



TM

PARIS 2024



Cahier des charges techniques

Fait par

KHABALI NAWAF

Date

05 - 12 - 2023

Version Blanc ici



-=Sommaire=-

1 Contexte du projet	3
1.1. Présentation du projet	3
1.2. Réponses aux attentes	3
2 Besoins fonctionnels	4
3 Ressources nécessaires à la réalisation du projet	4
3.1. Ressources matérielles	4
3.2. Ressources logicielles	4
4 Gestion du projet	5
5 Conception du projet	6
5.1. Le front-end	6
5.1.1. Wireframes	9
5.1.2. Maquettes	10
5.1.3. Arborescences	10
5.2. Le back-end	10
5.2.1. Diagramme de cas d'utilisation	10
5.2.2. Diagramme d'activités	11
5.2.3. Modèles Conceptuel de Données (MCD)	11
5.2.4. Modèle Logique de Données (MLD)	11
5.2.5. Modèle Physique de Données (MPD)	12
6 Technologies utilisées	12
6.1. Langages de développement Web	12
6.2. Base de données	12
7 Sécurité	13
7.1. Login	13
7.2. Cryptage des mots de passe	13
7.3. Protection des pages administrateurs	13
7.4. Protection contre les attaques XSS (Cross-Site Scripting)	13
7.5. Protection contre les injections SQL	13



1 *Contexte du projet*

1.1. Présentation du projet

Votre agence web a été sélectionnée par le comité d'organisation des jeux olympiques de Paris 2024 pour développer une application web permettant aux organisateurs, aux médias et aux spectateurs de consulter des informations sur les sports, les calendriers des épreuves et les résultats des JO 2024.

Votre équipe et vous-même avez pour mission de proposer une solution qui répondra à la demande du client.

Le projet doit être rendu au plus tard le **22 mars 2024**.

1.2. Réponses aux attentes

- Lien vers GitHub du Projet :

Github | <https://github.com/K02Nawaf/jo-2024-khabali-nawaf>

- Hébergement :

InfinityFree | <http://jo-2024-khabali-nawaf.free.nf/>



2 *Besoins fonctionnels*

Le site web devra avoir une partie accessible au public et une partie privée permettant de gérer les données.

Les données seront stockées dans une base de données relationnelle pour faciliter la gestion et la mise à jour des informations. Ces données peuvent être gérées directement via le site web à travers un espace administrateur.

3 *Ressources nécessaires à la réalisation du projet*

3.1. Ressources matérielles

- Ordinateur
- Connexion internet

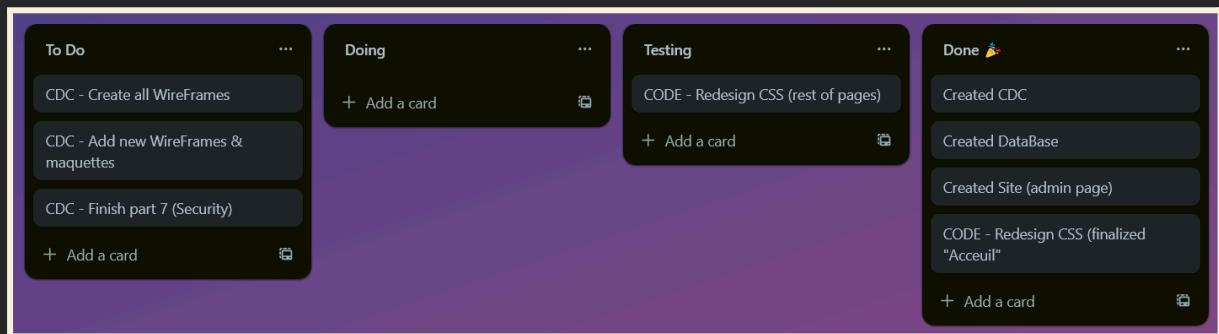
3.2. Ressources logicielles

- Visual studio code (l'IDE choisi)
- Github (enregistrement du projet dans le cloud)
- Mamp (Apache server, SQL database)
- Trello (organisation du projet)
- Mocodo (Creation du MCD et MLD)



4 Gestion du projet

Pour réaliser le projet, nous utiliserons la méthode Agile Kanban. Nous utiliserons également l'outil de gestion de projet en ligne Trello.



Nous travaillons également sur GitHub, plateforme de développement collaboratif.



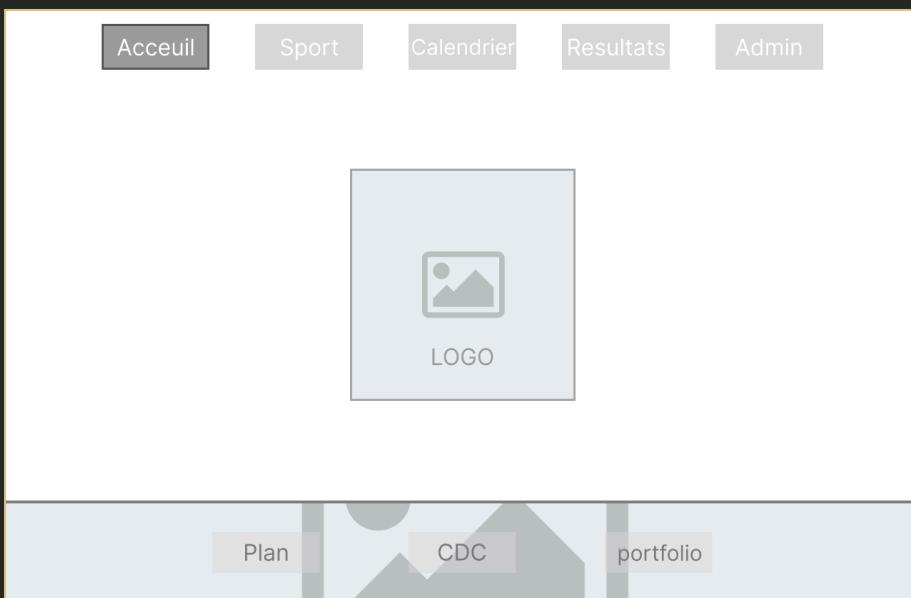
5 Conception du projet

5.1. Le front-end

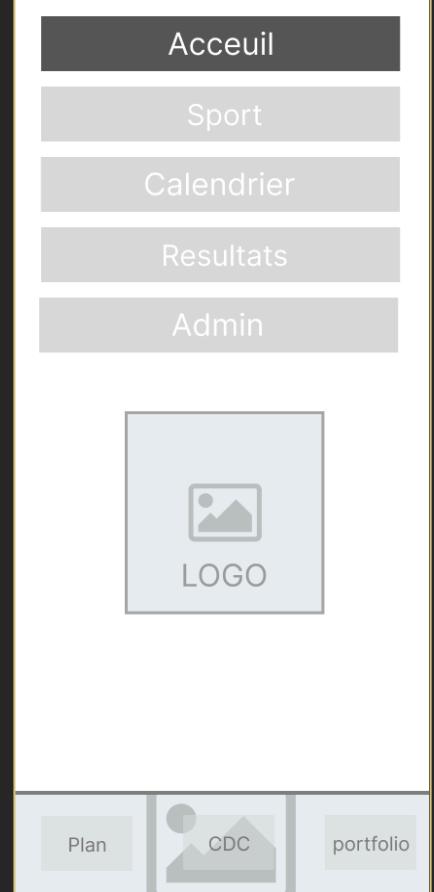
5.1.1. Wireframes

WireFrame.Index

PC



Responsive





WireFrame.Ajout-sports

PC

[Accueil](#)
[G-Sport](#)
[G-Linux](#)
[G-Calendrier](#)
[G-Pays](#)

[G-Genres](#)
[G-Athlete](#)
[G-Resultats](#)
[G-Users](#)
[Deconnexion](#)



Ajouter un Sport

Label :

Plan
CDC
portfolio

Responsive

[Accueil](#)

[Gestion du Sport](#)

[Gestion du Calendrier](#)

[Gestion du pays](#)

[\(...\)](#)



Ajouter un Sport

Label :

Plan
CDC
portfolio



WireFrame.Liste-Sport

PC

Acceuil Sport Calendrier Resultats Admin


Liste des sports

Sport

Plan CDC portfolio

Responsive

Acceuil

Sport

Calendrier

Resultats

Admin


Ajouter un Sport

Sport

Plan CDC portfolio



5.1.2. Maquettes

Maquette.Index

PC

The desktop view shows a dark-themed website. At the top is a navigation bar with five items: Accueil (highlighted in yellow), Sports, Calendrier des évènements, Résultats, and Accès administrateur. Below the navigation is the Paris 2024 logo (flame inside a circle) and the text "PARIS 2024". A large wheelchair accessibility icon is centered below the text. At the bottom is a yellow footer bar with two buttons: "Cahier de charge" and "Portfolio".

Responsive

The responsive view shows a dark-themed website. The navigation menu is displayed vertically on the left side, with rounded rectangular buttons for Accueil (highlighted in yellow), Sports, Calendrier des évènements, Résultats, and Accès administrateur. Below the navigation is the Paris 2024 logo and the text "PARIS 2024". A large wheelchair accessibility icon is centered below the text. At the bottom is a yellow footer bar with two buttons: "Cahier de charge" and "Portfolio".



Maquette.Liste-sport

PC

Sport

- Athlétisme
- Boxe
- Cyclisme
- Escalade
- Gymnastique
- Lancer
- Natation
- Running in the 90s
- Saut en hauteur
- Saut en longueur

[Cahier de charge](#) [Portfolio](#)

Responsive

[Accueil](#)

[Sports](#)

[Calendrier des évènements](#)

[Résultats](#)

[Accès administrateur](#)

Sport

- Athlétisme
- Boxe
- Cyclisme
- Escalade
- Gymnastique

[Cahier de charge](#) [Portfolio](#)



Maquette.Ajoute-sport

PC

- [Accueil Administration](#)
- [Gestion Sports](#)
- [Gestion Lieux](#)
- [Gestion Calendrier](#)
- [Gestion Pays](#)
- [Gestion Genres](#)
- [Gestion Athlètes](#)
- [Gestion Résultats](#)
- [Gestion Utilisateur](#)
- [Déconnexion](#)



Ajouter un Sport

Nom du Sport :

[Ajouter le Sport](#)

[Retour](#)

[Cahier de charge](#) [Portfolio](#)

Responsive

- [Accueil Administration](#)
- [Gestion Sports](#)
- [Gestion Lieux](#)
- [Gestion Calendrier](#)
- [Gestion Pays](#)
- [Gestion Genres](#)
- [Gestion Athlètes](#)
- [Gestion Résultats](#)
- [Gestion Utilisateur](#)
- [Déconnexion](#)



Ajouter un Sport

Nom du Sport :

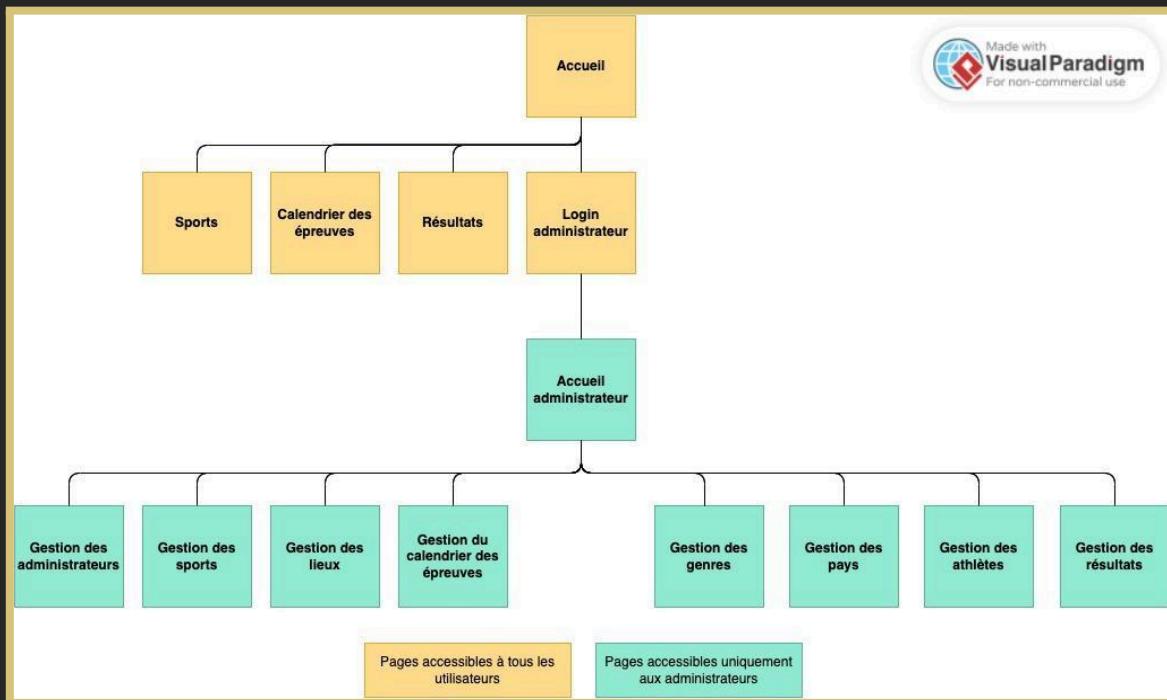
[Ajouter le Sport](#)

[Retour](#)

[Cahier de charge](#) [Portfolio](#)

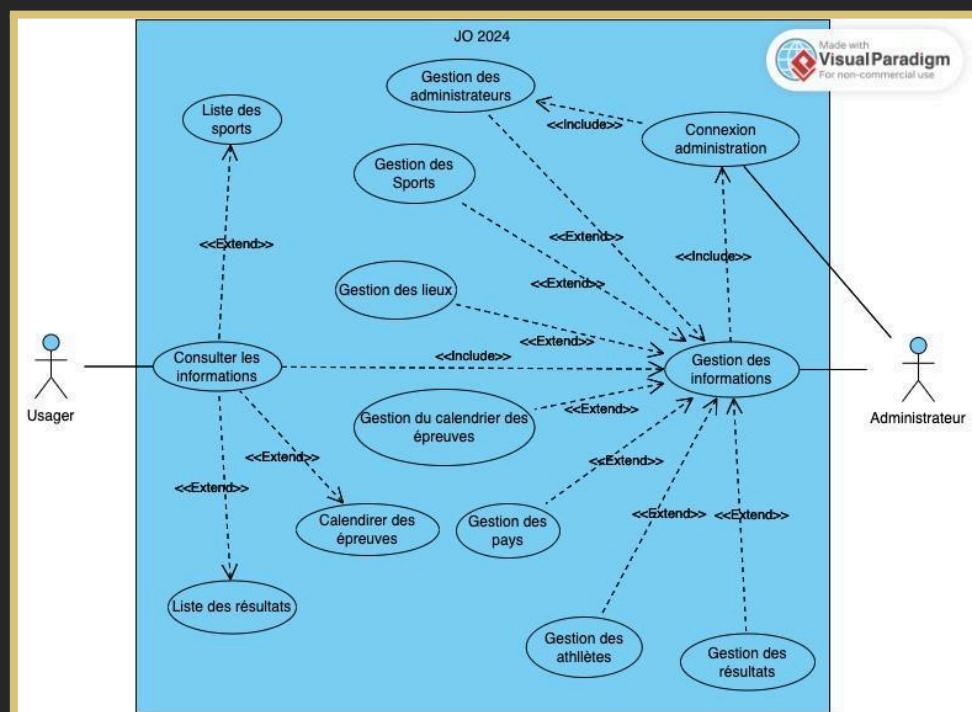


5.1.3. Arborescences



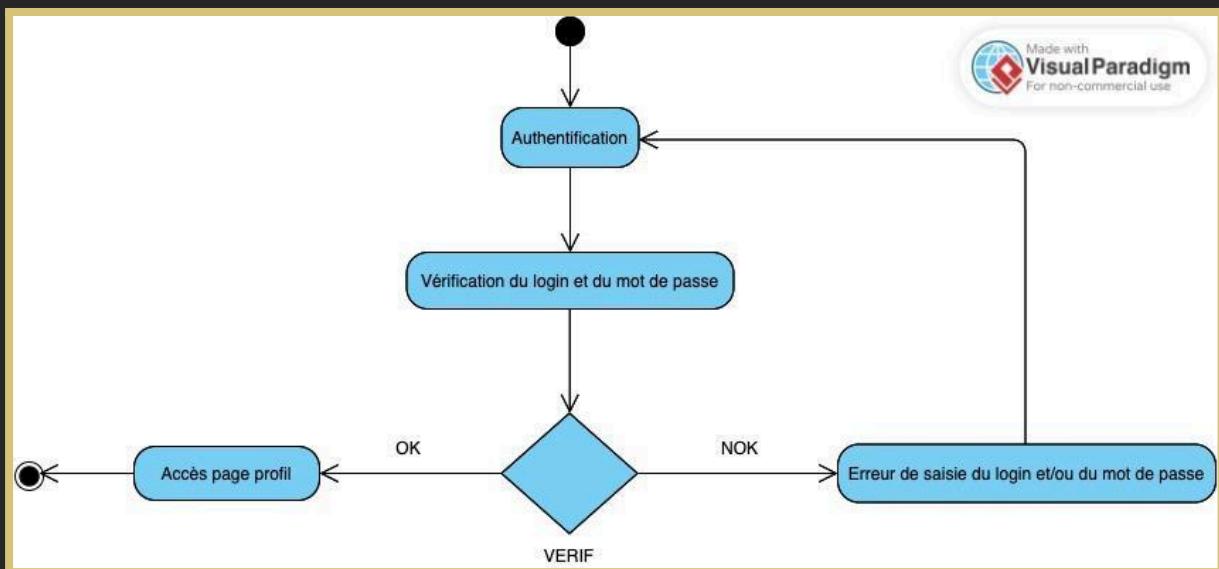
5.2. Le back-end

5.2.1. Diagramme de cas d'utilisation

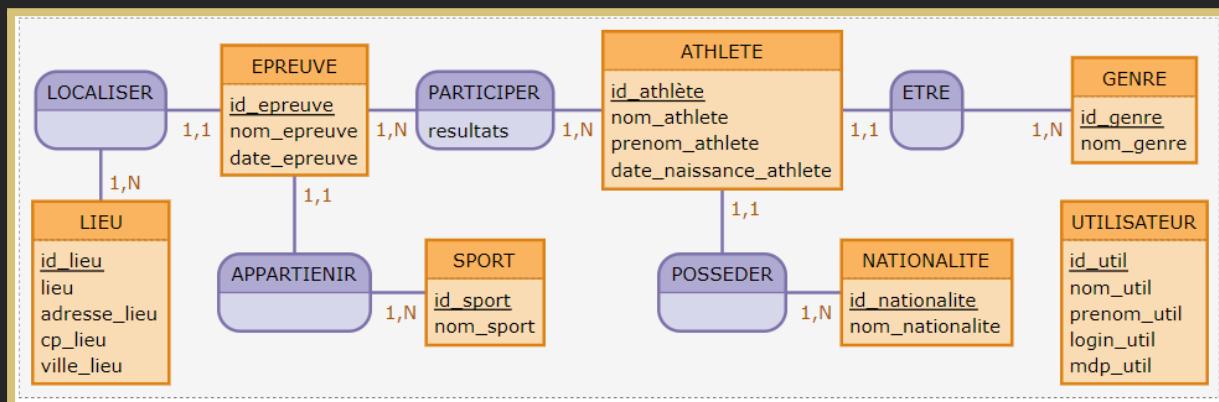




5.2.2. Diagramme d'activités



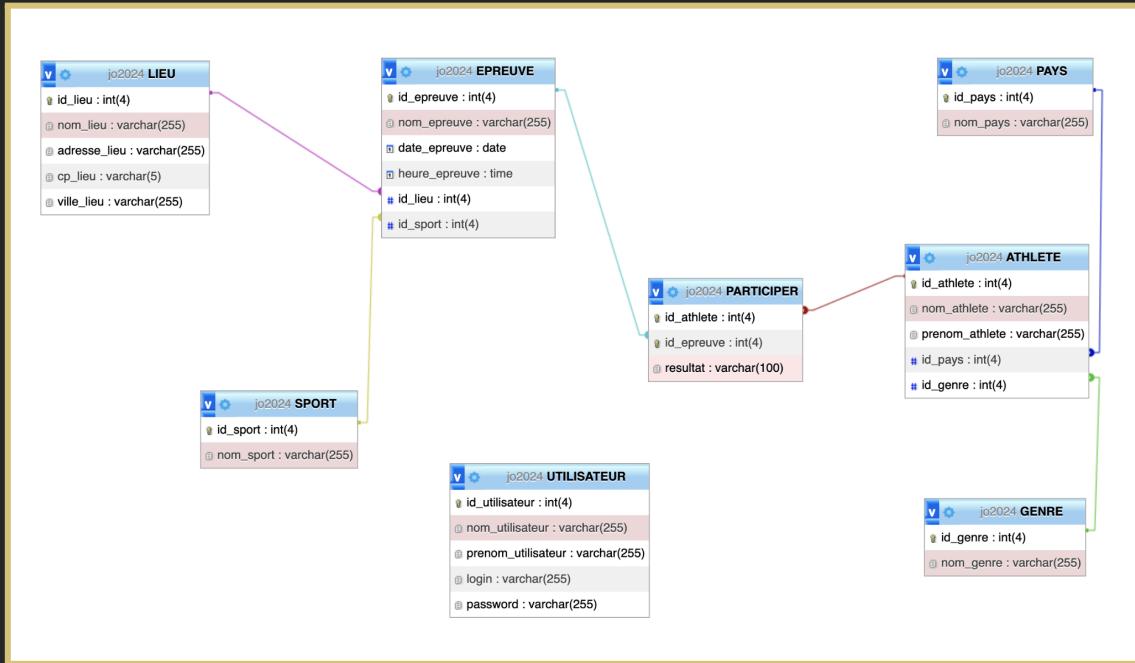
5.2.3. Modèles Conceptuel de Données (MCD)



5.2.4. Modèle Logique de Données (MLD)

- ATHLETE (id_athlète, nom_athlete, prenom_athlete, date_naissance_athlete, #id_genre, #id_nationalite)
- EPREUVE (id_epreuve, nom_epreuve, date_epreuve, #id_lieu, #id_sport)
- GENRE (id_genre, nom_genre)
- LIEU (id_lieu, lieu, adresse_lieu, cp_lieu, ville_lieu)
- NATIONALITE (id_nationalite, nom_nationalite)
- PARTICIPER (#id_athlète, #id_epreuve, résultats)
- SPORT (id_sport, nom_sport)
- UTILISATEUR (id_util, nom_util, prenom_util, login_util, mdp_util)

5.2.5. Modèle Physique de Données (MPD)



6 Technologies utilisées

6.1. Langages de développement Web

- HTML
- CSS
- JS
-
- SQL

6.2. Base de données

- PhpMyAdmin
- PDO



7 Sécurité

Assurer la sécurité d'un site web est l'un des piliers les plus importants lorsqu'il s'agit de créer des sites web dynamiques. Surtout lorsqu'ils contiennent des informations personnelles sur les utilisateurs, des données sensibles ou des informations de connexion.

C'est pourquoi nous avons mis en place les mesures de sécurité suivantes.

7.1. Login

Le site web contient une page de connexion avec un forum de connexion spécialement conçu pour les administrateurs du site web afin de pouvoir gérer les données affichées dans les pages principales.

The image shows a screenshot of an admin login form. At the top center is a stylized flame logo. Below it, the text "Connexion Admin" is displayed in a serif font. Underneath, there are two input fields: one for "Login:" and one for "Mot de passe:". Both fields have a dark grey placeholder area where the actual text would be entered. At the bottom of the form is a large, prominent yellow button labeled "Se connecter".



7.2. Cryptage des mots de passe

Pour la création d'un compte admin, cette action ne peut être effectuée que par le Super Admin, qui peut consulter tous les autres comptes administrateur et peut les créer, les modifier ou les supprimer.

Les comptes admin créés par le Super Admin sont enregistrés dans la base de données et les mots de passe sont hachés à l'aide de la méthode Bcrypt.

- Voici la ligne de code responsable du hachage :

```
// Hachage du mot de passe
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// Requête pour ajouter un utilisateur
$query = "INSERT INTO utilisateur (nom_utilisateur, prenom_utilisateur, login, password)
VALUES (:nomUser, :prenomUser, :login, :hashedPassword)";
```

- et voici à quoi ressemble un mot de passe haché :

nom_utilisateur	prenom_utilisateur	login	password
Admin	Super	admin	\$2y\$10\$WFxymbZ/gV2XfGy1We2bB.NZ9owdEU5QKUFWAicOY7q...
User	John	john_doe	\$2y\$10\$VSdvPWt4OQnuQdT2vrP1z.5PzBJ5FuJc/bJhIFL8TB2...



7.3. Protection des pages administrateurs

Les pages d'administration sont sécurisées de plusieurs manières, dont les suivantes :

- L'accès aux pages d'administration est limité au public et n'est accessible qu'aux administrateurs utilisant leurs identifiants de connexion, qui sont stockés en toute sécurité dans la base de données.
- Une fois qu'un administrateur s'est connecté, une session est ouverte, lui permettant de gérer les données. Lorsque l'administrateur se déconnecte, toutes les sessions actives sont rapidement détruites, ce qui permet d'empêcher tout accès non autorisé.
- La page "Gestion Utilisateurs" est conçue de manière à ce que seul le Super Admin ait accès à toutes les données des utilisateurs "administrateurs", ce qui inclut la modification, l'ajout et la suppression.

Pour les utilisateurs qui ne sont pas super-administrateurs, cette page ne contiendra que leurs propres informations de connexion, dans lesquelles ils pourront modifier leurs propres identifiants de connexion.



7.4. Protection contre les attaques XSS (Cross-Site Scripting)

La page de connexion est protégée contre les attaques XSS par la validation des entrées. L'entrée de l'utilisateur est assainie à l'aide de `htmlspecialchars` dans le backend PHP, ce qui garantit que toute injection de script potentielle est neutralisée avant le rendu.

- Exemple de auth.php :

```
if ($_SERVER["REQUEST_METHOD"] == "POST") { // vérifiez si la méthode de demande est POST (formulaire soumis).
    $loginRetrieve = $_POST["login"]; // Récupérez la valeur du champ "Login" du formulaire.
    $login = htmlspecialchars($loginRetrieve); // Assainissez l'entrée du login
    $password = $_POST["password"]; // Récupérez la valeur du champ "password" du formulaire.
```

7.5. Protection contre les injections SQL

L'utilisation d'instructions préparées dans les requêtes SQL garantit que les entrées de l'utilisateur sont traitées comme des données et non comme du code exécutable. Une autre méthode, "bindParam", est une méthode PDO permettant de lier des paramètres à une instruction préparée. Elle aide à prévenir les injections SQL en garantissant que les valeurs fournies sont traitées comme des paramètres et non comme une partie de la requête SQL elle-même, ce qui rend difficile l'injection de code SQL malveillant par les attaquants.

- Exemple de auth.php :

```
// Préparer la requête SQL pour récupérer les informations de l'utilisateur avec le Login spécifié.
$query = "SELECT id_utilisateur, nom_utilisateur, prenom_utilisateur, login, password FROM UTILISATEUR
WHERE login = :login";
$stmt = $connexion->prepare($query); // Préparer la requête avec PDO.
$stmt->bindParam(":login", $login, PDO::PARAM_STR);
// Lier la variable :login à l'identifiant assaini, empêchant ainsi les injections SQL.
```