CAHIER DES CHARGES TECHNIQUES

Projet JO 2024

Dossier rédiger par Aïssa Penichon

Sommaire

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.

Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

Matériel nécessaire à la réalisation du projet :

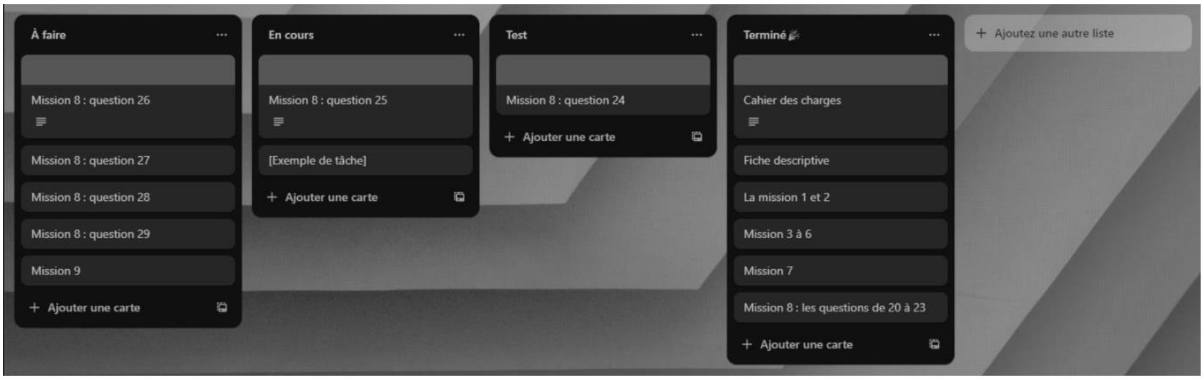
- Une Unité Central
- Clavier
- Écran et
- Souris

3.2. Ressources logicielles

Logiciel de développement utilisé à la réalisation, **MAMP → ServeurApach VisualStudio**, Platform dev:**GitHub** et Nous utiliserons également l'outil de gestion de projet en ligne **Trélllo**.
(Apach permet de executer le script php au lieu de VisualCode).

4. Gestion du projet

Nous travaillons également sur GitHub, plateforme de développement collaboratif.



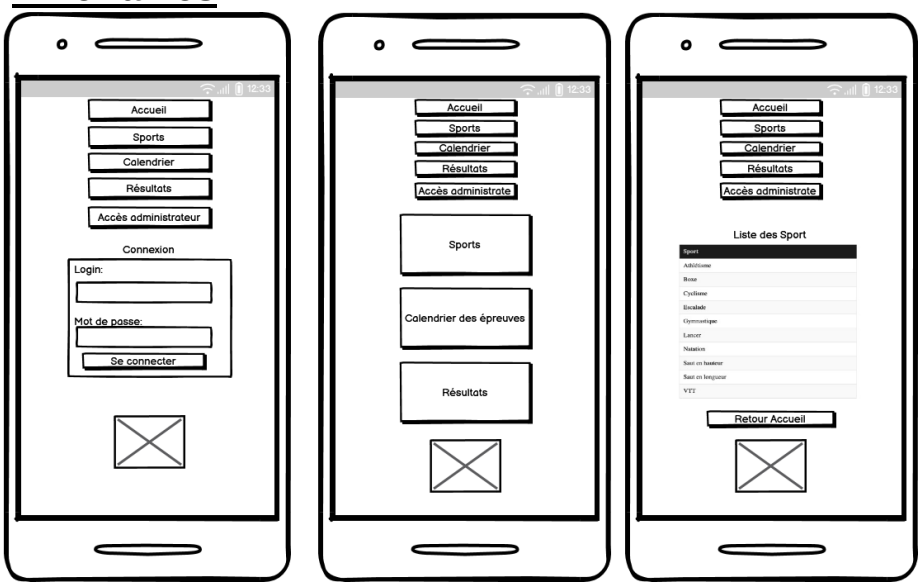
5. Conception du projet

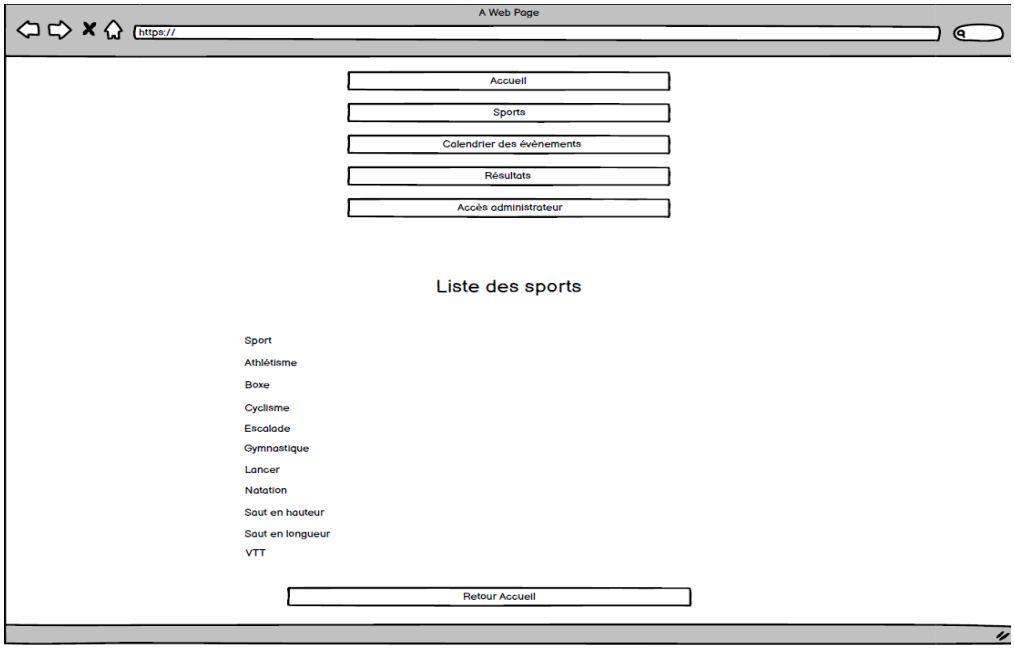
5.1. Le front-end

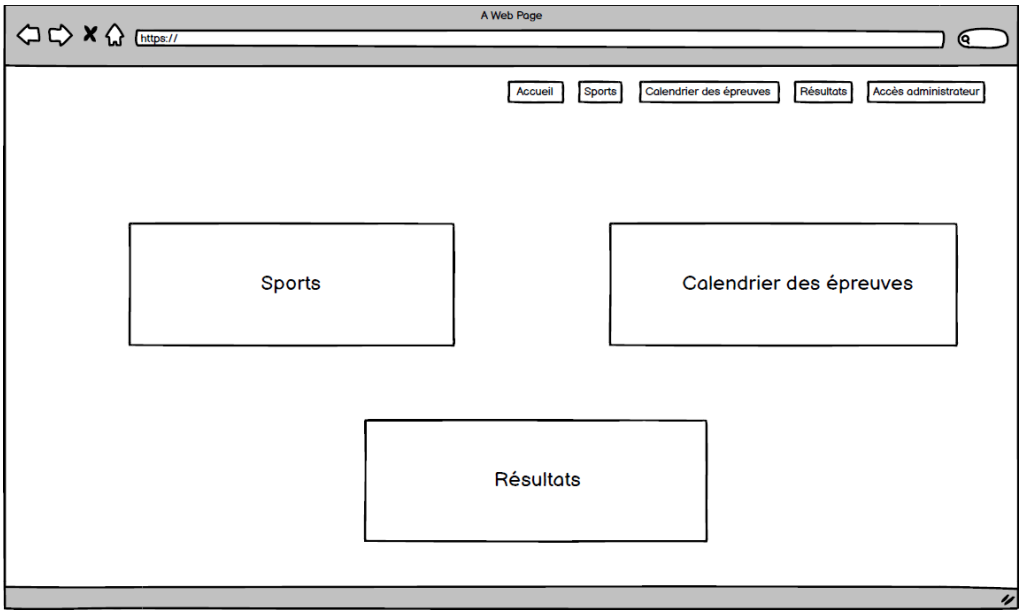
Les sites de renseignement pour aider à la mission :

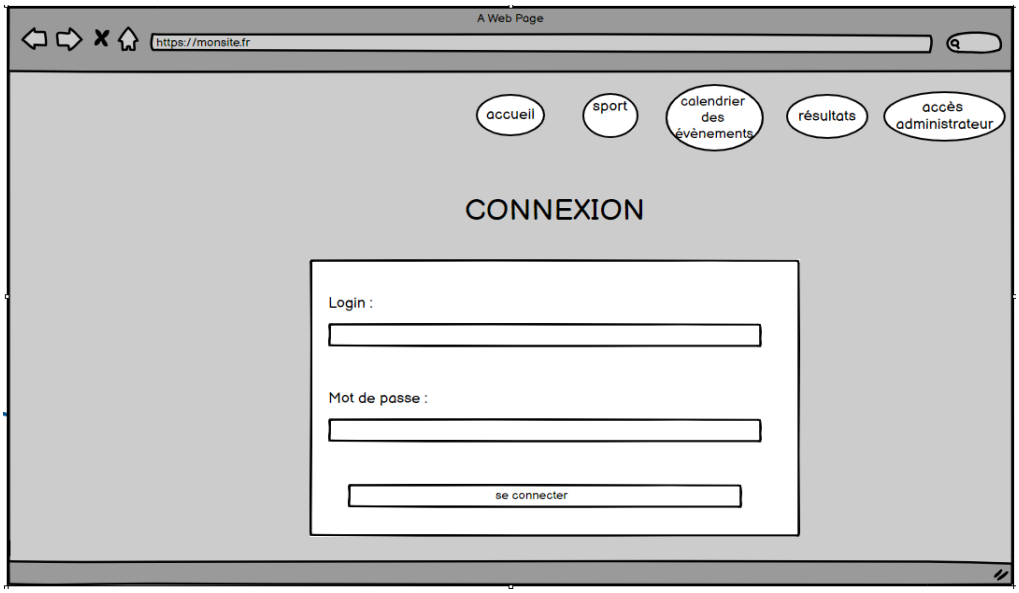
- **php.net**
- **OpenClassroom**
- **W3Schools**
- **Stackoverflow(us) où Developper.net**

5.1.1. Wireframes

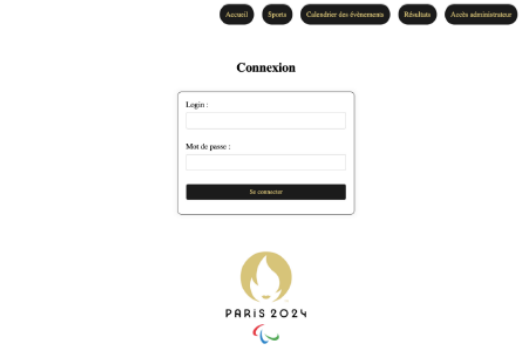



A Web Page
https://
Accueil
Sports
Calendrier des évènements
Résultats
Accès administrateur
Liste des sports
Sport
Athlétisme
Boxe
Cyclisme
Escalade
Gymnastique
Lancer
Natation
Saut en hauteur
Saut en longueur
VTT
Retour Accueil

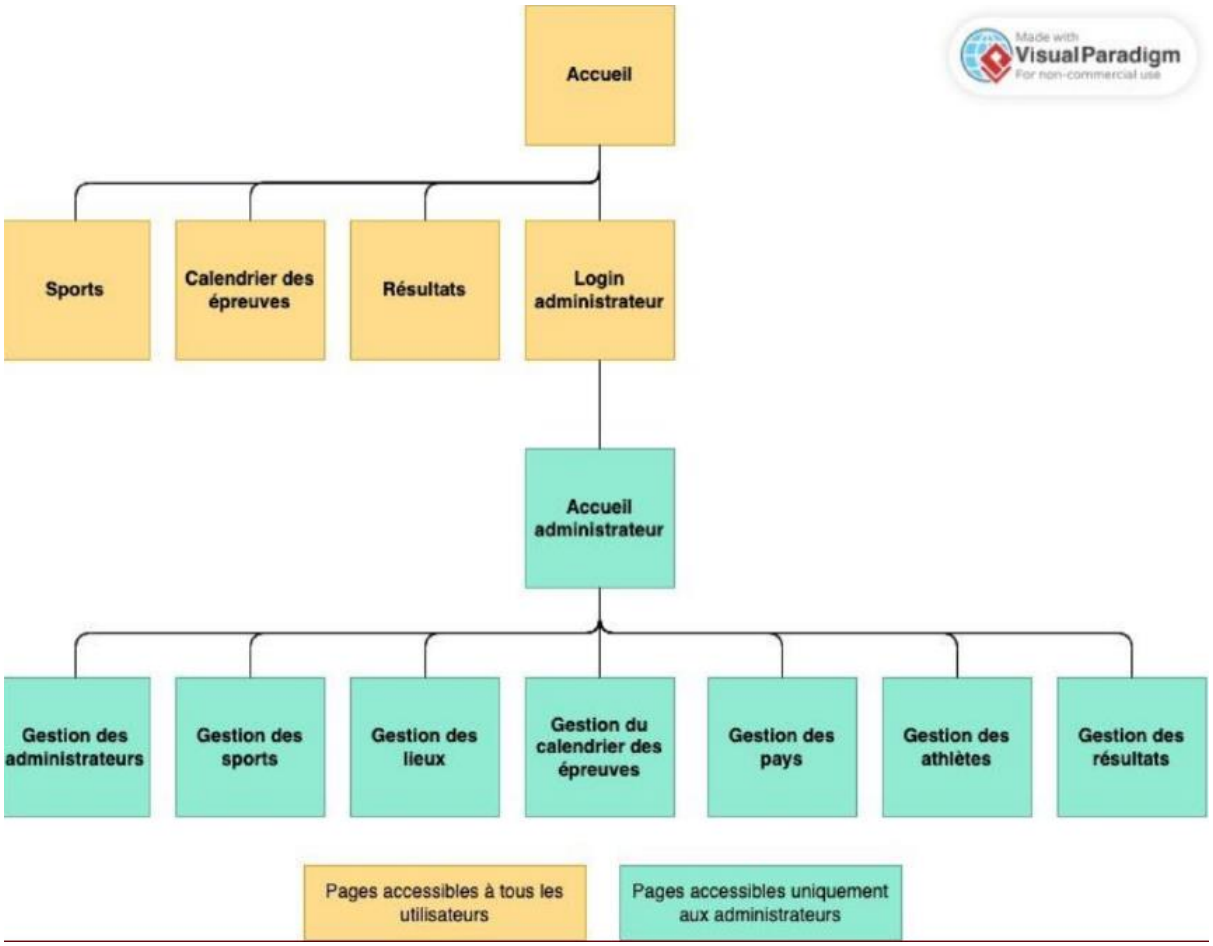

A Web Page
https://
Accueil
Sports
Calendrier des épreuves
Résultats
Accès administrateur
Sports
Calendrier des épreuves
Résultats


A Web Page
https://monsite.fr
accueil
sport
calendrier des évènements
résultats
accès administrateur
CONNEXION
Login :
Mot de passe :
se connecter

5.1.2. Maquettes

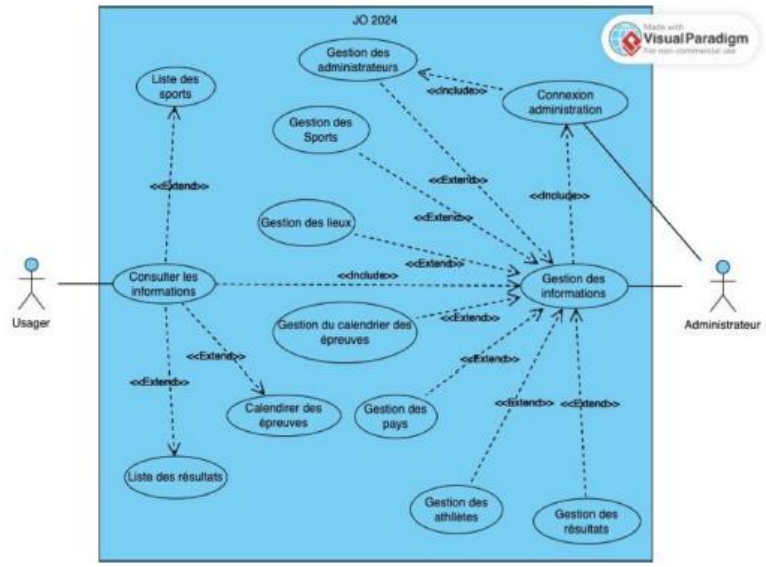


5.1.3. Arborescences

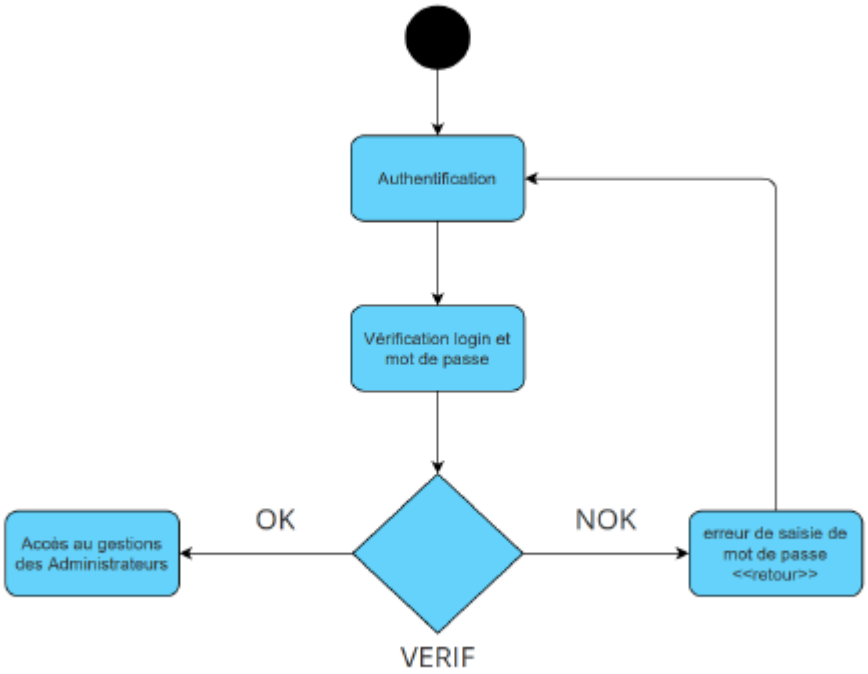


5.2. Le back-end

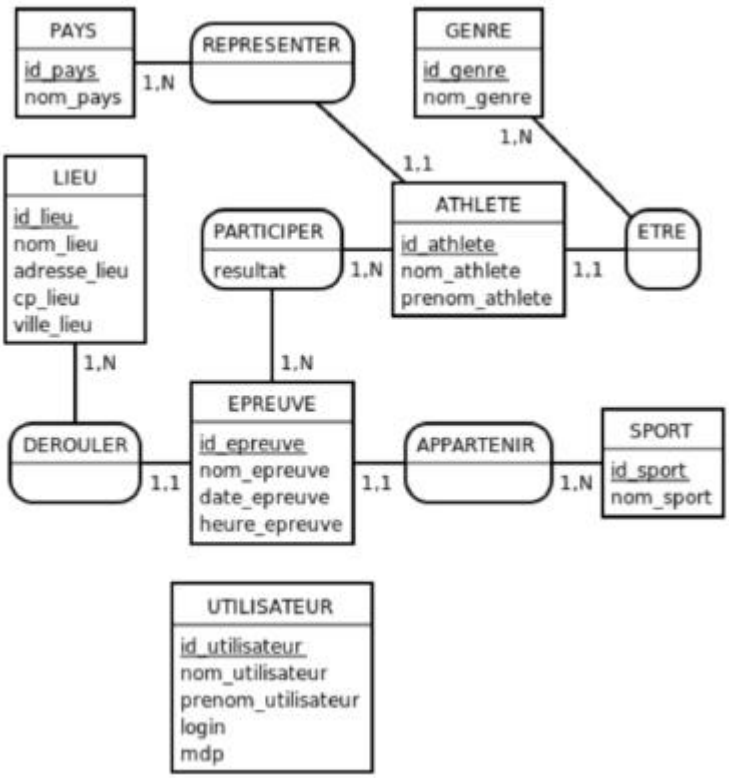
5.2.1. Diagramme de cas d'utilisation



5.2.2. Diagramme d'activités



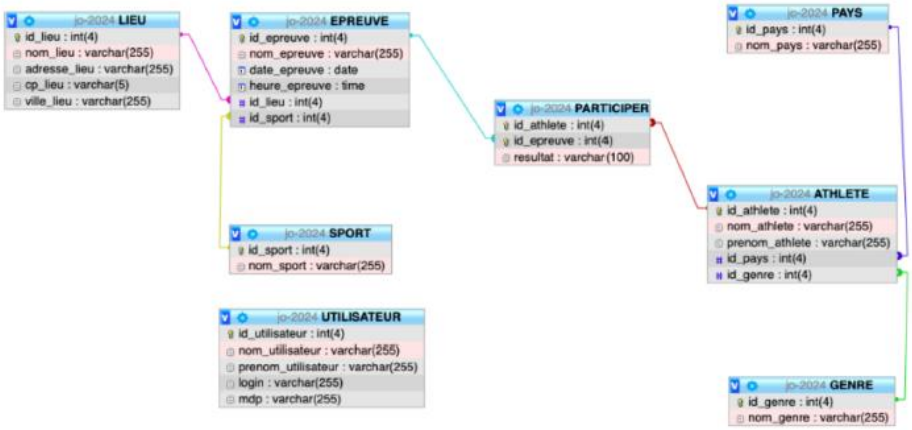
5.2.3. Modèles Conceptuel de Données (MCD)



5.2.4. Modèle Logique de Données (MLD)

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, mdp)

5.2.5. Modèle Physique de Données (MPD)



6. Technologies utilisées

6.1. Langages de développement Web

Les langages utilisés à la réalisation du projet sont **Html5.0,Css3.0**,**PHP7** où 8,**JS** et **Sql** pour les base de données. Bdd et une base de données relationnelle et non relationnelle Oracle).

6.2. Base de données

PHP8 et **Sql** pour les base de données la Bdd et une base de données relationnelle et non relationnelle Oracle.

7. Sécurité

7.1. Login

Le processus de connexion permet à un utilisateur de s'authentifier sur un site web en vérifiant son Identité, généralement par un nom d'utilisateur et un mot de passe.

- Utiliser HTTPS pour sécuriser les données transmises.
- Valider les entrées côté serveur.

● Limiter les tentatives de connexion infructueuses pour prévenir les attaques par force brute

7.2. Cryptage des mots de passe

Le cryptage des mots de passe consiste à transformer les mots de passe en une forme sécurisée qui ne peut pas être facilement déchiffrée, pour protéger les informations de l'utilisateur en cas de fuite de données.

● Utiliser des algorithmes de hachage robustes comme MD5 ou SHA_256.
● Implémenter des sels (valeurs aléatoires uniques) pour chaque mot de passe haché pour prévenir les attaques par dictionnaire et les attaques par tables de hachage.

7.3. Protection des pages administrateurs

Cela implique de s'assurer que seuls les utilisateurs autorisés peuvent accéder aux pages d'administration.

● Authentification forte (par exemple, 2FA).
● Vérification des rôles et des permissions avant d'accorder l'accès.
● Journalisation des accès pour le suivi et l'analyse des activités suspectes.

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

Les attaques XSS exploitent les vulnérabilités d'un site web en injectant des scripts malveillants dans les pages affichées aux autres utilisateurs.

● Échapper ou valider toutes les entrées utilisateur côté serveur et client.
● Utiliser des politiques de Content Security Policy (CSP).
● Ne jamais faire confiance aux données fournies par l'utilisateur.

7.5. Protection contre les injections SQL

Les injections SQL se produisent lorsqu'un attaquant insère ou manipule des commandes SQL dans une entrée de l'application pour accéder ou manipuler la base de données.

● Utiliser des requêtes préparées et des ORM pour éviter les injections SQL.
● Ne jamais construire des requêtes SQL directement à partir des entrées de l'utilisateur.
● Valider et échapper toutes les entrées utilisateurs.