5IDV2 Projet récapitulatif

MVC Objet et requêtes XHR

Créer un projet incluant la recherche des pokémons pour permettre à l'utilisateur de se créer une bibliothèque de pokémons favoris.

Pour se faire, vous devrez créer une base de données ne comprenant qu'une table, qui va enregistrer au minimum le nom, l'id et l'url de l'image (ou sprite) du pokémon sélectionné.

L'utilisateur pourra donc rechercher un pokémon grâce à une barre de recherche et choisir de le mettre en favori s'il le souhaite ainsi que de visionner sa bibliothèque de favoris.

Votre code PHP doit être structuré en MVC comme vu au cours, vos contrôleurs, modèles et DAO doivent être réalisés en objets.

Les requêtes permettant de récupérer des pokémons, d'enregistrer un favori ainsi que d'enlever un favori doivent être réalisées en XHR (Ajax). **GET** pour les demandes d'informations, **POST** pour les envois d'informations.

Récapitulatif des demandes

- Créer une base de données ainsi qu'une table pouvant enregistrer au minimum l'id du pokémon, son nom et l'url de son image/sprite
- Créer un MVC permettant d'ajouter / supprimer des pokémons dans votre table
- Afficher sur la page la liste des pokémons déjà présents dans la bibliothèque de favoris
- Permettre à l'utilisateur de chercher un Pokémon via une barre de recherche
- Permettre à l'utilisateur d'ajouter / supprimer un pokémon de ses favoris
- Les requêtes pour ajouter/supprimer un favori doivent être faites en XHR

Demandes complémentaires (bonus)

- Afficher le ou les types du pokémon recherché ainsi que la liste de ses compétences lorsque vous récupérez un pokémon par l'API
- Créer une page listant l'entièreté des pokémons disponible, montrant une * ou autre sur les pokémons qu'il a déjà en favori
- Créer une "bannière" ou défile de manière aléatoire, toutes les 5 secondes, un pokémon. Permettez à l'utilisateur de le mettre en favori lorsqu'il clique sur celui-çi
- Soigner la présentation visuelle (css)
- Libre à vous d'ajouter l'une ou l'autre fonctionnalité complémentaire, précisez le néanmoins lorsque vous me remettez votre travail

Récapitulatif des conventions

Une variable contenant un sélecteur jQuery commence par un $

Par exemple : $inputs = $('form input');

On attache les évènements au .on
Par exemple : $(‘input’).on(‘click’, function() { //… });

On attache les évènements sur le body si ceux-ci peuvent être déclenché sur du contenu injectés après la génération du document
$(‘body’).on(‘click’, **‘button#monBouton’**, function() { //… });

Click d’un bouton avec une ID ‘monBouton’ présent depuis le début du document :
$(‘**button#monBouton**’).on(‘click’, function(){// ...});

Click d’un bouton avec une ID ‘monBouton’ injecté à la suite d’une requête ajax :
$(‘body’).on(‘click’, **‘button#monBouton**’, function() {//...});

On traite une réponse réussie du serveur dans le .done() et une réponse mauvaise dans le .fail(). Pas de fourre-tout dans le .done().
Pour rappel, il suffit de changer le statut HTTP renvoyé par le serveur pour déclencher un appel au .fail de votre requête ajax.

Remise du travail
Date de remise : 03/06/2023 20h00

Rendez votre code au format ZIP ou RAR, nommé votre fichier avec votre nom et prénom, ainsi qu’un dump de votre base de données. Vous pouvez aussi m’envoyer un lien github à partir du moment où votre repository est publique pour que je puisse y accéder.

Résultat
Ce projet compte pour une partie des points de votre résultat final.