Rapport sur le projet de site d’escalade

Table des matières :

Introduction…………………………………………………………………………………………………………………………2

Partie 1: Organisation du Projet…………………………………………………………………………………………….2

1. Répartition des tâches dans le groupe………………………………………………………………….2

2. Modalités d’organisation………………………………………………………………………………………2

3. Difficultés rencontrées…………………………………………………………………………………………3

Partie 2 : Section technique de la création de la base de donnée…………………………..……………3

1. Conception des schéma entité relations………………………………………………………………3

2. Conception du schéma relationnel……………………………………………………………………..3-4

Partie 3 : Avancement du projet…………………………………………………………………………………………4-5

1. Interface du site web …………………………………………………………………………………………5
2. Quelques requêtes SQL du projet………………………………………………………………………6

Partie 4 : Amélioration du projet……………………………………………………………………………………….6

1. Amélioration du site web…………………………………………………………...........................7

2. Améliorations au niveau du code……………………………............………………………………8

3. Informations sur le code.……………………………………………………………………………………7

**Introduction**:

Le projet qui clôture l’UE Base de données simule un client qui souhaite engager des développeurs pour la conception d’un site web d’une association fictive d’escalade. Ici, notre groupe joue les développeurs qui créent le site web et la base de données.

L’objectif est d’alors de modéliser dans un premier temps la base de données en utilisant le modèle conceptuel de données puis de créer le schéma relationnel de données. Puis dans un second temps, l’implémentation en utilisant le langage html et css du site web et la connexion avec la base de données en utilisant les langages SQL (PostgreSQL) et python et les modules flask, passlib.context et psycopg qui sont des modules du langage python.

**Partie 1 : Organisation du projet**

1. **Répartition des tâches dans le groupe**

Afin de bien entamer le projet, le travail a été divisé en 3 parties : Conception de la base de données (schéma entité association et schéma relationnel), implémentation sous PostgreSQL et sous python/html/css du site web. Pour la création de la base de données, Christophe TARATIBU s’est occupé de l’édition du schéma entité relation et Benjamin PEGOURIE du schéma relationnel de la base. En ce qui concerne de la partie code du projet (PostgreSQL pour la base de données et python/html/css/SQL), Christophe TARATIBU s’en est entièrement occupé.

1. **Modalités d’organisation**

La majorité du travail a été fait à la maison et ces derniers ont servi à mettre en commun le travail fait à la maison. Des sessions sur Discord ont été menés pour structurer la suite du projet.

1. **Difficultés rencontrées**

En ce qui concerne la conception de la base de données, les problèmes encourus sont plutôt moyens l’ajout des guides d’escalade a été dure que ce soit dans la base ou dans le code final. Mais dans l’ensemble, le projet s’est fait de manière correcte dans la création du site web avec Jinja/html/css et la connexion avec la base de données avec le langage python.

**Partie 2 : Section technique sur la création de la base de données.**

1. **Conception du schéma entité association**

En ce qui concerne la création de ce schéma, plusieurs tables et relations ont été faites pour répondre aux besoins du client. Ces tables contiennent des clés primaires et les relations entre les tables leurs ajoutent des clés étrangères ou deviennent les tables en fonction des cardinalités présentes dans les tables.

1. **Conception du schéma relationnel**

Pour la conception du schéma relationnel, voici ici sa représentation :

1. site\_escalade(id\_site,nom\_site,localité)
2. niveau(nivFR, nivUS,nivGB)
3. type\_voie(code,nom)
4. adherent(id\_adherent,email,mdp,nom,prenom,niv\_grimpeur,niv\_guide)
5. voie(id\_voie,nom\_voie,longueur,id\_site,code\_voie,nivFR)
6. sortie(id\_sortie,court\_desc,date,id\_site,nivFR,organisateur)
7. voie\_suivante(id\_voie,autre\_id\_voie)
8. cordee(id\_cordee,nom\_cordee,participants)
9. exerce(id\_adherent,id\_site,liste\_localité)
10. inscrit(id\_adherent,id\_sortie,nbre\_inscriptions)
11. appartient(id\_adherent,id\_cordee)
12. voie\_grimpee(id\_cordee,id\_voie,manière,date\_escalade)
13. voie\_escaladees(id\_adherent, id\_voie, nbre\_de\_fois)

Clés étrangères :

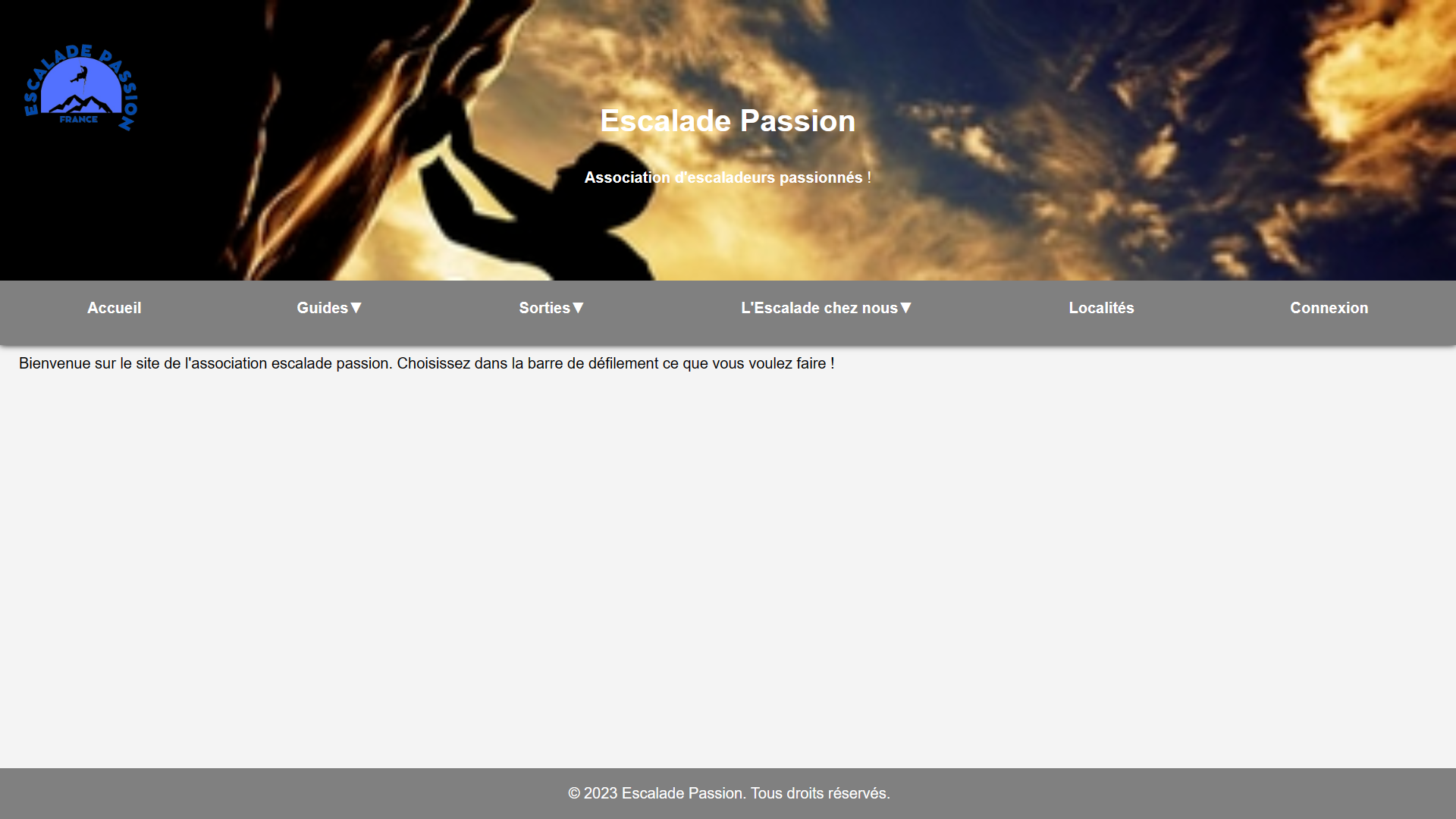
* voie(id\_site,code,nivFR) fait référence respectivement à site\_escalade(id\_site), type\_voie(code), niveau(nivFR)
* voie\_suivante(id\_voie,autre\_id\_voie) fait référence à voie(id\_voie)
* sortie(id\_site,nivFR,id\_adherent) fait référence respectivement à site\_escalade(id\_site), niveau(nivFR), adherent(id\_adherent)
* adherent(niv\_grimpeur,niv\_guide) fait référence respectivement à niveau(nivFR), niveau(nivFR)
* exerce(id\_adherent,id\_site) fait référence respectivement à adherent(id\_adherent) site\_escalade(id\_site)
* inscrit(id\_adherent,id\_sortie) fait référence respectivement à adherent(id\_adherent), sortie(id\_sortie)
* appartient(id\_adherent,id\_cordee) fait référence respectivement à adherent(id\_adherent) et cordee(id\_cordee)
* voie\_grimpee(id\_cordee,id\_voie) fait référence respectivement à cordee(id\_cordee), voie(id\_voie)
* voie\_escaladee(id\_adherent,id\_voie) fait référence respectivement à adherent(id\_adherent), voie(id\_voie)

Les attributs surlignés sont des clés étrangères qui sont présentes dans les tables.

Le code de la base à été faite en SQL sous PostgreSQL. Des requêtes de suppression de table (DROP TABLE CASCADE) permettent d’éviter des problèmes lors de la création de ces dernières. Une grande partie des clés primaires dans les tables sont de type SERIAL dans le but de générer facilement des d’identifiants et faciliter les jointures entre les tables. Pour la table des niveaux, la clé est de type VARCHAR (chaîne de caractères) 2 car il représente le niveau des voies ou des grimpeurs. Certains attributs des tables ont un attribut de type VARCHAR variable entre 30 et 80 afin de permettre une plus grande liberté pour les utilisateurs et lors de l’insertion de valeurs plus longues. Des tests d’insertion dans les tables ont été réalisés pour tester leur validité.

**Partie 3 : Petit manuel utilisateur**

**1. Interface du site web**



Img 1 : Accueil du site web escalade

L’image ci-dessus montre l’accueil du site web. En cliquant sur le logo bleu en haut ,à gauche, l’utilisateur retourne à l’accueil. La barre de de navigation (en gris sous l’image de fond) permet à l’utilisateur de naviguer sur les autres pages du site. La barre contient 6 directions vers d’autres pages du projet :

* Accueil : Permet à l’utilisateur de retourner à l’accueil.
* Guides : Permet à l’utilisateur de voir les guides existants et devenir guide s’il est adhérent.
* Sorties : Permet à l’utilisateur s’il est connecté de rejoindre ou créer une sortie. Il peut alors clôturer la sortie qui aura pour effet de créer une cordée ou la supprimer qui l’effacera de la base de données.
* L’escalade chez nous : Permet à l’utilisateur de consulter les adhérents, les sorties, voies grimpées, les niveaux et de rechercher les voies.
* Localités : Le lien contenu dans ce mot renvoie l’internaute sur une page où il pourra cliquer sur une image qui représentent les principales localités. Par la suite, il aura accès aux principaux sites d’escalades avec leurs voies incluses.
* Connexion : Permet à l’internaute de se connecter afin de créer des sorties et de les gérer.

**2. Quelques requêtes SQL du projet**

Pour ajout de guides nous avons décider d’inclure dans la table adhérent un attribut niv\_guide, (niveau de guide) qui est inférieur au niveau de grimpeur.

Ainsi la requête SQL : "UPDATE adherent SET niv\_guide = %s WHERE id\_adherent = %s" octroie à l’adhérent un niveau dès lors que l’utilisateur s’il est inscrit et souhaite devenir guide.

L’affichage des voies est effectué lorsque l’internaute veut voir les voies contenues dans un site d’escalade après avoir cliqué sur Localité dans le site web. Ce choix est fait par cette requête : "SELECT id\_voie,nom\_voie,longueur\_voie,nivfr,code\_voie FROM voie NATURAL JOIN site\_escalade WHERE localite = %s AND nom\_site = %s". Les données sont récupérées par les clics sur les noms de voies qui contiennent des liens avec pour information le nom du site et le nom de la voie.

Si un utilisateur connecté veut s’inscrire à une sortie, il pourra voir se voir et les autres personnes qui sont inscrites avec lui grâce à cette requête : "SELECT adherent.id\_adherent,nom,prenom FROM adherent JOIN inscrit ON adherent.id\_adherent = inscrit.id\_adherent WHERE inscrit.id\_sortie = %s" ; L’identifiant de sortie est récupéré dès lors que l’internaute choisit d’y s’inscrire.

Aussi, l’utilisateur peut voir le nombre d’adhérents dans une sortie peut y participer s’il considère comme étant un argument de popularité. Cette fonctionnalité est faite par cette requête : 'SELECT id\_sortie,COUNT(id\_sortie) As nbe\_inscrits,adherent.prenom,adherent.nom,adherent.id\_adherent,sortie.site\_souhaite,sortie.court\_description,sortie.nivfr,sortie.date\_sortie FROM inscrit NATURAL JOIN sortie JOIN adherent ON adherent.id\_adherent = sortie.organisateur GROUP BY (id\_sortie,prenom,nom,adherent.id\_adherent,site\_souhaite,court\_description,nivfr,date\_sortie)' ;

L’internaute peut s’il le souhaite, effectuer des recherches sur les voies selon ses préférences et découvrir la première cordée à avoir ouverte la voie. Cependant il se pourrait qui n’y ai pas de cordées qui ont escaladée telles voies ou que des voies soient existantes. La requête qui exécute ce choix dépend du filtre de longueur choisie par l’utilisateur :

Si l’utilisateur veut une longueur choisie entre x,y mètres (x,y des nombres) :

SELECT nom\_voie,id\_voie,longueur\_voie,code\_voie,nivfr FROM voie NATURAL JOIN site\_escalade WHERE %s <longueur\_voie AND longueur\_voie< %s AND nom\_site = %s AND nivfr = %s"

Si l’utilisateur veut une voie plus grande que x :

"SELECT nom\_voie,id\_voie,longueur\_voie,code\_voie,nivfr FROM voie NATURAL JOIN site\_escalade WHERE longueur\_voie>%s AND nom\_site = %s AND nivfr = %s"

Dans ces requêtes ci-dessus, les données sont récupérées par les filtres choisis par l’utilisateur.

**Partie 4 : Amélioration du projet :**

**1. Amélioration du site web :**

Le site web est composé d’une barre de navigation qui contient une barre de navigation. Cette dernière fait apparaître un menu déroulant dès lors que l’utilisateur y met la souris des mots qui ont une flèche. Le logo du site a été fait par Christophe TARATIBU et redirige l’utilisateur lorsqu’il clique dessus. Un logo du profil apparaît si un utilisateur s’y connecte au site. En cliquant, il a accès à son profil et peut se déconnecter. Les connexions sont gérées par les sessions de Flask. Des crédits ont été ajoutés au site.

**2. Améliorations au niveau du code :**

Le projet final est contenu dans un dossier. Dans ce même dossier, 2 fichiers : main.py qui connecte la base de données au site web. Puis 2 sous dossiers :

* Static, qui possède 8 fichiers : 7 images et le code css du site web
* Templates, ayant 29 pages html qui structurent le site web.

Les connexions sont gérées par des sessions proposées par Flask. Une variable booléenne permet d’afficher si c’est vrai le profil indiquant qu’une personne est connectée. Les données récupérées par les requêtes permettent le bon fonctionnement du site web. L’onglet du site web est dimensionnable. Cependant, L’image de fond est parfois déformée.

**Conclusion :**

Ce projet était quelque chose de très intéressante à travailler. Il nous a permis de mettre en valeur la création de la base de données (schéma entité relation et schéma relationnel) et du site en rapport à l’escalade. Ce projet nous à donné beaucoup de liberté notamment en ce qui concerne le design du site et de répondre aux mieux aux attentes du client.