

# Projet escalade

## **presentation :**

le but du projet etait de lier une base donnée a un site qui doit permettre une inscription et connection pour ses membres et ainsi leur proposer des services comme pouvoir voir le top des sites les plus escaladés ,pouvoir rechercher des guides pour certaines voies ,voie que l'on peut aussi rechercher avec plusieurs criteres dans une section qui y est dedié.  
Les membres ,une fois connecté et depuis l'onglet profil ,doivent pouvoir créer et rejoindre des sorties , voir les profils des autres membres ainsi que le leur et peuvent voir , rejoindre et créer des cordées.

## **Manuel utilisateur :**

une fois le site web lancé vous aurez acces a l'accueil sur lequel vous retrouverez le top 5 des sites les plus escaladés.

### **Acces sans inscription :**

- section pour rechercher des guides ,qui vous affiche les guides n'ayant rien proposé depuis plus d'un an et vous pourrez rechercher un guide pour une localité en particulier ou pour etre plus precis pour une voie en particuliere directement,
- section pour rechercher des voies, dans laquelle vous aurez le choix de rechercher en entrant une longueur minimum ou/et maximum , une difficulté minimum et/ou maximum , par le type de la voie que vous chercher ,ou directement par la localité ou le site d'escalade dans lesquelles vous cherchez des voies.

### **Acces seulement apres inscription et connection en tant que membre :**

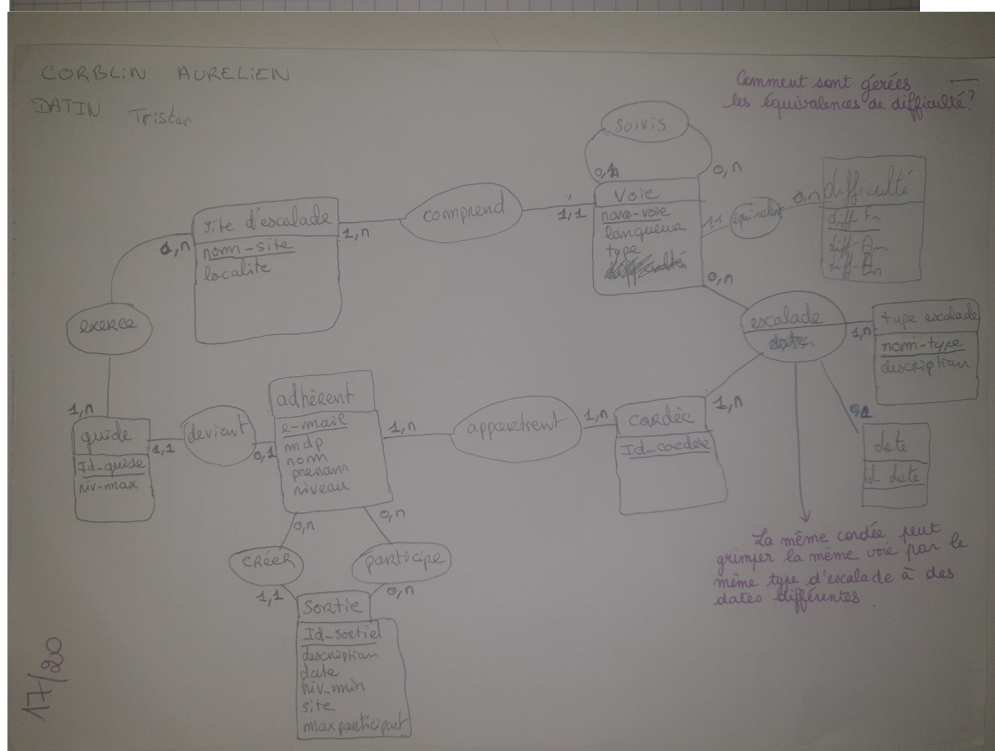
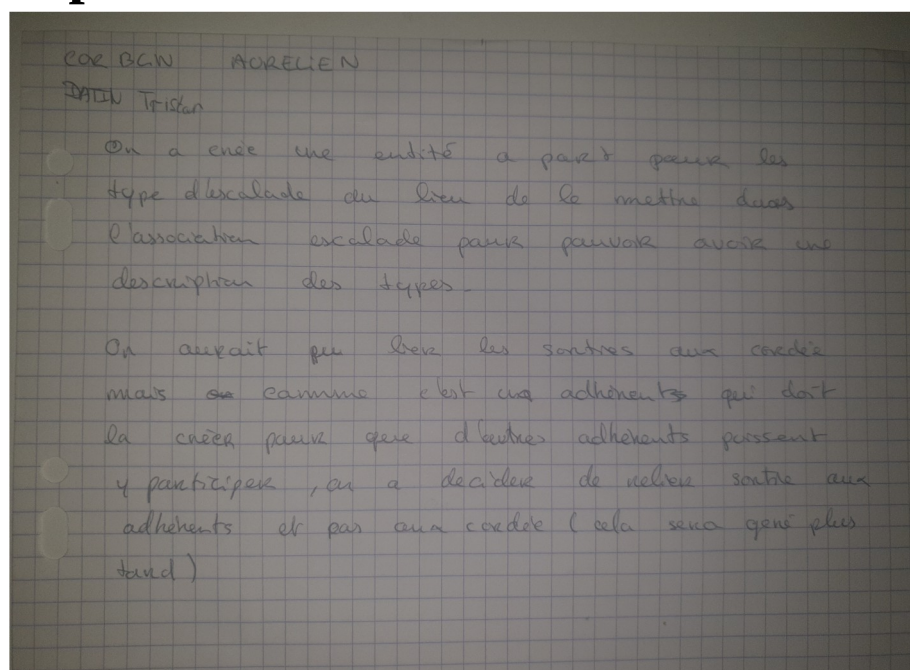
une fois connecté vous avez acces aux sections :

- de sorties , dans laquelle vous pouvez créer de nouvelle sorties , valider ou annuler vos proposition de sorties ou encore rejoindre des sorties créer par d'autre membres.
- de profil , accessible depuis l'icone en haut a droite de la page, vous pourrez donc modifier vos informations personnelle tel que votre nom, prenom et niveau. Vous pourrez egalement consulter votre historique de voies grimpés , les cordées auxquels vous appartenez, et vous avez acces a une section pour devenir guide ,dans laquelle vous pouvez entrer votre niveau maximum dans laquelle vous pouvez etre guide (inferieur ou egal a votre niveau d'escalde)

puis vous pourrez entrer les localité ou vous voulez exercer,  
 enfin sur votre profil vous pouvez aussi vous deconnecter.  
 Une fois sur votre profil deux nouvelle section apparaissent en haut de la page :

- les membres : dans cette section vous avez acces a la liste de tout les membres inscrit et ainsi en cliquant dessus vous pouvez voir leurs profils avec leurs historiques des voies grimpees et des cordées.
- les cordées : dans cette section vous pouvez voir toute les cordées créer par les autres membres et vous pouvez les rejoindre.  
 Vous pouvez egalement créer vos propres cordées.

## Etape 1 :



## Etape 2 :

Schema relationnel :

difficulte(diff\_FR, diff\_AM, diff\_EN)  
adherents(email, mdp, nom, prenom, niveau)  
sorties(id\_sorties, sortie\_desc, date\_sortie, niv\_min, site\_sortie, max\_part, createur)  
guides(id\_guide, niv\_max)  
siteEscalade(nom\_site, localite)  
cordee(nom\_cordee)  
typesEscalade(nom\_type, desc\_type)  
voie(nom\_voie, longueur, type\_voie, nom\_site, diff)  
suit(voie\_suivi, voie2)  
participe(id\_ad, id\_sortie)  
exerce(id\_guide, localite)  
appartient(id\_ad, nom\_cordee)  
Escalade(esc\_date, nom\_cordee, nom\_voie, nom\_type)

FK : adherents(niveau) fait ref à difficulte(diff\_FR)  
FK : sorties(niv\_min) fait ref à difficulte(diff\_FR)  
FK : guides(id\_guide) fait ref à adherents(email)  
FK : guides(niv\_max) fait ref à difficulte(diff\_FR)  
FK : voie(nom\_site) fait ref à siteEscalade(nom\_site)  
FK : voie(diff) fait ref à difficulte(diff\_FR)  
FK : voie(suit) fait ref à voie(nom\_voie)  
FK : participe(id\_ad) fait ref à adherents(email)  
FK : participe(id\_sortie) fait ref à sorties(id\_sortie)  
FK : exerce(id\_guide) fait ref à guides(id\_guide)  
FK : exerce(localite) fait ref à sitesEscalade(localite)  
FK : appartient(id\_ad) fait ref à adherents(email)  
FK : appartient(nom\_cordee) fait ref à cordee(nom\_cordee)  
FK : Escalade(nom\_cordee) fait ref à cordee(nom\_cordee)  
FK : Escalade(nom\_voie) fait ref à voie(nom\_voie)  
FK : Escalade(nom\_type) fait ref à typeEscalade(nom\_type)

reponse aux demandes supplementaires :

pour les 5 sites avec le plus de voies escaladées depuis le debut de l'année :

```
CREATE VIEW top_5_sites AS
(
select nom_site from escalade natural join voie natural join appartient
where (SELECT EXTRACT(YEAR FROM esc_date)) = (SELECT EXTRACT(YEAR FROM
CURRENT_DATE))
group by nom_site order by count(nom_voie) desc limit 5
);
```

pour les guides n'ayant pas proposé de sortie depuis plus d'un an :

```
CREATE VIEW controle_guide AS
(
select id_guide, nom, prenom from guides join adherents on id_guide = email
except
select id_guide, nom, prenom from (guides join adherents on id_guide = email)
join sorties on id_guide = createur
where date_sortie >= (CURRENT_DATE - INTERVAL '1 YEAR')
);
```

texte justificatif :

on a decidé de mettre on update en cascade pour les tables : adherents, sorties, guides, voie, participe, exerce, appartient, Escalade. Pour que si on modifie un niveau parce que on s'est trompé par exemple , ca le modifie partout ou bien si un utilisateur veut changer d'email, il faut que le changement se fasse partout.

Pour la suppression on a laissé en restrict car on ne veut pas les historiques des voies escalader en supprimant un adherents ou perdre des limite de niveau en supprimant une difficulté.

On a également mis des contrainte not null pour le mdp des adherents car il ne peuvent pas se connecter sans mdp, pour niv\_max des guides car on veut que les guides est forcément un niveau limite pour pouvoir accompagner des gens du meme niveau ou niveau inferieur, et on a mis not null pour toutes les colonnes de la table sorties car on veut absolument toute les informations de la sortie pour pouvoir la créer.

On ne peut pas encore garantir qu'un guide n'exerce pas dans une localite ou il n'a pas le niveau pour exercer ou que des adherents ne participent pas des sorties dont ils n'ont pas le niveau.

### **Description technique :**

- si on essaye de créer une sortie avec une information manquante cela ne fonctionnera pas.
- pour la recherche de voie si aucun champs n'est rempli , la recherche affichera toute les voies existante ,idem si deux champs sont contradictoire (ex : site : autoire et localite : nice). Vous n'etes ,par contre , pas obliger de remplir tout les champs pour faire une recherche de voie
- dans les sorties auquel vous participez et celles a rejoindre seule celles qui sont encore en attente ou déjà valider avec une date qui plus tard que le jour actuelle (ou bien une date de sorite du jour actuelle)
- en visitant les differents profils des membres vous pouvez cliquer directement sur les voies dans leur historique ca renvera directement sur la page des caracteristique de la voie en question.
- dans les cordees il vous est proposé dans les cordees a rejoindre seulement celle auxquels vous n'appartenez pas déjà , et quand vous creez une nouvelle cordee vous y appartenez directement(idem quand creez une proposition de sortie).

Dans la fonction gest\_sortie() :

au moment d'une annulation on fait bien attention de faire :

```
DELETE FROM participe WHERE id_sortie = %s
```

```
DELETE FROM sorties WHERE id_sortie = %s
```

et pas l'inverse , cela permet de ne pas avoir de probleme cote base de donnee.

On supprime donc d'abord tout les participants puis on supprime la sortie pour l'annuler.

Dans la fonction sortie :

on utilise cette requete qui permet de recuperer toute les sorties en dehors de celle ou nous somme le createur de cette derniere ou que l'on y participe déjà (grace au except)

```
select * from sorties natural left join (select id_sortie,count(id_ad) as nbpart from participe group by id_sortie)as
```

```
nb where niv_min <= (select niveau from adherents where email = %s) and createur <> %s and max_part > nbpart and date_sortie > now()
```

```
except select distinct
```

```
sorties.id_sortie,sortie_desc,date_sortie,niv_min,site_sortie,max_part,createur,nbpart from sorties natural left join \
```

```
(select id_sortie,count(id_ad) as nbpart from participe group by id_sortie)as nb,participe \ where participe.id_sortie=sorties.id_sortie and id_ad = %s
```

dans la fonction prof\_voie (qui permet l'affiche des 'profil' des voie)

```
cur.execute("select distinct nom_type from typesescalade")
```

```
type = [elem.nom_type for elem in cur.fetchall()]
```

```
for elem in type:
```

```
cur.execute("select distinct nom_type,esc_date,nom_cordee from voie natural join sitesescalade natural join escalade where id_voie = %s and nom_type = %s order by esc_date limit 1",(id,elem))
```

```
tmp = cur.fetchall()
```

```
if tmp:
```

```
d.append(tmp)
```

nous avons ce morceau de code qui fait une boucle sur les differents types d'escalade , pour lesquelles , a chaque fois , on recupere le type , la date et la cordee qui a escaladé (en premiere pour ce type) en faisant un trie par la date pour avoir de la plus acienne a la plus recente et en limitant a 1 pour ne garder que la plus ancienne.

Les autres requetes ou bout de code utilisé sont soit 'basique' ,soit du meme style que celle ci , il a donc pas ete trouver necessaire d'en mettre encore plusieurs pour les expliquer.

### Eventuelles piste d'amelioration :

pour ameliorer tout ca , on aurait pu ajouter une possible annulation de participation a un sortie , de pouvoir quitter une cordee ou bien pouvoir faire une suppression de compte (pour se desinscrire)

### **organisation :**

Pour les deux premieres partie elles ont été faite totalement directement ensemble.

Pour ce qui est de la derniere partie pour le site web , le rapport etc :

je l ai faite tout seul car je n'est pas eu de nouvelle de Tristan pendant les deux dernieres semaines(pendant lesquels il n'était pas en cours non plus ,donc je ne l'ai pas vu en presentiel egalelement )