Cahier des charges techniques

E5 - Jeux Olympiques

PARIS 2024

Cahier des charges techniques créé par MTIMET Rayanne

À rendre le 22 mars 2024

Cahier des charges techniques

Sommaire

1. Contexte du projet

a. 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.

b. 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.

3. Ressources nécessaires à la réalisation du projet

a. Ressources matérielles

Pour assurer la mise en œuvre efficace du projet, les ressources matérielles indispensables comprennent :

- **Ordinateur :** Un équipement informatique essentiel pour exécuter les logiciels et effectuer les tâches de développement, de conception et de gestion du projet.
- **Moniteur :** Un écran d'ordinateur nécessaire pour visualiser et interagir avec les données, les applications et les interfaces utilisateur lors du processus de travail.
- **Périphériques :**
    - **Clavier :** Utilisé pour saisir des commandes, du texte et des données, facilitant ainsi la communication avec l'ordinateur et l'exécution des tâches.
    - **Souris :** Un dispositif d'entrée permettant de contrôler le curseur à l'écran, de cliquer sur des éléments et d'interagir avec les applications de manière intuitive.

- **Casque Audio :** Utile pour écouter des instructions, des rétroactions audio, des vidéos ou pour participer à des réunions virtuelles, assurant ainsi une communication claire et efficace pendant le processus de travail.

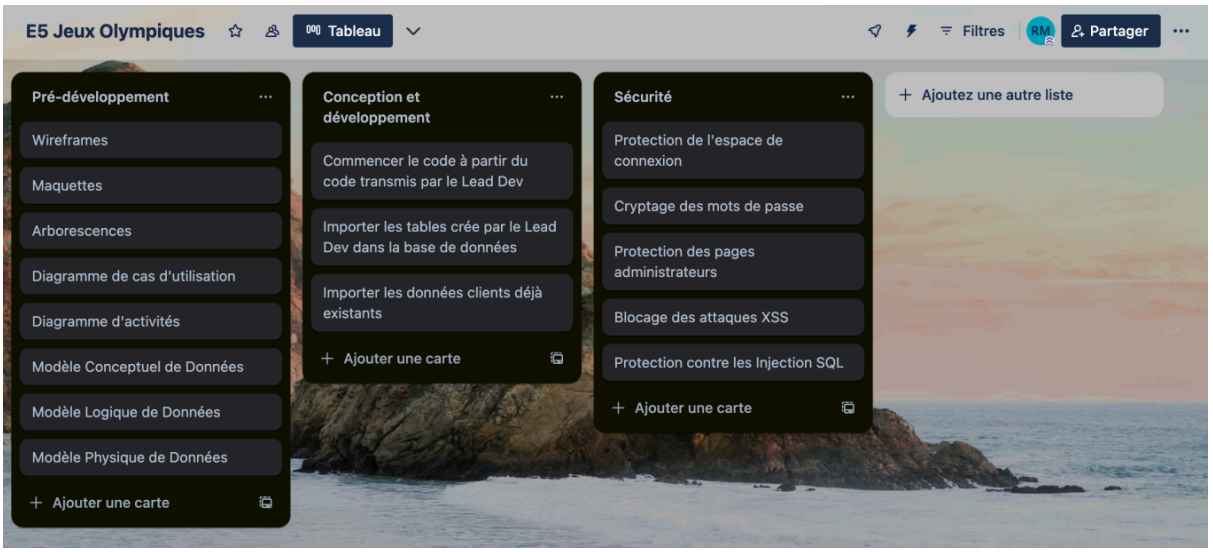
b. Ressources logicielles

Les outils logiciels essentiels pour mener à bien le projet incluent :

- **Visual Studio Code (IDE) :** Pour le développement et la programmation du code.
- **MAMP :** Un environnement de développement local pour la gestion de bases de données MySQL et la configuration de serveurs web Apache.
- **Figma :** Utilisé pour la création des wireframes et des maquettes afin de visualiser et conceptualiser l'interface utilisateur.
- **Trello :** Un outil de gestion de projet en ligne permettant l'organisation et le suivi des tâches, ainsi que la collaboration au sein de l'équipe.
- **Visual Paradigm :** Employé pour établir une arborescence structurée du projet, notamment pour la modélisation et la conception des processus.
- **Moccodo :** Pour simplifier la création du modèle conceptuel de données, facilitant ainsi la compréhension et la structuration des informations.
- **Github :** Une plateforme de développement collaboratif utilisée pour l'hébergement de code source, le suivi des versions, et la gestion des modifications apportées au projet.

4. Gestion du projet

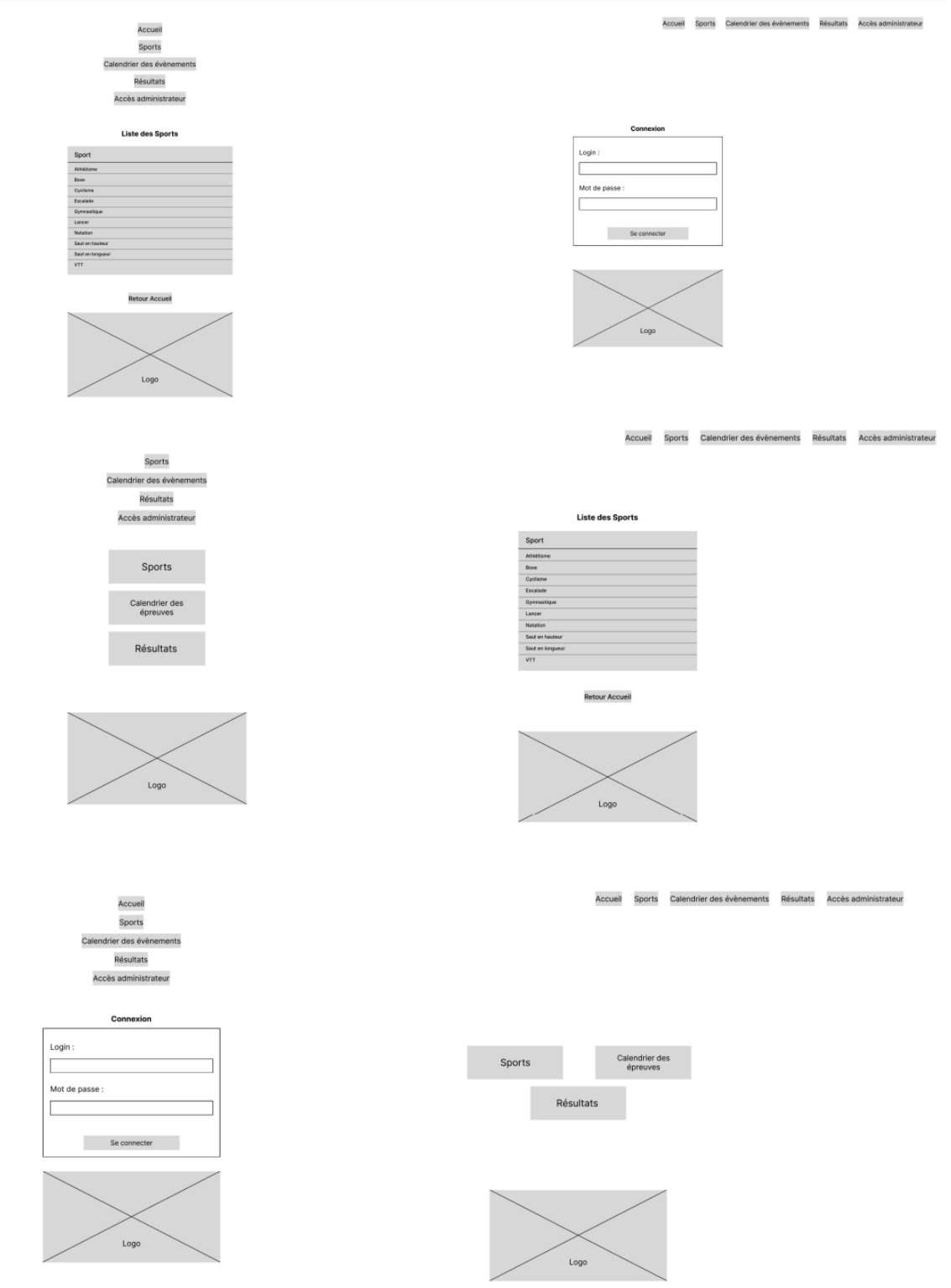
Pour la concrétisation de notre projet, nous opterons pour l'approche méthodologique Agile Kanban. De plus, nous ferons usage de la plateforme de gestion de projet en ligne Trello pour faciliter la coordination et le suivi des tâches.



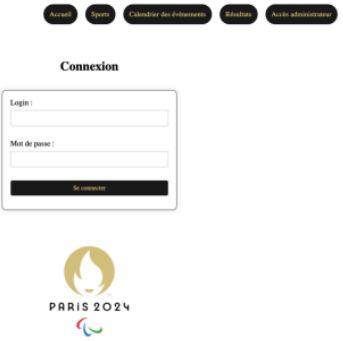
5. Conception du projet

a. Le front-end

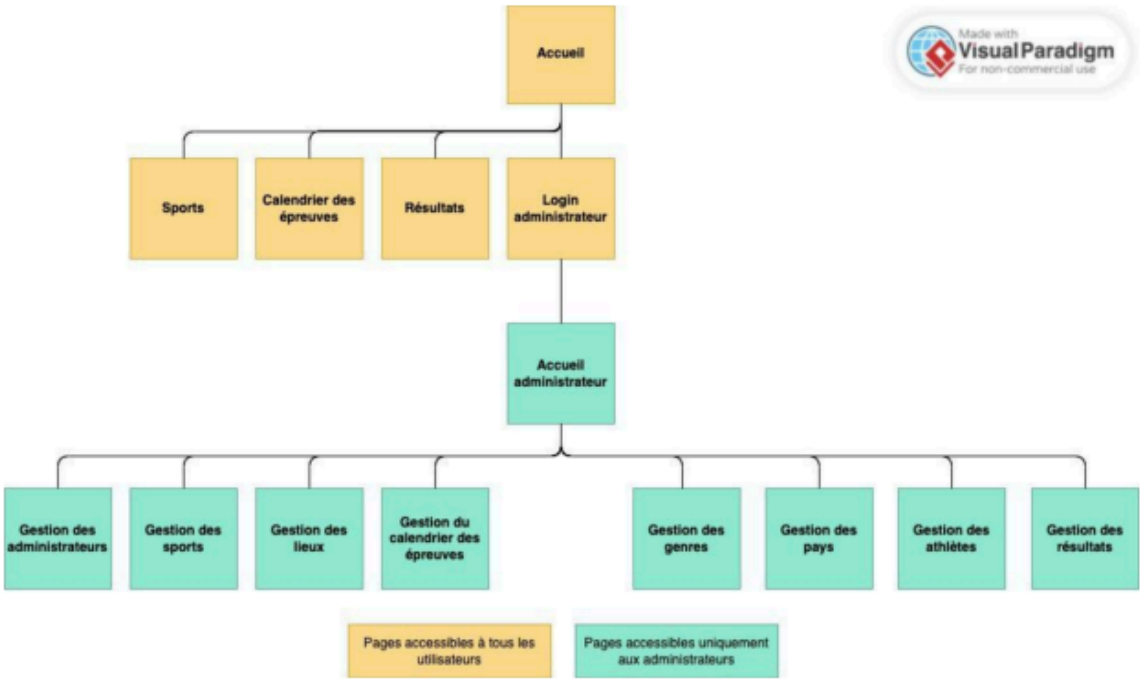
i. Wireframes



ii. Maquettes

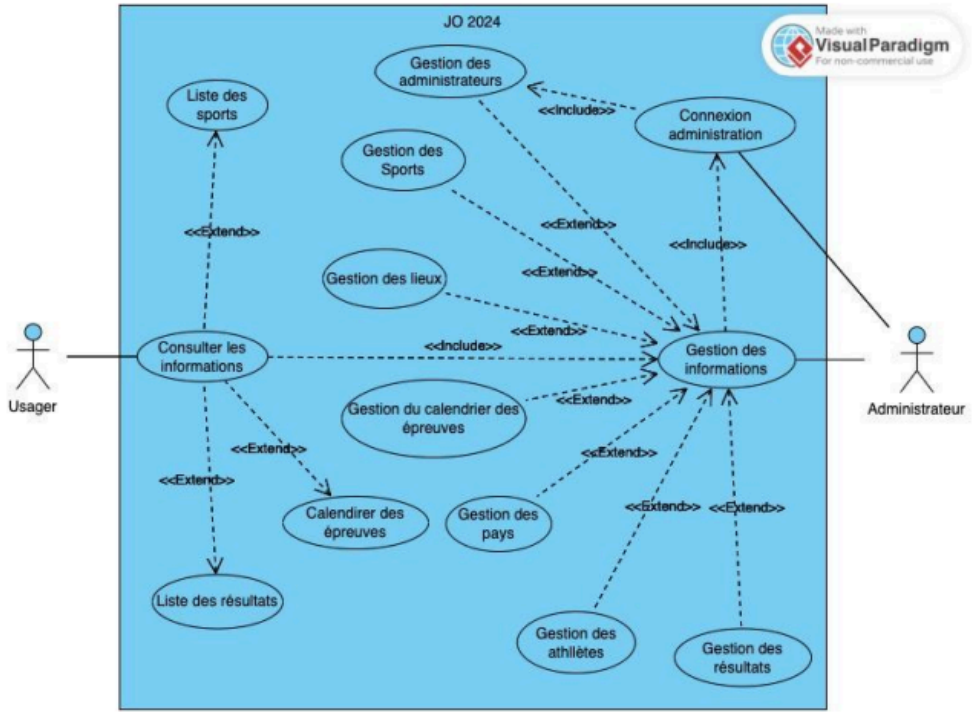


iii. Arborescences

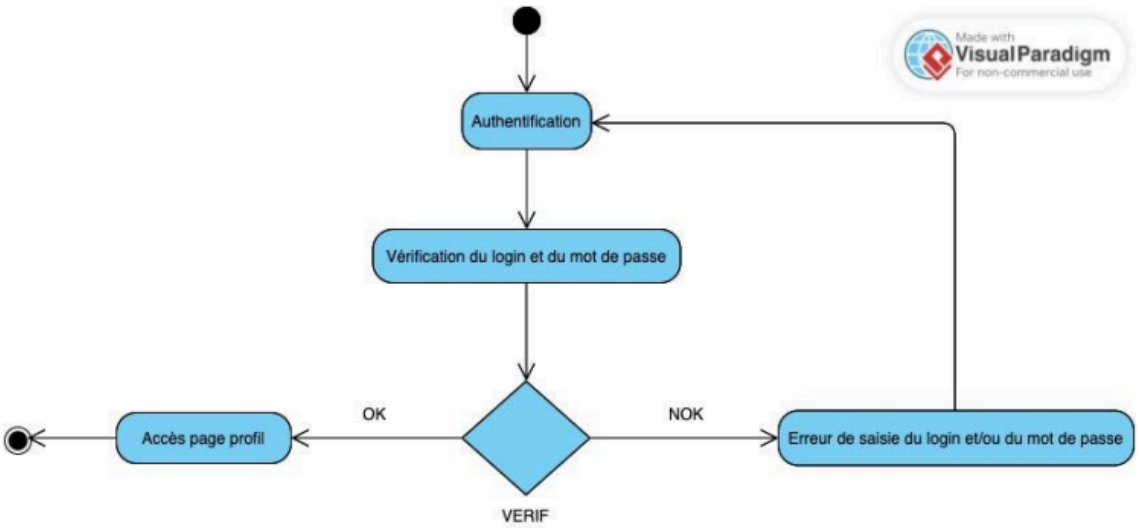


b. Le back-end

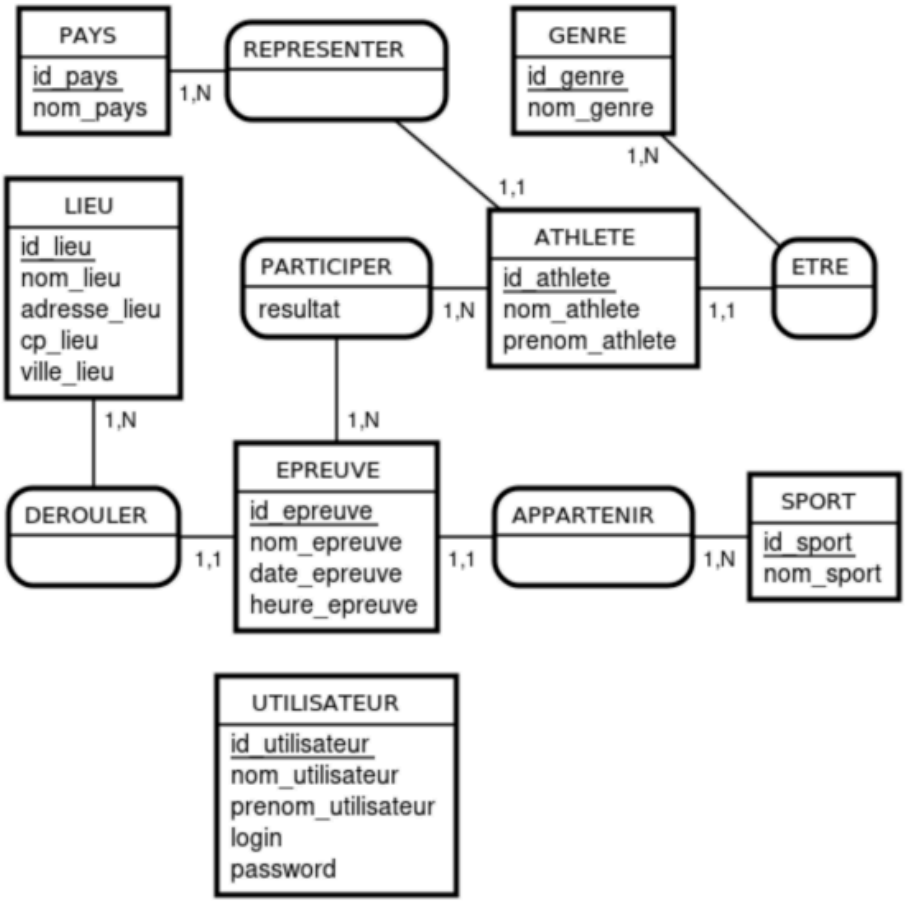
i. Diagramme de cas d'utilisation



ii. Diagramme d'activités



iii. Modèles Conceptuel de Données (MCD)

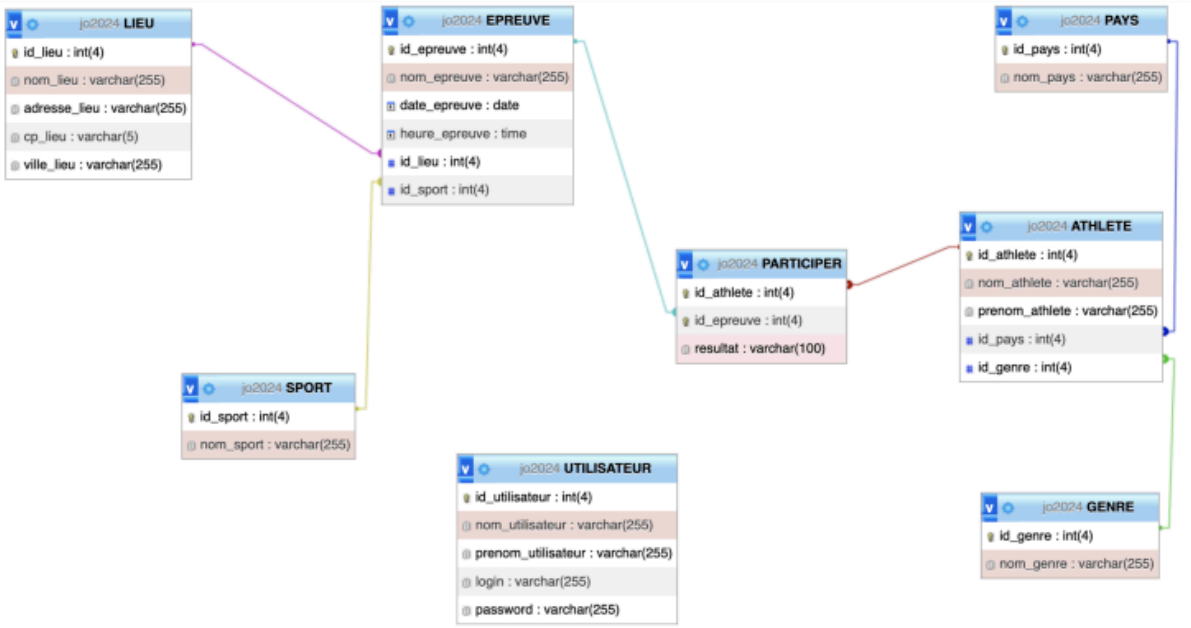


iv. Modèle Logique de Données (MLD)

Modèle Logique de données (MLD) en utilisant mocodo

- ATHLETE (id_athlète, nom_athlete, prenom_athlete, date_naissance_athlete, #id_genre, #id_nationalite)
- EPREUVE (id_epreuve, nom_epreuve, date_epreuve, #id_lieu, #id_sport)
- GENRE (id_genre, nom_genre)
- LIEU (id_lieu, lieu, adresse_lieu, cp_lieu, ville_lieu)
- NATIONALITE (id_nationalite, nom_nationalite)
- PARTICIPER (#id_athlète, #id_epreuve, resultats)
- SPORT (id_sport, nom_sport)
- UTILISATEUR (id_util, nom_util, prenom_util, login_util, mdp_util)

v. Modèle Physique de Données (MPD)



6. Technologies utilisées

a. Langages de développement web

Pour la réalisation du projet, nous prévoyons d'utiliser les langages de développement web suivants, chacun jouant un rôle spécifique dans la création et le fonctionnement des pages web interactives :

- **HTML5 :** Langage de balisage indispensable pour structurer le contenu des pages web, fournissant un cadre organisationnel pour les textes, les images, les liens et autres éléments.
- **CSS3 :** Langage de feuilles de style en cascade essentiel pour la présentation visuelle des pages web, permettant de définir le style, la mise en forme et la disposition des éléments HTML, ce qui contribue à l'aspect esthétique et à l'expérience utilisateur.
- **JavaScript :** Langage de programmation côté client incontournable pour ajouter des fonctionnalités interactives et dynamiques aux pages web, permettant une manipulation en temps réel du contenu et des éléments de la page, ainsi que l'interaction avec l'utilisateur.

b. Base de données

Pour la gestion de la base de données, nous avons planifié l'utilisation des technologies suivantes, combinant à la fois un langage de programmation serveur et un langage de requête structuré :

- **PHP8 :** Utilisé comme langage de programmation côté serveur, PHP8 permettra de dynamiser les interactions entre la base de données et le front-end de l'application web, facilitant ainsi le traitement des données et la génération de contenu dynamique.
- **SQL :** Langage de requête structuré essentiel pour la manipulation des données stockées dans la base de données, SQL sera employé pour exécuter des requêtes, des mises à jour et des opérations de gestion de données afin d'assurer la cohérence et l'intégrité des informations stockées.
- **MAMP :** En utilisant MAMP, une plateforme de développement local, nous serons en mesure de créer et de gérer efficacement une base de données MySQL. Cette solution offre un environnement de développement complet, intégrant un serveur Apache, une base de données MySQL et le langage de programmation PHP, ce qui facilitera le processus de développement et de test de notre application web.

7. Sécurité

L'objectif fondamental de la sécurité des sites web réside dans la prévention de diverses formes d'attaques et de compromissions. De manière plus précise, la sécurité des sites web se définit comme l'ensemble des mesures mises en œuvre pour garantir la protection contre l'accès non autorisé, l'utilisation abusive, la modification malveillante, la destruction ou la perturbation des sites web. Ces mesures visent à maintenir l'intégrité, la confidentialité et la disponibilité des données, des services et des fonctionnalités associés au site web, assurant ainsi la confiance des utilisateurs et la pérennité de l'activité en ligne.

a. Login

Pour gérer l'authentification des utilisateurs, nous prévoyons d'implémenter un formulaire de connexion dédié qui recueillera les informations de connexion, notamment le nom d'utilisateur et le mot de passe. Ces données seront ensuite comparées aux informations stockées dans la base de données afin de vérifier l'existence d'un compte correspondant. Il est à noter que cette fonctionnalité sera principalement utilisée pour l'accès à l'espace administrateur, soulignant ainsi l'importance de la sécurisation de ce processus.

b. Cryptage des mots de passe

Dans le cadre de la sécurisation des mots de passe, nous mettons en œuvre la fonction "password_hash", laquelle génère un hachage sécurisé des mots de passe. Ainsi, les mots de passe stockés dans la base de données sont cryptés, offrant une couche supplémentaire de sécurité. Cette approche garantit que les mots de passe ne sont pas stockés en texte brut, mais plutôt sous forme de hachages irréversibles, renforçant ainsi la protection des informations sensibles des utilisateurs contre les attaques potentielles.

c. Protection des pages administrateurs

Dans le souci de renforcer la sécurité des pages administrateurs, nous mettons en place une procédure de déconnexion rigoureuse à l'aide des fonctions "session_unset" et "session_destroy". Cette approche permet de nettoyer et de détruire toutes les variables associées à la session en cours dès que l'utilisateur quitte la page. Cette mesure préventive vise à éviter toute possibilité d'exploitation par des tiers malveillants, notamment en cas de copie d'URL de la page connectée et de tentative d'utilisation sur une session déconnectée. En effaçant les données de session dès la sortie de la page, nous réduisons significativement les risques de compromission de la sécurité et préservons ainsi l'intégrité des comptes administrateurs et des données sensibles associées.

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

Afin de prévenir les attaques XSS (Cross-Site Scripting), nous avons mis en place une série de mesures de sécurité. Tout d'abord, nous utilisons la fonction "htmlspecialchars" pour filtrer et échapper les caractères spéciaux présents dans les données utilisateur. Cela permet de neutraliser les tentatives d'injection de code malveillant dans les pages web.

De plus, nous utilisons des requêtes préparées lors de l'interaction avec la base de données. Les requêtes préparées sont une technique de programmation sécurisée qui permet de séparer les instructions SQL des données utilisateur, réduisant ainsi le risque d'injection SQL et d'autres formes d'attaques.

En complément, nous maintenons un backlog de sécurité afin de détecter et de corriger rapidement les vulnérabilités potentielles. Ce backlog comprend des audits de sécurité réguliers, des correctifs de sécurité et des mises à jour système pour garantir un niveau de sécurité optimal pour nos services.

En appliquant ces mesures de sécurité de manière proactive et continue, nous renforçons la protection de nos applications contre les attaques XSS et autres menaces potentielles, assurant ainsi la sécurité et la confidentialité des données de nos utilisateurs.

e. Protection contre les injections SQL

Dans le cadre de la protection contre les injections SQL, nous mettons en œuvre une stratégie qui comprend l'utilisation de la fonction "htmlspecialchars". Cette fonction joue un rôle crucial en convertissant les caractères spéciaux présents dans les données utilisateur en entités HTML, ce qui réduit le risque d'injection de code malveillant dans les requêtes SQL.

En convertissant les caractères spéciaux tels que les guillemets, les apostrophes et les signes inférieurs en entités HTML équivalentes, nous empêchons toute interprétation incorrecte de ces caractères par le moteur SQL. Ainsi, même si des données utilisateur contiennent des caractères potentiellement dangereux, ils seront traités comme du texte ordinaire et ne seront pas interprétés comme des éléments de syntaxe SQL.

Cette approche constitue une mesure préventive efficace pour éviter les attaques par injection SQL, en renforçant la sécurité de nos applications et en protégeant les données sensibles stockées dans nos bases de données contre les tentatives d'exploitation malveillante.