
Cahier des Charges Techniques



PETHIYAGODA HIRUSHA
BTS SIO 2

SOMMAIRE

Contexte du projet	3
1.1 Présentation du projet.....	3
1.2 Date de rendu du projet	3
2. Besoins fonctionnels.....	3
3. Ressources nécessaires à la réalisation du projet	4
3.1 Ressources matérielle	4
3.2 Ressources logicielles	4
4. Gestion du projet	5
5. Conception du projet.....	6
5.1. Le Front-End.....	6
5.1.1. Wireframes	6
5.1.2. Maquettes	9
5.1.3. Arborescence	10
5.2 Le Back-End.....	10
5.2.1. Diagramme de cas d'utilisation	10
5.2.2. Diagramme d'activité	11
5.2.3. Dictionnaire des données	11
5.2.4. Modèle Conceptuel des Données (MCD).....	12
5.2.5. Modèle Logique des Données (MLD)	13
5.2.6. Modèle Physique des Données (MPD)	14
6. Technologies utilisées.....	14
6.1. Langages de développement Web	14
6.2. Base de données	14
7. Sécurité.....	15
7.1. Login et protection des pages administrateurs.....	15
7.2. Cryptage des mots de passe avec Bcrypt.....	15
7.3. Protection contre les attaques XXS	15
7.4. Protection contre les injections SQL.....	15

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 08/11/2024.

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 nécessaires à la réalisation du projet

3.1 Ressources matérielle

Les ressources matérielles dont nous avons besoins sont :

- PC Fixe
- Connexion Internet (par câble ou wifi)
- Ecran
- Clavier
- Souris

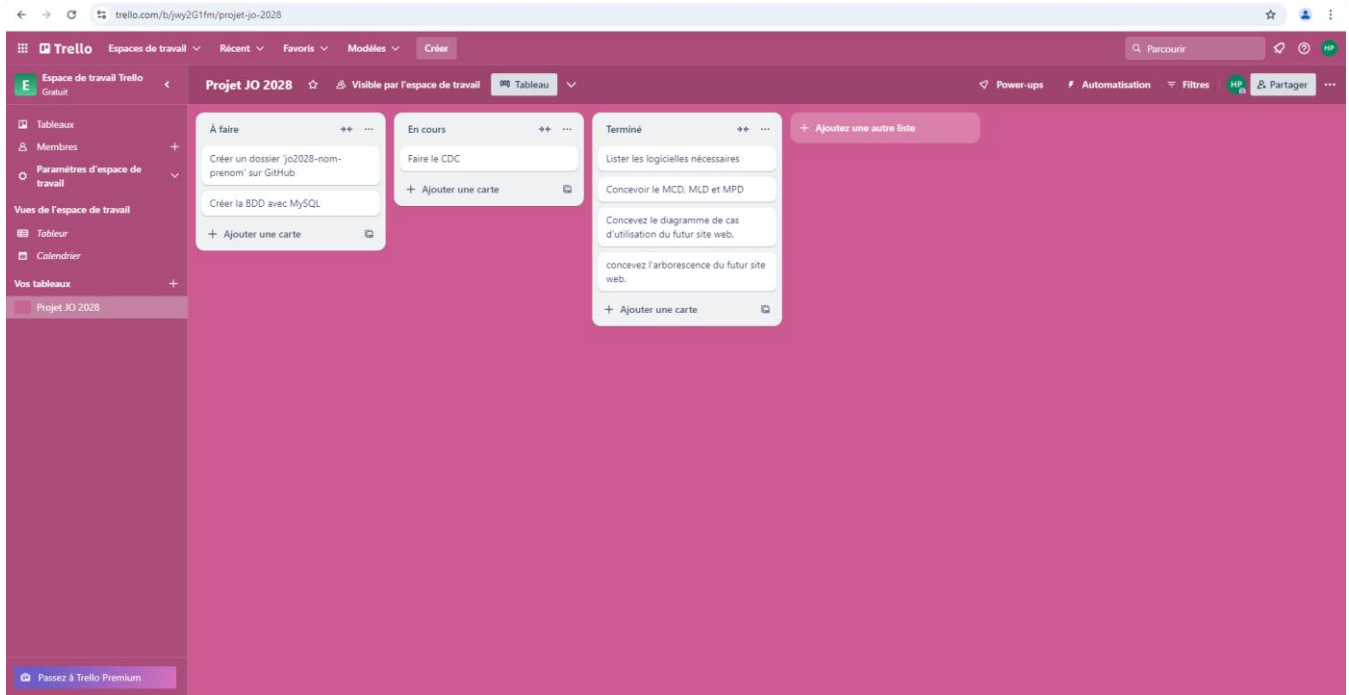
3.2 Ressources logicielles

Les ressources logicielles dont nous avons besoins sont :

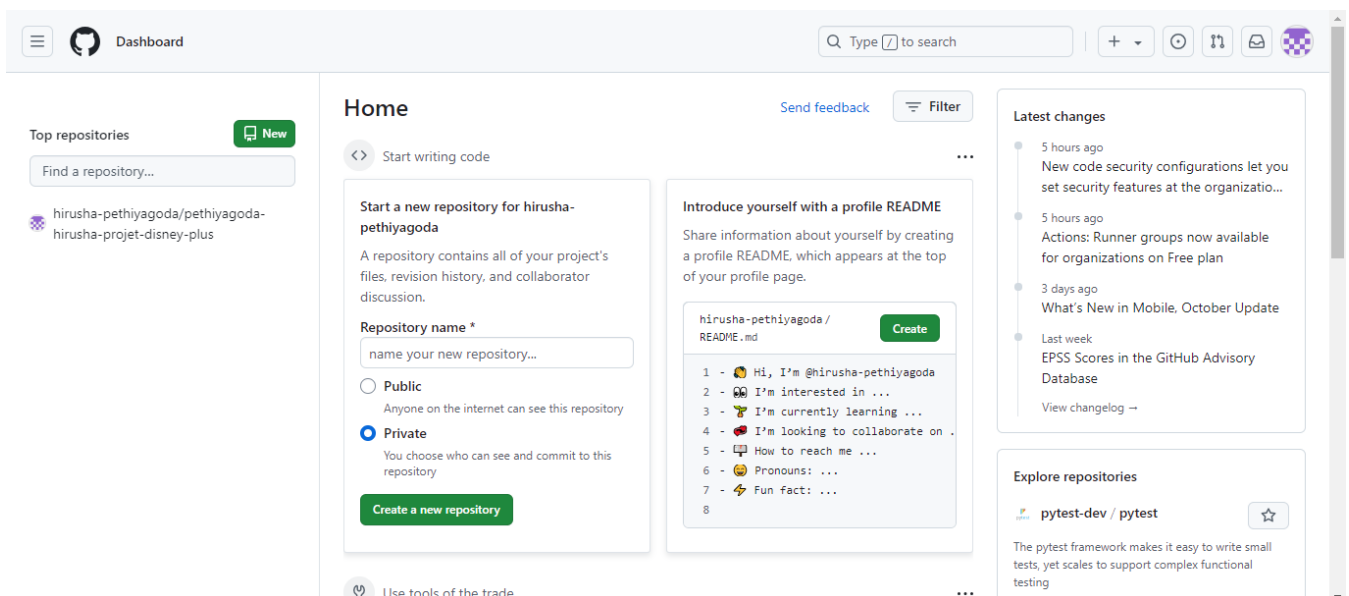
- IDE (Environnement de Développement) : Visual Studio Code
- Plateforme de développement collaboratif : GitHub
- Conception de Base des Données : MOCODO
- Outil de gestion des projets : Trello
- Conception UML et arborescence : Visual Paradigm Online
- Maquettage : Figma
- MAMP
 - Serveur Web (Contenu dans MAMP) : Apache
 - SGBDR (Contenu dans MAMP) : MySQL

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.

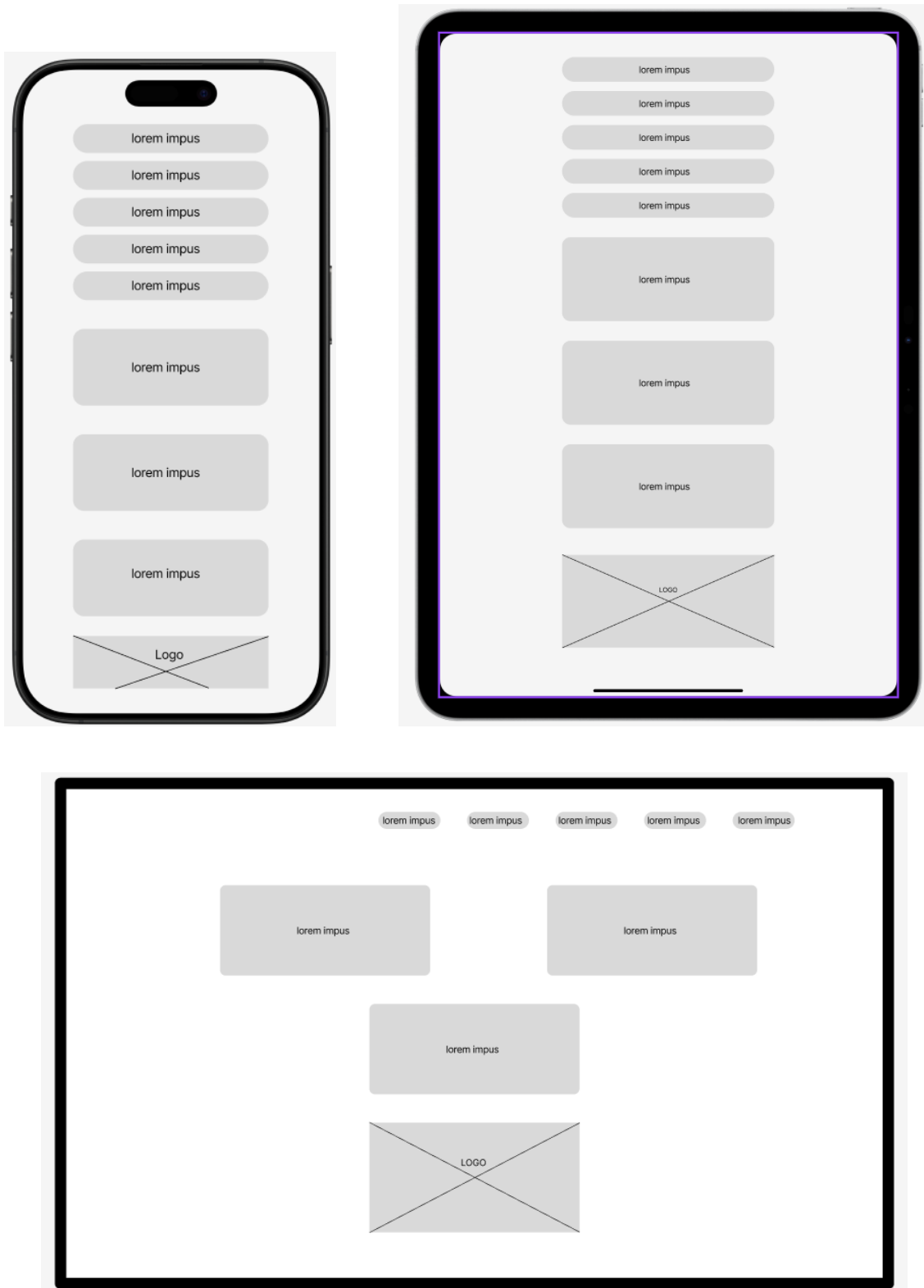


5. Conception du projet

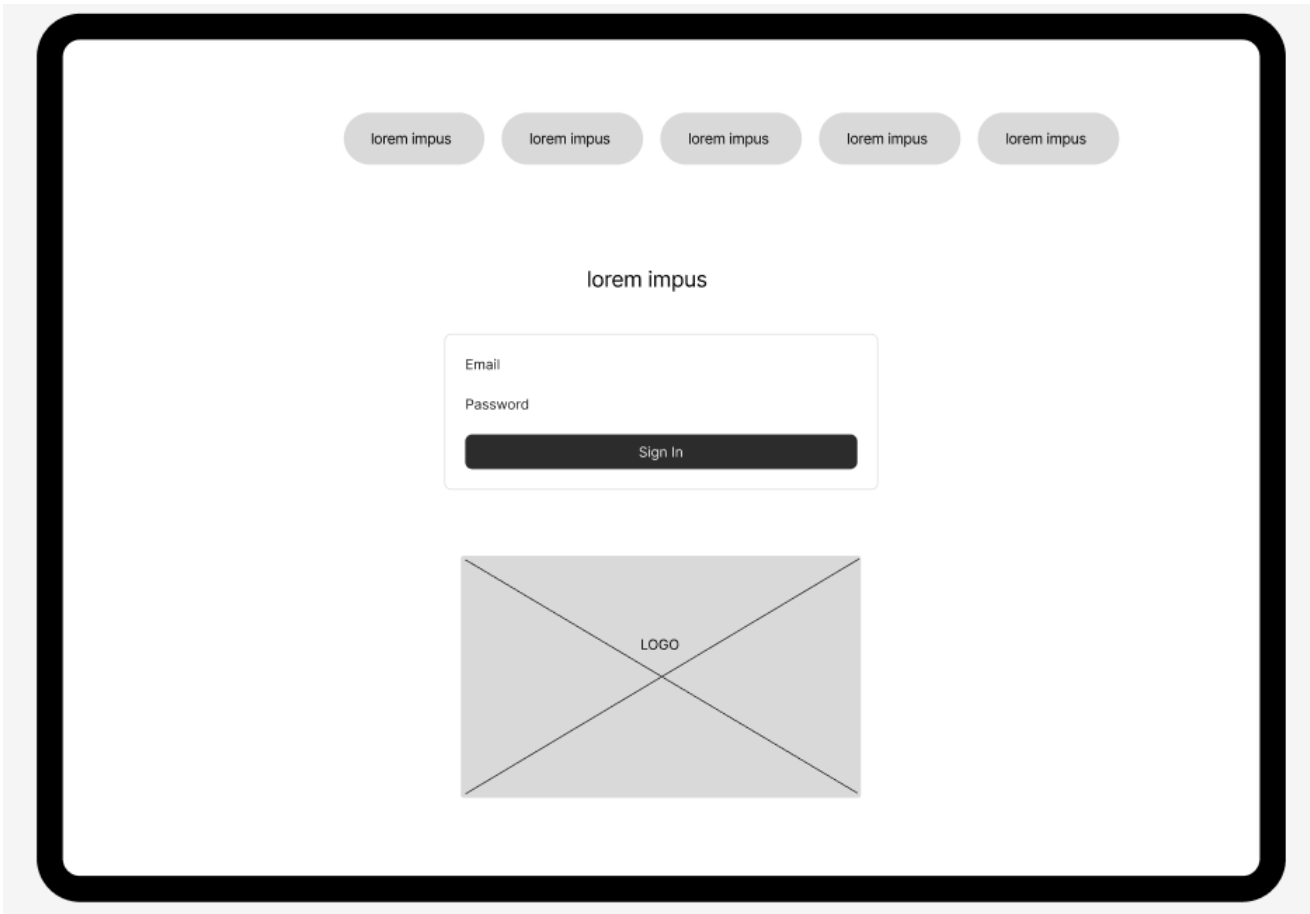
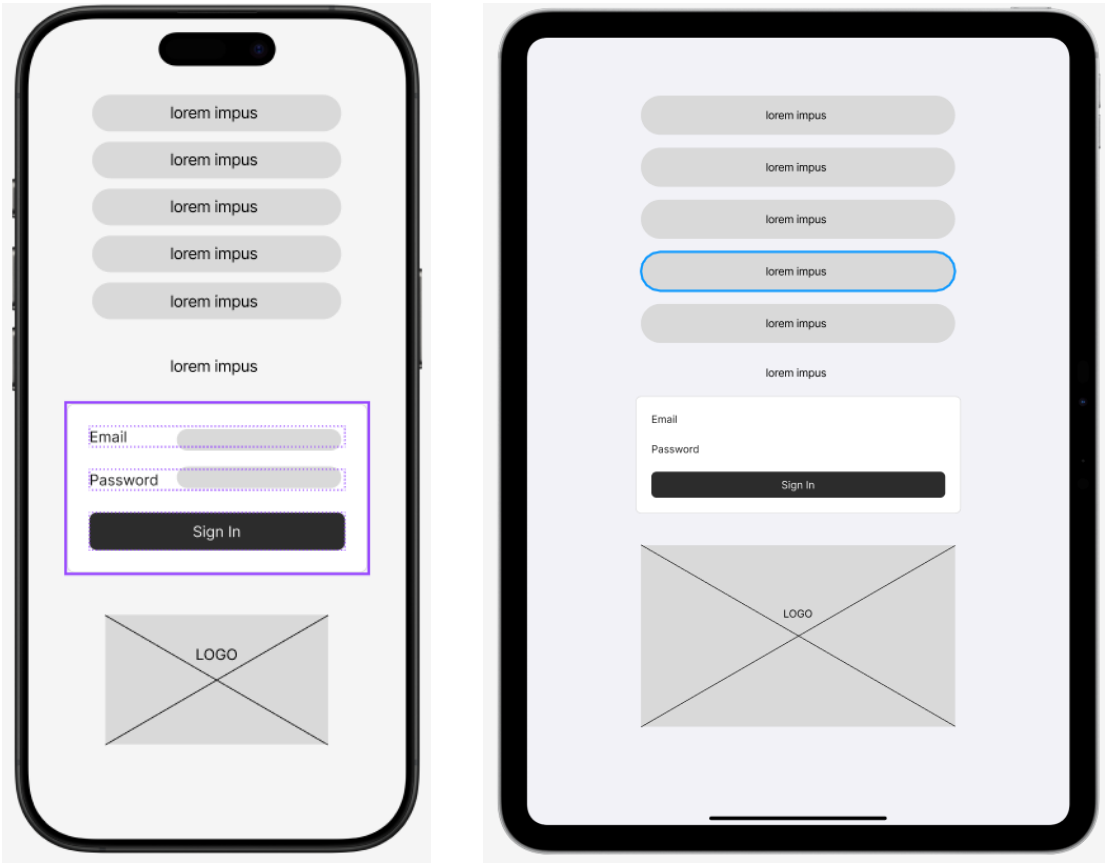
5.1. Le Front-End

5.1.1. Wireframes

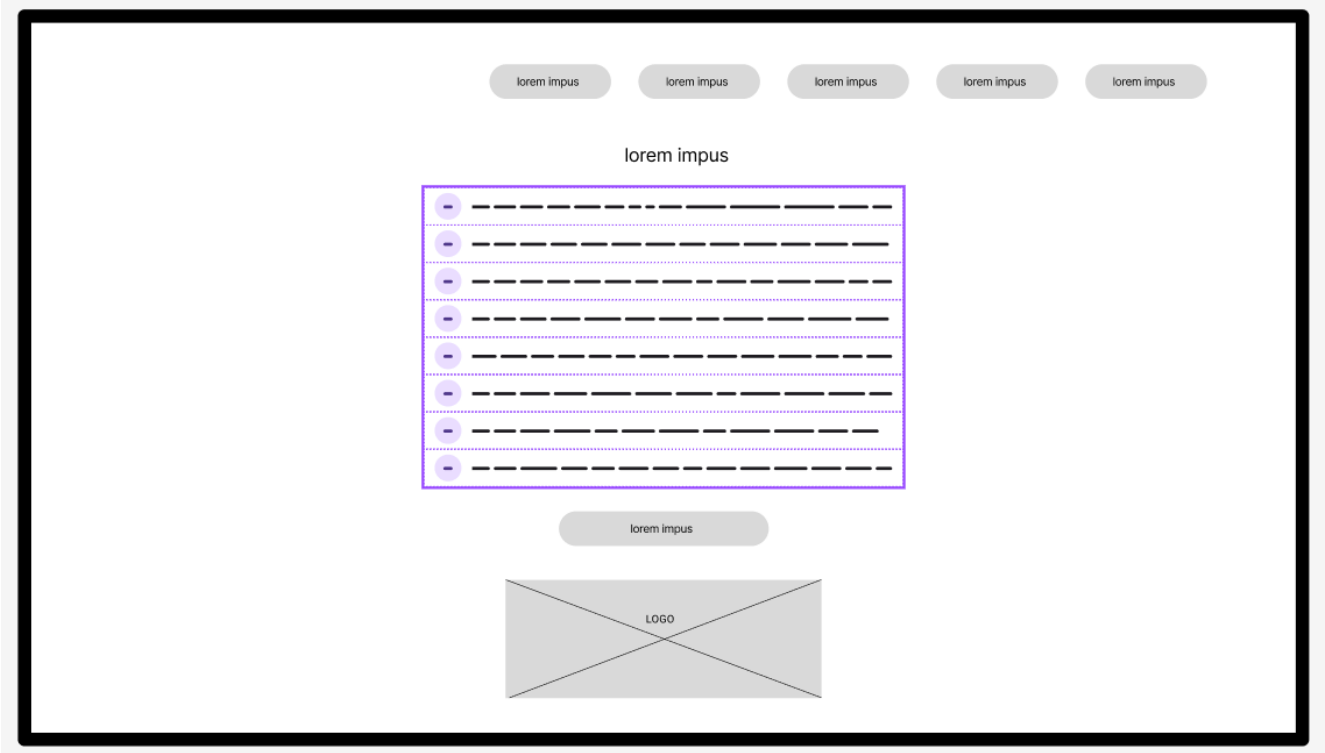
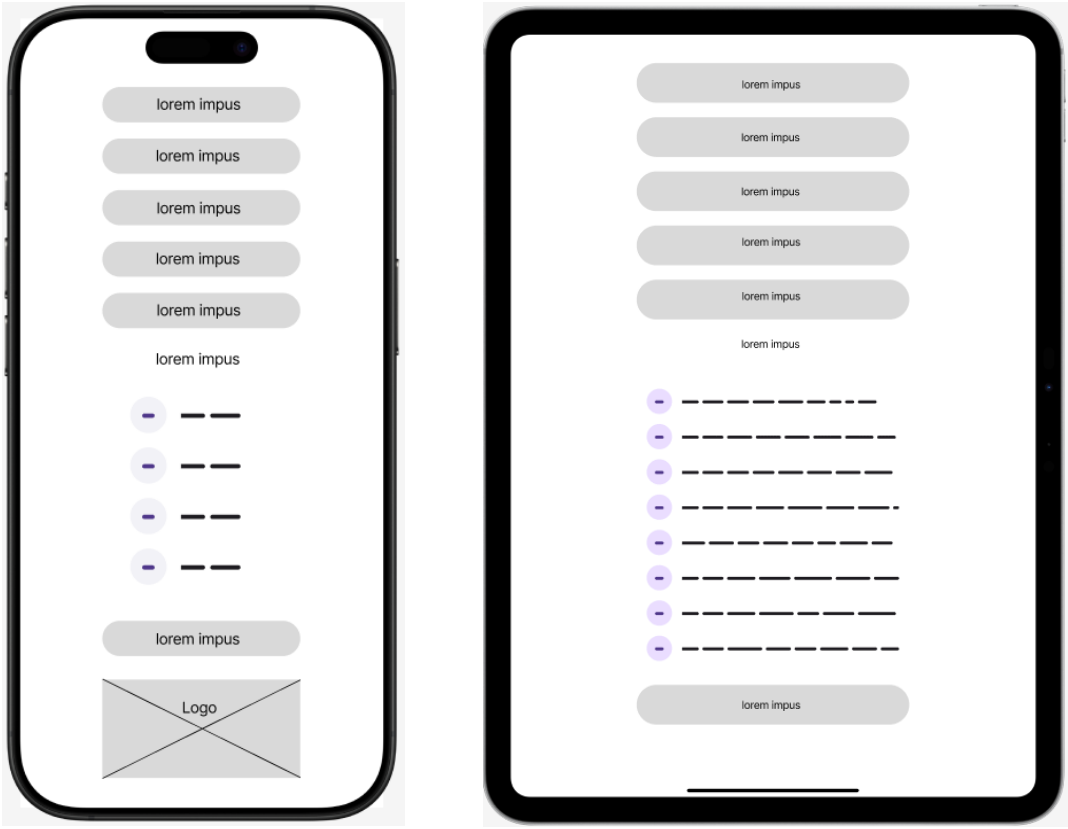
Accueil (index.html)



Connexion (login.php)



Liste Sport (manage-sports.php)



5.1.2. Maquettes

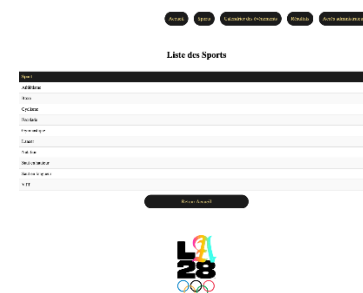


Connexion

Login :

Mot de passe :

Se connecter



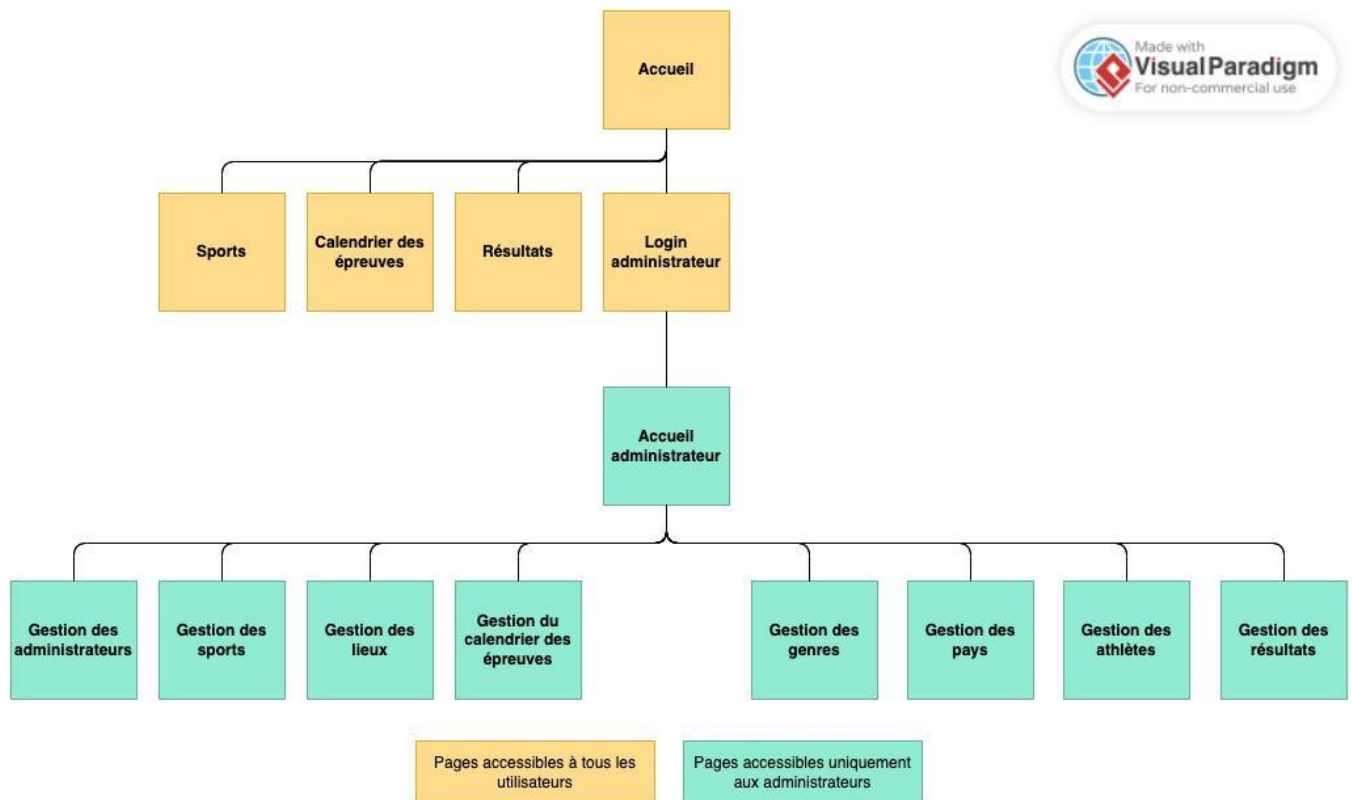
Liste des Sports

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

Retour Accueil

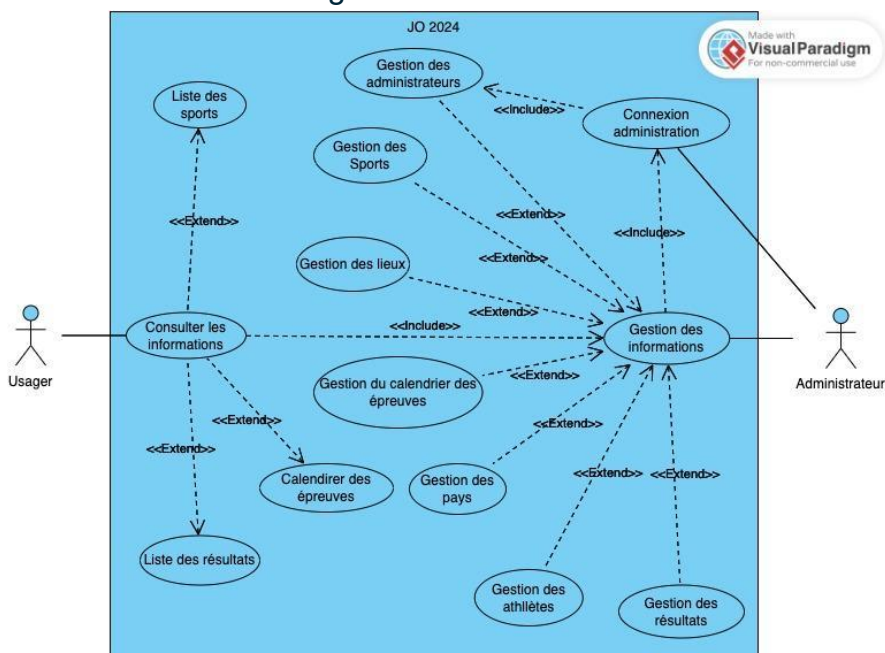


5.1.3. Arborescence

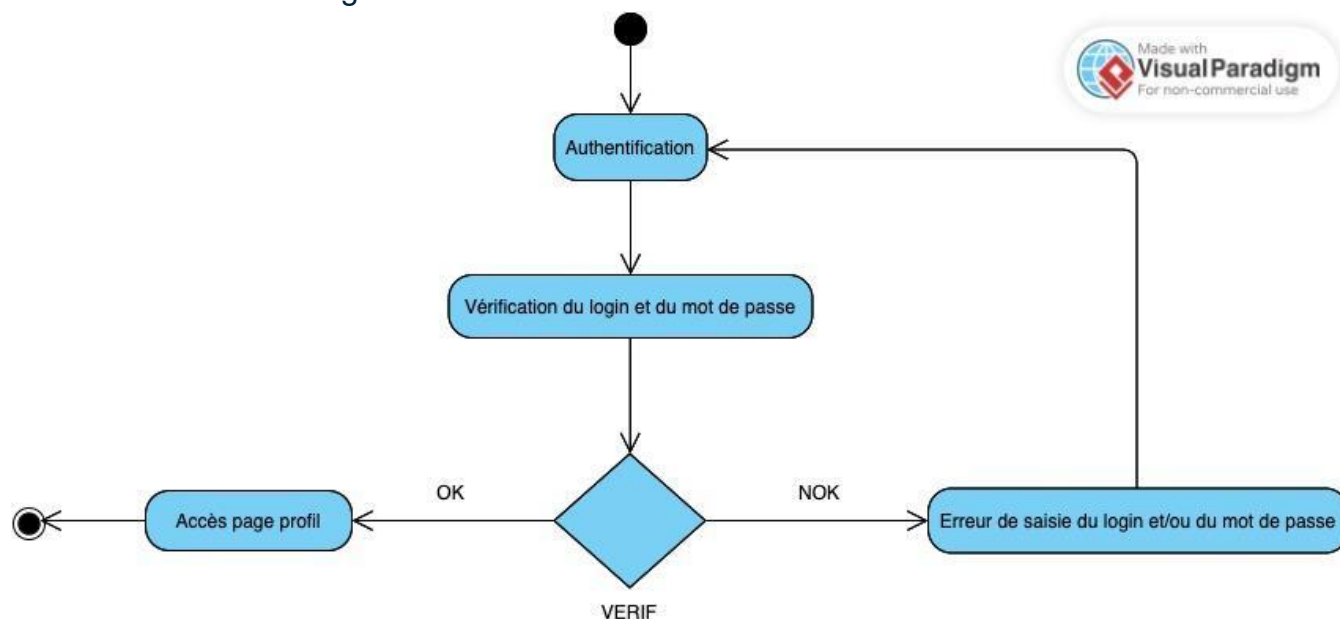


5.2 Le Back-End

5.2.1. Diagramme de cas d'utilisation



5.2.2. Diagramme d'activité

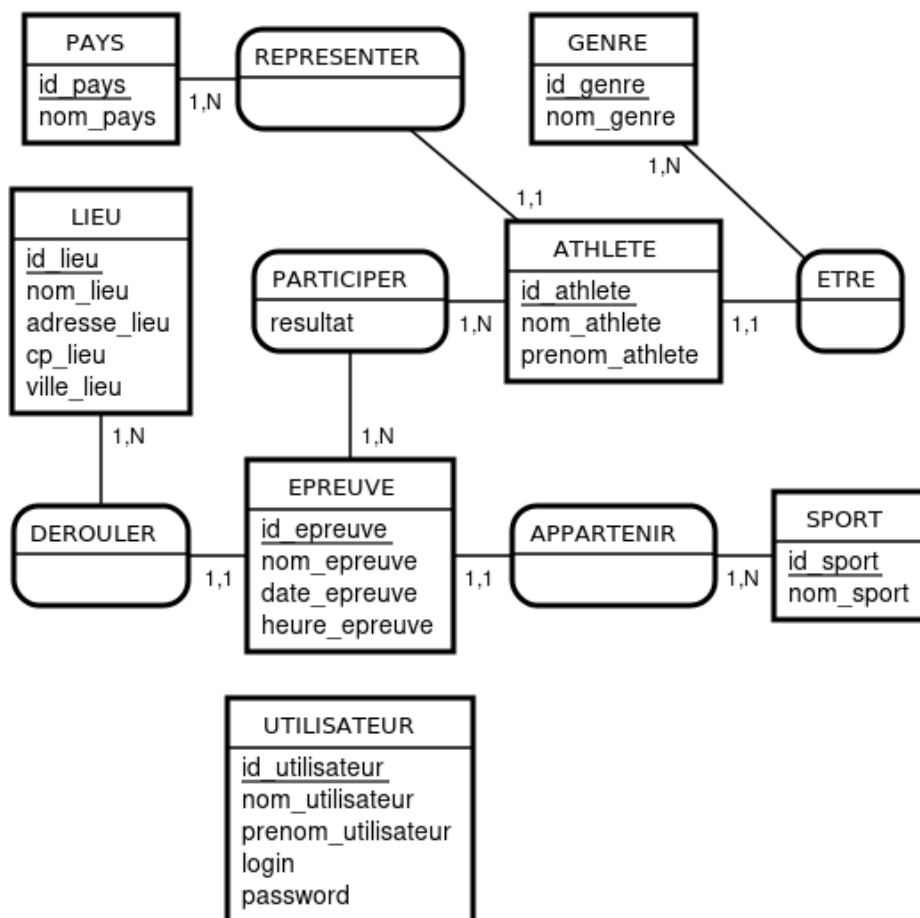


5.2.3. Dictionnaire des données

Table	Nom de l'attribut	Type	Description
Pays	id_pays	INTEGER(4)	Identifiant unique du pays
Pays	nom_pays	VARCHAR(255)	Nom du pays
Genre	id_genre	INTEGER(4)	Identifiant unique du genre
Genre	nom_genre	VARCHAR(255)	Nom du genre
Athelete	id_athlete	INTEGER(4)	Identifiant unique de l'athlète
Athelete	nom_athlete	VARCHAR(255)	Nom de l'athlète
Athelete	prenom_athlete	VARCHAR(255)	Prénom de l'athlète
Athelete	id_pays	INTEGER(4)	Identifiant unique du pays (Clé étrangère)
Athelete	id_genre	INTEGER(4)	Identifiant unique du genre (Clé étrangère)
Lieu	id_lieu	INTEGER(4)	Identifiant unique du lieu
Lieu	nom_lieu	VARCHAR(255)	Nom du lieu
Lieu	adresse_lieu	VARCHAR(255)	Adresse du lieu
Lieu	cp_lieu	VARCHAR(5)	Code Postale du lieu
Lieu	ville_lieu	VARCHAR(255)	Ville du lieu
Epreuve	id_epreuve	INTEGER(4)	Identifiant unique de l'épreuve
Epreuve	nom_epreuve	VARCHAR(255)	Nom de l'épreuve
Epreuve	date_epreuve	DATE	Date de l'épreuve
Epreuve	heure_epreuve	TIME	Heure de l'épreuve
Epreuve	id_lieu	INTEGER(4)	Identifiant unique du lieu (Clé étrangère)
Epreuve	id_sport	INTEGER(4)	Identifiant unique du sport (Clé étrangère)

Sport	id_sport	INTEGER(4)	Identifiant unique du sport
Sport	nom_sport	VARCHAR(255)	Nom du sport
Utilisateur	id_utilisateur	INTEGER(4)	Identifiant unique de l'utilisateur
Utilisateur	nom_utilisateur	VARCHAR(255)	Nom de l'utilisateur
Utilisateur	prenom_utilisateur	VARCHAR(255)	Prénom de l'utilisateur
Utilisateur	login	VARCHAR(255)	Login de l'utilisateur
Utilisateur	password	VARCHAR(255)	Mot de passe de l'utilisateur
Participer	id_athlete	INTEGER(4)	Identifiant unique de l'athlète (Clé étrangère)
Participer	id_epreuve	INTEGER(4)	Identifiant unique de l'épreuve (Clé étrangère)
Participer	resultat	VARCHAR(100)	Résultat des épreuves

5.2.4. Modèle Conceptuel des Données (MCD)



5.2.5. Modèle Logique des Données (MLD)

Pays (id_pays, nom_pays)

- Clé primaire : id_pays

Genre (id_genre, nom_genre)

- Clé primaire : id_genre

Athlete (id_athlete, nom_athlete, prenom_athlete, id_pays, id_genre)

- Clé primaire : id_athlete
- Clé étrangère : id_pays en reference à id_pays de PAYS
id_genre en reference à id_genre de GENRE

Lieu (id_lieu, nom_lieu, adresse_lieu, cp_lieux, ville_lieux)

- Clé primaire : id_lieu

Epreuve (id_epreuve, nom_epreuve, date_epreuve, heure_epreuve, id_lieu, id_sport)

- Clé primaire : id_epreuve
- Clé étrangère : id_lieu en reference de id_lieu de LIEU
id_sport en reference de id_sport de SPORT

Sport (id_sport, nom_sport)

- Clé primaire : id_sport

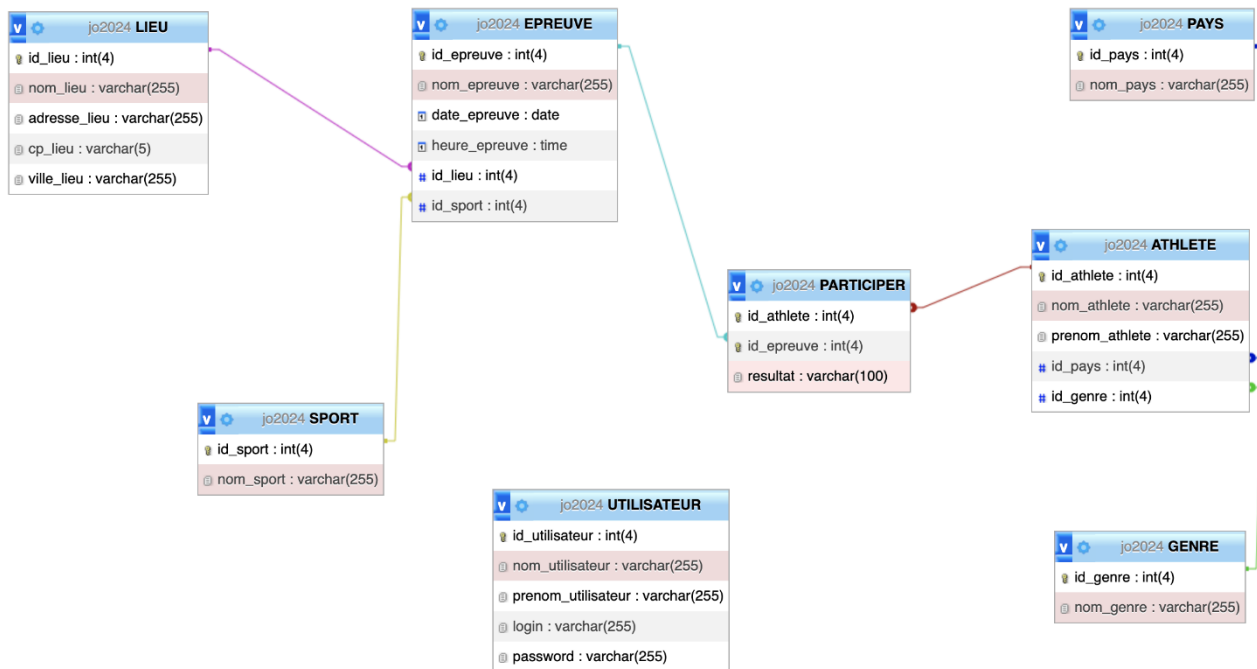
Utilisateur (id_utilisateur, nom_utilisateur, prenom_utilisateur, login, password)

- Clé primaire : id_utilisateur

Participer (id_athlete, id_epreuve, resultat)

- Clé primaire : id_epreuve en reference de id_epreuve de EPREUVE
id_athlete en reference à id_athlete de ATHLETE
- Clé étrangère : id_epreuve en reference de id_epreuve de EPREUVE
id_athlete en reference à id_athlete de ATHLETE

5.2.6. Modèle Physique des Données (MPD)



6. Technologies utilisées

6.1. Langages de développement Web

Afin de réaliser le site web nous allons utiliser des langages de programmations suivants :

- HTML
- CSS
- PHP
- JAVA SCRIPT

6.2. Base de données

Nous allons utiliser le langage de programmation SQL afin de créer et gérer notre base de données en passant par le Système de Gestion de Base de Données Relationnel (MySQL) de MAMP.

7. Sécurité

7.1. Login et protection des pages administrateurs

- Mettre des POST et non pas GET dans les formulaires de Login.
- Utiliser des : `session_start()`
`Session_unset()`
`Sessio_destroyer()`

7.2. Cryptage des mots de passe avec Bcrypt

Bcrypt est une technique de hachage utilisée pour se protéger du mot de passe contre les attaques des hackers en stockant les mots de passe sous un format « bcrypté ».

La fonction `password_hash` de PHP est une fonction intégrée qui est utilisée pour créer un nouveau hachage de mot de passe.

7.3. Protection contre les attaques XSS

La faille XSS, de son nom complet Cross-Site Scripting, est une faille qui permet d'injecter du code HTML et/ou Javascript dans des variables ou bases de données mal protégées.

Différentes solutions permettent de corriger la vulnérabilité du PHP aux attaques de scripts intersites. Entre autres, vous pouvez :

Inclure une protection anti XSS dans le header afin de désamorcer ces attaques.

Convertir les caractères spéciaux des entrées de formulaires en entités HTML grâce aux fonctions PHP `htmlspecialchars()` et `htmlentities()`.

Retirer les éléments entre les balises HTML grâce à la fonction `strip_tags()`.

Utiliser un pare-feu d'application web (WAF).

7.4. Protection contre les injections SQL

Une injection SQL, parfois abrégée en SQLi, est un type de vulnérabilité dans lequel un pirate utilise un morceau de code SQL (« Structured Query Language », langage de requête structuré) pour manipuler une base de données et accéder à des informations potentiellement importantes. C'est l'un des types d'attaques les plus répandus et menaçants, car il peut potentiellement être utilisé pour nuire à n'importe quelle

application Web ou n'importe quel site Web qui utilise une base de données SQL (soit la plupart).

Pour se protéger des injections SQL, nous allons donc utiliser la fonction **prepare** de PDO.

```
$sql = "SELECT `id`,`title` FROM `article` WHERE `title` LIKE ':%search%'";  
$data = [  
    'search' => $_POST['search']  
];  
$prep = $conn->prepare($sql);  
$result = $prep->execute( $data );
```