

TP PHP/HTML/Bootstrap: Mini YouTube

Objectif du TP:

Le but du tp est de créer une mini-plateforme de streaming vidéo (comme YouTube).

Les fonctionnalités du site seront les suivantes :

- La page d'accueil affiche une mosaïque avec toutes les miniatures des vidéos qui sont sur le site
- Les visiteurs peuvent s'inscrire sur le site
- Les visiteurs peuvent se connecter sur le site une fois inscrits
- Les visiteurs inscrits peuvent ajouter une vidéo, qui apparaîtra pour tout le monde sur la page d'accueil
- Cliquer sur une miniature affiche une autre page où la vidéo est visible

Etape 1 :

Réaliser la page d'accueil index.php avec Bootstrap.

La page doit ressembler au fichier cibleIndex.png.

Les liens dans la barre de menu envoient le visiteur vers les pages connexion.html et inscription.html

Etape 2 :

Créer les pages connexion.html et inscription.html.

L'utilisateur devra choisir un pseudo et un mot de passe (qu'il répètera 2 fois pour l'inscription). Pas d'image de profil dans le formulaire.

Les pages doivent ressembler aux fichiers cibleInscription.png & cibleConnexion.png

Etape 3 : Créons la BDD

Réfléchissez à la structure de la base de données (nombre de tables, nom des tables, attributs des tables).

Faites un schéma et demander une confirmation à l'enseignant avant de continuer.

Etape 4

Codez les pages inscription.php et connexion.php.

Elles sont appelées quand on remplit les formulaires inscription.html et connexion.html (donc dans les pages HTML on aura un form action= ""). Et elles recevront les infos sous forme de POST (method = "post" dans les formulaires).

Pour l'inscription :

- Le pseudo devra être unique (pas encore utilisé)
- Le mot de passe devra être long d'au moins 8 caractères et contenir un chiffre
- Le mot de passe devra être sauvegardé de manière hashée dans la BDD (utilisez password_hash)

Pour la connexion :

- En cas de succès de connexion, on créera une variable de session pour considérer que la personne est connectée
- On redirigera l'utilisateur vers le de formulaire page envoyer-video.php

Vous pouvez vous baser sur les exercices similaires déjà vus ensemble.

Etape 5

Faire la page de formulaire envoyer-video.php.

Avant d'afficher le formulaire, cette page doit vérifier que l'utilisateur est bien connecté (regarder si la variable de session est présente), sinon le renvoyer vers connexion.html

Le formulaire doit contenir 4 champs :

- Un titre
- Une description
- Une image de miniature
- Une vidéo

Et un bouton d'envoi.

Ce formulaire va être transmis à la page enregistrer-video.php. Etant donné qu'elle va transmettre de fichiers, ne pas oublier de mettre le bon type de formulaire.

Etape 6

Créer le fichier enregistrer-video.php

Ce fichier doit tout d'abord vérifier que l'utilisateur qui tente d'y accéder est connecté (vérifiez la variable de SESSION).

Ensuite il doit vérifier que les 4 champs du formulaire ont été transmis et qu'il n'y a pas eu d'erreur d'envoi (pour les 2 fichiers).

Il faut vérifier que la miniature envoyée a bien pour extension jpg et que la vidéo a bien pour extension mp4.

Si c'est le cas, on insère dans la base de données le titre et la description de la vidéo.

Il faut aussi enregistrer définitivement les fichiers miniatures et vidéo. On les mettra dans un dossier uploads/miniatures/ et uploads/videos.

Les fichiers s'appelleront miniatureN.jpg videoN.mp4 où N est l'ID de la vidéo que l'on vient d'insérer. Comme l'ID est généré automatiquement, il existe une technique pour récupérer l'ID du dernier élément inséré. A vous de chercher.

A ce stade on affiche un message de succès.

Si une des étapes décrite juste avant échoue, il faut rediriger l'utilisateur vers envoyer-video.php.

Etape 7

Adaptez la page index.php.

Celle-ci doit désormais aller chercher les vidéos depuis la base de données et créer une miniature pour chacune d'entre elles (pensez fetchAll et foreach).

L'image de miniature à afficher a donc pour chemin /uploads/miniatures/miniatureN.jpg (ou N est l'ID de la vidéo en question).

Le lien « voir la vidéo » doit rediriger l'utilisateur vers la page video.php?id=N (ou N est l'ID de la vidéo), cela va nous servir par la suite.

Etape 8 : Finalisation

Créer la page video.php.

Celle-ci récupère l'ID de la vidéo que l'on souhaite afficher depuis l'URL (id=N, donc \$_GET['id']).

Et affiche la vidéo grâce aux balises HTML5 que vous trouverez sur internet.

Le titre de la vidéo et sa description devront également figurer sur la page.

Exercices bonus:

- Mettre des messages d'erreurs décrivant l'erreur pour chaque redirection que peut subir l'utilisateur.
- Faire la page de déconnexion et l'afficher dans le menu uniquement pour les utilisateurs connectés.
- Les personnes connectées doivent pouvoir ajouter des commentaires sur les vidéos des autres (adapter la BDD en ajoutant une table commentaires avec deux clés étrangères, une vers les utilisateurs et une vers la vidéo ; faire un formulaire en dessous de la page pour visionner les vidéos, afficher les formulaires en dessous de la vidéo
- Créer une page pour supprimer les vidéos si on est connecté (on peut supprimer celles des autres pour des raisons de simplicité).
- Ca serait pas mal de pouvoir mettre d'autres formats d'images pour les miniatures et les vidéos non ?