**Projet BDW1**

***Application mini-Pinterest***

**2020-2021**

***Rendu***

Nom : RANDRIAMITANDRINA Prénom(s) : Finaritra Groupe :C1

Nom : HOXHA Prénom(s) : Benereta Groupe :C1

Date :05/05/2021

**Description de la démarche/méthodologie (2 pages maximum, *Police Times New Roman 11*)**

//Décrire comment vous vous êtes organisés pour faire le travail demandé et les grandes étapes du travail réalisé.

1iere étape, l’affichage de la page principale et mise en place de bdd:

* On a codé la page du site **index.html** (déterminer les textes et le css).
* Créer la base de données **bdd** dans myphpadmin SQL (puis les tables photo et catégorie)
* Définit 4 catégories distincts : animaux, vêtement, nourriture, voiture.
* Chercher les liens de 3 ou 4 images correspondants à une catégorie spécifique.
* Enregistrer toutes les images (sous format jpg), les renommer pour avoir un nom plus court et les mettre dans un répertoire **data**.
* Faire des INSERT dans la table catégorie : (animaux, vêtement, nourriture, voiture.)
* Faire des INSERT dans la table photo : pour les 15 images (compléter les attributs photoId, nomFich, description, catId)
* Enfin, on a exporté **bdd.sql** et on l’a mis dans le répertoire **bdd.**

2ieme étape, le code php du formulaire pour filtrer les images :

* Pour l’environnement de travail, on a opté pour la deuxième option pour le serveur, puisque nous avions toutes les deux Windows, donc **XAMPP.** On a consacré une journée entière à essayer de bien faire fonctionner php (vidéo tuto, forum, etc.…).
* Puis, on a codé l’affichage de *« toutes les images »* en premier, à partir de **data**, avec des images brutes.
* On a codé la partie PHP du formulaire : *« Quelles photos souhaitez-vous affichés ?».*
* Entre temps, on a codé toutes les fonctions mysql\_...() pour accéder à la base de donnée **bdd**.
* Enfin, on a organisé tout ça dans les fonctions affic($cate), qui affiche les images par catégorie et affic\_def() qui affiche toutes les photos par défauts (l’accueil).

3ieme étape, la page des détails d’une image :

* Regarder des tutos « Comment envoyer et recevoir des variables via un lien » (La méthode get).
* Puis même démarche que pour **index.html** : coder l’affichage de **detail.html** avec des textes bruts, l’accès à la base de données puis manipuler des données.

4ieme étape, la page d’inscription et connexion :

* L’une devait s’occuper de rajouter les boutons *inscription* et *connexion* dans index.html et l’autre, à coder le formulaire d’**inscription.html,** le code PHP de la validation de mot de passe et le login, et l’envoyer à bdd.
* Entre temps, écrire le code SQL pour créer une nouvelle table validation (CREATE TABLE validation), où il y avait comme attribut id, pseudo et passe.
* Coder **connexion.html** (l’html estplus rapide car copier/coller de **inscription.html)** etajuster la validation de login et mot de passe.
* Regarder des tutos « Comment ouvrir une session en PHP » (La méthode $\_SESSION).

5ieme étape, index.html en mode utilisateur connecté :

* Au départ, on a créé un fichier à part **index\_utilisateur.html**, (une interface utilisateur) mais finalement, c’est exactement **index.html** mais avec des « boutons en plus ». Donc, on a modifié **index.html** de façon que si un utilisateur se connecte($\_SESSION[‘logged’]), on ajuste l’affichage (bouton *« Ajouter une photo »* n’est plus caché, *login* à la place du bouton inscription et « *Déconnexion »* à la place de « *Connexion »*).
* On a également codé **deconnexion.php** qui permetsimplement de réinitialiser la connexion et de revenir à la page d’accueil.

6ieme étape, quand un utilisateur quelconque veut ajouter des images :

* Coder : **ajouter.html** : Page de formulaire, qui ajoute une photo à la base de données
* Même démarche que **inscription.html (**envoie de donnée**s à bdd).**
* Vérifications des conditions de validation du formulaire avec PHP (voir annexe pour plus de détail)

7ieme étape, distinction deux profils utilisateur :

* Rajouter un attribut admin, un booléen (TINYINT(1)) dans la table validation
* Insérer 2 tuples d’utilisateurs (fina et bena) en tant qu’administrateur.
* Adapter à nouveau le code de index.html pour les administrateurs (rajouter des boutons, adapter l’affichage en fonction des nouveaux items...etc.)

Nos logins administrateurs : **fina** ou **bena**

Le même mot de passe pour les deux: **12345678**

8ieme étape, des options pour l’administrateur :

* Coder : **statistic.html** : Page de statistiques dédiée au site et accessible seulement à l’administrateur. (Nombre d’utilisateurs et nombre d’image de chaque catégorie)
* Coder : **modifier.html :** Page affichant les données d’une photo + des inputs pour les nouvelles valeurs de la photo (description, nomFich…etc.)
* Coder : **supprimer.php :** Supprime une image, accessible depuis la page modifier.html
* Ecrire sur chaque page le login de l’utilisateur connecté et son temps de connexion (qu’il soit administrateur ou non).

**Décrire le fonctionnement de l’application sous forme d’un workflow commenté (1 page maximum, *Police Times New Roman 11*)**

Page d’accueil du site (toutes les photos sont affichés par défaut) **index.html**

**Utilisateur non connecté** **index.html**

Connexion ? Session ouverte ?

* *Bouton connexion*

Déconnexion (code php : réinitialise la session) **deconnexion.php**

Si statut administrateur en plus :

**Utilisateur connecté** (login affiché, bouton déconnexion et bouton ajouter une photo)

**index.html**

* *Bouton déconnexion*
* *Bouton valider*

Connexion (validation de login et mdp : connexion) **connexion.html**

* *Bouton accueil*
* *Les champs sont vides*
* *Login et le mot de passe ne sont pas corrects.*

Animaux

**Filtre** : affichage des photos correspondant à une catégorie **index.html**

Vêtements

* *Filtre*

Toutes les photos

* *Filtre*

Page des statistiques

statistic.html

Voitures

Nourriture

* *Bouton ajouter une photo*
* *Bouton inscription*
* *Bouton accueil*
* *Bouton catégorie*
* *Cliquer sur une photo*

Modifier photo

photo.html

Inscription (champs de donnée à envoyer) **inscription.html**

Détail d’une photo (Page dédiée à une photo particulière) **detail.html**

Ajouter une photo (formulaire à remplir, puis envoyer à la base) **ajouter.html**

* *Des champs sont vides*
* *Des champs sont vides*
* *Le login existe déjà*
* *Le mot de passe n’a pas 8 caractères*
* *Les 2 mots de passe ne sont pas les mêmes.*
* *Des champs sont vides*
* *Le fichier choisis n’est pas une image*

Supprimer la photo **supprimer.php**

**Annexe (*Police Times New Roman 11*)**

//Fournir les composants (code) de votre application et les documentant correctement

Pour ce projet, nous avons 7 fichiers html et 2 fichiers PHP :

* **index.html :** la page principale du site
  + Si pas de connexion :
    - Cacher le bouton « Ajouter une photo »
    - Afficher les boutons connexion et inscription
  + Si connexion :
    - Affiche le login de l’utilisateur et le temps de connexion
    - Afficher le bouton « Ajouter une photo »
    - Désactiver le bouton inscription et afficher le login
    - Remplacer bouton « Connexion » par « Déconnexion »
* Si clique sur déconnexion, renvoie au fichier deconnexion.php
* Si connexion ET administrateur
  + Affiche « Compte administrateur » près du login.
  + Affiche le bouton « Statistiques »
  + Affiche des boutons « Modifier la photo » près de chaque photo.
  + En général :
    - Affiche le formulaire filtre « Quelle photo ? »
      * Appelle la fonction affic\_def(), pour afficher l’accueil ou toutes les photos
      * Appelle la fonction affic\_cat($cate) , pour afficher selon la catégorie
    - Si une photo est cliquée, la page renvoie à la page de detail.html et envoie les variable idphoto et idcat de la photo cliquée, via le lien.
* **detail.html** : la page dédiée à une photo
  + Affiche le login de l’utilisateur et le temps de connexion
  + Reçoit idphotoetidcat de la photo via $\_GET.
  + Affiche la photo à gauche
  + A droite le tableau des informations de l’image (récupérer dans la BD bdd grâce à la foncion recup\_info() )
  + Bouton cliquable :MN retour à l’Accueil et le nom de la catégorie en bleu
* **inscription.html :** la page où l’utilisateur s’inscrit
  + Est ouvert dans un nouvel onglet depuis index.html
  + 4 champs input : login, mot de passe 1 et 2, et valider
  + Condition de validation : login n’est pas déjà dans la table validation de la base de données et les 2 mdp sont les mêmes.
  + Si valider, envoie login et mot de passe à la bdd (INSERT…)
* **connexion.html :** la page où les utilisateurs se connectent
  + Est ouvert dans un nouvel onglet depuis index.html
  + 3 champs input : login, mot de passe et valider
  + Condition de validation : il faut que login et mot de passe soit dans la table validation de bdd.
  + Si valider, renvoie à la page index.html (où $\_SESSION [‘logged’] prend la valeur login) et ouvre une session.

Nos logins administrateurs : **fina** ou **bena**

Le même mot de passe pour les deux: **12345678**

* **ajouter.html** : la page de formulaire d’une photo à envoyer
  + Affiche le login de l’utilisateur et le temps de connexion
  + Ne s’ouvre que si un utilisateur s’est connecté depuis index.html.
* Vérifications des conditions de validation du formulaire avec PHP (voir annexe)
  + La taille de l'image ne doit pas dépasser les 100ko en vérifiant avec $\_FILES['nomFich']['size'] °L'extension doit être jpg, jpeg, gif ou png en vérifiant grâce à $\_FILES['nomFich']['name'] °La description doit contenir au moins une lettre en vérifiant grâce à $\_POST['description']
  + Une catégorie doit être choisie en vérifiant que $\_POST['categorie'] est non nulle.
  + Une fois que toutes les vérifications ont été faites, le fichier est téléversé dans le répertoire data qui contient les images et est ajouté à la table photo en portant le nom demandé
* **modifier.html :** la même page que detail.html mais avec un formulaire.
  + Affiche le login de l’utilisateur et le temps de connexion
  + Accessible seulement à l’administrateur
  + Affiche tous les détails actuels de la photo cliquée.
  + Des inputs des nouvelles valeurs de la photo.
  + Envoie les modifications à bdd (le code ne marche pas encore)
  + Bouton, supprimer la photo
  + Redirige vers la page d’accueil index.html.
* **supprimer.php :** 
  + Affiche seulement « Le photo va être supprimée... »
  + Envoie la suppression à bdd
  + Redirige vers la page d’accueil index.html.
* **static.html :**une page html affichant des informations du site.
  + Affiche le login de l’utilisateur et le temps de connexion
  + Accessible seulement à l’administrateur
  + Affiche nombre d’utilisateurs et nombre d’image de chaque catégorie
  + Redirige vers la page d’accueil index.html.
* **deconnexion.php :** le code PHP permettant de réinitialiser la session.
  + Détruit la variable $\_SESSION [‘logged’] et détruit la session.
  + Redirige vers la page d’accueil index.html

On a également 2 répertoires :

* **data :** il contient toutes les photos de la base de données.
* **bdd :** il contient **bdd.sql ,** la base de données contenant les tables :
  + *Categorie*(*catId*, nomCat) // La table des catégorie de photo
  + *Photo*(*photoId*, nomFich, description, #catId) // la table des informations de photo
  + *Validation(id, pseudo, passe, admin) //La table des informations sur les utilisateurs*

***bdd.sql*** est un fichier à importer dans ***http://localhost/phpmyadmin/*** *avec* ***le serveur XAMPP.*** *Et bien nommée la base de donnée* ***bdd.***