Cahier des charges techniques

Sommaire

[1. Contexte du projet](#_heading=h.1d6mma2ci7xl)

[1.1. Présentation du projet](#_heading=h.wek20lqal41w)

[1.2. Date de rendu du projet](#_heading=h.hsbf2s9v0gpe)

[2. Besoins fonctionnels](#_heading=h.72adhlswac0n)

[3. Ressources nécessaires à la réalisation du projet](#_heading=h.lr47i5krzqwp)

[3.1. Ressources matérielles](#_heading=h.moe5ly27tmhr)

[3.2. Ressources logicielles](#_heading=h.cvyk6p742phf)

[4. Gestion du projet](#_heading=h.wj6e27o1xw4s)

[5. Conception du projet](#_heading=h.4jjvlmxvhcof)

[5.1. Le front-end](#_heading=h.b3pf1ig00kap)

[5.1.1. Wireframes](#_heading=h.xbdoysf41ddh)

[5.1.2. Maquettes](#_heading=h.taj1xh3jo3sr)

[5.1.3. Arborescences](#_heading=h.7cpah6l43vkf)

[5.2. Le back-end](#_heading=h.ri4n2nlk4x5y)

[5.2.1. Diagramme de cas d’utilisation](#_heading=h.4hfnlk8rv9if)

[5.2.2. Diagramme d’activités](#_heading=h.d1ay6n1m7wn9)

[5.2.3. Modèles Conceptuel de Données (MCD)](#_heading=h.mhbyfyrbmn9)

[5.2.4. Modèle Logique de Données (MLD)](#_heading=h.rr2kq3n12ss)

[5.2.5. Modèle Physique de Données (MPD)](#_heading=h.5jowy0y259aj)

[6. Technologies utilisées](#_heading=h.6j2nbpnxpppw)

[6.1. Langages de développement Web](#_heading=h.uhwlwqptkq2s)

[6.2. Base de données](#_heading=h.q8i476rr9oys)

[7. Sécurité](#_heading=h.15p9a6rgigad)

[7.1. Login](#_heading=h.q04sq9cnrw82)

[7.2. Cryptage des mots de passe](#_heading=h.ct1wz6q0jyd1)

[7.3. Protection des pages administrateurs](#_heading=h.fopygvrobvcw)

[7.4. Protection contre les attaques XSS (Cross-Site Scripting)](#_heading=h.cimahn54kest)

[7.5. Protection contre les injections SQL](#_heading=h.cg22ajiezi4v)

# 

# 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.

## 1.2. Date de rendu du projet

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

# 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.

# Ressources nécessaires à la réalisation du projet

Besoins logiciels et plateformes de développement :

Environnement de développement (IDE) : Visual Studio Code

Plateforme de développement collaboratif : Github

Serveur Web : Apache (contenu dans MAMP)

Base de données relationnelle : MySQL (contenu dans MAMP)

Outil de gestion de projet : Trello

Visual Paradigm Online (lien) : conception UML et arborescence

FIGMA : maquettage

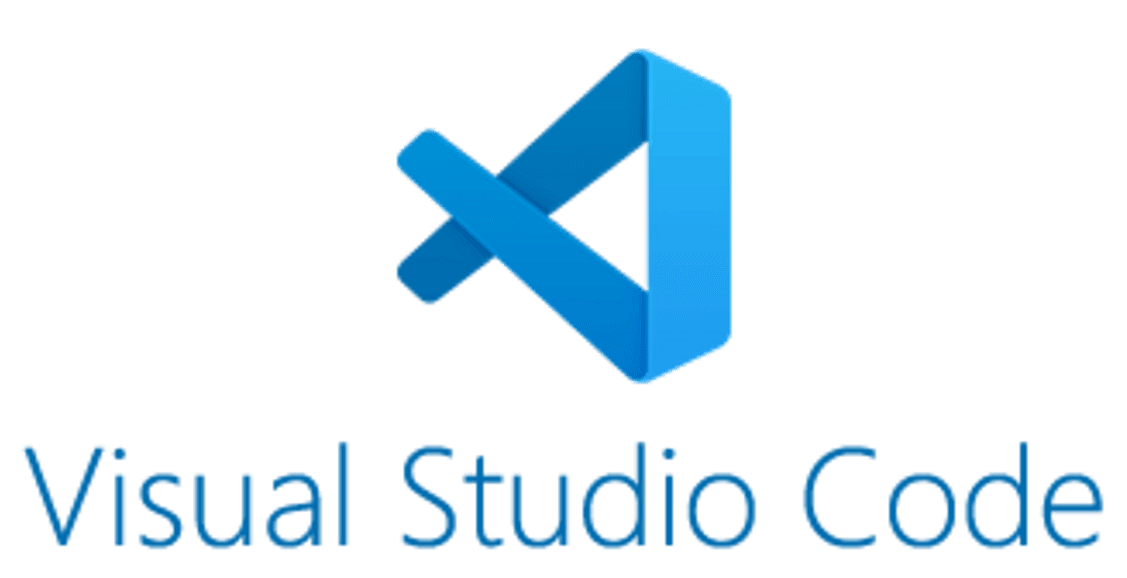
Mocodo: conception de la base de données

## 3.1. Ressources matérielles

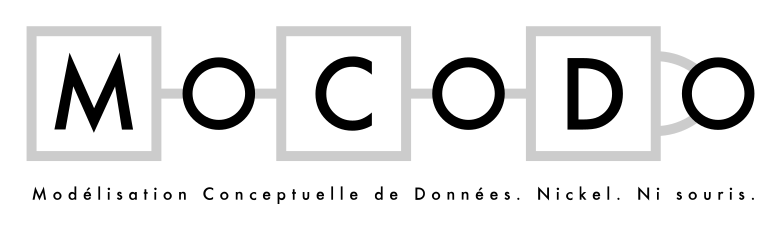
Un ordinateur fixe (écran, unité centrale, souris et clavier) ou un ordinateur portable en fonction des étudiants.

## 3.2. Ressources logicielles









# 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.

**INSÉREZ UNE CAPTURE D'ÉCRAN DE VOTRE TRELLO**

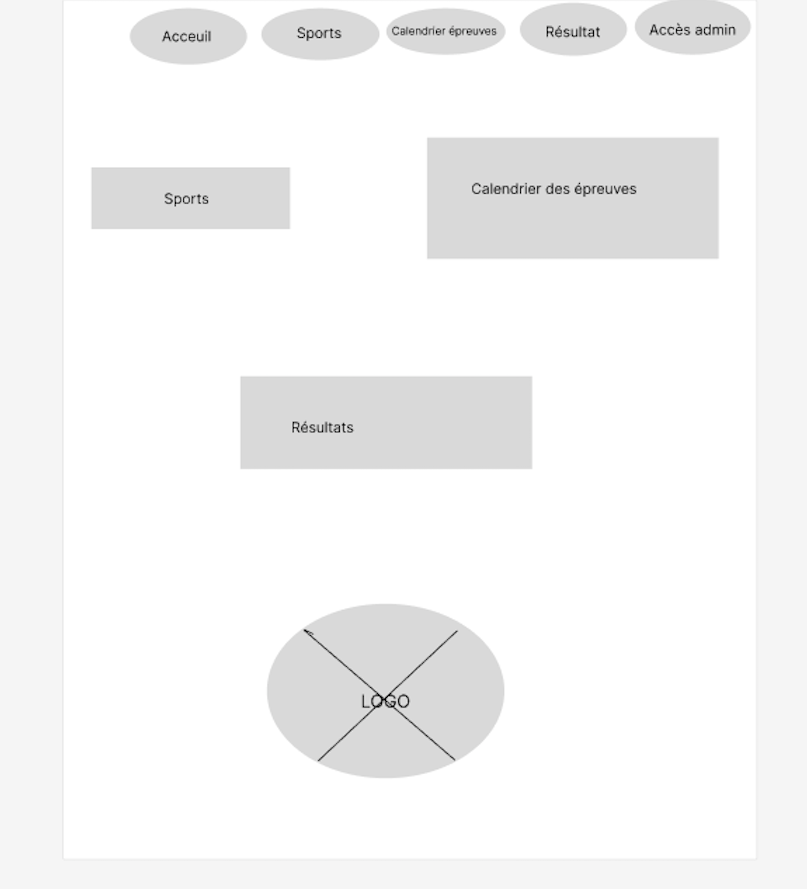
Nous travaillons également sur GitHub, plateforme de développement collaboratif.

# Conception du projet

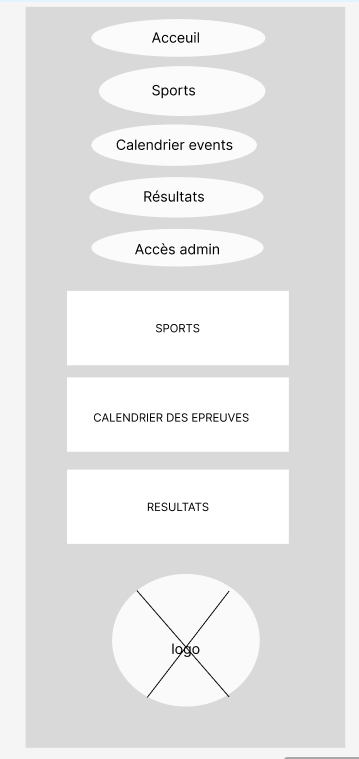
## 5.1. Le front-end

### 1.1. Wireframes

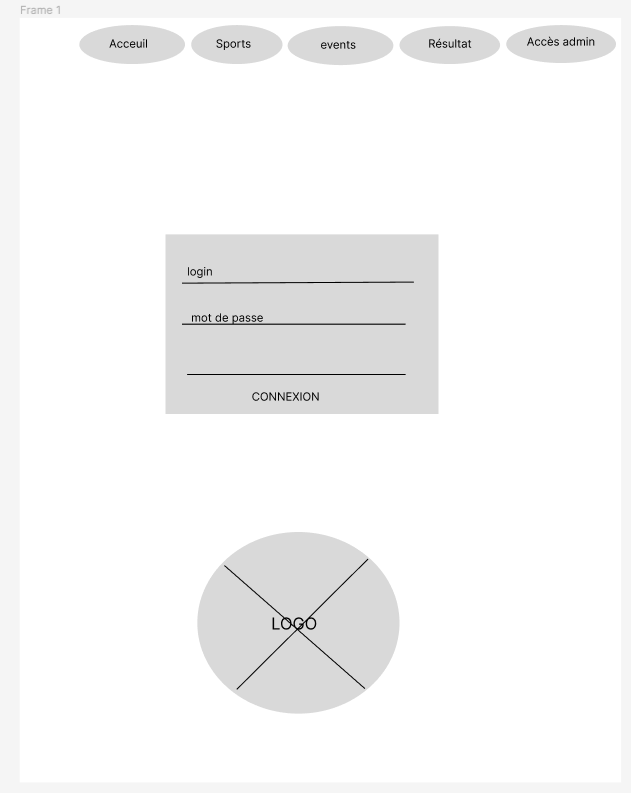
1. Index



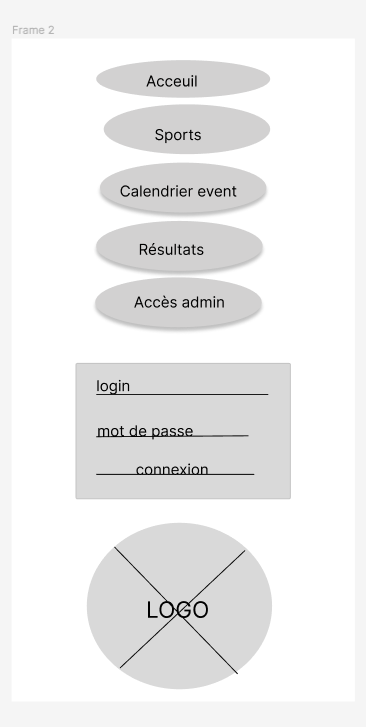
Index Responsive



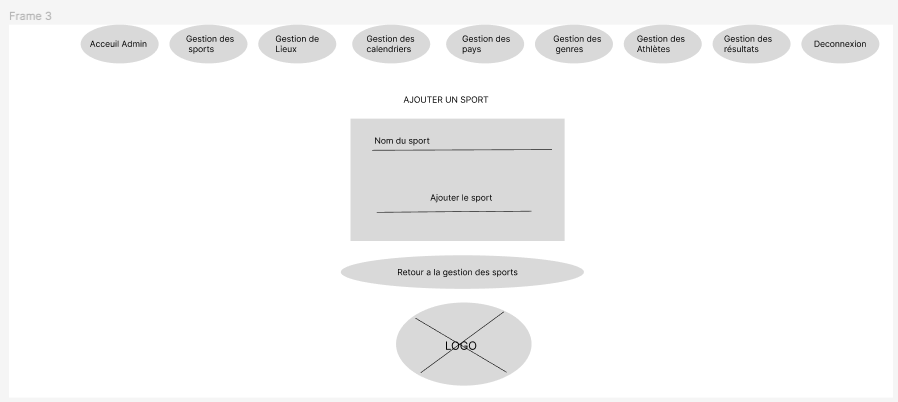
1. Connexion



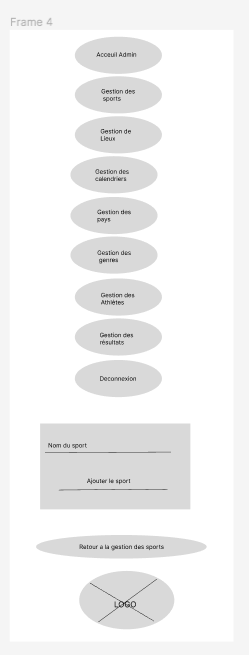
Connexion Responsive



1. Ajout d’un sport

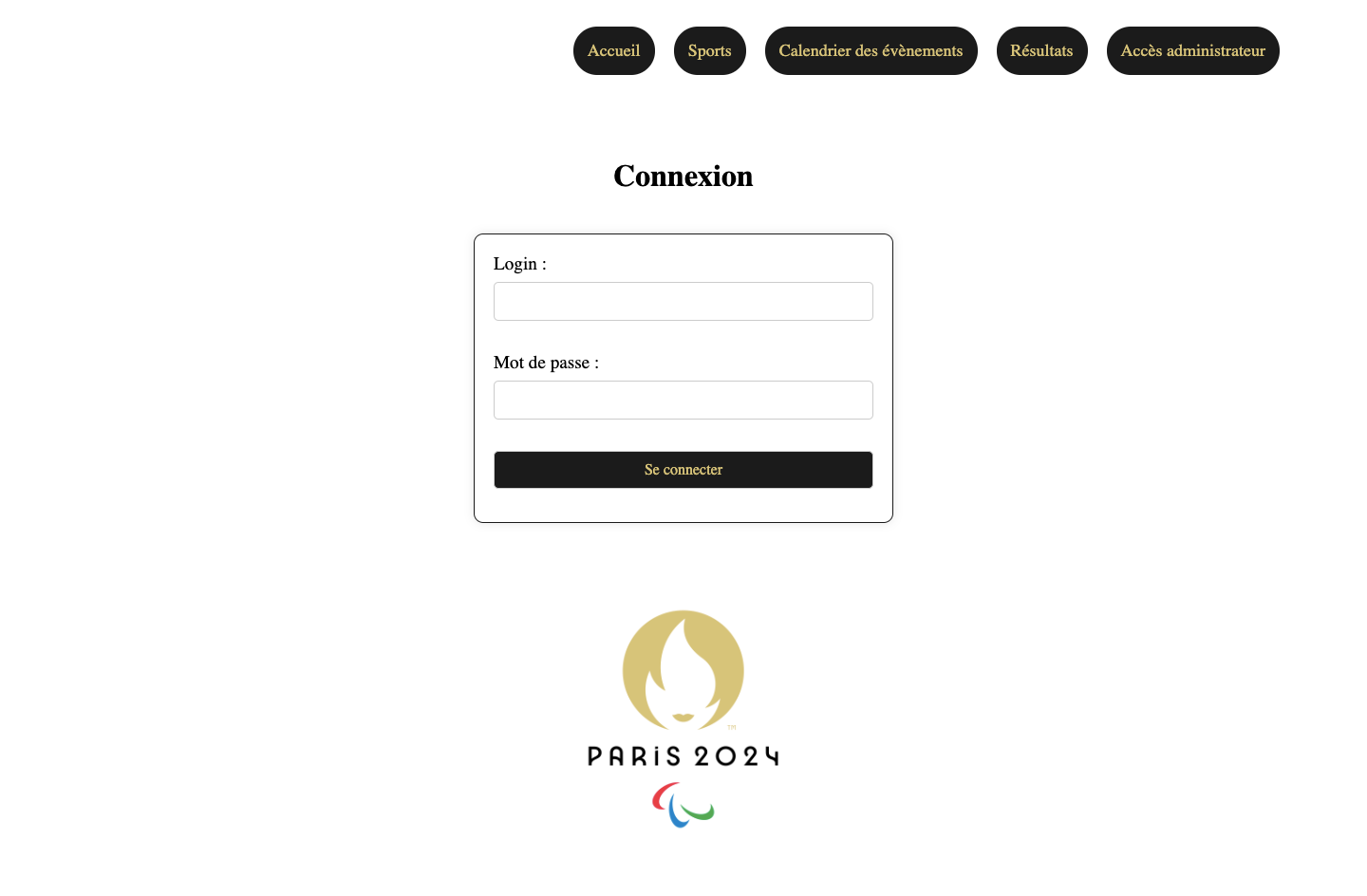


Ajout d’un sport Responsive



### 5.1.2. Maquettes

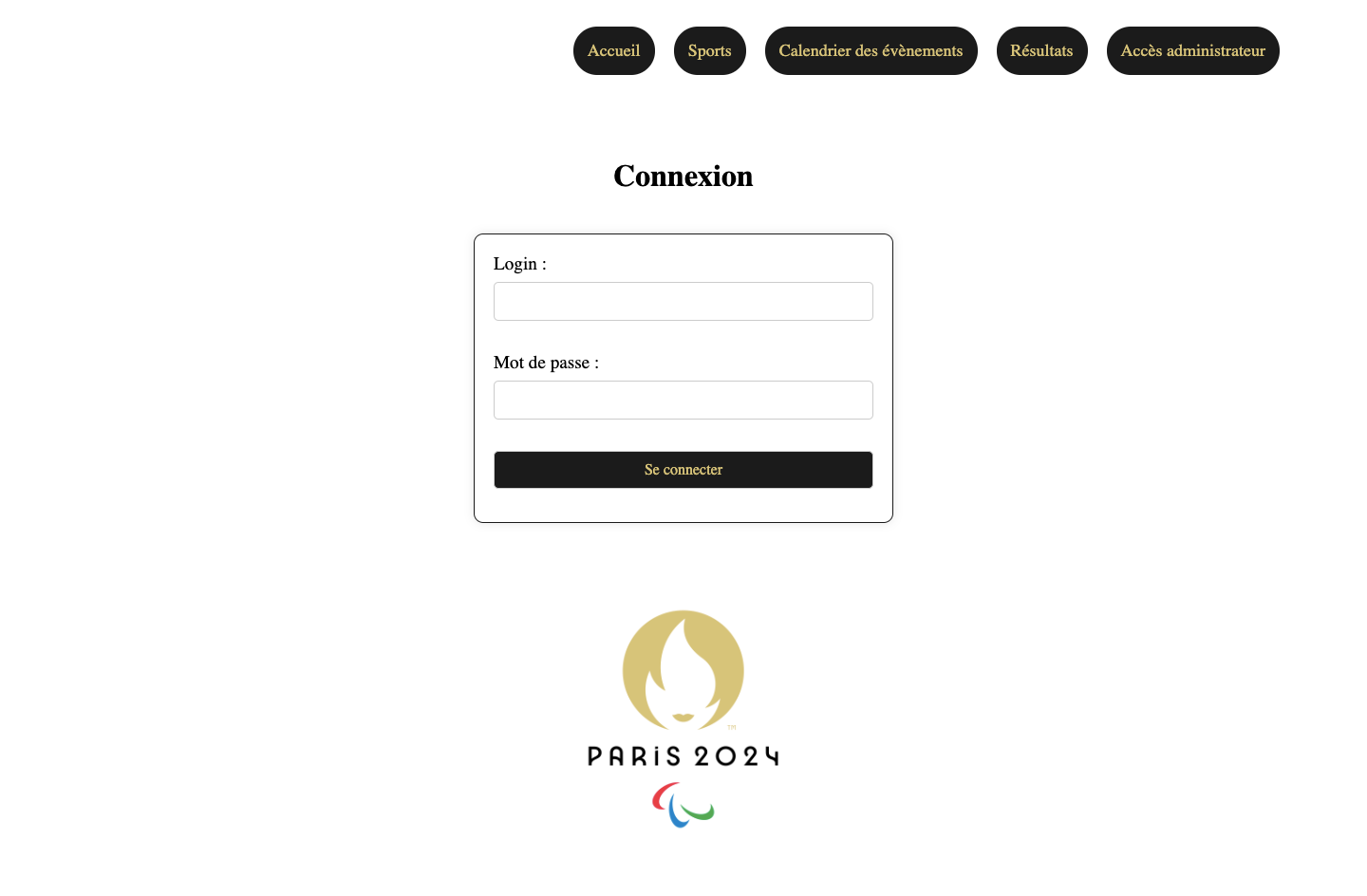
1. Index



Index Responsive



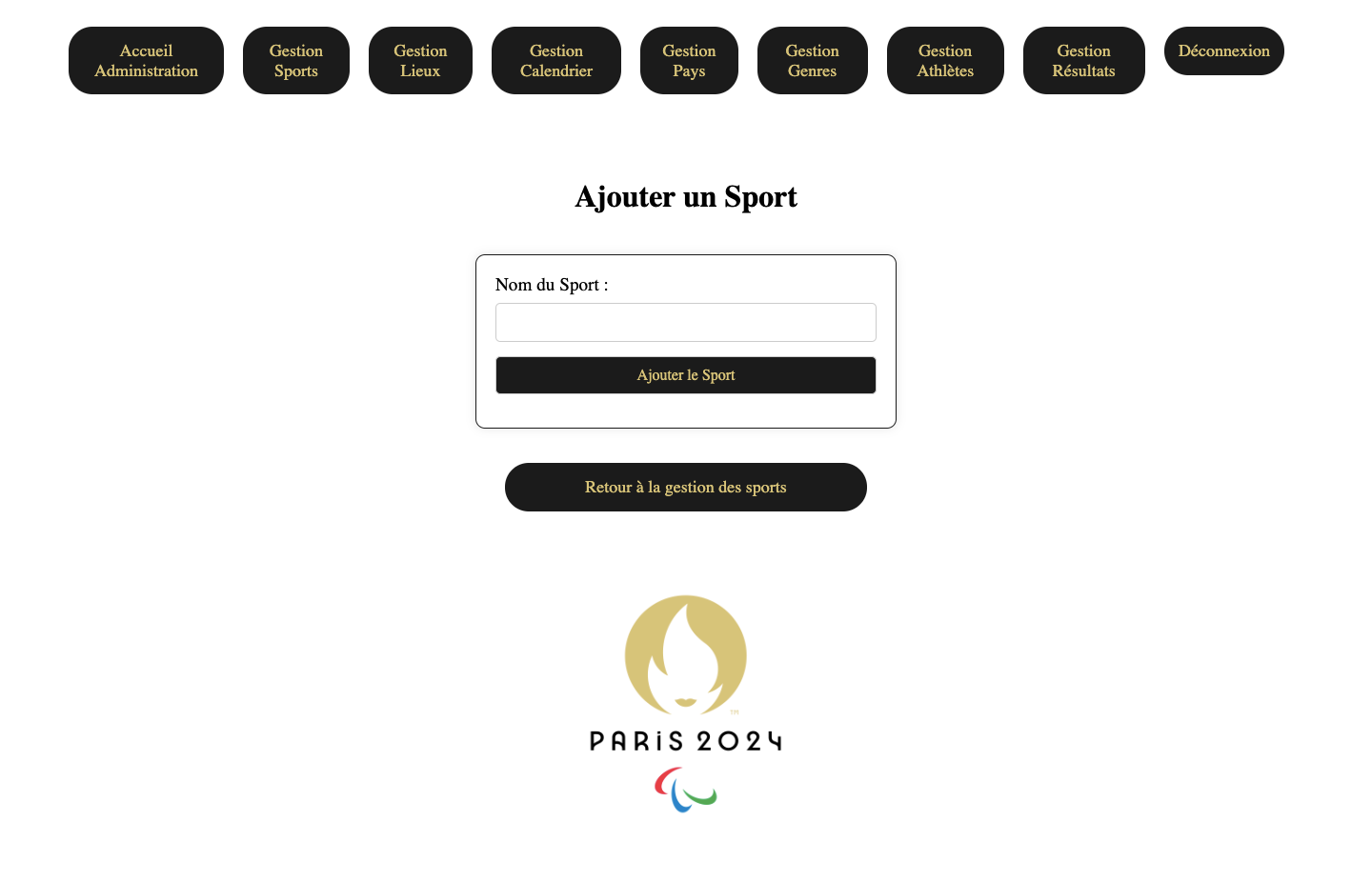
1. Connexion



Connexion Responsive



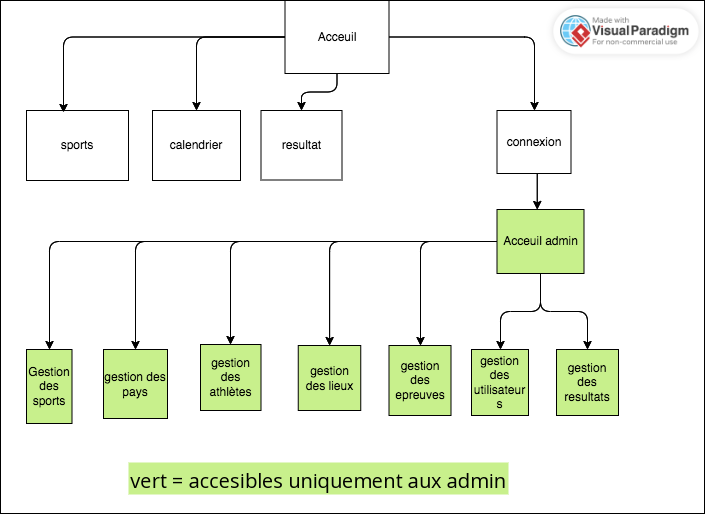
1. Ajout d’un Sport



Ajout d’un Sport Responsive



### 5.1.3. Arborescence



## 

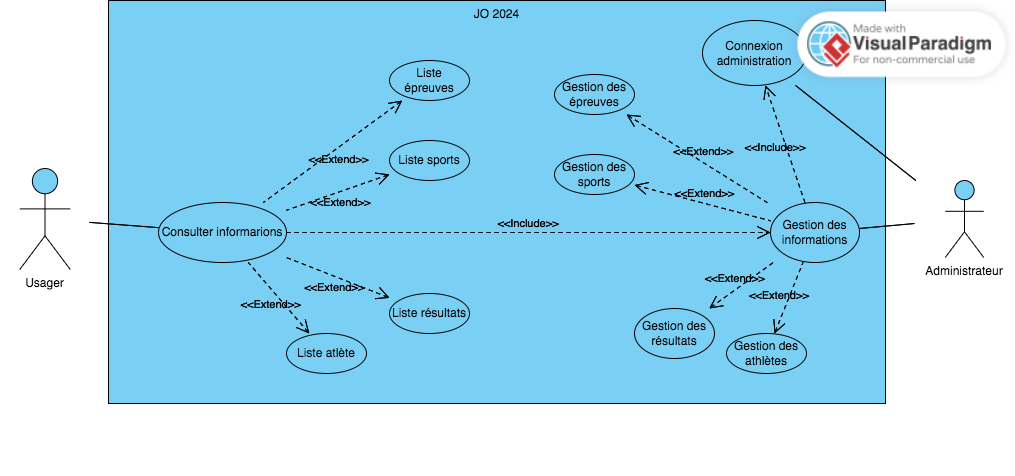
## 

## 

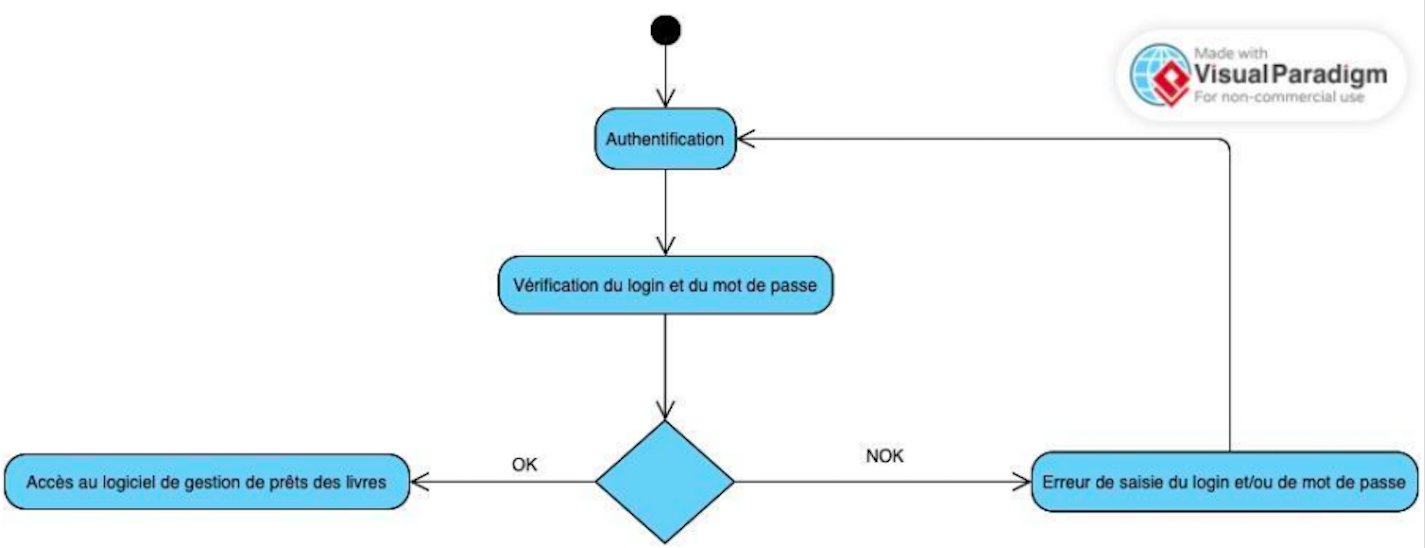
## 

## 5.2. Le back-end

### 5.2.1. Diagramme de cas d’utilisation



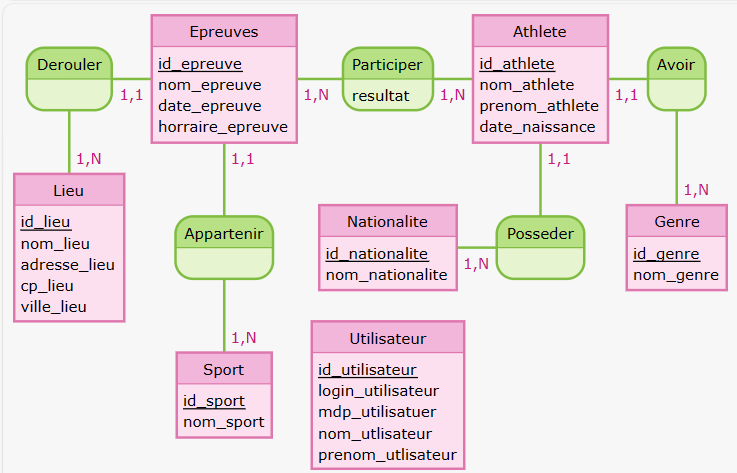
### 5.2.2. Diagramme d’activités



### 

### 

### 5.2.3. Modèles Conceptuel de Données (MCD)

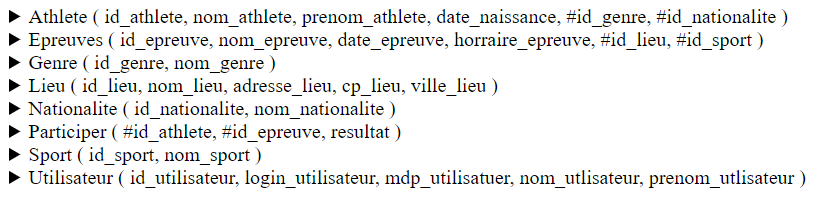


### 

### 

### 

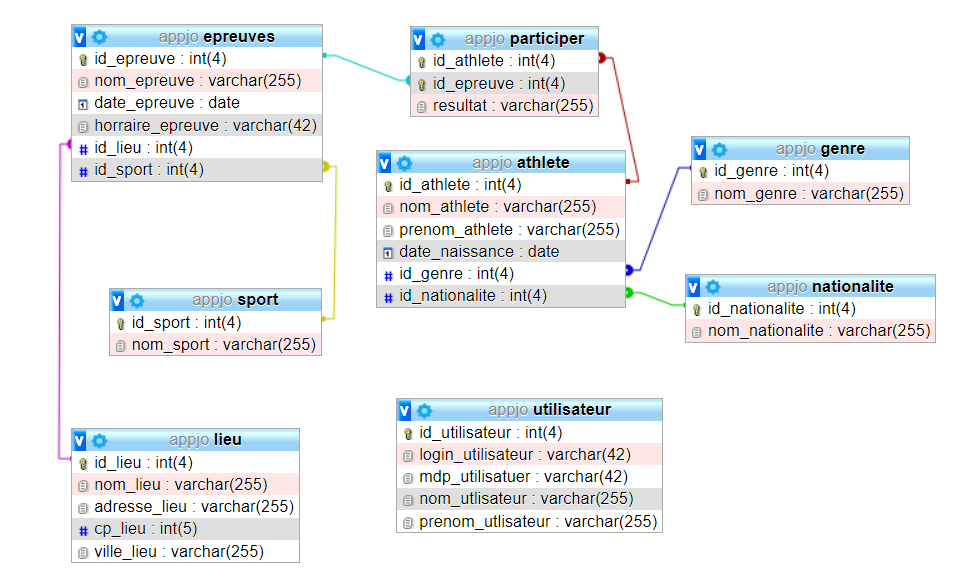
### 5.2.4. Modèle Logique de Données (MLD)



### 

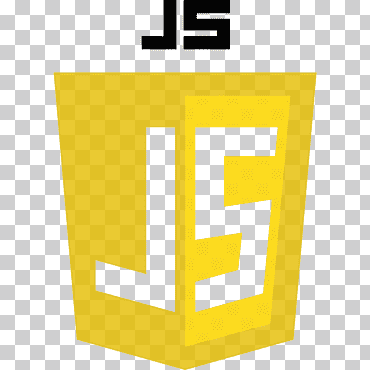
### 

### 5.2.5. Modèle Physique de Données (MPD)

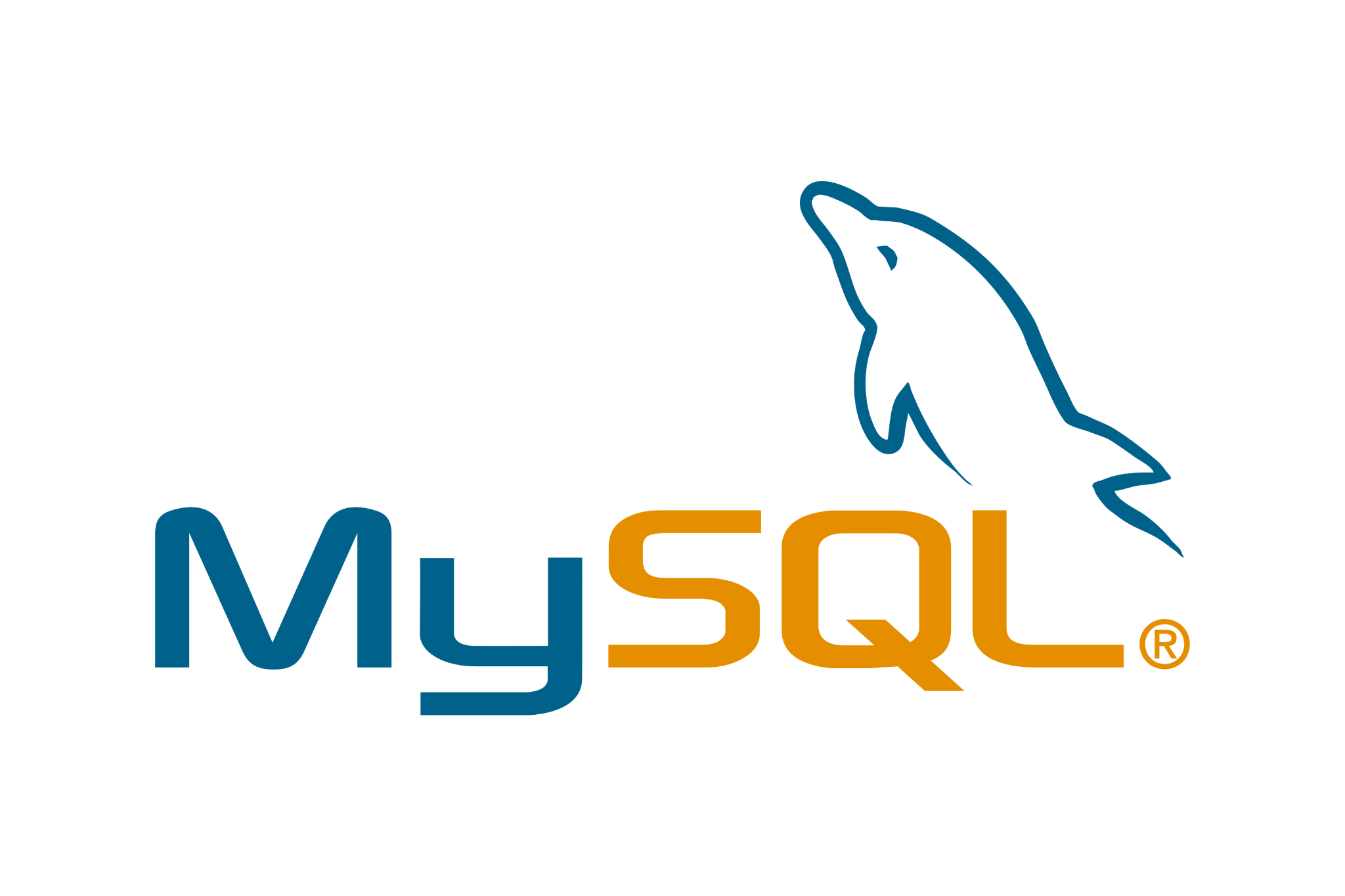


# Technologies utilisées

## 6.1. Langages de développement Web



## 6.2. Base de données



# 

6.Sécurité

6.1. Login

1. Connexion sécurisée (HTTPS).

Utilisation de certificats SSL/TLS.

2. Protection contre les attaques par force brute.

Limiter les tentatives de connexion.

Imposer des délais entre les tentatives.

Verrouiller les comptes après un certain nombre d'échecs.

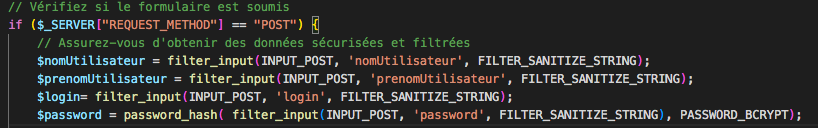
3. Gestion des sessions.

Utilisation de mécanismes de gestion de session sécurisés.

Expiration des sessions après une période d'inactivité.

6.2. Cryptage des mots de passe

1. Hachage des Mots de Passe Utilisez une fonction de hachage sécurisée comme bcrypt, Argon2, ou scrypt.



Ces algorithmes sont conçus spécifiquement pour le stockage sécurisé des mots de passe.

2. Sel de Hachage Intégrez un "sel" (salt) unique pour chaque utilisateur.

Le sel renforce la sécurité en générant des hachages uniques même pour les mêmes mots de passe.

3. Politique de Mots de Passe Forts

Mettez en place des règles pour des mots de passe forts : longs, avec des caractères spéciaux, des chiffres et des lettres en majuscules et minuscules.

4. Stockage Sécurisé

Assurez-vous que les mots de passe hachés et les sels sont stockés de manière sécurisée. Évitez de les stocker en texte brut ou dans un format facilement déchiffrable.

6.3. Protection des pages administrateurs

1. Contrôle d'Accès Basé sur les Rôles

Mettez en œuvre un modèle de contrôle d'accès basé sur les rôles en attribuant des rôles spécifiques aux administrateurs et en autorisant l'accès en fonction de ces rôles.

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

1. Validation des Entrées Utilisateurs

Validez et filtrez toutes les entrées utilisateur du côté serveur pour vous assurer qu'elles correspondent aux attentes.

2. Échappement des Données Échappez correctement toutes les données dynamiques avant de les afficher dans les pages HTML. Utilisez des fonctions d'échappement fournies par le langage de programmation ou le framework utilisé (ex : htmlspecialchars en PHP).

3. Utilisation de HTTPOnly Cookies Marquez les cookies sensibles comme HTTPOnly pour les rendre inaccessibles aux scripts côté client, réduisant ainsi les risques d'attaques XSS.

6.5. Protection contre les injections SQL

Utilisation de requêtes préparées.

Utilisation de procédures stockées.

Évitement de la concaténation directe des entrées utilisateur dans les requêtes SQL.

**DÉFINISSEZ (SI POSSIBLE) ET EXPLIQUEZ BRIÈVEMENT VOTRE SOLUTION.**

**AJOUTEZ SI VOUS LE SOUHAITEZ DES COURS EXTRAITS DE CODE.**

## 7.1. Login

## 7.2. Cryptage des mots de passe

## 7.3. Protection des pages administrateurs

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

## 7.5. Protection contre les injections SQL