**Cahier des charges techniques**



Louay CHARNI

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

Nous avons été sélectionnés par le comité d’organisation des Jeux Olympiques de Los Angeles 2028 pour développer une application web. Cette application permettra aux organisateurs, aux médias et aux spectateurs de consulter des informations sur les sports, les calendriers des épreuves et les résultats des JO 2028. Notre équipe et moi avons pour mission de proposer une solution qui répondra aux besoins de notre client.

## 1.2. Date de rendu du projet

Nous devons rendre le projet au plus tard le 7 novembre 2024.

# 2. Besoins fonctionnels

Le site web comportera une partie accessible au public ainsi qu'une section privée pour la gestion des données. Nous allons stocker ces informations dans une base de données relationnelle, ce qui facilitera leur gestion et mise à jour. Les données pourront être gérées directement via le site à travers un espace administrateur.

# 3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

Ordinateur portable (Connecter à internet)Fixe (Connecter à internet)

## 3.2. Ressources logicielles

Environnement de développement (IDE) : Visual Studio Code

Plateforme de développement collaboratif : Github

Serveur web : Apache (contenue dans MAMP)

Système de base de données relationnel : MySQL (contenue dans MAMP)

Outils de gestion de projet : Trello

Conception UML et arborescence : Visual Paradigme online

Maquettage : Figma

Conception de base de données : Mocodo

Langages utilisés :

Front-end web :

* HTML 5
* CSS 3
* JS VANILLA

Back-end web :

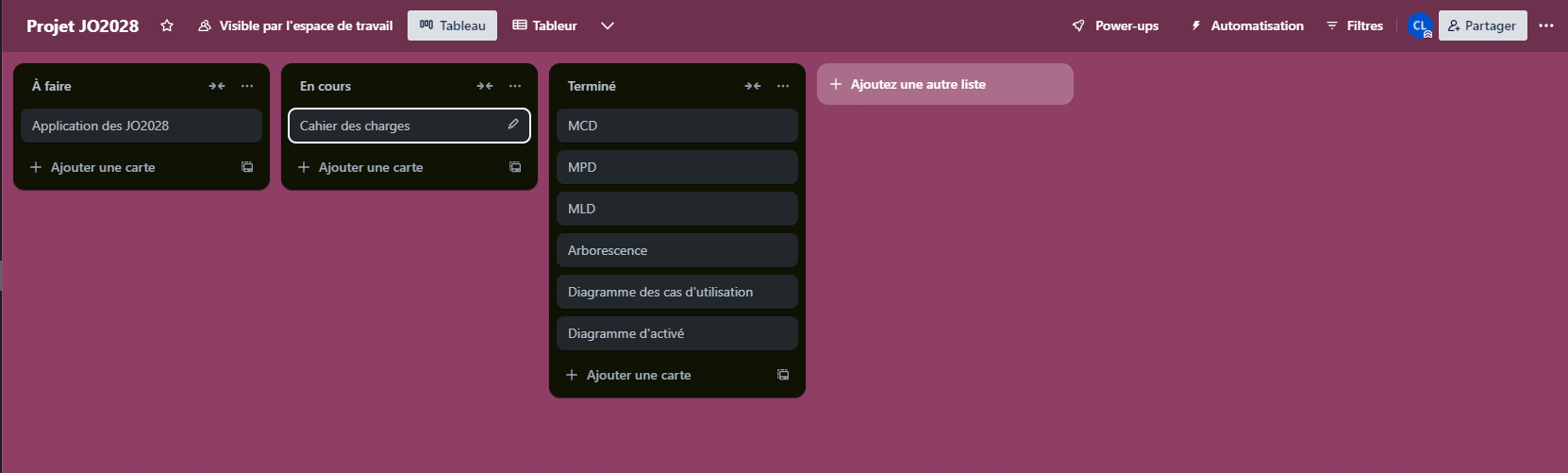
* PHP 8

Sources documentaire possible : StackOverFlow

# 

# 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.

# 

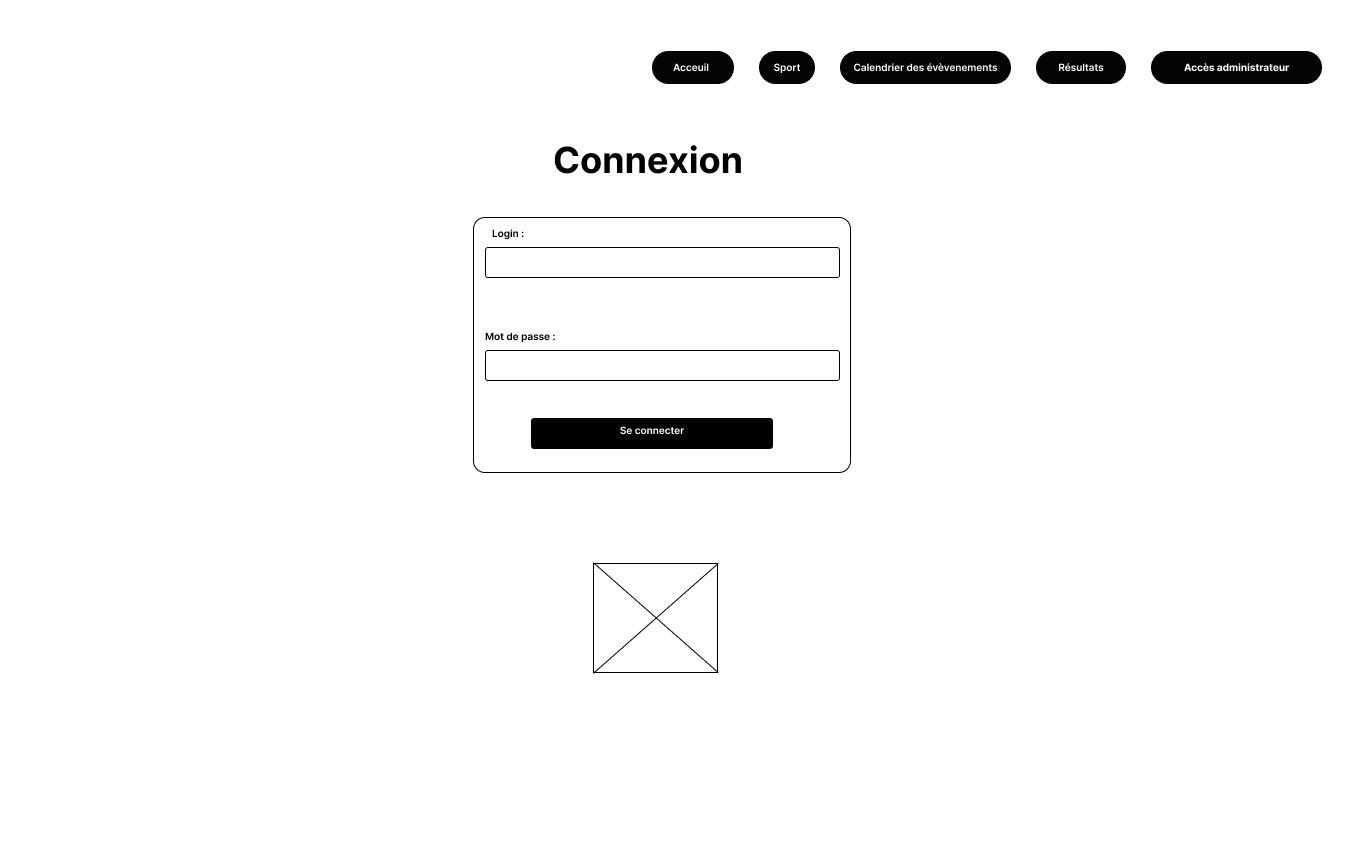
# 

# 5. Conception du projet

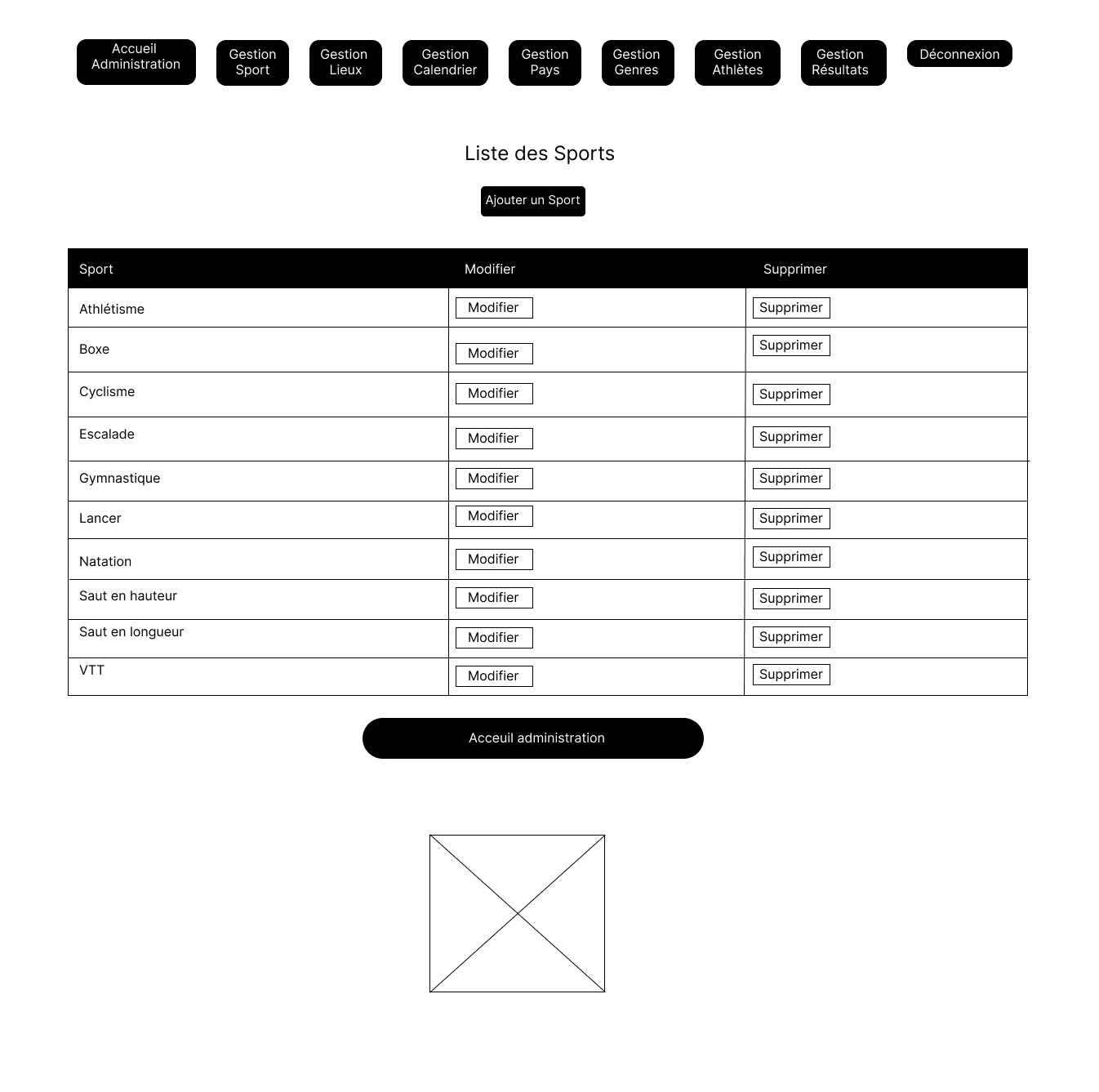
## 5.1. Le front-end

### 5.1.1. Wireframes

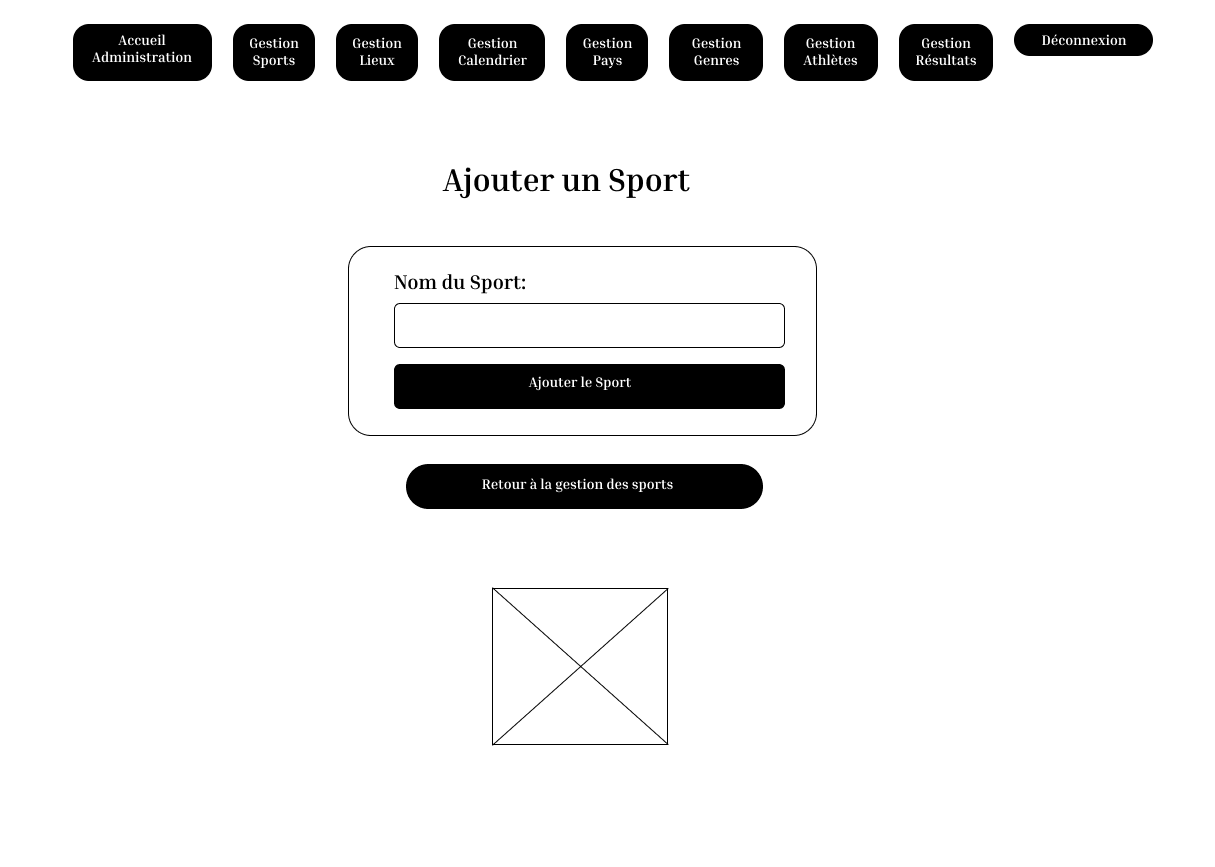
La page de login version ordinateur :



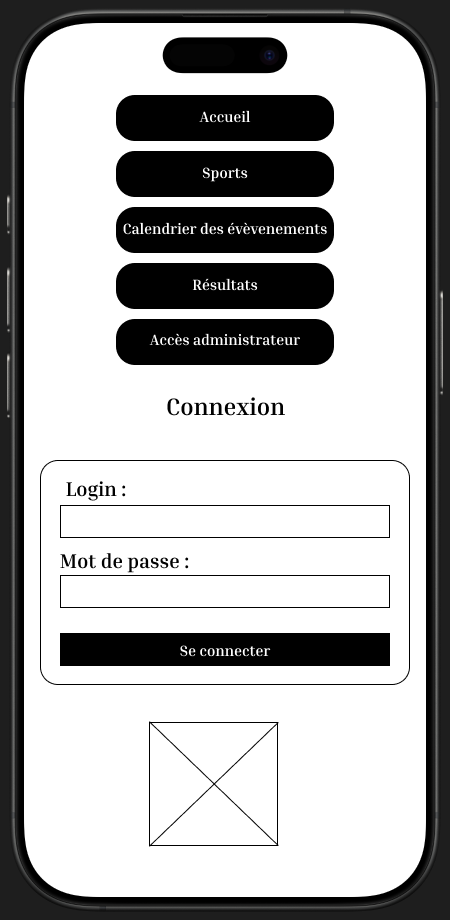
La page de la liste des sports version ordinateur :



La page de l’ajout des sports version ordinateur :



La page de login version téléphone:



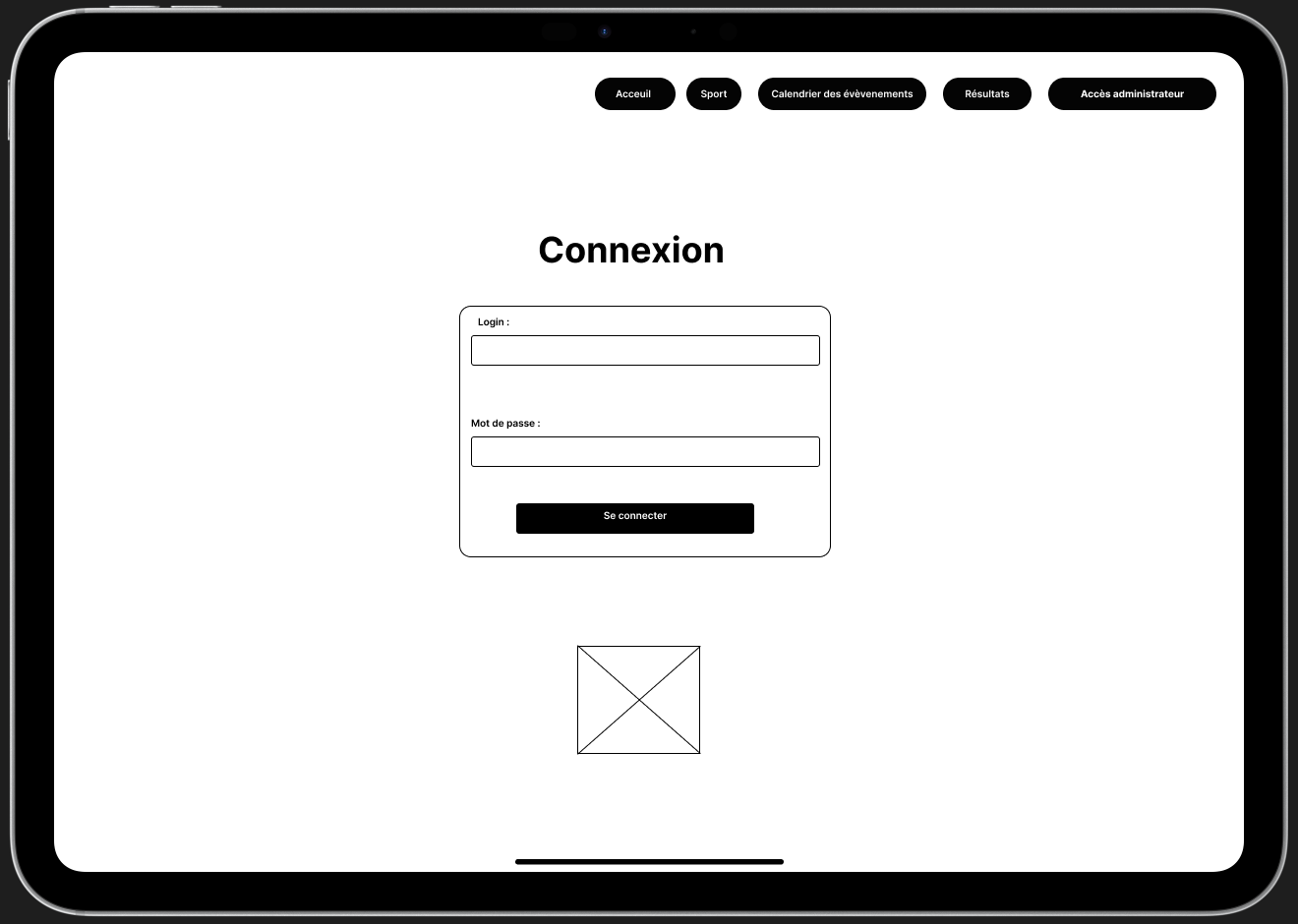
La page de l’ajout des sports version téléphone :



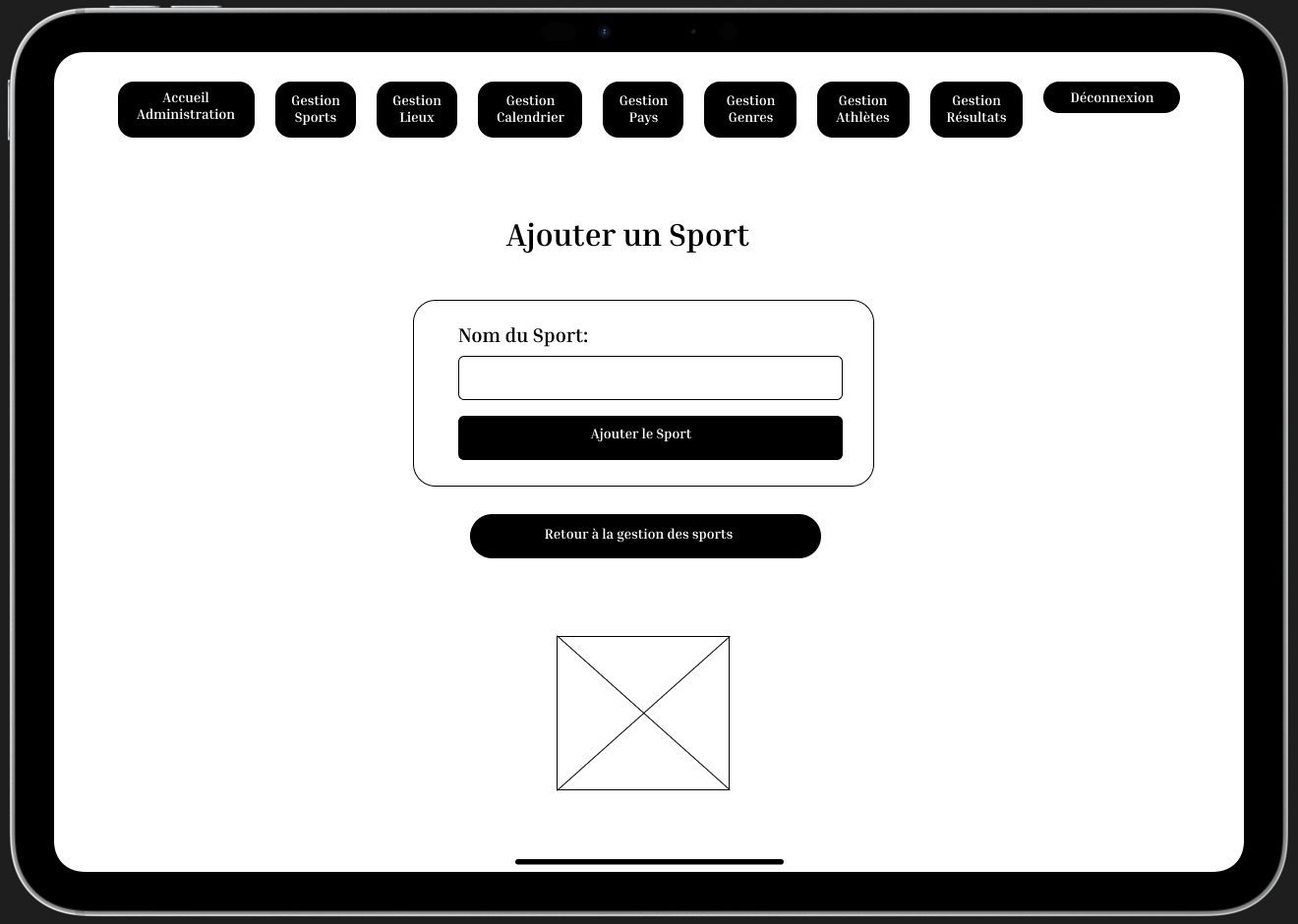
La page de la liste des sports version téléphone :



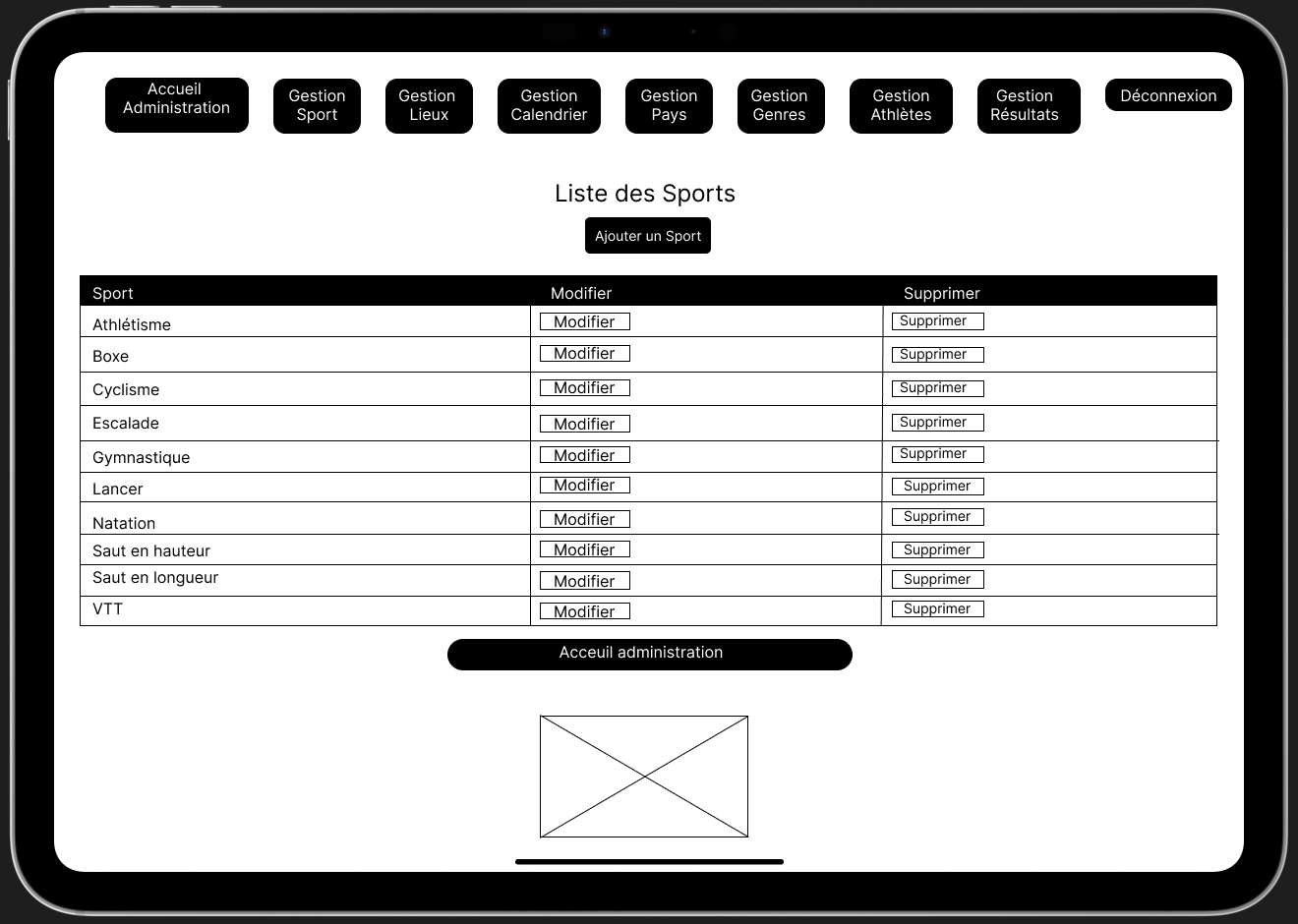
La page de login version tablette :



La page de l’ajout des sports version tablette :

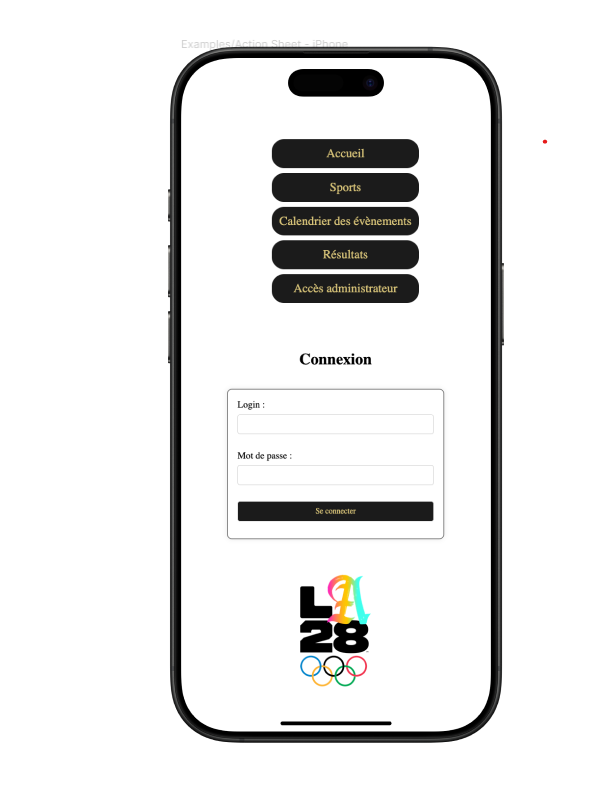


La page de la liste des sports version tablette :

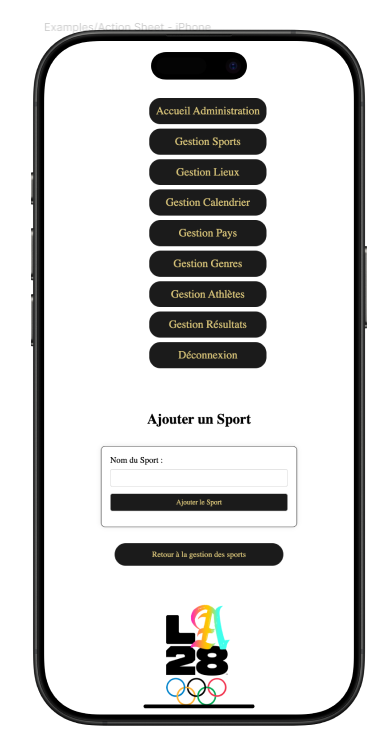


### 5.1.2. Maquettes

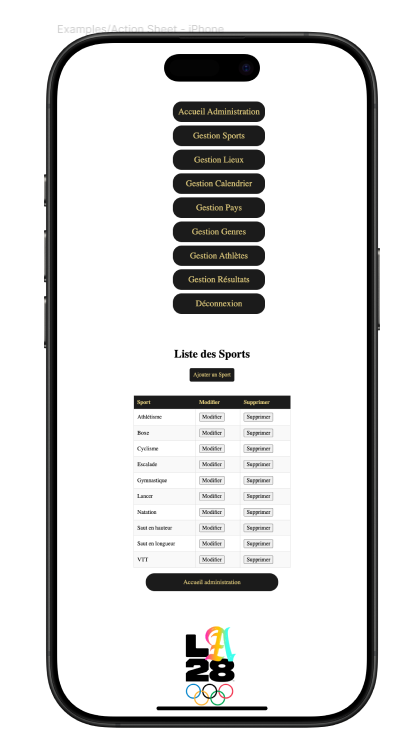
La page de login version téléphone :



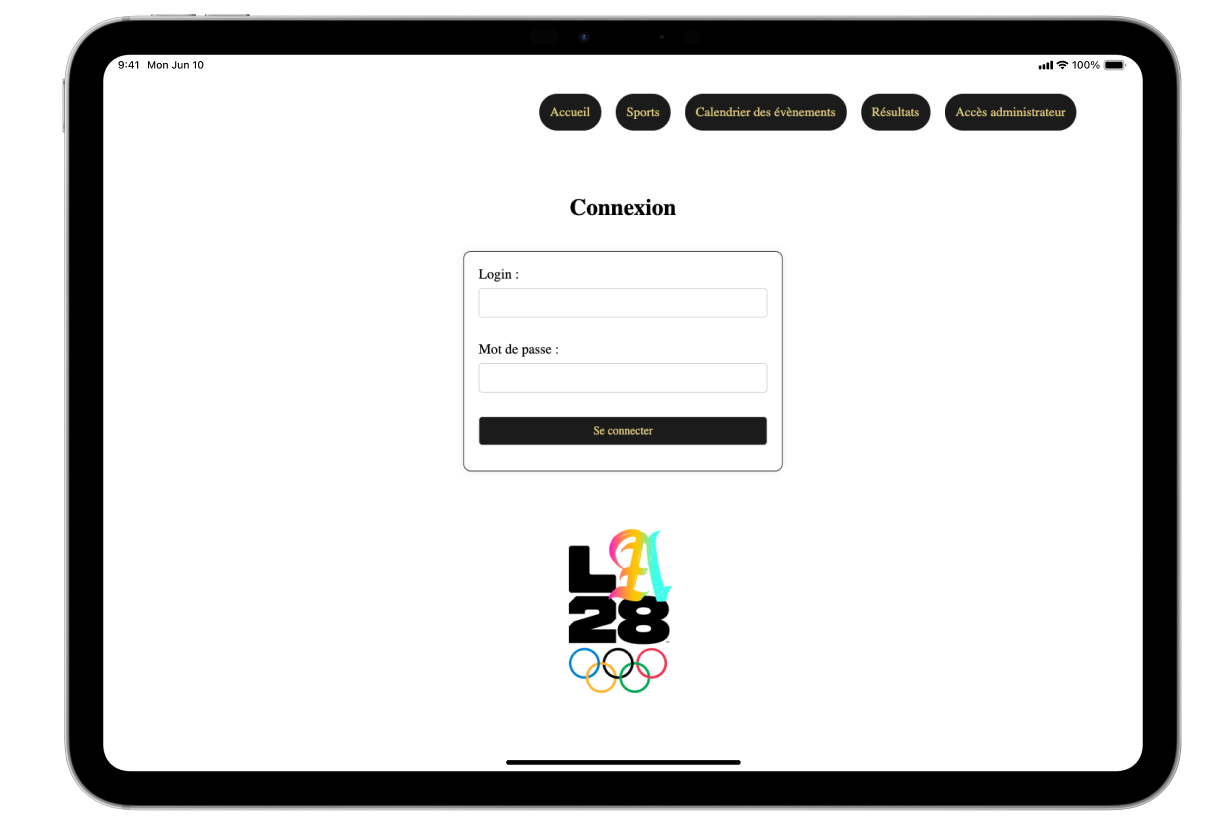
La page d’ajout des sports version téléphone :



La page de la liste des sport version téléphone :



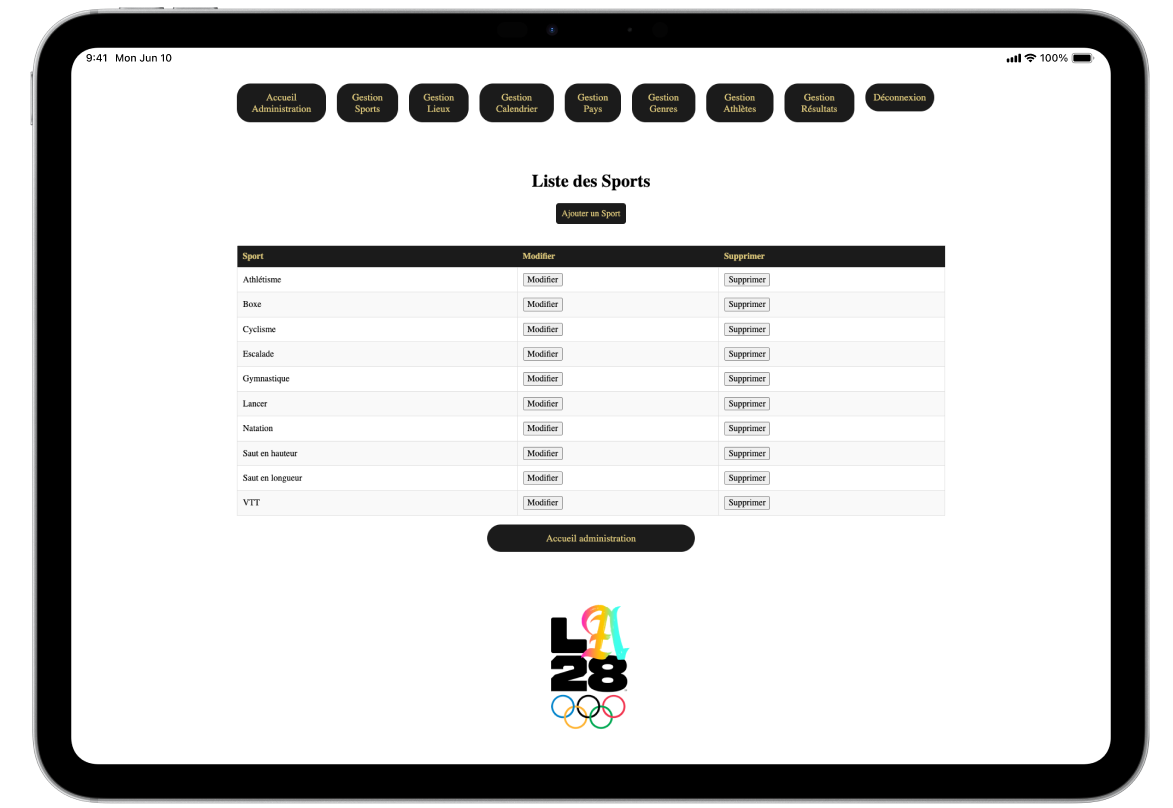
La page de login version tablette :



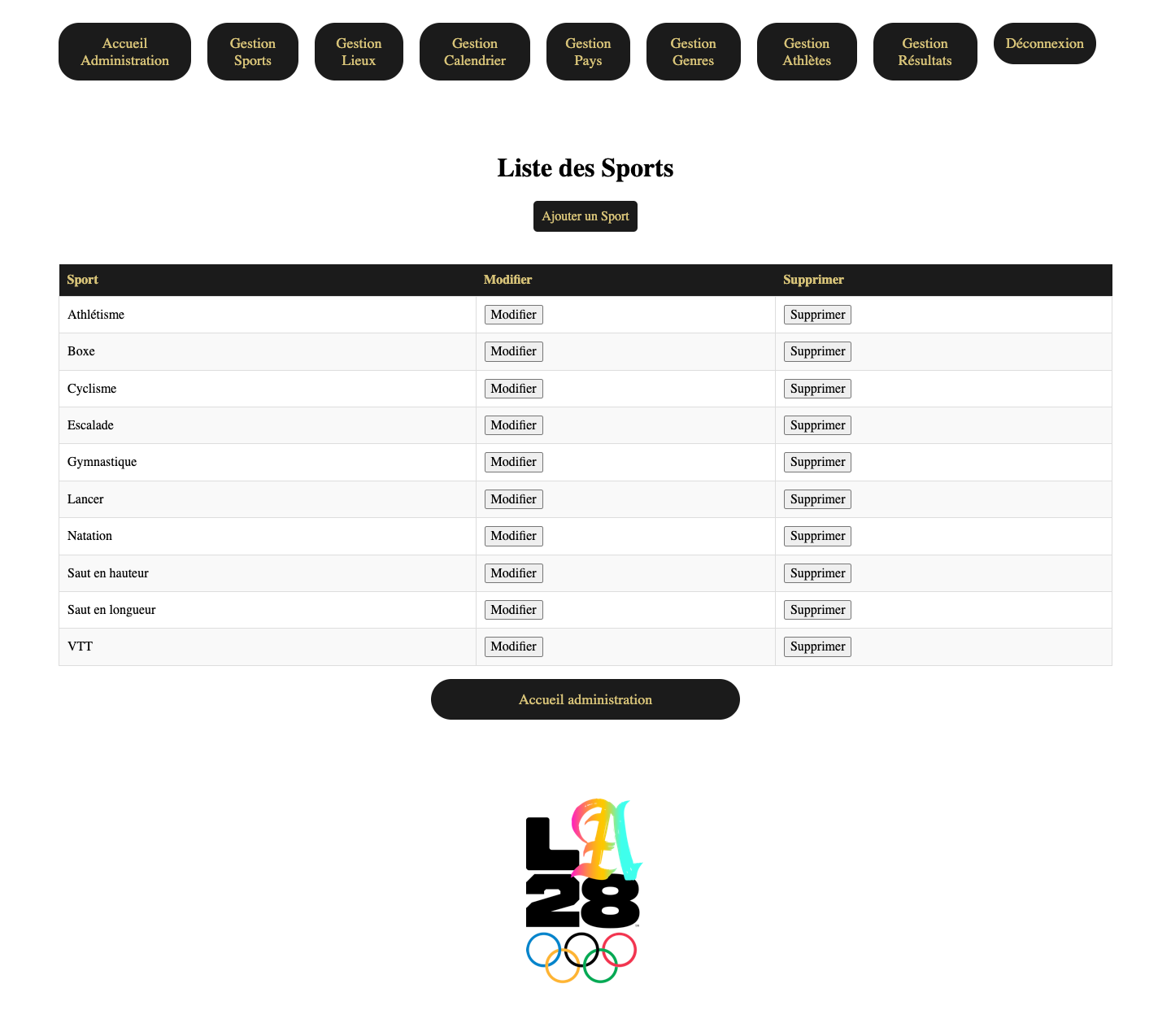
La page d’ajout des sport version tablette :



La page de la liste des sport version tablette :



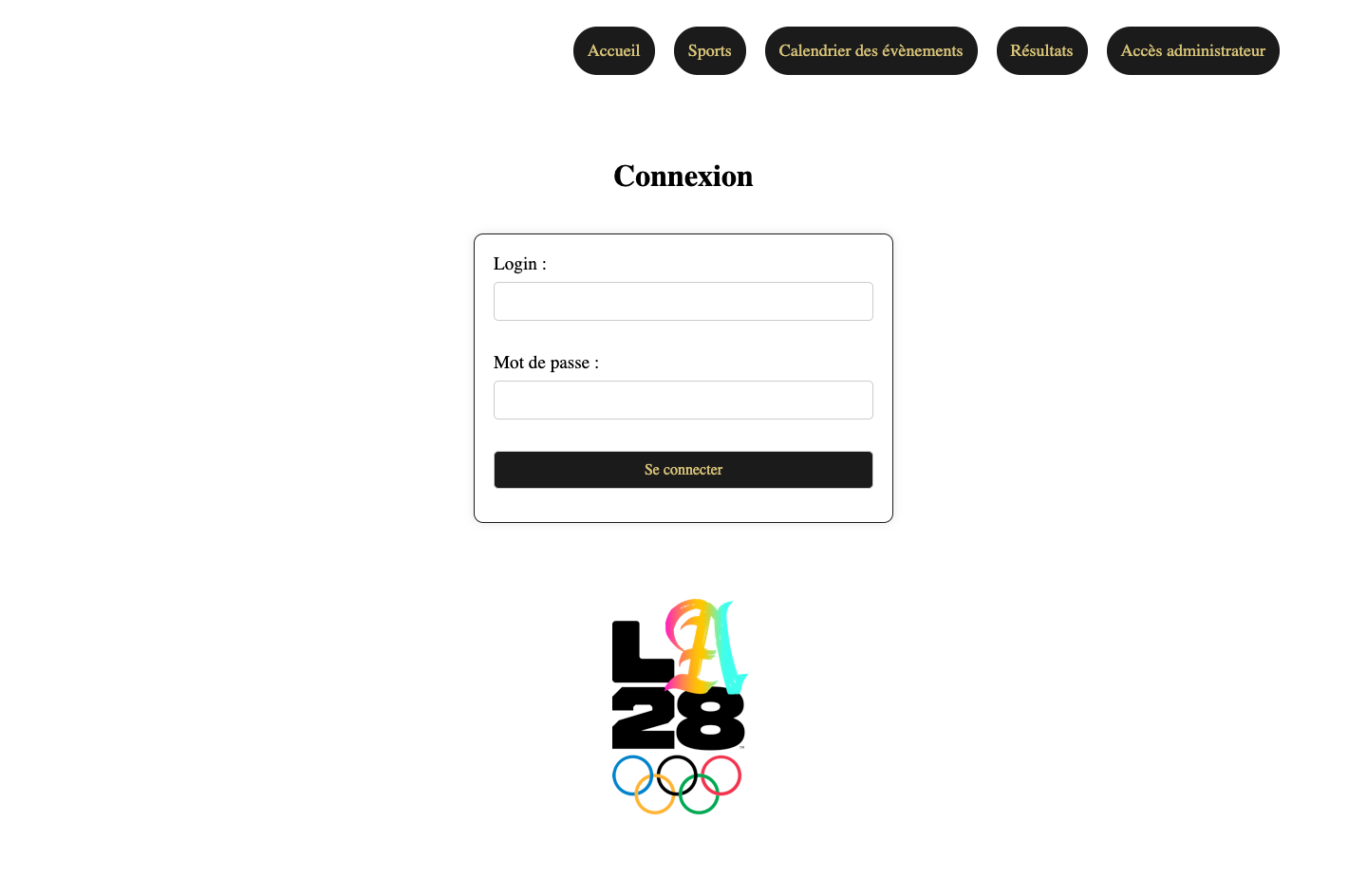
La page de la liste des sport version ordinateur :



La page d’ajout des sport version ordinateur :

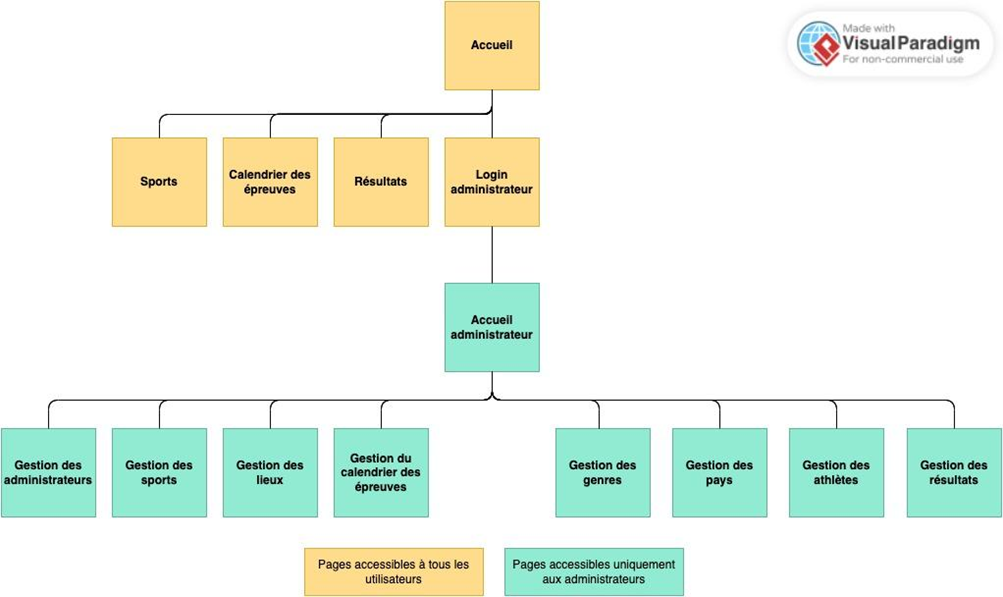
### 

La page de login version ordinateur :

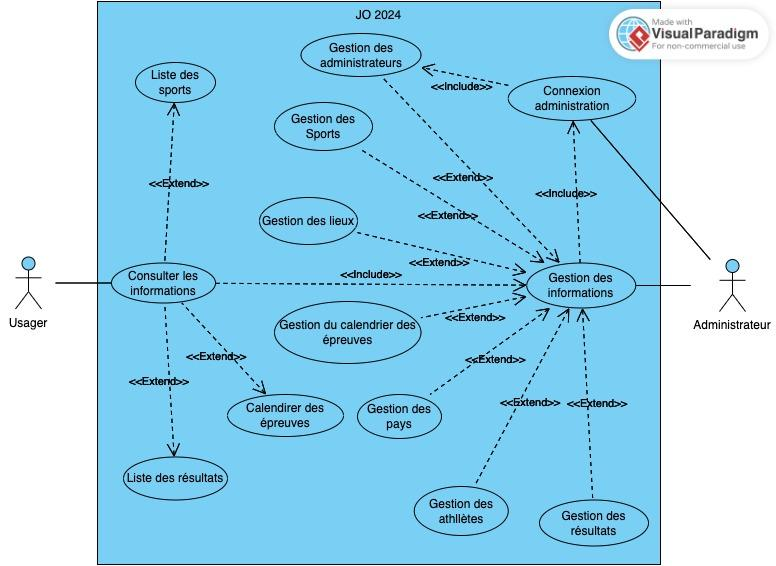


### 

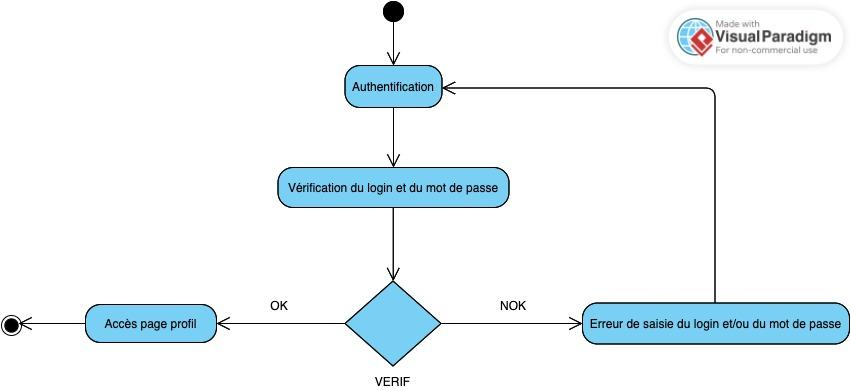
### 5.1.3. Arborescences



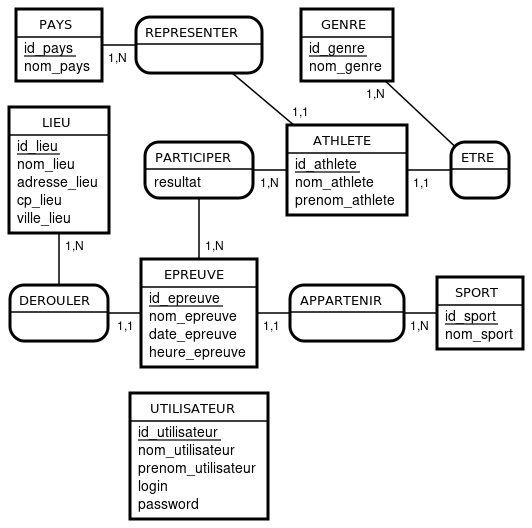
### 5.2.1. Diagramme de cas d’utilisation



### 5.2.2. Diagramme d’activités



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



### 

### 

### 

### 

### 

### 

### 

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

● ATHLETE (id\_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)

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

# 

# 

# 

# 

# 

# 

# 

# 

# 6. Technologies utilisées

## 6.1. Langages de développement Web

Pour le développement de l'application web, nous utiliserons principalement :

* HTML/CSS : pour la structure et le design du site.
* JavaScript : pour l'interactivité et le dynamisme des pages.
* Backend : PHP pour la logique serveur.

## 6.2. Base de données

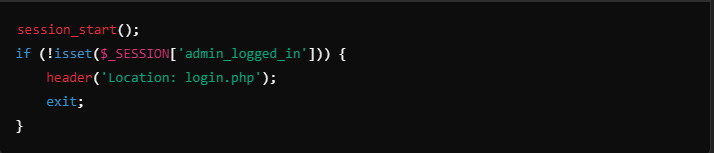
# Nous opterons pour une base de données relationnelle, telle que :

# MySQL: pour stocker les données de manière structurée, avec des relations entre les tables.

# 7. Sécurité

7.1. Login et protection des pages administrateurs

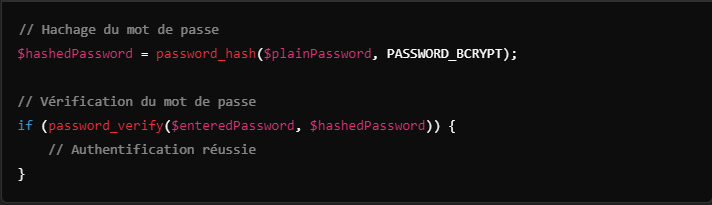
Pour protéger les pages administrateurs en PHP, on commence par mettre en place un système d'authentification basé sur des sessions. Voici comment on procède :

1. Formulaire de connexion : On crée un formulaire où l'utilisateur peut entrer son nom d'utilisateur et son mot de passe.
2. Vérification des identifiants : Lorsque le formulaire est soumis, on vérifie les identifiants dans la base de données.
3. Gestion des sessions :
   * On utilise session\_start() pour initialiser une session.
   * Si les identifiants sont valides, on stocke une valeur dans la session, par exemple $\_SESSION['admin\_logged\_in'] = true.
4. Protection des pages : Sur chaque page d'administration, on vérifie si l'utilisateur est connecté. Sinon, on le redirige vers la page de connexion.
5. 

7.2. Cryptage des mots de passe avec Bcrypt

On utilise Bcrypt pour hacher les mots de passe de manière sécurisée. Voici comment on fait :

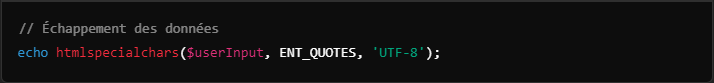
1. Création d'un mot de passe : Lorsqu'un utilisateur s'enregistre, on utilise password\_hash() pour hacher le mot de passe.
2. Vérification du mot de passe : Lors de la connexion, on utilise password\_verify() pour comparer le mot de passe saisi avec le hachage stocké.



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

Les attaques XSS se produisent lorsque des scripts malveillants sont injectés dans des pages web. Pour se protéger :

1. Échappement des données : On utilise htmlspecialchars() pour échapper les caractères spéciaux dans le HTML.
2. Validation des entrées : On valide et filtre toutes les données entrantes.
3. Utilisation des en-têtes de sécurité : On peut ajouter des en-têtes comme Content-Security-Policy pour restreindre les sources de contenu.



## 

## 

## 

## 

## 7.4. Protection contre les injections SQL

Les injections SQL se produisent lorsque des entrées non validées sont intégrées dans des requêtes SQL. Pour se protéger :

1. Utilisation de requêtes préparées : On utilise des instructions préparées avec PDO ou MySQLi pour éviter l'injection.
2. Validation des entrées : On valide et filtre toutes les données entrantes.
3. Limitation des privilèges de la base de données : On accorde uniquement les permissions nécessaires aux comptes de base de données.

