Une image contenant Police, texte, logo, Graphique

Description générée automatiquement

2024-2025

Alexia Ly - Antoine Court - Lucas Noel - Ethan Soucher

Sommets

Projet Sommets

Table des matières

**Introduction2**

**La page d’accueil et catalogue3**

**Formulaire de connexion4-5**

**Le mode éditeur6**

**Le mode non-éditeur6-7**

**Nos difficultés et résolution des problèmes7**

**Conclusion8**

Introduction :

Le projet a pour objectif de développer une application web permettant de gérer le catalogage et l’étiquetage d’images. Ce site a pour but d’alterner entre un utilisateur client et éditeur tout en ayant une page dynamique grâce au javascript. Les deux profils distincts de ce projet est l’éditeur qui a la possibilité de se connecter pour sélectionner des images qu’il peut étiqueter et associer des informations correspondantes et la gestion libre des catalogues , hors l’utilisateur client qui est non-éditeur peut consulter le contenu de chaque catalogues d’images avec les étiquettes qui permet de consulter les informations détaillées.

Ce projet repose alors sur les langages étudiés lors de notre semestre, HTML, CSS, JAVASCRIPT avec JQUERY, PHP, BOOSTRAP et notre base de données fournie pour perfectionner ce projet.

Tout d’abord, nous allons présenter et expliquer la partie Accueil et Catalogue de notre site. Puis, nous allons vous montrer et commenter la création du formulaire de connexion au site. Ensuite, nous allons commenter nos démarches sur nos pages éditeur et non-éditeur. Enfin, nous allons exprimer nos points faibles et nos points forts sur notre projet, et par quels moyens nous avons résolu les différents problèmes rencontrés.

1-Page d’accueil et la page catalogue

Tout d’abord, nous avons fait une page d’accueil, qui est la première interface visible pour accueillir l’utilisateur. Cette page a pour but d’introduire à l’utilisateur nos différentes fonctionnalités disponibles. On y retrouve une barre de navigation réalisée avec Bootstrap pour nous aider à la mise en page. Cette barre de navigation possède deux interfaces visibles selon si l’utilisateur est connecté ou non.

Ensuite, la page en elle-même présente 3 interfaces visibles selon plusieurs critères :

* Si l’utilisateur n’est pas connecté, nous lui demandons de se connecter.
* Si l’utilisateur est connecté sur un compte non-éditeur, il a juste accès à la visualisation du site, de ses catalogues et étiquettes (bouton mode exécution).
* Enfin, si l’utilisateur est connecté sur un compte éditeur, il a accès à la visualisation et la modification du site, il peut créer des catalogues et des étiquettes, mais aussi les supprimer (bouton mode édition).

Ces conditions sont codées en php, langage propice à la gestion de BDD.

Cette page permet de bien distinguer les éditeurs des non-éditeurs avec la variable $\_SESSION[‘role’]. Ce choix nous permet de bien différencier que chaque utilisateur a une bonne interface adaptée à ses fonctionnalités. Cette page d’accueil permet d’interdire certaines fonctionnalités ou informations à des utilisateurs non-autorisés à les utiliser.

La page Catalogue est conçue pour s’adapter selon le mode (édition ou exécution) sélectionné par l’utilisateur connecté . Elle permet d’afficher les catalogues existants, de créer des catalogues et d’assurer la transition vers des pages de gestion ou visualisation des catalogues selon le mode actif.

Dans cette page, la session est initialisé avec début-session(), qui permet de personnaliser le contenu de la page selon le compte de l’utilisateur. Puis, nous avons fait une variable $SESSION[‘mode’] qui permet de déterminer si l’utilisateur navigue en mode édition ou exécution avec une condition pour basculer entre les deux modes. Cette gestion du mode actif permet d’adapter la page sans faire une duplication de code.

On retrouve ainsi une page qui utilise une connexion MySQL avec un fichier de configuration dbconfig.php pour assurer une abstraction et une sécurité importante. Ses interactions avec la base de donnés sont mises dans des instructions précises comme : $statement=$connexion->prepare(…). Une requête SQL (SELECT name, id FROM catalog) est exécutée pour récupérer les noms et identifiants des catalogues disponibles dans la base de données.

Pour récupérer nos données, nous utilisons JSON et javascript pour que les données du tableau soient bien rendu dans le catalogue de façon dynamique. Chaque catalogue est présenté sous forme de bouton cliquable généré par du Javascript. En cliquant sur un bouton, l’utilisateur est dirigé vers une autre page pour consulter ou modifier le contenu du catalogue.

2-Formulaire de connexion

Ensuite, nous allons nous pencher sur les fichiers de gestion utilisateur ainsi que leurs formulaires. Pour le formulaire d’inscription, les types de champs à remplir sont spécifiques de manière à éviter des erreurs de saisis par l’utilisateur (exemple mot de passe et email), nous utilisons la méthode « POST » pour des raisons de confidentialité par rapport à la méthode « GET » et le formulaire est lié avec le formulaire de récupération des données, de plus nous avons intégré à la page de navigation pour que l’utilisateur permet d’accéder directement à la page de connexion ou d’inscription et elle permet de s’adapter à la page selon l’état de connexion.

Le formulaire de connexion suit le même schéma que celui d’inscription avec cette fois-ci seulement deux entrées : pseudo et password, et un bouton inscription si l’utilisateur s’est trompé en cliquant sur connexion alors qu’il n’est pas encore inscrit dans notre base de données.

Le fichier déconnexion permet de se déconnecter en appuyant sur le bouton Déconnexion. On y retrouve la gestion des sessions, il permet aussi de supprimer les variables avec la fonction session\_unset et entraine la redirection de l’utilisateur vers la page d’accueil.

Enfin, on retrouve les fichiers de configuration qui permettent de récupérer les informations nécessaires pour se connecter à la base de données, qui sont stockées dans une table. Grâce à cela, si l’utilisateur fait des modifications, elles vont directement s’appliquer aux autres fichiers. Dans le fichier recup.php une connexion est établie avec la base de données puis les données du formulaire d’inscription sont récupérées via la méthode « POST » ,et une requête cherche l’utilisateur dans la table useraccount, si un utilisateur est trouvé, on vérifie le mot de passe et le hachage avec password\_verify, si c’est correct les variables de session sont mises à jour et l’utilisateur est redirigé vers la page d’accueil, si l’utilisateur n’est pas trouvé dans la base de données ou si le mot de passe est incorrect, des messages personnalisés sont affichés pour informer l’utilisateur de l’erreur rencontrée.

Le fichier modif.php vérifie l’état de connexion de l’utilisateur et l’invite à se connecter si ce n’est pas le cas, le formulaire de création de l’étiquette récupère les variables catalogue (id du catalogue) et position (position de l’élément à modifier) par la méthode « POST » (toujours pour des raisons de confidentialité) et permet de modifier le nom de l’étiquette obligatoirement (avec required), la description et les coordonnées et un bouton permet de valider les modifications. Le fichier sommets.js permet de mettre à jour les coordonnées sur le canvas de manière dynamique via le formulaire.

Concernant la création des étiquettes, comme le sujet semblait assez ambigu **sur** la création des étiquettes, nous avons mis en place un dossier pour le nom et la description ainsi que du code HTML pur à insérer. Celui-ci sera contrôlé et affiché à la place de la description. Nous avons aussi ajouté la création d'un tableau dont la taille peut augmenter. Il sera intégré à la variable HTML, mais tout cela reste facultatif dans la création de l'étiquette.

3- Mode éditeur

Le mode éditeur est conçu pour permettre aux utilisateurs d’avoir des droits pour gérer et modifier les catalogues, les images et les étiquettes associées. Il est conçu pour donner la responsabilité de création, de modification, de personnalisation et de gestion de données.

Le cheminement du site si l’on est éditeur, débute sur la page d’accueil en étant connecté avec deux boutons soient exécution soit édition. Lorsque l’utilisateur clique sur un des boutons, il se retrouve sur la page Catalogues, avec différentes informations selon son mode sélectionné. S’il est dans le mode édition, il retrouvera tous les catalogues, et pourra en créer de nouveaux. En cliquant sur un catalogue, il accède à la liste d’images associées où il peut les supprimer, réorganiser ou encore en ajouter des nouvelles.

Dans ce mode édition, on retrouve une fonctionnalité de création d’étiquettes via le bouton édition. L’utilisateur accède à une interface non proposée au non-éditeur qui lui permet de placer des points sur l’image pour tracer un polygone. Il peut aussi ajouter ou modifier les descriptions associées à ces zones. Il enregistre ensuite ses étiquettes dans la base de données.

L’utilisateur peut, et ce avant de valider son étiquette, modifier son tracé. Il lui suffit de cliquer sur reset pour l’effacer. Quand le tracé lui convient et qu’il a rempli tous les champs obligatoires, il peut sauvegarder son étiquette en cliquant sur le bouton valider.

4- Mode non-éditeur

Le mode éditeur, également appelé mode exécution, est conçu pour les utilisateurs qui n’ont pas les droits d’édition sur les catalogues. Dans ce mode, l’utilisateur peut uniquement visualiser les catalogues et les images, et obtenir les informations associées aux images. Contrairement au mode éditeur, ce mode permet uniquement une gestion des catalogues, images et étiquettes.

Le cheminement de l’utilisateur connecté à un compte non-éditeur, est très similaire à celui d’un utilisateur connecté à un compte éditeur, il ne possède juste pas les mêmes fonctionnalités. Il n’a pas le choix entre différents modes, il n’a que le mode exécution, qui lui permet de consulter les catalogues.

Il peut ensuite explorer différents catalogues, où il trouvera différents images. Si des utilisateurs ont créés des étiquettes, il pourra les consulter en cliquant sur l’image, puis sur le nom de l’étiquette. Sinon, il pourra toujours consulter les images, mais aucunes étiquettes n’apparaitras. Sur cette page, les images du catalogue sont présentées dans l’ordre défini par l’éditeur. L’utilisateur peut naviguer entre elles en utilisant le boutons précédant et suivant représentés par des flèches.

5-Nos difficultés et résolutions de nos problèmes

Nos premiers défis ont été liés à la manipulation de la base de données. Bien que ces difficultés aient été initiales, elles ont été surmontées grâce à un temps d’adaptation et à une meilleure compréhension des outils et du langage php.

Puis, un de nos gros soucis était l’automatisation de l’affichage images et des étiquettes créées par les éditeurs, afin qu’elles soient intégrées directement dans les catalogues et visibles en temps réel par les non-éditeurs dès leur ajout. Nous avons également rencontré des difficultés techniques pour dessiner les points sur les images et former correctement les polygones. Ces problèmes ont été réglés après de nombreuses recherches.

Par la suite, la mise en place de la fonctionnalité permettant de créer un polygone et de stocker ses coordonnées dans le tableau a également posé des problèmes, nécessitant une attention particulière pour garantir la précision des tracés et leur intégration correcte dans le système.

Enfin, on retrouve une grosse difficulté d’affichage du code html associé aux labels en cliquant sur le bouton car ça me demandait de relier une variable ne js à php, pour ce problème, nous avons utilisé une requête Ajax.

Conclusion :

Le projet nous a permis de développer notre site ainsi que nos connaissances fournir un outil interactif permettant l’étiquetage d’images. Grâce à une interface bien pensée et des fonctionnalités adaptées aux différents profils d’utilisateurs (éditeurs et non-éditeurs), l’application offre une expérience utilisateur fluide et efficace.

Nous avons mis en avant la distinction entre le mode éditeur, destiné à la création et la modification des catalogues, et le mode non-éditeur, orienté vers la consultation, ce qui garantit la possibilité pour chaque utilisateur d’interagir avec le système en fonction de ses besoins et permissions.

Ce projet nous a offert l’opportunité d’approfondir nos compétences en HTML, CSS, JavaScript et PHP, tout en enrichissant notre apprentissage grâce à des ressources en ligne et une entraide active entre les membres de l’équipe. Il a favorisé une collaboration efficace, nous permettant d’unir nos connaissances et de répartir équitablement les tâches pour la création et la réalisation du site. Néanmoins, nous sommes conscients des imperfections visibles. En effet, même si le code est fonctionnel, il reste très vulnérable aux intentions malveillantes, comme les injections SQL ou autres. De plus, nous n’avons pas implémenté la modification des labels, qui était un bonus, en partie à cause du manque de temps. Cependant, ajouter une page pour la modification des labels n’aurait pas représenté une grande difficulté.