GIROUD Antoine

BOIDIN Benoît

Projet de Programmation Web

MIASHS L3

Table des matières

[II. Introduction 3](#_Toc71922763)

[III. Architecture du site 3](#_Toc71922764)

[IV. Schéma conceptuel de la base de données 4](#_Toc71922765)

[A. Modèle conceptuel 4](#_Toc71922766)

[B. Modèle logique 4](#_Toc71922767)

[V. Méthodes et algorithmes significatifs 4](#_Toc71922768)

[VI. Conclusion 5](#_Toc71922769)

# Introduction

Pendant notre projet, nous avions pour but de créer un site web répertoriant les sites d’escalade présents aux alentours de Lyon. Les internautes peuvent, depuis l’accueil, se rendre sur une page présentant les détails de la salle ou de la falaise, tels que le lieu, une photo, la difficulté des voies présentes ainsi que leur nombre, et enfin le type d’escalade, qui permet de connaitre le matériel à emporter.

Afin de rendre l’ensemble plus dynamique et communautaire, les utilisateurs ont la possibilité de s’inscrire, ce qui leur permet de poster un commentaire, de le supprimer et de le modifier.

# Architecture du site

Pendant toute la navigation, l’utilisateur a accès à un menu dynamique, qui s’adapte en fonction de la connexion pour proposer les options cohérentes, ainsi que le pied de page avec les mentions légales.

Pour l’affichage dynamique des pages, nous avons utilisé le même moteur de template que lors des séances de TD.

Nous avons utilisé, pour la forme de notre projet, un template CSS qui donne un aspect professionnel au site.

Le site comprend tout d’abord une page d’accueil qui regroupe tous les sites répertoriés dans une grille, avec leurs images : une requête permet de récupérer les noms et les photos pour les afficher grâce à un modèle.

Pour avoir plus de renseignements, l’utilisateur peut cliquer sur un site pour accéder à une page qui affiche des informations plus précises, ainsi que les commentaires des utilisateurs. Les informations sont, de la même manière que dans la page d’accueil, requises dans la base de données.

Les commentaires peuvent être ajoutés par les utilisateurs à conditions qu’ils soient connectés. La personne à l’origine d’un commentaire peut également le supprimer ou e modifier, depuis un formulaire en bas de la page d’informations.

Le cas échant, les personnes connectées peuvent également ajouter un site d’escalade à la base de données.

Deux pages permettent respectivement la connexion et l’inscription. L’utilisateur doit renseigner un pseudonyme et un mot de passe pour avoir accès aux fonctions disponibles aux membres.

# Schéma conceptuel de la base de données

## Modèle conceptuel

## Modèle logique

Site (Idsite, Nomsite, Localisation, Niveau, Nbvoies, Image, #Idtype)

Type (Idtype, Nomtype)

Message (Idmessage, Date, Contenu, #Idgrimpeur, #Idsite)

Grimpeur (Idgrimpeur, Login, Passwd)

# Méthodes et algorithmes significatifs

J’ai mis ce à quoi je pensais mais si tu trouves que ce n’est pas pertinent ou qu’il manque des chose n’hésite pas à modifier.

## Requêtes afficher.

* Connexion à la base de données.
* Préparation puis exécution de la requête.
* Association des noms et valeurs pour l’affichage du modèle, en fonction du template voulu.

## Ajout d’un site (avec l’image etc…).

* Déplacement du fichier original dans un dossier d’images.
* Insertion des informations et du nom de l’image dans la base.

## Connexion.

* Requête pour chercher les identifiants dans la base ;
* Si les identifiants sont bons, on les met dans les variables de session.
* Redirection vers le menu.

## Inscription.

* Requête pour ajouter les identifiants
* Appel à la classe connexion.

## Affichage du template des commentaires avec les boutons « supprimer » et « afficher ».

* Vérifier que l’utilisateur est connecté.
* Ajout / modification puis redirection vers la page d’infos.

# Conclusion

## Problèmes rencontrés.

* Conflit entre les commentaires du modèle de style et le moteur de template.
* Problème pour inclure le fichier des identifiants (connect.inc.php) dans les classes qui s’en servent.

## Bilan.

* Séparer le plus possible tous les langages pour plus de clarté.
* Faire en sorte d’utiliser le moins de lignes de code possible.