



CAHIER DES CHARGES

TECHNIQUE



Dossier rédiger par Bah Ibrahima

2023-2024



Sommaire

<u>1. Contexte du projet</u>	3
<u>1.1. Présentation du projet</u>	3
<u>1.2. Date de rendu du projet</u>	3
<u>2. Besoins fonctionnels</u>	3
<u>3. Ressources matérielles nécessaires à la réalisation du projet</u>	3
<u>4. Ressources logiciels nécessaires à la réalisation du projet</u>	3
<u>5. Gestion du projet</u>	4
<u>6. Conception du projet</u>	4
<u>6.1. Le front-end</u>	4
<u>6.1.1. Wireframes</u>	4
<u>6.1.2. Maquettes</u>	6
<u>6.1.3. Arborescences</u>	9
<u>6.2. Le back-end</u>	10
<u>6.2.1. Diagramme de cas d'utilisation</u>	10
<u>6.2.2. Diagramme d'activités</u>	10
<u>6.2.3. Modèles Conceptuel de Données (MCD)</u>	11
<u>6.2.4. Modèles Logique de Données (MLD)</u>	11
<u>6.2.5. Modèle Physique de Données (MPD)</u>	12
<u>7. Technologies utilisées</u>	12
<u>7.1. Langages de développement Web</u>	12
<u>7.2. Base de données</u>	13
<u>8. Sécurité</u>	13
<u>8.1. Login et protection des pages administrateurs</u>	13
<u>8.2. Cryptage des mots de passe avec Bcrypt</u>	14
<u>8.3. Protection contre les attaques XSS (Cross-Site Scripting)</u>	15
<u>8.4. Protection contre les injections SQL</u>	16



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.

1.2. Date de rendu du projet

Le projet doit être rendu au plus tard le 22 mars 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 matérielles nécessaires à la réalisation du projet

- CLAVIER
- ECRAN
- SOURIS
- PC



4. Ressources logicielles nécessaires à la réalisation du projet



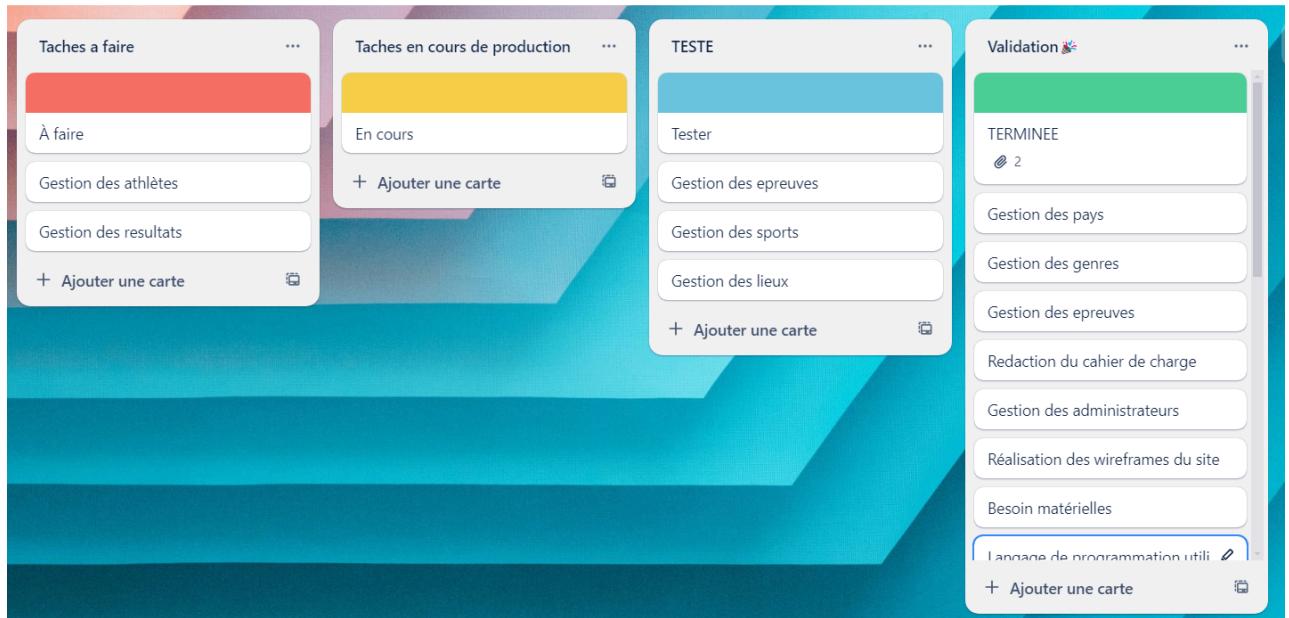
Dossier rédiger par Bah Ibrahima

2023-2024



5. Gestion du projet

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



Dossier rédiger par Bah Ibrahima

2023-2024



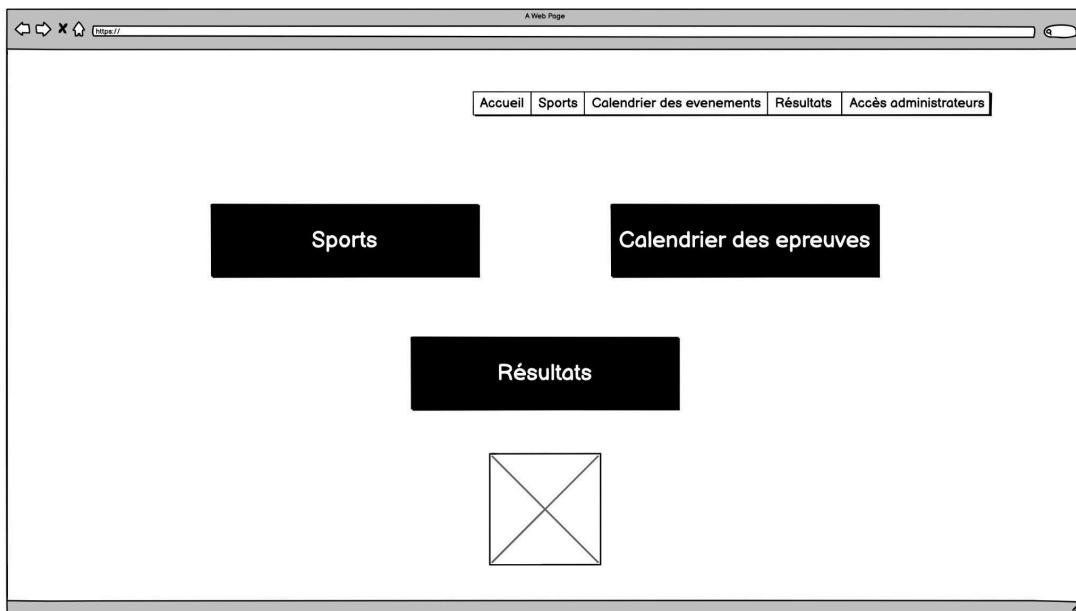
6. Conception du projet

6.1. Le front-end

La conception du projet repose sur le développement d'un front-end dynamique et intuitif, élément essentiel dans l'expérience utilisateur globale.

6.1.1. Wireframes

Wireframe de la page index





Wireframe de la page des sports

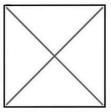
A Web Page

Accueil | Sports | Calendrier des evenements | Résultats | Accès administrateurs

Liste des sports

Sport
Athlétisme
Boxe
Cyclisme
Escalade
Gymnastique
Lancer
Natation
Saut en hauteur
Saut en longueur
VTT

Retour Accueil



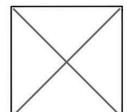
Wireframe de la page Connexion

A Web Page

Accueil | Sports | Calendrier des evenements | Résultats | Accès administrateurs

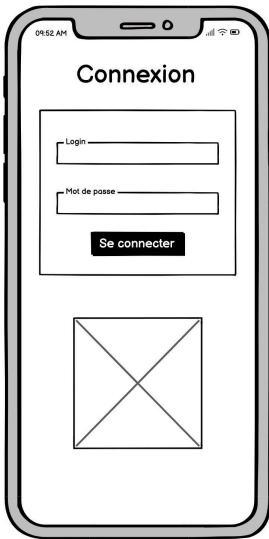
Connexion

— Login —	<input type="text"/>
— Mot de passe —	<input type="password"/>
<input type="button" value="Se connecter"/>	





Wireframe Responsive de la page connexion



Wireframe Responsive de la page index



Wireframe Responsive de la page sports



Dossier rédiger par Bah Ibrahima

2023-2024



6.1.2. Maquettes

Maquette responsive de la page sports Maquette responsive de la page index Maquette responsive de la page connexion



Liste des Sports

Sport
Athlétisme
Boxe
Cyclisme
Escalade
Gymnastique
Lancer
Natation
Saut en hauteur
Saut en longueur
VTT

[Retour Accueil](#)

Sports

Calendrier des épreuves

Résultats

Connexion

Login :

Mot de passe :

[Se connecter](#)



Dossier rédiger par Bah Ibrahima

2023-2024



Maquette de la page connexion

Connexion

Login :

Mot de passe :



Maquette de la page index

Accueil Sports Calendrier des épreuves Résultats Accès administrateur

Accueil Sports Calendrier des événements Résultats Accès administrateur

Sports

Calendrier des épreuves

Résultats



Maquette de la page sports

Accueil Sports Calendrier des événements Résultats Accès administrateur

Liste des Sports

Sport
Athlétisme
Boxe
Cyclisme
Escalade
Gymnastique
Lancer
Natation
Saut en hauteur
Saut en longueur
VTT



Dossier rédiger par Bah Ibrahima

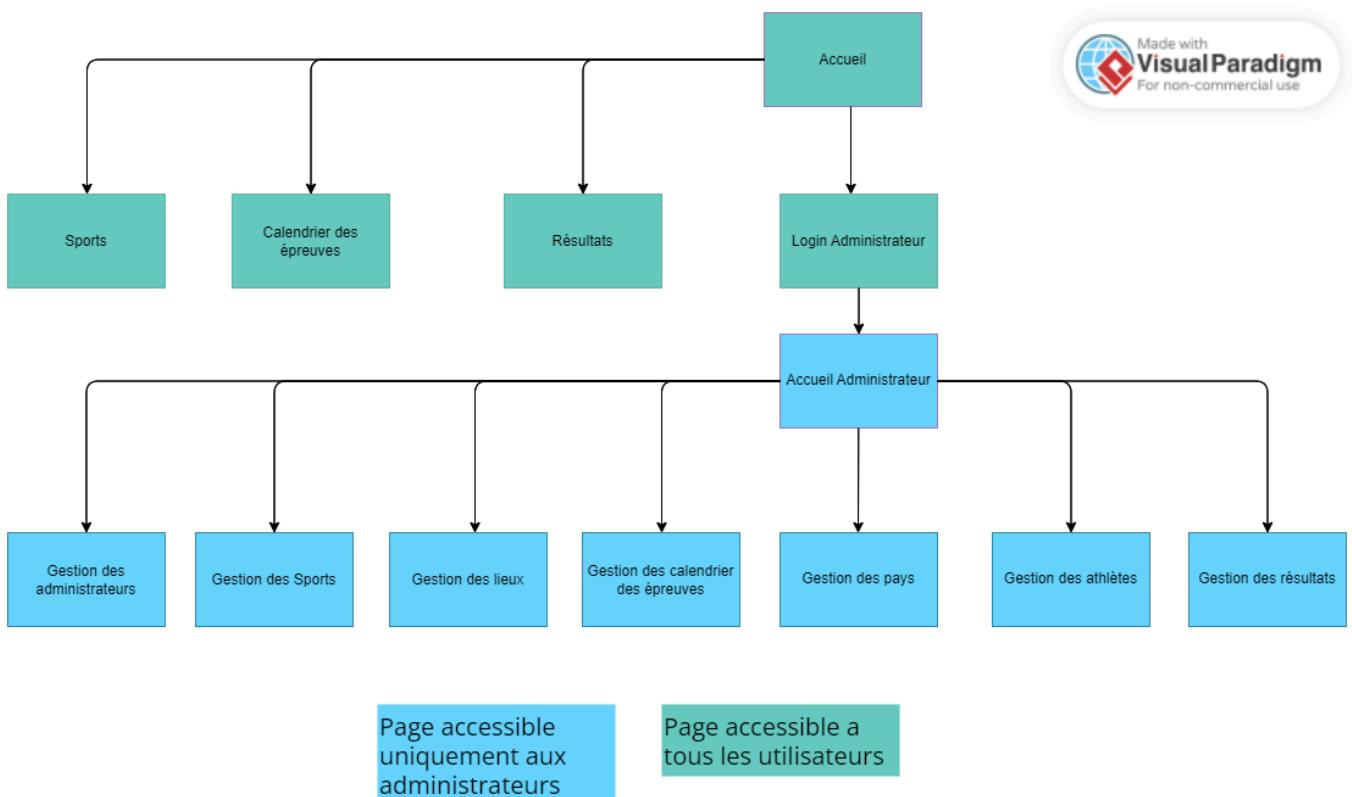
2023-2024



6.1.3. Arborescences

La page d'accueil, ouverte à tous les utilisateurs, présente un menu clair comprenant des sections telles que SPORT, CALENDRIER DES EPREUVES et RESULTAT. Les visiteurs peuvent également accéder à une section sécurisée, LOGIN ADMINISTRATEUR, réservée aux administrateurs du site.

Une fois connecté, l'administrateur est dirigé vers une nouvelle page d'accueil dédiée. Cette interface propose différentes options de gestion, notamment GESTION D'ADMINISTRATEUR, GESTION DES SPORTS, GESTION DES LIEUX, CALENDRIER DES ÉPREUVES, GESTION DES PAYS, GESTION DES ATHLÈTES et GESTION DES RÉSULTATS.



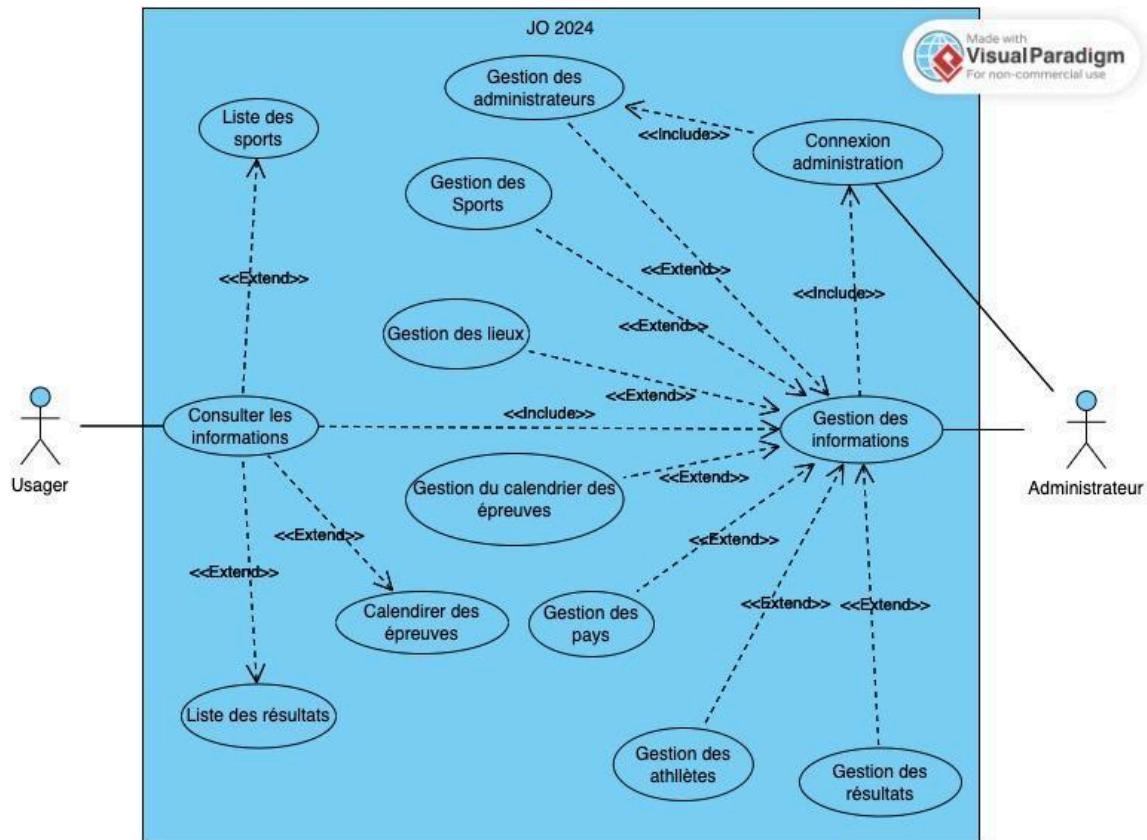
Made with
Visual Paradigm
For non-commercial use



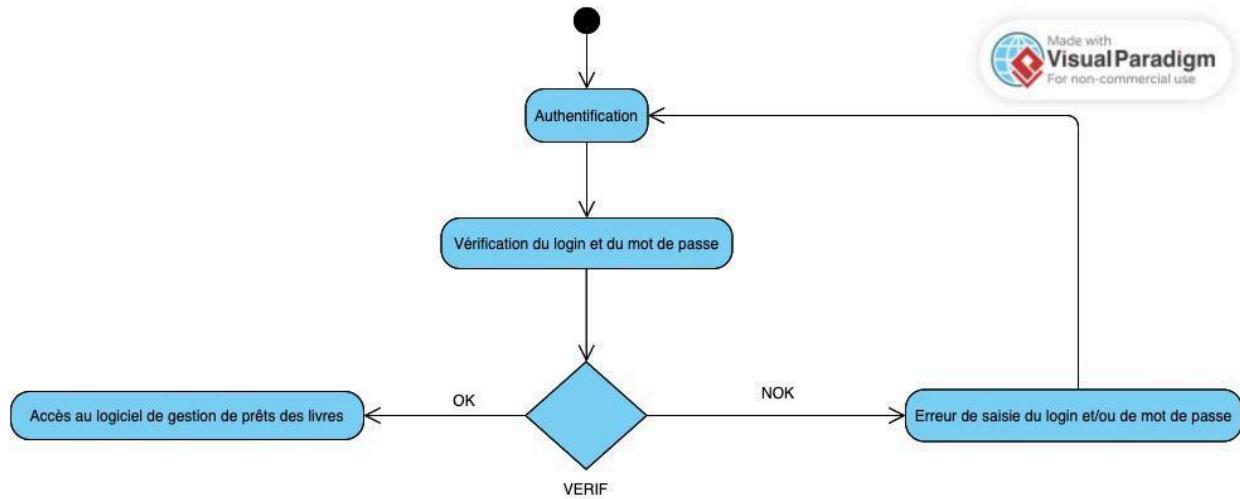
6.2. Le back-end

6.2.1. Diagramme de cas d'utilisation

Réalisation du diagramme de cas d'utilisation du site web réalisé sur Visual Paradigm.

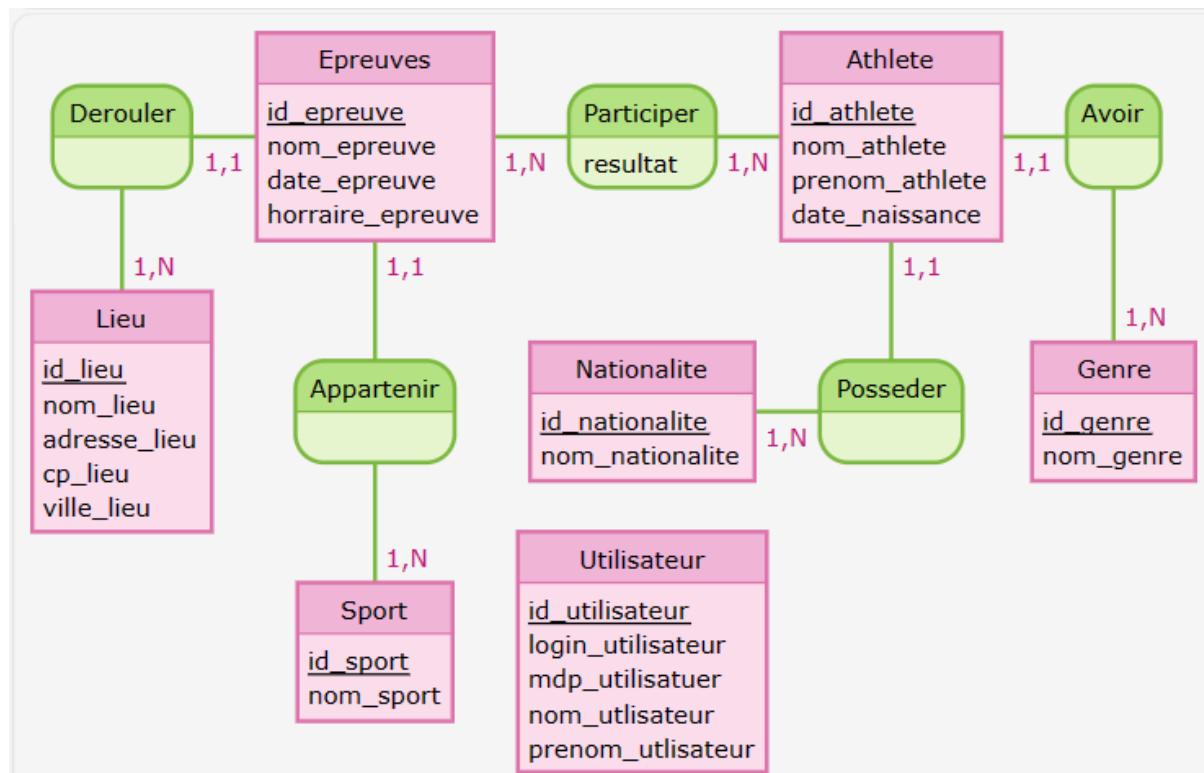


6.2.2. Diagramme d'activités



6.2.3. Modèles Conceptuel de Données (MCD)

Création d'un modèle conceptuel de données pour représenter de manière transparente les entités, les relations et les contraintes du système d'information lié à l'événement.



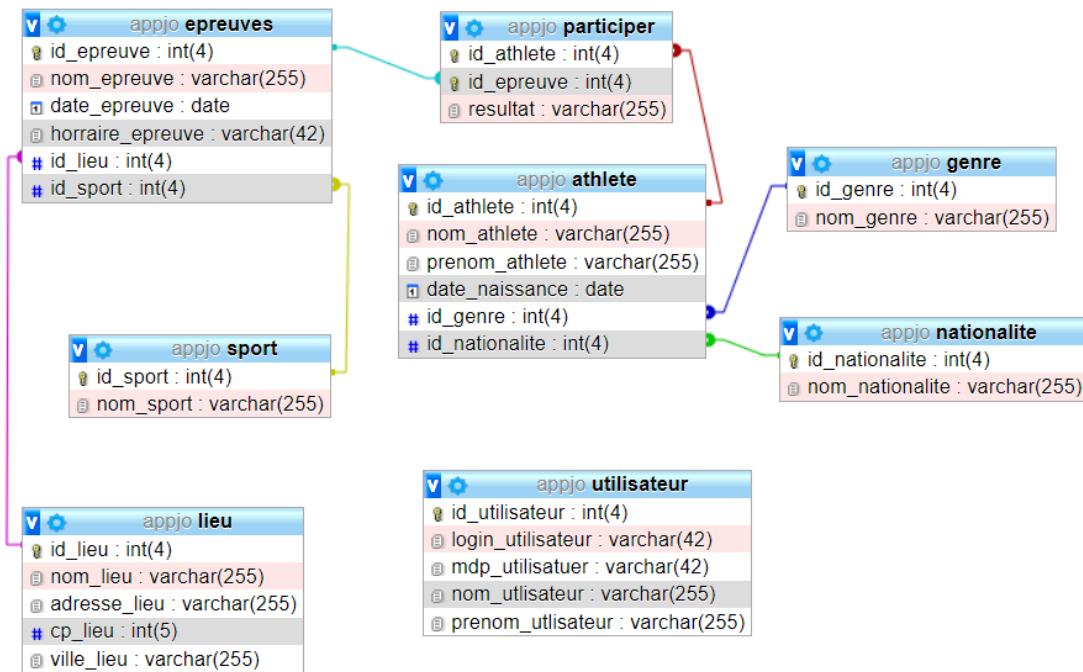
6.2.4. Modèles Logique de Données (MLD)

Conception du modèle logique de données, pour définir plus précisément la structure des données, en utilisant des concepts tels que les tables, les clés primaires et étrangères, et les relations entre les entités.

- Athlete (id_athlete, nom_athlete, prenom_athlete, date_naissance, #id_genre, #id_nationalite)
- Epreuves (id_epreuve, nom_epreuve, date_epreuve, horraire_epreuve, #id_lieu, #id_sport)
- Genre (id_genre, nom_genre)
- Lieu (id_lieu, nom_lieu, adresse_lieu, cp_lieu, ville_lieu)
- Nationalite (id_nationalite, nom_nationalite)
- Participer (#id_athlete, #id_epreuve, resultat)
- Sport (id_sport, nom_sport)
- Utilisateur (id_utilisateur, login_utilisateur, mdp_utilisatuer, nom_utlisateur, prenom_utlisateur)

6.2.5. Modèle Physique de Données (MPD)

Élaboration d'un modèle physique de données qui spécifie les détails techniques de mise en œuvre du MLD dans un système de gestion de base de données spécifique. Le MPD a inclus des éléments tels que les types de données, les index, les contraintes d'intégrité, et d'autres aspects techniques nécessaires à la création effective de la base de données pour les Jeux Olympiques.



7. Technologies utilisées

Dans le cadre des technologies utilisées pour notre projet, nous avons adopté une approche soigneusement planifiée, couvrant divers aspects du développement web.

7.1. Langages de développement Web



7.2. Base de données



8. Sécurité

8.1. Login et protection des pages administrateurs

Pour garantir la sécurité d'une page de connexion administrateur, voici les étapes à suivre :

- ❖ Démarrer une session PHP pour stocker les variables de session et inclure le fichier de connexion à la base de données.
- ❖ Vérifier si la requête est une méthode POST afin de récupérer les identifiants soumis par l'utilisateur, puis préparer une requête SQL pour obtenir les informations de l'utilisateur spécifié.
- ❖ Utiliser PDO pour lier la variable :login à la valeur du login, ce qui évite les injections SQL. Ensuite, exécuter la requête préparée et vérifier si le mot de passe correspond à celui stocké dans la base de données en utilisant la fonction password_verify.
- ❖ Si les identifiants sont corrects, stocker les informations de l'utilisateur dans la session et rediriger l'utilisateur vers la page d'administration. Sinon, stocker un message d'erreur dans la session et rediriger l'utilisateur vers la page de connexion.
- ❖ Optimiser le code en libérant les ressources associées à la requête préparée et en fermant la connexion à la base de données.

Exemple de code résumant ce texte:

```
<?php
session_start(); // Démarre la session PHP pour stocker des variables de session.

require_once("database.php"); // Inclut le fichier de connexion à la base de données.

if ($_SERVER["REQUEST_METHOD"] == "POST") { // Vérifie si la requête est une méthode POST (formulaire soumis).
    $login = $_POST["login"]; // Récupère la valeur du champ "login" du formulaire.
    $password = $_POST["password"]; // Récupère la valeur du champ "password" du formulaire.

    // Prépare 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); // Prépare la requête avec PDO.
    $stmt->bindParam(":login", $login, PDO::PARAM_STR); // Lie la variable :login à la valeur du login, évitant les injections SQL.

    if ($stmt->execute()) { // Exécute la requête préparée.
        $row = $stmt->fetch(PDO::FETCH_ASSOC); // Récupère la première ligne de résultat de la requête.

        if ($row && password_verify($password, $row["password"])) {
            // Si une ligne est récupérée et le mot de passe correspond à celui stocké dans la base de données.
            $_SESSION["id_utilisateur"] = $row["id_utilisateur"]; // Stocke l'ID utilisateur dans la session.
            $_SESSION["nom_utilisateur"] = $row["nom_utilisateur"]; // Stocke le nom de l'utilisateur dans la session.
            $_SESSION["prenom_utilisateur"] = $row["prenom_utilisateur"]; // Stocke le prénom de l'utilisateur dans la session.
            $_SESSION["login"] = $row["login"]; // Stocke le login de l'utilisateur dans la session.

            header("location: ../pages/admin/admin.php"); // Redirige vers la page d'administration.
            exit(); // Termine le script.
        } else {
            $_SESSION['error'] = "Login ou mot de passe incorrect.";
            header("location: ../pages/login.php"); // Redirige vers la page de login avec un message d'erreur.
        }
    }
}
```

Code permettant la connexion sécurisé de l'espace administration

```

1  <?php
2
3  // page_admin.php
4
5  session_start();
6
7  if (!isset($_SESSION['id_utilisateur'])) {
8      // Rediriger vers la page de connexion si l'utilisateur n'est pas authentifié
9      header('Location: login.php');
10     exit();
11 }
12
13 // Le reste du code de la page d'administration...
14
15 ?>

```

Code permettant de sécuriser l'accès aux pages administrateur

8.2. Cryptage des mots de passe avec Bcrypt

Bcrypt est un algorithme de hachage spécialement conçu pour le stockage sécurisé des mots de passe. Son fonctionnement repose sur une itération paramétrable, ce qui le rend résistant aux attaques par force brute et par dictionnaire.

Pour utiliser Bcrypt de manière sécurisée, il est recommandé de passer par la fonction PHP "password_hash()". Voici un exemple d'utilisation de l'algorithme Bcrypt avec cette fonction.

```

if ($stmt->execute()) { // Exécute la requête préparée.
    $row = $stmt->fetch(PDO::FETCH_ASSOC); // Récupère la première ligne de résultat de la requête.

    if ($row && password_verify($password, $row["password"])) {
        // Si une ligne est récupérée et le mot de passe correspond à celui stocké dans la base de données.
        $_SESSION["id_utilisateur"] = $row["id_utilisateur"]; // Stocke l'ID utilisateur dans la session.
        $_SESSION["nom_utilisateur"] = $row["nom_utilisateur"]; // Stocke le nom de l'utilisateur dans la session.
        $_SESSION["prenom_utilisateur"] = $row["prenom_utilisateur"]; // Stocke le prénom de l'utilisateur dans la session.
        $_SESSION["login"] = $row["login"]; // Stocke le login de l'utilisateur dans la session.

        header("location: ../pages/admin/admin.php"); // Redirige vers la page d'administration.
        exit(); // Termine le script.
    } else {
        $_SESSION['error'] = "Login ou mot de passe incorrect.";
        header("location: ../pages/login.php"); // Redirige vers la page de login avec un message d'erreur.
    }
}

```

Utilisation Vérification Du Mot De Passe

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

- ❖ Validation des Entrées Utilisateurs
- ❖ Validez et filtrez toutes les entrées utilisateur du côté serveur pour vous assurer qu'elles correspondent aux attentes.
- ❖ 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).
- ❖ Utilisation de HTTPOnly Cookies

8.4. Protection contre les injections SQL

L'injection SQL est une technique d'injection de code utilisée pour attaquer les applications basées sur les données, dans laquelle des instructions SQL malveillantes sont insérées dans un champ de saisie pour exécution. Il est donc crucial de protéger le code de toutes injections SQL possible, plusieurs méthode doivent donc être appliquée:

L'utilisation de requêtes préparées:

En manipulant la base de données depuis le code, il est préférable d'utiliser PDO.

```
// Prépare 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); // Prépare la requête avec PDO.
$stmt->bindParam(":login", $login, PDO::PARAM_STR); // Lie la variable :login à la valeur du login, évitant les injections SQL.
```

utilisation de requêtes préparées

Filtrer les données de saisie:

Enfin, on peut valider les données d'entrée pour assurer qu'elles correspondent au format attendu. On peut également ajouter des filtres ou des expressions régulières pour garantir que les données respectent les critères définis.

```
// Assurez-vous d'obtenir des données sécurisées et filtrées
$nomUtilisateur = filter_input(INPUT_POST, 'nomUtilisateur', FILTER_SANITIZE_STRING);
$prenomUtilisateur = filter_input(INPUT_POST, 'prenomUtilisateur', FILTER_SANITIZE_STRING);
$login = filter_input(INPUT_POST, 'login', FILTER_SANITIZE_STRING);
$password = password_hash(filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING), PASSWORD_BCRYPT);
```

utilisation du filtrage des données