Développer en JavaScript

Formateur : Youssouf SAGAF

sagafysf@gmail.com

Projet : Application de Gestion de Tâches

Dans ce projet, vous allez créer une application de gestion de tâches qui permet aux utilisateurs d'ajouter, de marquer comme terminées, et de supprimer des tâches. De plus, l'application fera appel à une API pour afficher des citations inspirantes. Utilisez `npm` pour la gestion des packages. Vous pouvez créer un dossier **task-manager** et utilisez la commande `npm init -y` pour initialiser un projet `npm`.

Partie 1 : Développement d'une application de gestion de tâche avec JavaScript et Ajax:

Étape 1 : Structure de l'application

Télécharger la structure de base de l'application à cette adresse : (https://github.com/yosagaf/task-manager). L'application contient un formulaire pour l'entrée des tâches, une liste pour afficher les tâches ajoutées, et un espace avec un **div** pour afficher les citations inspirantes.

Étape 2 : Ajout de tâches

Utilisez JavaScript pour ajouter la fonctionnalité d'ajout de tâches. Quand un utilisateur soumet le formulaire, une nouvelle tâche doit être créée et ajoutée à la liste des tâches avec un **button** supprimer et un checkbox.

Étape 3 : Marquer les tâches comme terminées

Ajoutez la possibilité pour les utilisateurs de marquer les tâches comme terminées. Ceci pourrait être réalisé en utilisant une case à cocher pour chaque tâche. Quand l'utilisateur coche cette case, la tâche correspondante doit être marquée comme terminée d'une manière visible (par exemple, en la barrant).

Étape 4 : Suppression de tâches

Ajoutez une option pour supprimer les tâches de la liste. Cela pourrait être réalisé en ajoutant un bouton "**Supprimer**" à chaque tâche. Quand ce bouton est cliqué, la tâche correspondante est retirée de la liste.

Étape 5 : Stockage des tâches

Utilisez le localStorage pour stocker les tâches de manière persistante. Ainsi, même après le rechargement de la page, les tâches précédemment ajoutées resteront visibles.

Étape 6 : Validation du formulaire

Implémentez une validation du formulaire pour empêcher les utilisateurs d'ajouter des tâches vides. Si l'utilisateur essaie de soumettre une tâche vide, affichez un message d'erreur approprié.

Étape 7 : Intégration d'une API

En utilisant AJAX, intégrez une API qui fournit des citations inspirantes. Affichez une nouvelle citation sur votre application chaque fois que l'utilisateur la visite ou recharge la page.

Partie 2 : Développement d'une application de gestion de tâche avec JavaScript et JQuery:

Dans cette partie, la correction de la partie 1 vous est fourni. Le but ici est de transformer le code JS en utilisant une approche orientée objet avec JQuery. Le code qui vous est fourni gère une liste de tâches en utilisant JavaScript pur.

Voici les étapes à suivre :

1. Analysez le code existant et comprenez comment il fonctionne (Vous pouvez copier le code et demander à ChatGPT de vous l'expliquer simplement).
2. Créez une classe appelée "**Task**" qui encapsulera toute la logique de gestion des tâches.
3. Utilisez les sélecteurs jQuery pour sélectionner les éléments du DOM au lieu des méthodes JavaScript pure.
4. Transformez les fonctions existantes en méthodes de la classe **Task**.
5. Utilisez les méthodes de la classe **Task** pour ajouter des écouteurs d'événements, générer des tâches, sauvegarder et charger les tâches, etc.
6. Testez votre code en ajoutant, supprimant et marquant des tâches comme terminées.

Conseils :

- Consultez la documentation jQuery pour comprendre comment utiliser les sélecteurs et les méthodes jQuery.
- Divisez votre code en méthodes logiques et réutilisables pour faciliter la maintenance et la compréhension.
- Utilisez des commentaires pour expliquer chaque étape importante de votre code.