Cahier des charges techniques

Sommaire

Table des matières

1. [1. Contexte du projet 2](#_Toc216181851)

[1.1. Présentation du projet 2](#_Toc216181852)

[1.2. Date de rendu du projet 2](#_Toc216181853)

1. [2. Besoins fonctionnels 2](#_Toc216181854)
2. [3. Ressources nécessaires à la réalisation du projet 2](#_Toc216181855)

[3.1. Ressources matérielles 2](#_Toc216181856)

[3.2. Ressources logicielles 2](#_Toc216181857)

1. [4. Gestion du projet 3](#_Toc216181858)
2. [5. Conception du projet 4](#_Toc216181859)

[5.1. Le front-end 4](#_Toc216181860)

[5.1.1. Wireframes 4](#_Toc216181861)

[5.1.2. Maquettes 4](#_Toc216181862)

[5.1.3. Arborescences 5](#_Toc216181863)

[5.2. Le back-end 5](#_Toc216181864)

[5.2.1. Diagramme de cas d’utilisation 5](#_Toc216181865)

[5.2.2. Diagramme d’activités 5](#_Toc216181866)

[5.2.3. Modèles Conceptuel de Données (MCD) 5](#_Toc216181867)

[5.2.4. Modèle Logique de Données (MLD) 5](#_Toc216181868)

[5.2.5. Modèle Physique de Données (MPD) 5](#_Toc216181869)

1. [6. Technologies utilisées 5](#_Toc216181870)

[6.1. Langages de développement Web 5](#_Toc216181871)

[6.2. Base de données 5](#_Toc216181872)

1. [7. Sécurité 6](#_Toc216181873)

[7.1. Login et protection des pages administrateurs 6](#_Toc216181874)

[7.2. Hachages des mots de passe avec Bcrypt 6](#_Toc216181875)

[7.3. Protection contre les attaques XSS (Cross-Site Scripting) 7](#_Toc216181876)

[7.4. Protection contre les injections SQL 7](#_Toc216181877)

# 

# 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 30 novembre 2025.

# 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

Dans les ressources matérielles, nous avons besoin :

* Un pc (fixe ou portable) disposant de minimum 8Go de RAM.
* Logiciel
* Une bonne connexion réseau

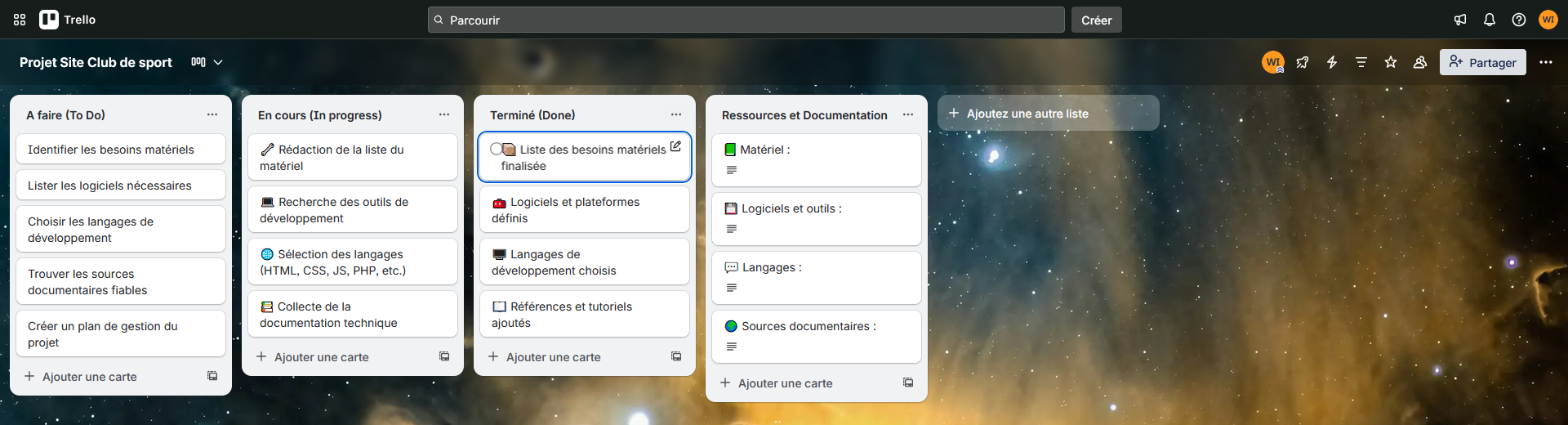
## 3.2. Ressources logicielles

Dans les ressources logicielles, nous avons besoin :

* Html
* Css
* Mocodo
* Mamp
* MySQL
* PHP
* Visual Studio Code
* Visual Paradigm

# 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.

# 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)

# Technologies utilisées

## 6.1. Langages de développement Web

## 6.2. Base de données

# Sécurité

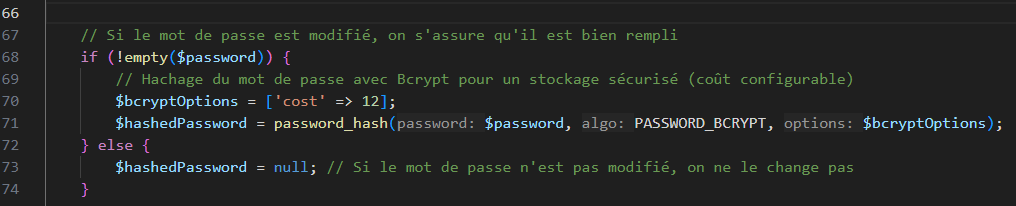
## 7.1. Login et protection des pages administrateurs

Un système d’authentification et de protection des pages administratives consiste à vérifier l’identité de l’utilisateur éventuellement par un formulaire de connexion, à établir une session lorsque celui-ci est identifié, puis à autoriser l’accès aux pages réservées selon si son rôle le lui permet (ex. admin).



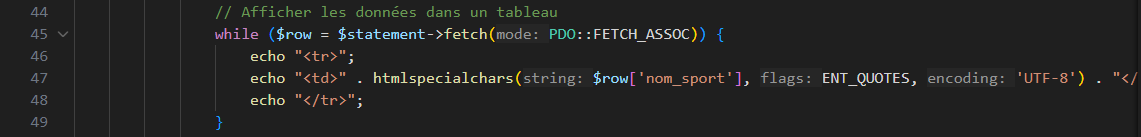
## 7.2. Hachages des mots de passe avec Bcrypt

Le processus de hachage du mot de passe avec Bcrypt vise à transformer un mot de passe en empreinte cryptographique, sécurisée, salée, et difficilement réversible pour le stockage fiable et la protection des utilisateurs en cas de fuite de base de données.



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

## Ces mécanismes permettent d’empêcher les attaques XSS en transformant une série de caractères potentiellement dangereux, à savoir <, >, les guillemets, tout début de script, etc., en simple texte, ce qui a pour conséquence de les afficher tels quels dans le navigateur, au lieu d’être interprétés comme le code JavaScript éventuellement exécutable que pourrait exploiter un attaquant.



## 7.4. Protection contre les injections SQL

L’injection SQL se caractérise par des attaques dans lesquelles un utilisateur malintentionné va insérer des commandes SQL compromettantes dans une requête, ce qui peut d’ores et déjà compromettre la base de données (lire, modifier ou même supprimer des enregistrements non autorisés).

