

Cahier des charges techniques



Réalisé par KANOUNI Sofiane

Sommaire

1. Contexte du projet

1.1. Présentation du projet

1.2. Date de rendu du projet

2. Besoins fonctionnels

3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

3.2. Ressources logicielles

4. Gestion du projet

5. Conception du projet

5.1. Le front-end

5.1.1. Wireframes

5.1.2. Maquettes

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)

5.2.5. Modèle Physique de Données (MPD)

6. Technologies utilisées

6.1. Langages de développement Web

6.2. Base de données

7. Sécurité

7.1. Login et protection des pages administrateurs

7.2. Cryptage des mots de passe avec Bcrypt

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

7.4. Protection contre les injections SQL

1. Contexte du projet

1.1. Présentation du projet

Votre agence web a été sélectionnée par le comité d'organisation des jeux olympiques de Los Angeles 2028 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 2028.

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 7 novembre 2024.

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

Les matériels nécessaires à la réalisation de la mission sont un ordinateur fixe et portable (écran, souris, unité centrale). Ces ordinateurs sont reliés à Internet par câble ou Wi-Fi.

3.2. Ressources logicielles

Environnement de développement (IDE) : Visual Studio Code

Plateforme de développement collaboratif : Github

Outils de gestion de projet : Trello

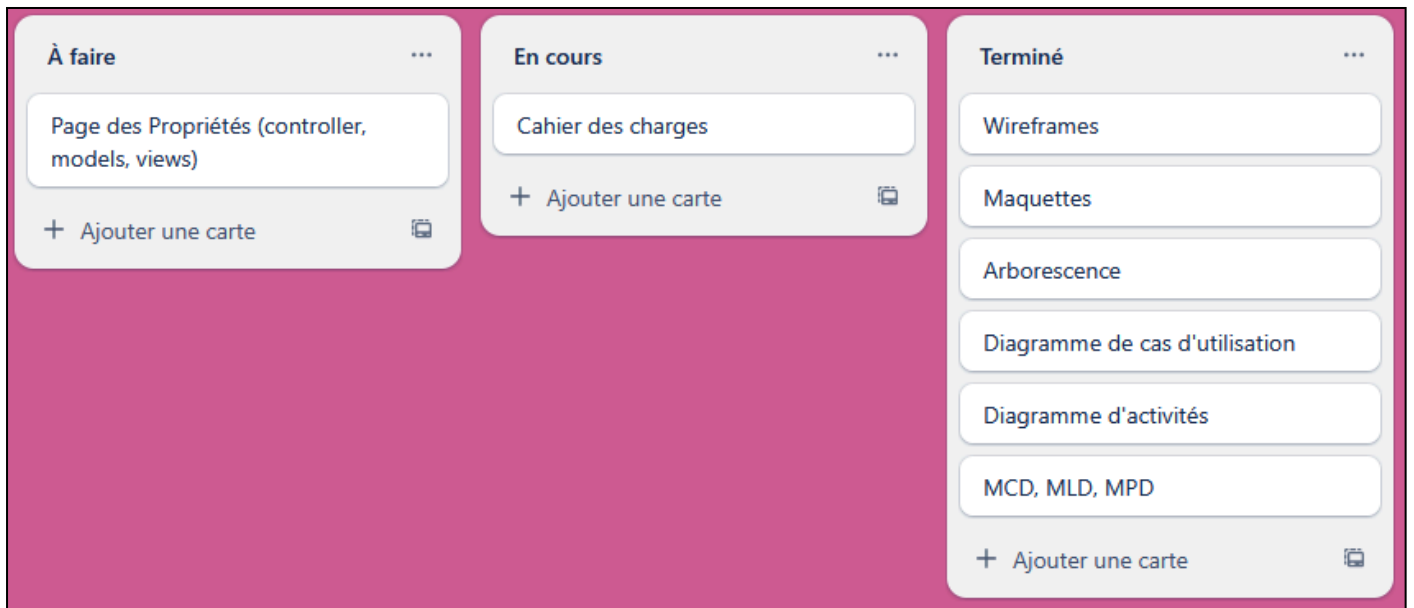
Conception UML et arborescence : Visual Paradigm online

Maquettage : Figma

Conception de base de données : Mocodo

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.



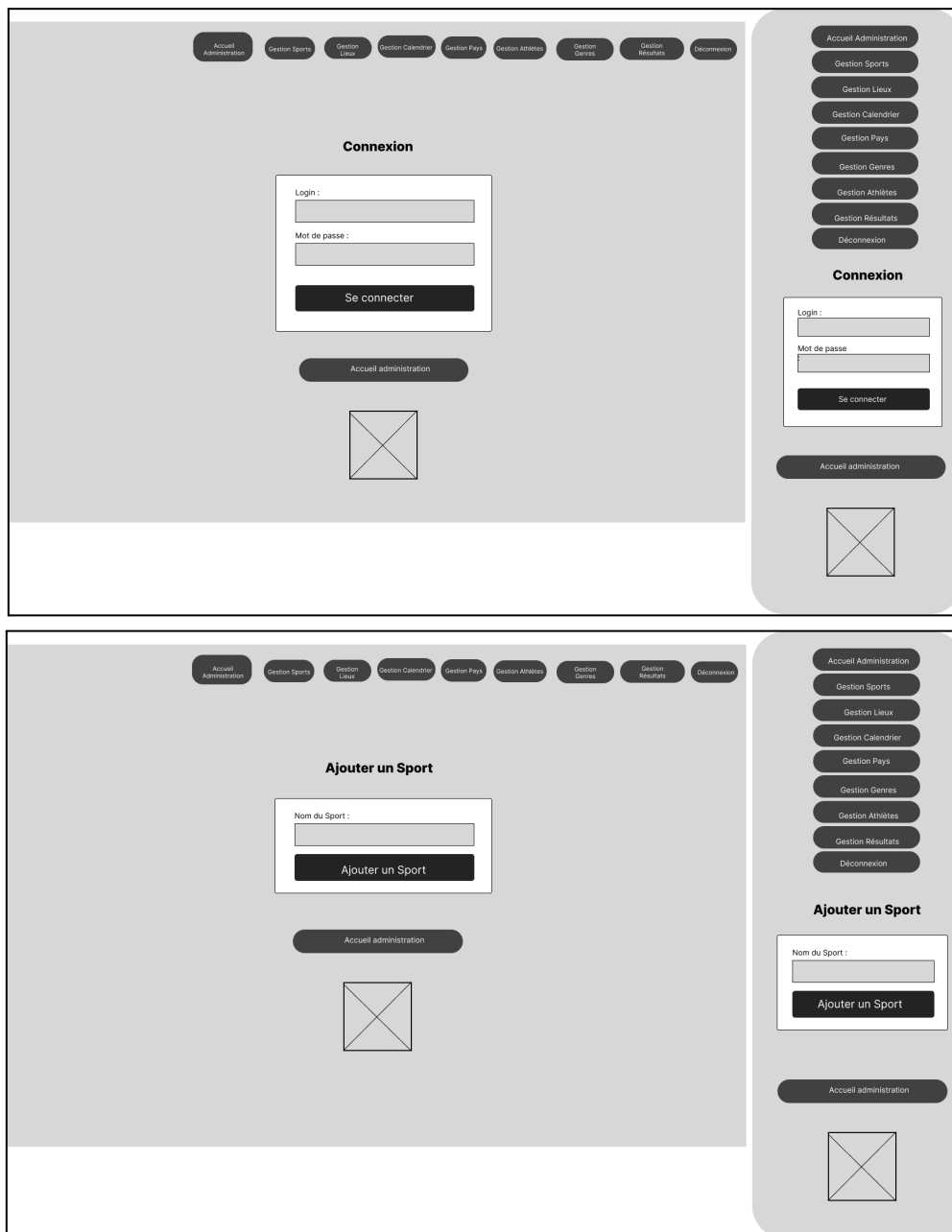
Nous travaillons également sur GitHub, plateforme de développement collaboratif.



5. Conception du projet

5.1. Le front-end

5.1.1. Wireframes



5.1.2. Maquettes

Version PC


AccueilSportsCalendrier des événementsRésultatsAccès administrateur

Connexion

Login :

Mot de passe :

Se connecter




Accueil AdministrationGestion SportsGestion LieuxGestion PaysGestion CalendrierGestion AthlètesGestion GenresGestion RésultatsDéconnexion

Ajouter un Sport



Nom du Sport :

Ajouter le Sport

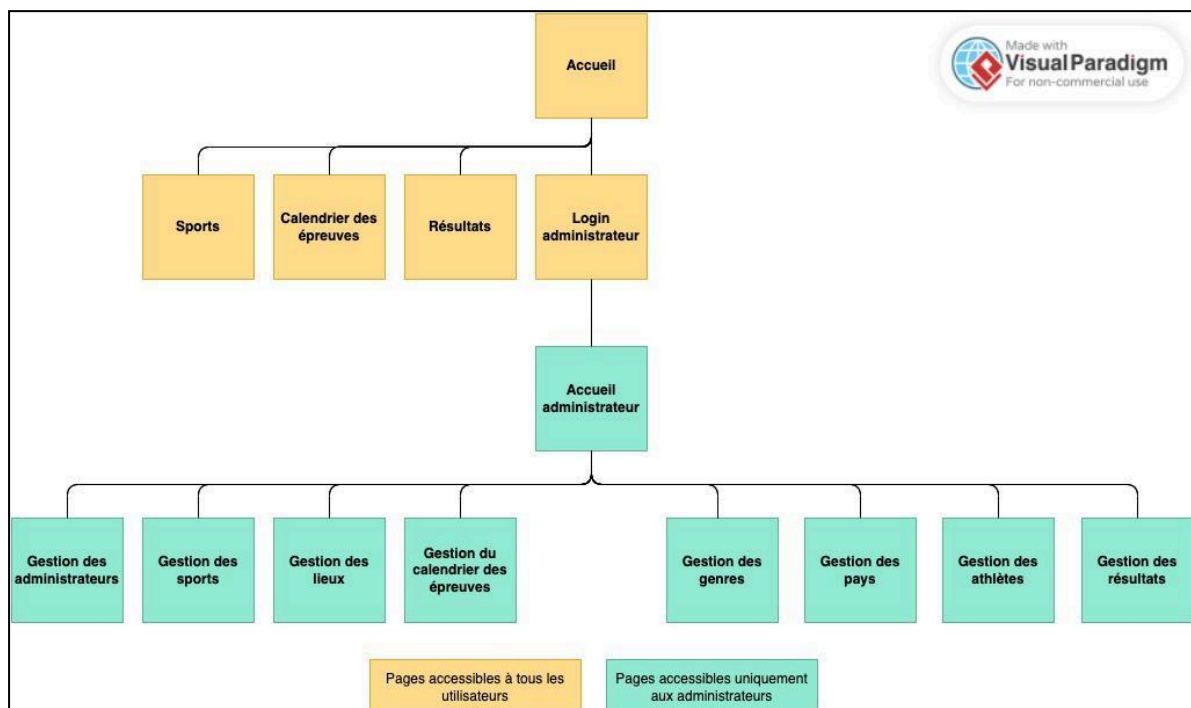
Retour à la gestion des sports



Version Responsive

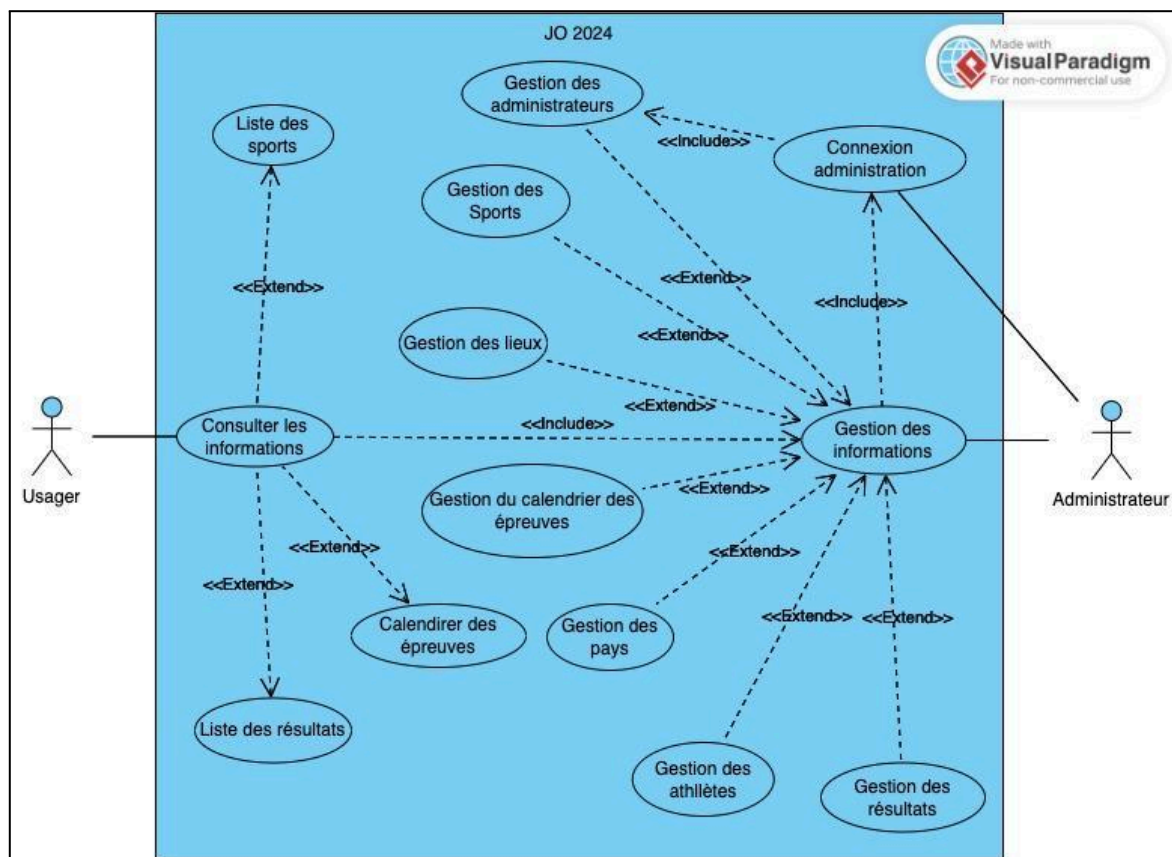
<div><div>Accueil Administration</div><div>Gestion Sports</div><div>Gestion Lieux</div><div>Gestion Calendrier</div><div>Gestion Pays</div><div>Gestion Genres</div><div>Gestion Athlètes</div><div>Gestion Résultats</div><div>Déconnexion</div></div> <div><div>Ajouter un Sport</div><div><div>Nom du Sport :</div><div><input type="text"/></div><div>Ajouter le Sport</div></div><div>Retour à la gestion des sports</div></div> <div></div>	<div><div>Accueil</div><div>Sports</div><div>Calendrier des évènements</div><div>Résultats</div><div>Accès administrateur</div></div> <div><div>Connexion</div><div><div>Login :</div><div><input type="text"/></div><div>Mot de passe :</div><div><input type="password"/></div><div>Se connecter</div></div></div> <div></div>
--	---

5.1.3. Arborescences

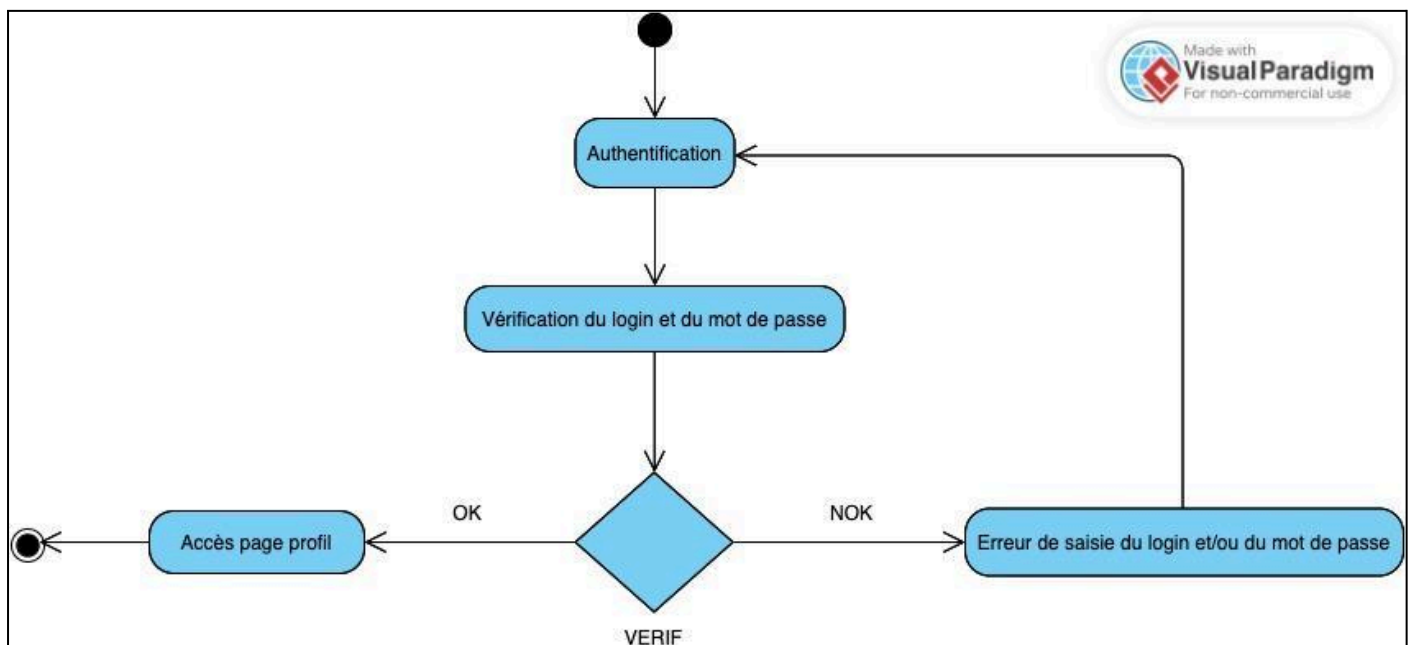


5.2. Le back-end

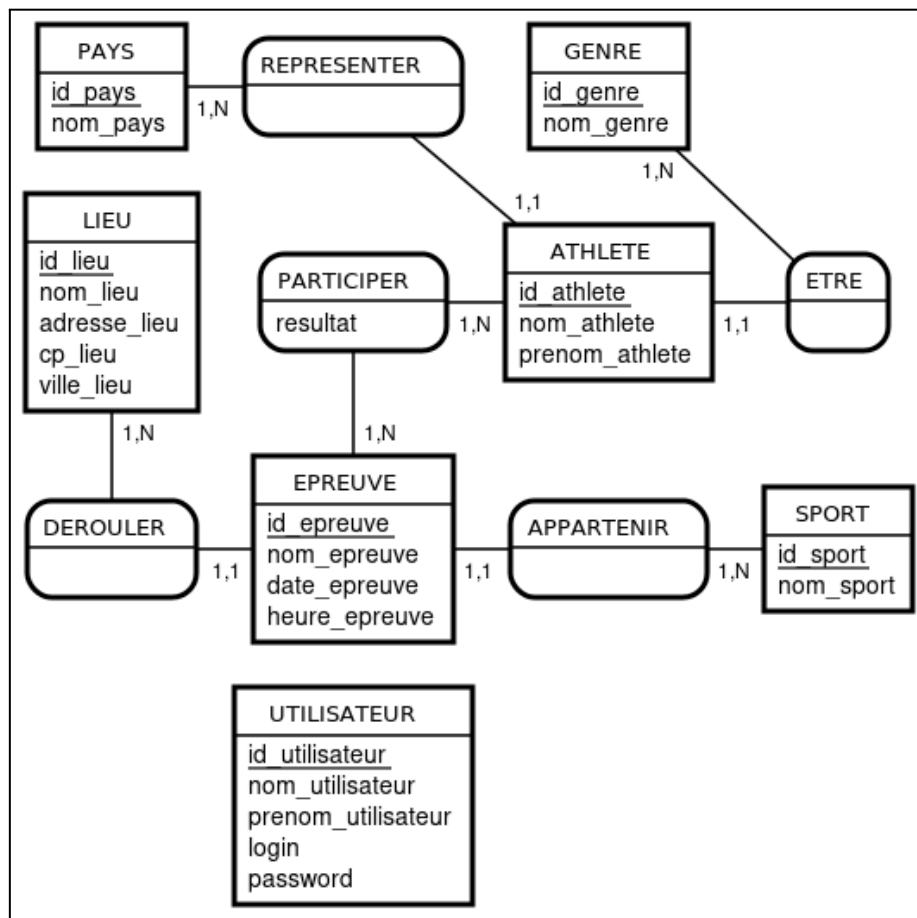
5.2.1. Diagramme de cas d'utilisation



5.2.2. Diagramme d'activités



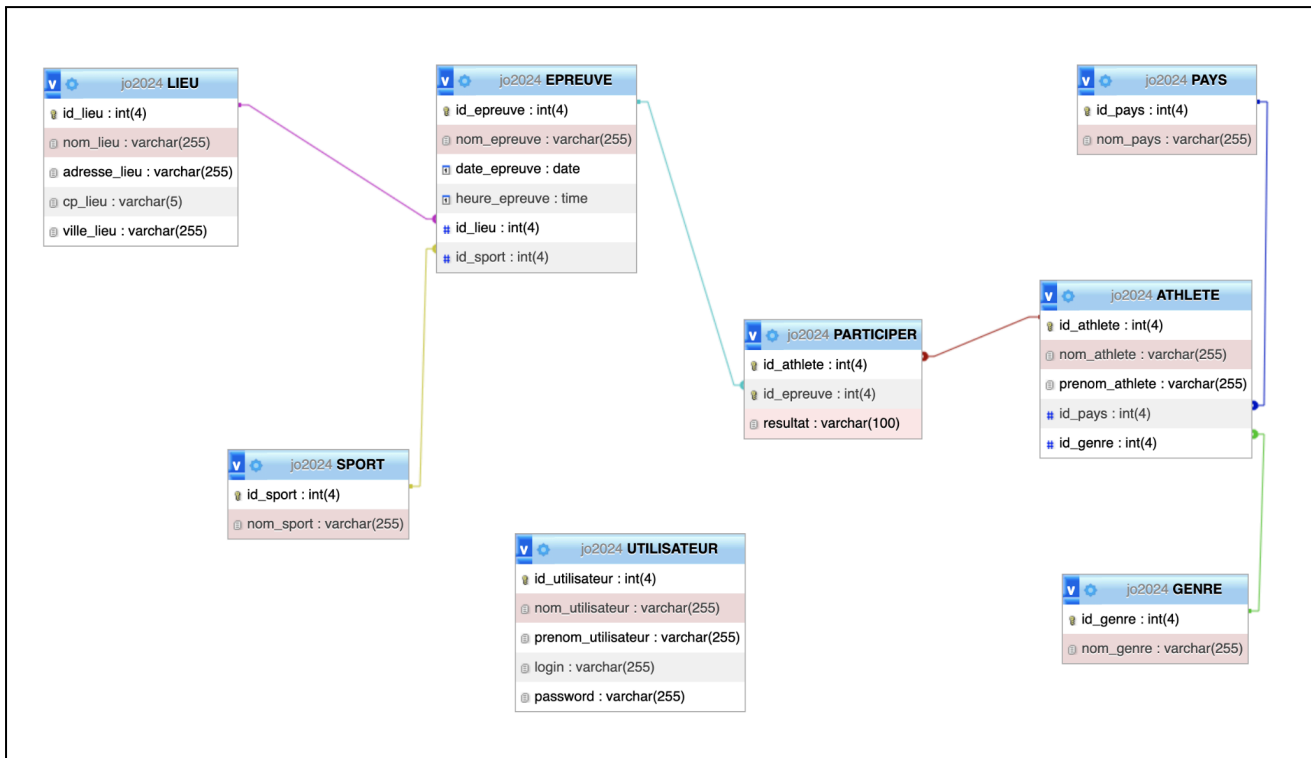
5.2.3. Modèles Conceptuel de Données (MCD)



5.2.4. Modèle Logique de Données (MLD)

- PAYS(id_pays, nom_pays)
 - **Clé primaire : id_pays**
- GENRE(id_genre, nom_genre)
 - **Clé primaire : id_genre**
- LIEU(id_lieu, nom_lieu, adresse_lieu, cp_lieu, ville_lieu)
 - **Clé primaire : id_lieu**
- PARTICIPER(id_athlete, id_epreuve, resultat)
 - **Clé primaire : id_athlete, id_epreuve**
 - **Clé étrangère : id_athlete en référence à id_athlete de ATHLETE**
 - **id_epreuve en référence à id_epreuve de EPREUVE**
- ATHLETE(id_athlete, nom_athlete, prenom_athlete)
 - **Clé primaire : id_athlete**
- EPREUVE(id_epreuve, nom_epreuve, date_epreuve, heure_epreuve)
 - **Clé primaire : id_epreuve**
- SPORT(id_sport, nom_sport)
 - **Clé primaire : id_sport**
- UTILISATEUR(id_utilisateur, nom_utilisateur, prenom_utilisateur, login, password)
 - **Clé primaire : id_utilisateur**

5.2.5. Modèle Physique de Données (MPD)



6. Technologies utilisées

6.1. Langages de développement Web

HTML : Pour structurer le contenu des pages web.

CSS : Langage de style pour la mise en forme et l'apparence des pages.

JavaScript Vanilla : Langage qui permet de rendre les pages interactives.

PHP : Langage de script côté serveur pour générer du contenu dynamique.

Outil : Visual Studio Code pour le développement et l'édition de code.

6.2. Base de données

Mocodo : Outil de modélisation conceptuelle des données.

MAMP : Environnement de développement local pour exécuter PHP et MySQL.

MySQL : Système de gestion de base de données relationnelles (SGDBR) pour stocker et gérer les données.

Visual Paradigm : Outil de modélisation UML.

7. Sécurité

7.1. Login et protection des pages administrateurs

- Formulaire de connexion : Il demande à l'utilisateur d'entrer son nom d'utilisateur et son mot de passe, ces derniers sont envoyés à un script PHP pour validation.

- Vérification des informations d'identification : Le script PHP cherche l'utilisateur dans la BDD. S'il le trouve, il compare le mot de passe soumis par celui stocké avec la fonction `password_verify()`, qui le valide (ou non).

- Protection des pages administrateurs : Pour chacune des pages auxquelles on ne peut accéder que par connexion, on vérifie si l'utilisateur est bien connecté avant de lui donner accès.

Pour cela, on utilise une session. Si la session (par exemple : 'admin_sess') n'existe pas, l'utilisateur est redirigé vers la page de connexion.

Ex :

```
session_start();
if (!isset($_SESSION['admin_sess'])) {
    header("Location: login.php");
    exit;
}
```

7.2. Cryptage des mots de passe avec Bcrypt

Bcrypt : Algorithme de hachage sécurisé en PHP utilisé pour protéger les mots de passe.

Hachage : Processus convertissant un mot de passe en une chaîne de caractères fixe, rendant impossible la récupération du mot de passe initial.

Bcrypt a deux principaux avantages :

- Le Salage : Un sel (chaîne aléatoire) est ajouté à chaque mot de passe, garantissant que les hachages générés sont uniques, même si deux utilisateurs choisissent le même mot de passe.

- Coût ajustable : Le nombre d'itérations du hachage peut être ajusté, rallongeant ainsi le processus et rendant les attaques plus difficiles.

En PHP :

Hachage d'un mot de passe : On utilise `password_hash()`.

Exemple :

```
$hashed_password = password_hash($plain_password, PASSWORD_DEFAULT);
```

Vérification du mot de passe : On utilise `password_verify()`.

Exemple :

```
if (password_verify($submitted_password, $stored_hash)) {  
    // Mot de passe correct  
} else {  
    // Mot de passe incorrect  
}
```

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

La faille XSS (Cross-Site Scripting) permet l'injection de code malveillant (HTML, JavaScript) dans des variables ou des bases de données insuffisamment sécurisées.

Cette attaque peut être de nature persistante (stockée) ou temporaire. Son objectif est d'insérer un script qui amène le site à se connecter à une source externe malveillante, exposant ainsi des données sensibles comme les cookies des utilisateurs.

Pour se prémunir contre les attaques XSS, deux principales approches existent :

- **Filtrage du contenu HTML** : L'utilisation de fonctions telles que `strip_tags()` permet de supprimer les balises HTML des entrées utilisateur, garantissant ainsi que seules des données textuelles sont enregistrées.
- **Échappement des caractères spéciaux** : Les fonctions comme `htmlspecialchars()` ou `htmlentities()` transforment les caractères spéciaux en entités HTML, empêchant ainsi l'exécution de scripts tout en conservant l'affichage du texte d'origine.

7.4. Protection contre les injections SQL

Les injections SQL permettent à un attaquant d'introduire du code SQL malveillant pour altérer la base de données.

Protection avec PHP :

- Utilisation de requêtes préparées (PDO)
- Validation des données utilisateur pour garantir leur bon format

Exemple : `$id = filter_var($id_saisi, FILTER_VALIDATE_INT);`