client pour afficher les différentes pages de votre application. Le routage doit être basé sur l'URL que vous devez analyser pour déclencher la bonne fonction du gestionnaire d'application. Il doit être possible de revenir en arrière et de naviguer en avant dans l'historique du navigateur avec les boutons "Back" et "Forward".

Affichage des erreurs et des messages

Une classe `ToastModale` vous a été fournie. Vous devez l'utiliser pour afficher les messages d'erreur et de succès pour les actions liées à la base de données. Ex: affichage, ajout et suppression d'un exercice. Le HTML est déjà placé dans la page index.html. Le toast s'auto-détruit après 2.5 secondes.

Programmation orientée objet

Vous devez utiliser la programmation orientée objet pour structurer votre application. Vous devez avoir une classe pour chaque page de l'application. Vous devez également avoir une classe pour gérer les requêtes FETCH. Un Schéma UML vous est fourni pour vous aider à structurer votre application.

Lorsque possible, mettez les variables et les fonctions privées. Utilisez les méthodes de classe et les propriétés statiques lorsque cela est nécessaire.

Commentez vos classes et vos méthodes pour expliquer leur fonctionnement.

Base de données

Assurez-vous d'inscrire les bons identifiants à votre base de données dans chaque fichier PHP. Testez les requêtes avec Postman pour vous assurer que tout fonctionne correctement avant de commencer à programmer votre application.

La base de données et la table doivent garder le même nom svp. C'est plus simple à corriger.

Remise

Vous devez remettre le travail avant le 21 août 2024 à 23h59. Vous devez remettre un fichier ZIP contenant les fichiers de votre application. Vous devez également inclure un fichier README.md qui explique comment installer et utiliser votre application. Utilisez le lien suivant pour remettre votre travail:

<https://classroom.github.com/a/eqRtf334/>

Vous n'avez pas à déposer votre projet sur Github Page car il n'y a pas de serveur PHP. Pour ceux qui le souhaite, je pourrai vous montrer à déposer sur WebDev mais ce n'est pas nécessaire.

Critères d'évaluation

- Juste compréhension des fonctionnalités demandées à l'intérieur du devis (10 %)
- Utilisation correcte d'un langage de programmation orienté objet côté client (30 %)
- Gestion adéquate de l'affichage dynamique (20 %)
- Traitement et exécution adéquates des réponses synchrones et asynchrones (25 %)
- Structure et qualité optimale de la programmation (15 %)

Plagiat

L'utilisation de génération de code par intelligence artificielle est strictement interdite. Tout plagiat entraînera une note de 0 tel que défini dans la politique d'évaluation.

Si vous empruntez du code, vous devez citer vos sources dans un commentaire dans votre code. Vous devez également expliquer le code que vous avez emprunté dans votre rapport.