

Cahier des charges techniques

Sommaire

I. Contexte du projet

I.I. Présentation du projet

I.II. Date de rendu du projet

II. Besoins fonctionnels

III. Ressources nécessaires à la réalisation du projet

III.I. Ressources matérielles

III.II. Ressources logicielles

IV. Gestion du projet

V. Conception du projet

V.I. Le front-end

V.I.I. Wireframes

V.I.II. Maquettes

V.I.III. Arborescences

V.II. Le back-end

V.II.I. Diagramme de cas d'utilisation

V.II.II. Diagramme d'activités

V.II.III. Modèles Conceptuel de Données (MCD)

V.II.IV. Modèle Logique de Données (MLD)

V.II.V. Modèle Physique de Données (MPD)

VI. Technologies utilisées

VI.I. Langages de développement Web

VI.II. Base de données

VII. Sécurité

VII.I. Login et protection des pages administrateurs

VII.II. Cryptage des mots de passe avec Bcrypt

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

VII.IV. Protection contre les injections SQL

I.Contexte du projet

I.I. 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.

I.II. Date de rendu du projet

Le projet doit être rendu au plus tard le 7 novembre 2024.

II. 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.

III. Ressources nécessaires à la réalisation du projet

III.I. Ressources matérielles

Les ressources matérielles nécessaires pour réaliser le projet sont :

- Pc portable, connexion internet (Wi-Fi)

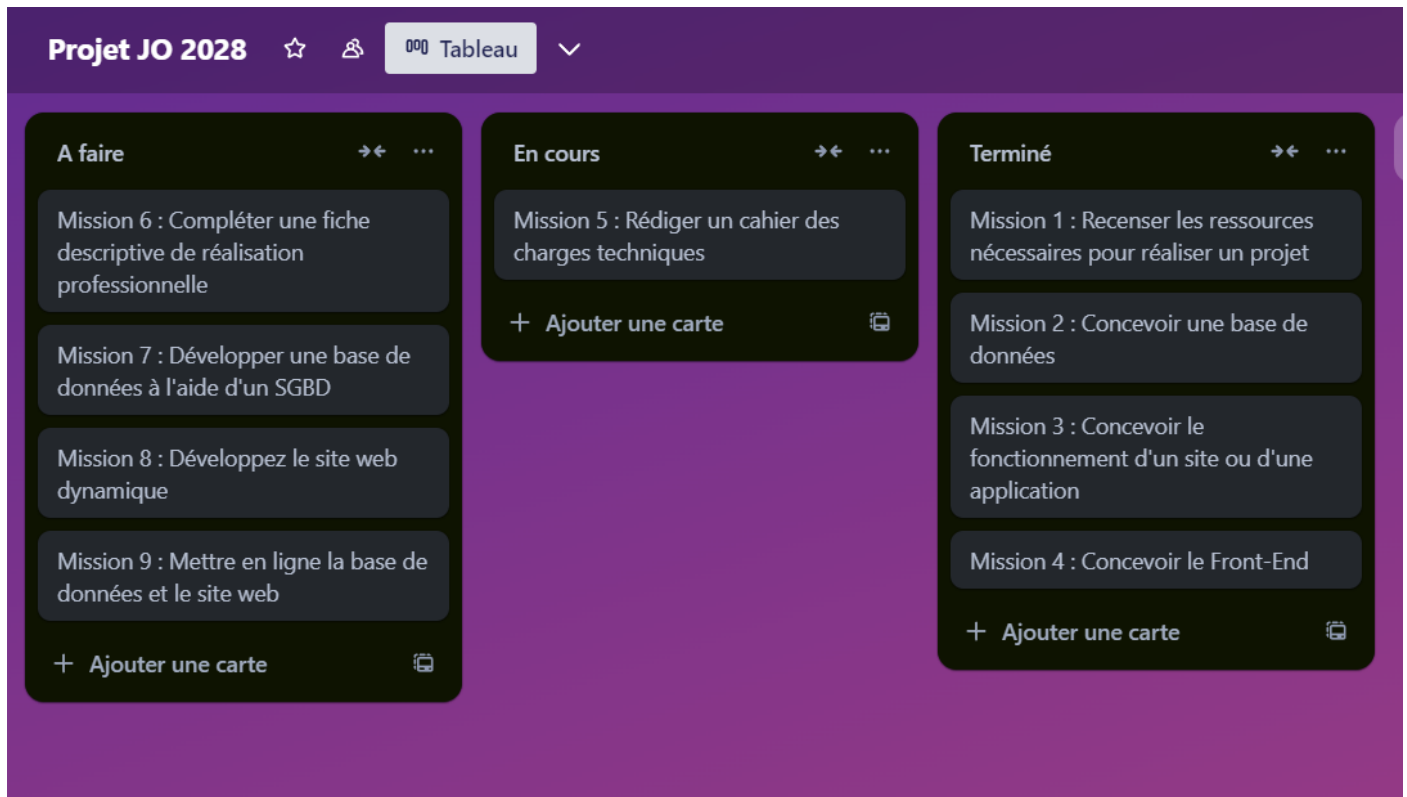
III.II Ressources logicielles

Les ressources logicielles nécessaires pour réaliser le projet sont :

- Un IDE : Visual Studio Code
- Une plateforme de développement collaborative : Github
- MAMP :
 - o Serveur web Apache
 - o SGBDR : MySQL

IV. 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.

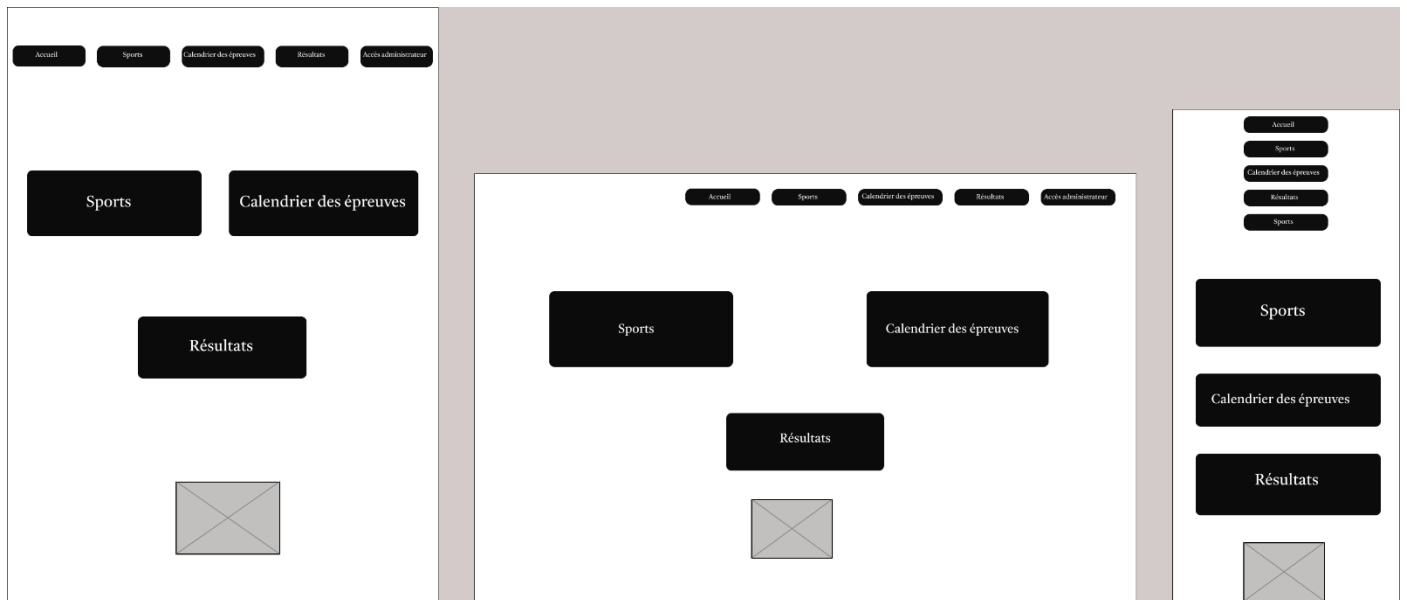


V. Conception du projet

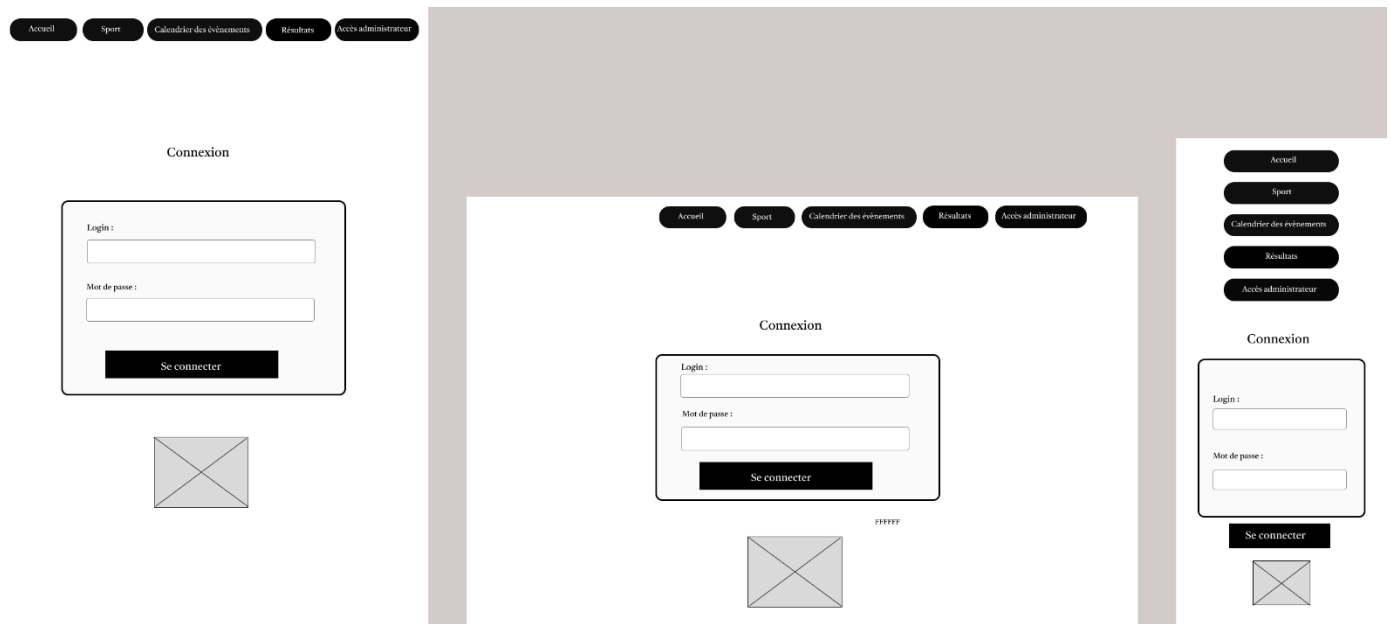
V.I. Le front-end

V.I.I. Wireframes

Afin de réaliser les Wireframes, on a utilisé l'outil Figma.



Page de Connexion :



Page de Gestion des Sports :

Liste des sports

Ajouter un Sport

Sport	Modifier	Supprimer
Athlétisme	Modifier	Supprimer
Bone	Modifier	Supprimer
Cyclisme	Modifier	Supprimer
Escalade	Modifier	Supprimer
Gymnastique	Modifier	Supprimer
Lancer	Modifier	Supprimer
Natation	Modifier	Supprimer
Saut en hauteur	Modifier	Supprimer
Saut en longueur	Modifier	Supprimer
VTT	Modifier	Supprimer

Accueil administration



Liste des sports

Ajouter un Sport

Sport	Modifier	Supprimer
Athlétisme	Modifier	Supprimer
Bone	Modifier	Supprimer
Cyclisme	Modifier	Supprimer
Escalade	Modifier	Supprimer
Gymnastique	Modifier	Supprimer
Lancer	Modifier	Supprimer
Natation	Modifier	Supprimer
Saut en hauteur	Modifier	Supprimer
Saut en longueur	Modifier	Supprimer
VTT	Modifier	Supprimer

Accueil administration



Accueil Administration

Gestion résultats

Gestion Athlètes

Gestion Genres

Gestion Pays

Gestion Calendrier

Gestion Lieux

Gestion Sports

Gestion Sports

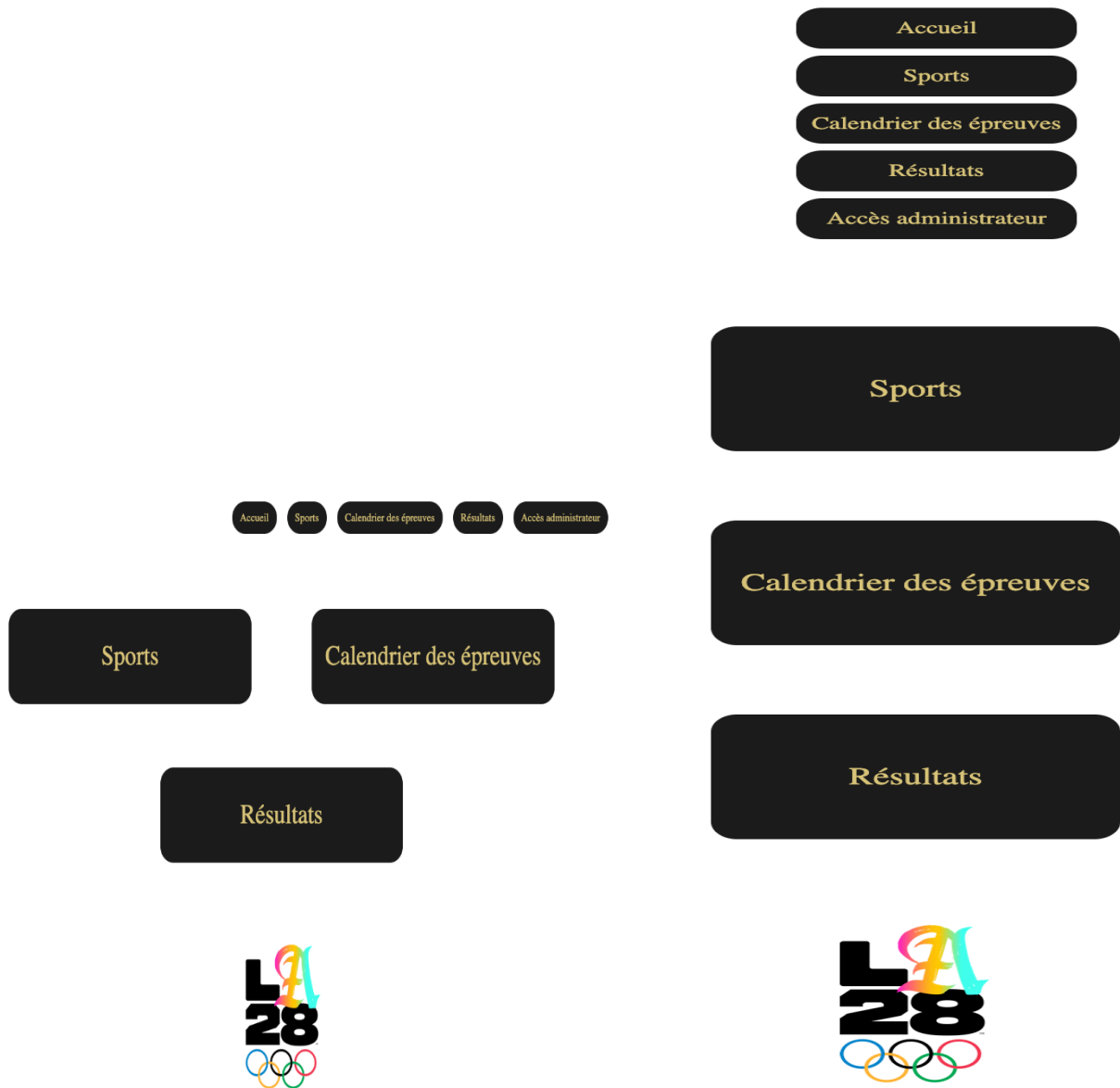
Deconnexion

Sport	Modifier	Supprimer
Athlétisme	Modifier	Supprimer
Bone	Modifier	Supprimer
Cyclisme	Modifier	Supprimer
Escalade	Modifier	Supprimer
Gymnastique	Modifier	Supprimer
Lancer	Modifier	Supprimer
Natation	Modifier	Supprimer
Saut en hauteur	Modifier	Supprimer
Saut en longueur	Modifier	Supprimer
VTT	Modifier	Supprimer

Accueil administration



Maquettes de la page d'accueil (index) ;



Maquettes de la page de Connexion :

[Accueil](#)[Sports](#)[Calendrier des évènements](#)[Résultats](#)[Accès administrateur](#)

Connexion

Login :

Mot de passe :

Se connecter

[Accueil](#)[Sports](#)[Calendrier des évènements](#)[Résultats](#)[Accès administrateur](#)

Connexion

Login :

Mot de passe :

Se connecter



Page de gestion des sports :

- Accueil Administration
- Gestion Sports
- Gestion Lieux
- Gestion Calendrier
- Gestion Pays
- Gestion Genres
- Gestion Athlètes
- Gestion Résultats
- Déconnexion

Liste des Sports

Ajouter un Sport

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

Accueil administration



- Accueil Administration
- Gestion Sports
- Gestion Lieux
- Gestion Calendrier
- Gestion Pays
- Gestion Genres
- Gestion Athlètes
- Gestion Résultats
- Déconnexion

Liste des Sports

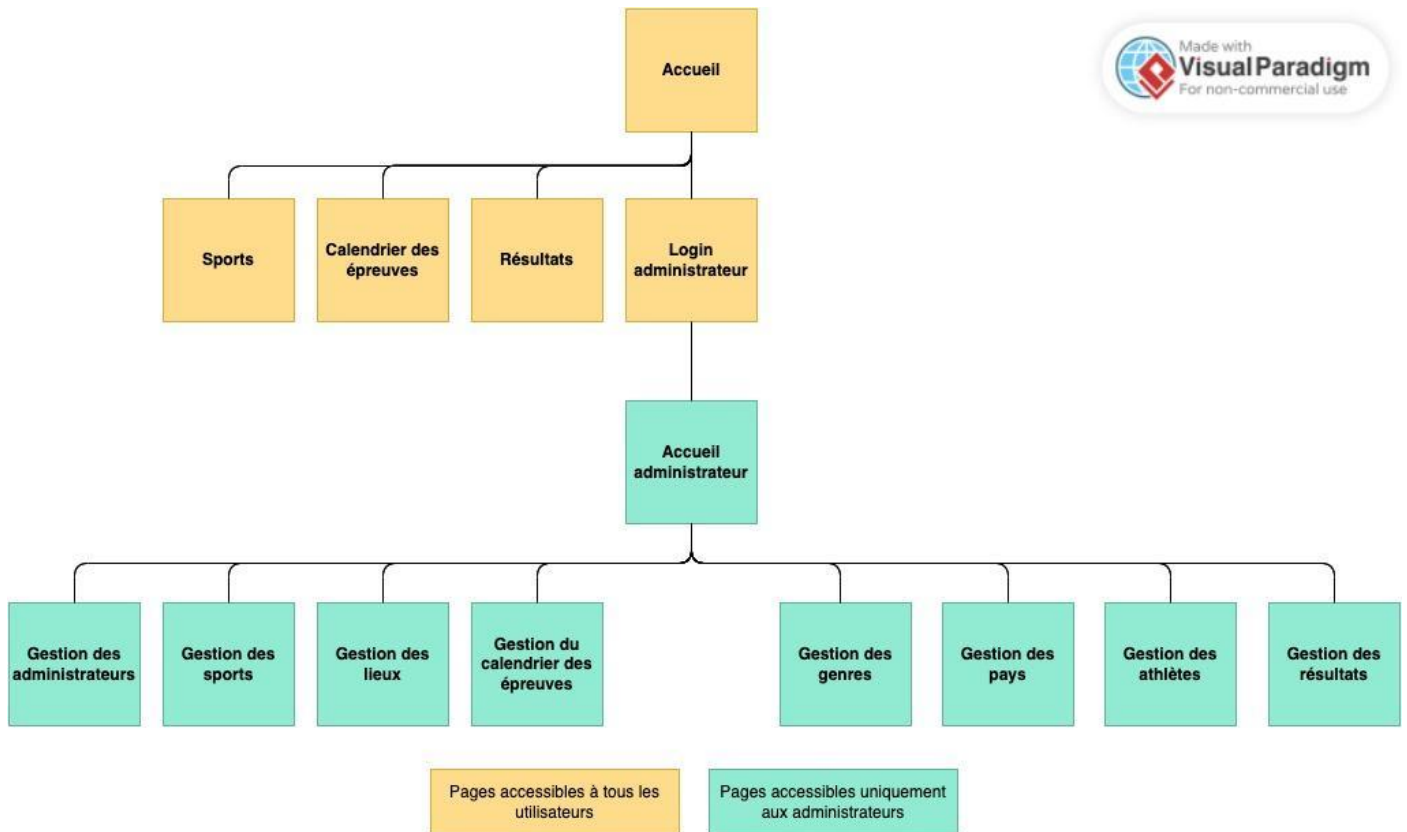
Ajouter un Sport

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

Accueil administration

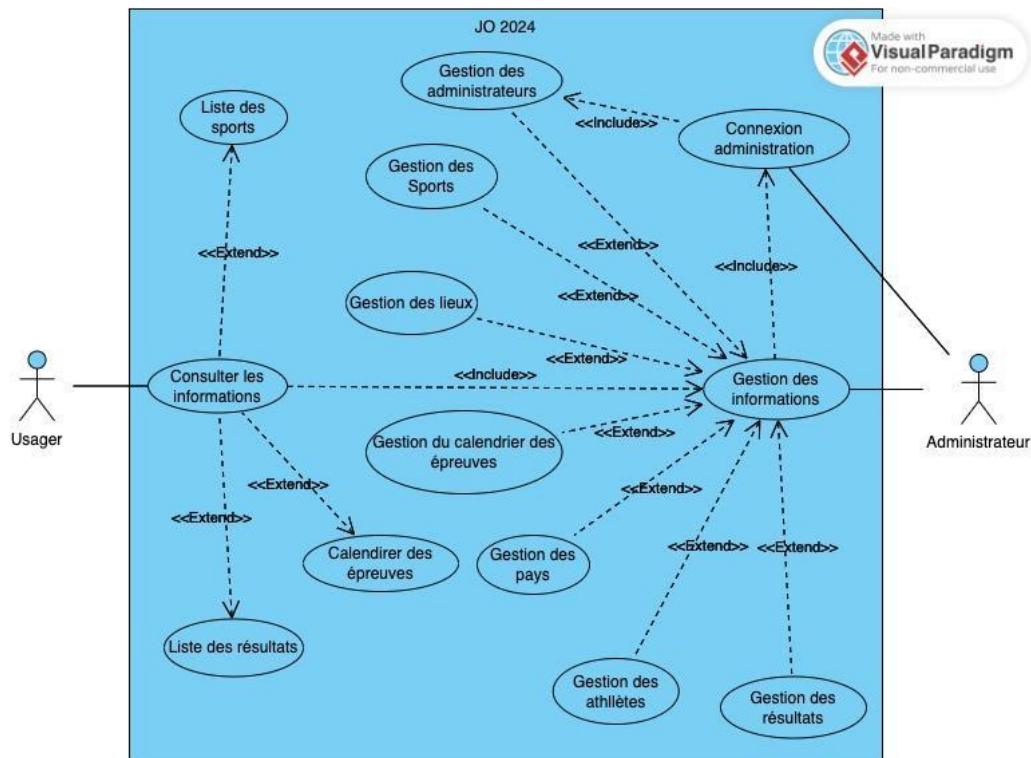


V.I.III. Arborescences

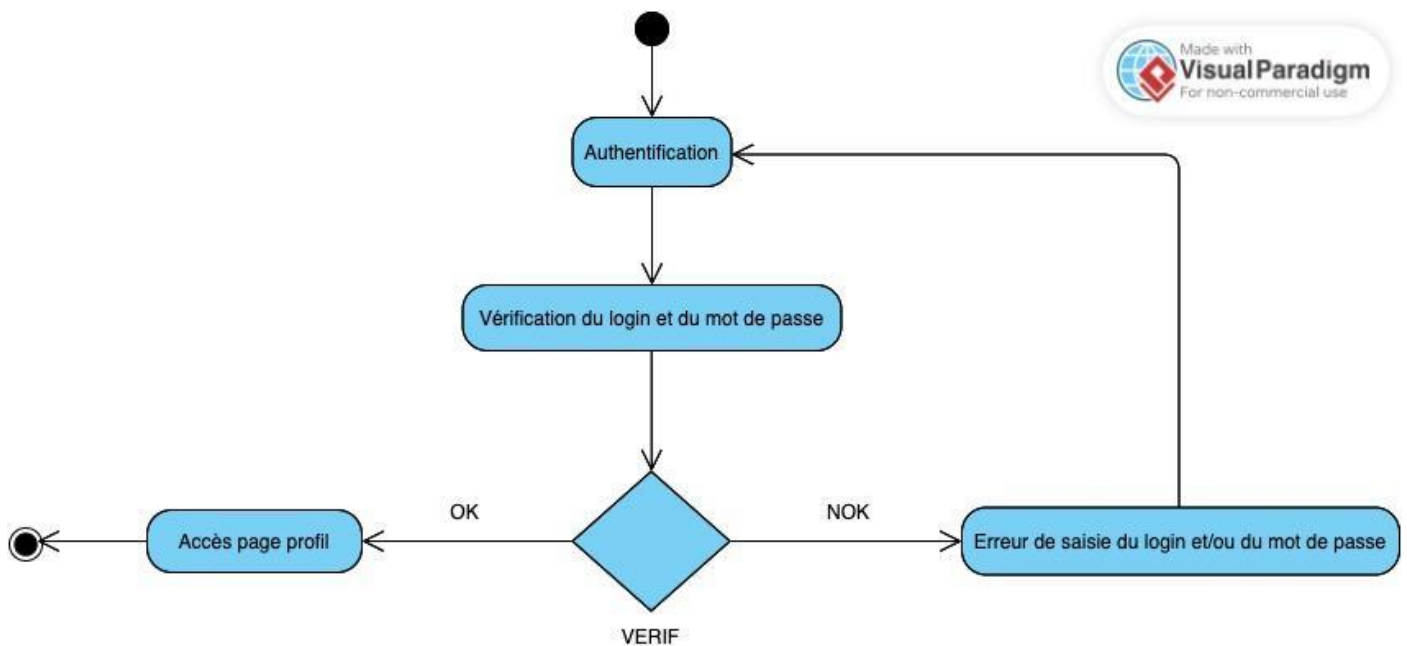


V.II. Le back-end

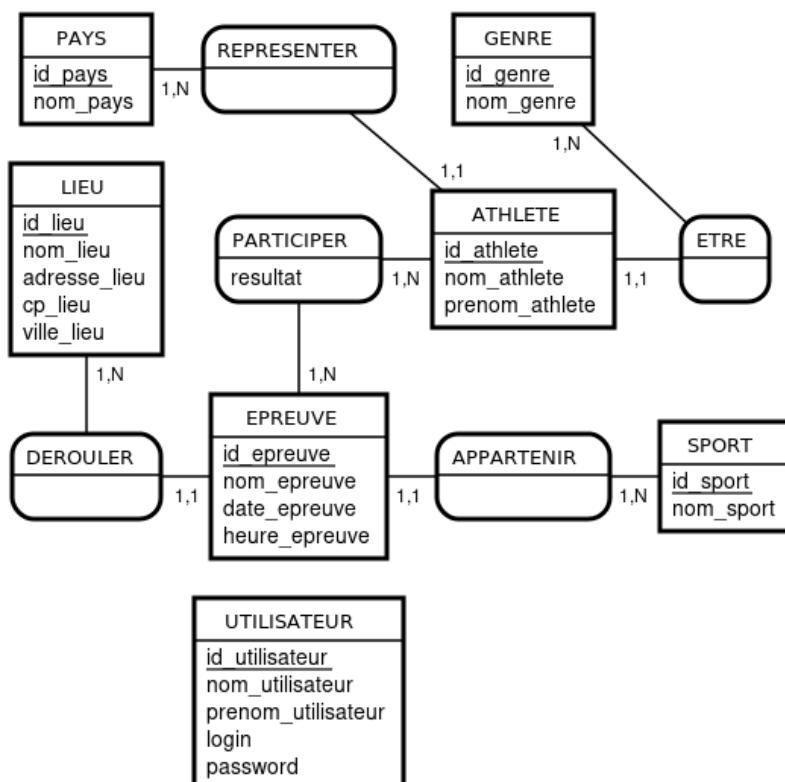
V.II.I. Diagramme de cas d'utilisation



V.II. II. Diagramme d'activités



V.II.III. Modèles Conceptuel de Données (MCD)



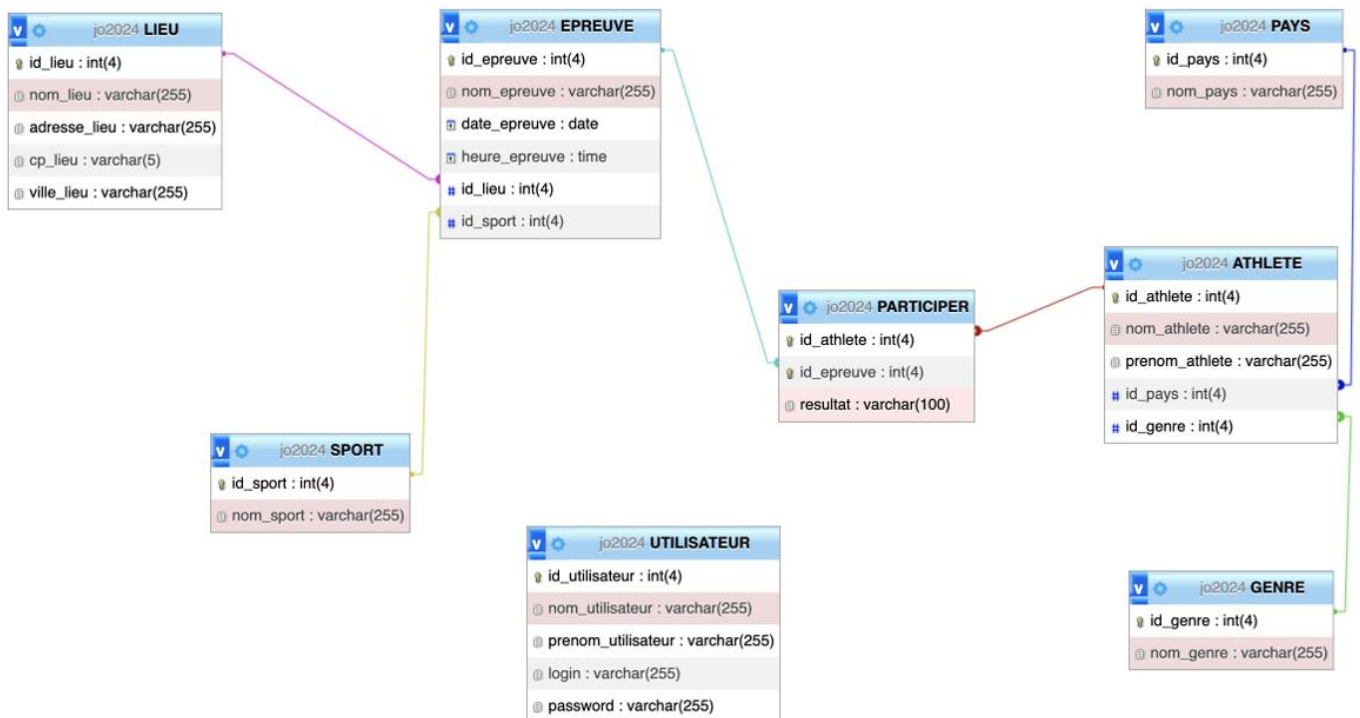
V.II. IV. Modèle Logique de Données (MLD)

- **ATHLETE** (id_athlete, nom_athlete, prenom_athlete, #id_pays, #id_genre)
- **EPREUVE** (id_epreuve, nom_epreuve, date_epreuve, heure_epreuve, #id_lieu, #id_sport)

- GENRE (id_genre, nom_genre)
- LIEU (id_lieu, nom_lieu, adresse_lieu, cp_lieu, ville_lieu)
- PARTICIPER (#id_athlete, #id_epreuve, resultat)
- PAYS (id_pays, nom_pays)
- SPORT (id_sport, nom_sport)

UTILISATEUR (id_utilisateur, nom_utilisateur, prenom_utilisateur, login, password)

V.II.V. Modèle Physique de Données (MPD)



VI. Technologies utilisées

VI.I. 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 : 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.

VI.II. 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

VII. Sécurité

VII.I. 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.

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

VII.II. Cryptage des mots de passe avec Bcrypt

Bcrypt : algorithme php de hachage sécurisé pour protéger les mots de passe.

Hachage : Processus qui transforme un mot de passe en une chaîne de caractères fixe, ce qui rend le mot de passe original non récupérable.

Bcrypt a deux principaux avantages :

- **Le Salage** : chaque mot de passe est associé à un sel (chaîne aléatoire), ce qui fait que les hachages sont différents même pour les mots de passe identiques.

- **Coût ajustable** : On peut définir le nombre de hachage, ce qui rend le processus plus long et plus difficile pour les attaquants.

En PHP :

Hachage d'un mot de passe : on utilise **password_hash()**

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

Vérification du mot de passe : on utilise `password_verify()`

Ex :

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

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

La faille XSS permet d'injecter du code (HTML, JavaScript) dans des variables où des bases de données mal protégées.

L'attaque peut être permanente (stockée) ou non, le principe est d'injecter du code pour que le site de connecte à un site distant contenant un code malveillant.

Ceci peut permettre à un attaquant d'accéder aux cookies ou autres informations sensibles d'un utilisateur.

Pour se protéger contre les attaques CSS, deux principales méthodes existent :

- **Supprimer le contenu HTML** : utiliser des fonctions comme « `strip_tags()` » pour éliminer les balises HTML des saisies de l'utilisateur, ce qui fait que seules des données textuelles sont stockées.
- **Neutraliser les caractères** : Utiliser des fonctions comme « `htmlspecialchars()` » ou « `htmlentities()` » pour remplacer les caractère spéciaux par leur entités HTML, ainsi on empêche l'exécution de scripts en conservant le texte d'origine.

VII.IV. Protection contre les injections SQL

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

Protection avec PHP :

- Requêtes préparées (PDO)
- Filtrer les données utilisateur pour s'assurer qu'elles sont au bon format
 - Exemple : `$id = filter_var($id_saisi, FILTER_VALIDATE_INT);`