

Cahier des Charges

1. Présentation du Projet

1.1 Contexte et Objectifs

L'objectif de ce projet est de développer une application web permettant aux utilisateurs de créer, gérer et participer à des événements sportifs. L'application fonctionnera comme un réseau social où les membres pourront :

- Publier des événements sportifs (matches, tournois, entraînements, courses, etc.).
- S'inscrire aux événements d'autres utilisateurs.
- Laisser des commentaires et des likes sur les événements.
- Modifier ou supprimer les événements qu'ils ont créés.

1.2 Public Cible

- Sportifs amateurs souhaitant organiser des rencontres.
- Clubs sportifs cherchant à promouvoir leurs événements.
- Associations sportives.

2. Cahier des Charges Fonctionnel

2.1 Fonctionnalités Principales

Fonctionnalité	Description
Inscription et Connexion	Les utilisateurs doivent pouvoir s'inscrire avec un pseudo, email et mot de passe, puis se connecter à leur compte.
Création d'événements	Un utilisateur peut créer un événement en renseignant un titre, une description, une date, un lieu et éventuellement une image.
Modification/Suppression d'un événement	L'utilisateur ayant créé l'événement peut le modifier ou le supprimer.
Affichage des événements	Une page liste tous les événements avec des filtres par sport, date, localisation, etc.
Inscription aux événements	Les utilisateurs peuvent s'inscrire aux événements via un bouton dédié.
Like et Commentaires	Les utilisateurs peuvent liker un événement et laisser des commentaires.
Profil utilisateur	Chaque utilisateur a une page de profil affichant ses événements créés et auxquels il est inscrit.
Recherche et filtres (optionnel)	Une barre de recherche permet de trouver des événements par mots-clés, date ou localisation.

Interface administrateur (optionnel)	Un administrateur peut gérer les utilisateurs et modérer les événements si nécessaire.
--------------------------------------	--

3. Cahier des Charges Technique

3.1 Technologies Utilisées

- **Backend** : PHP (avec MySQL pour la base de données).
- **Frontend** : HTML, CSS, JavaScript (pour dynamiser certaines pages).
- **Base de données** : MySQL avec des tables pour les utilisateurs, événements, inscriptions, commentaires, likes.

3.2 Architecture de l'Application

L'application suivra l'architecture **MVC (Modèle-Vue-Contrôleur)** pour une meilleure organisation du code.

- **Modèle (Model)** : Gestion des données (utilisateurs, événements, inscriptions, etc.).
- **Vue (View)** : Affichage des pages HTML et dynamique avec JavaScript.
- **Contrôleur (Controller)** : Traitement des requêtes (ajout d'événements, modification, suppression, etc.).

4. Base de Données

Voici une première ébauche des tables nécessaires :

4.1 Tables principales

Table users (Utilisateurs)

Champ	Type	Description
id	INT (AUTO_INCREMENT)	Identifiant unique
pseudo	VARCHAR(50)	Nom d'utilisateur
email	VARCHAR(100)	Adresse email
password	VARCHAR(255)	Mot de passe hashé

Table events (Événements)

Champ	Type	Description
id	INT (AUTO_INCREMENT)	Identifiant unique
title	VARCHAR(100)	Titre de l'événement
description	TEXT	Détails de l'événement
date_event	DATETIME	Date et heure de l'événement
location	VARCHAR(100)	Lieu
user_id	INT	Créateur de l'événement (FK vers users.id)

Table inscriptions (Inscriptions aux événements)

Champ	Type	Description
id	INT (AUTO_INCREMENT)	Identifiant unique
event_id	INT	Événement concerné (FK vers events.id)
user_id	INT	Utilisateur inscrit (FK vers users.id)

Table comments (Commentaires)

Champ	Type	Description
id	INT (AUTO_INCREMENT)	Identifiant unique
event_id	INT	Événement concerné
user_id	INT	Auteur du commentaire
content	TEXT	Texte du commentaire
created_at	DATETIME	Date du commentaire

Table likes (Likes)

Champ	Type	Description
id	INT (AUTO_INCREMENT)	Identifiant unique
event_id	INT	Événement aimé
user_id	INT	Utilisateur ayant liké

5. Interfaces Utilisateurs (UX/UI)

5.1 Pages Principales

- **Page d'accueil** : Liste des événements récents avec possibilité de recherche.
- **Page de connexion / inscription.**
- **Page de création/modification d'événements.**
- **Page d'un événement** : Affiche les détails, les participants, les likes et les commentaires.
- **Page profil utilisateur** : Historique des événements créés et suivis.

6. Contraintes et Exigences

6.1 Contraintes Techniques

- Le projet doit être développé en PHP et MySQL pour la gestion des données.
- JavaScript sera utilisé pour améliorer l'expérience utilisateur (validation de formulaire, AJAX, etc.).
- Responsive Design pour s'adapter aux écrans mobiles et tablettes.

6.2 Contraintes de Sécurité

- Hashage des mots de passe avec `password_hash()`.

- Validation et nettoyage des données en entrée pour éviter les injections SQL (mysqli_real_escape_string() ou requêtes préparées).
- Protection contre le Cross-Site Scripting (XSS) en échappant les sorties HTML avec htmlspecialchars().

7. Planification du Projet :

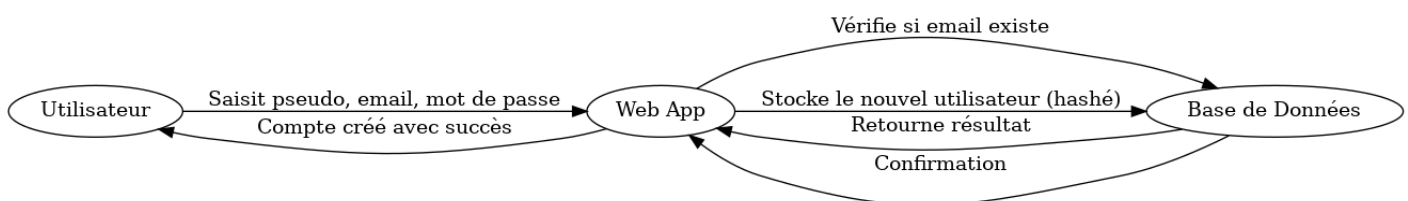
7.1 Étapes de Développement

Étape	Description	Durée estimée
Conception	Cahier des charges, maquettes et base de données	2 jours
Développement Backend	PHP + MySQL (CRUD des événements, utilisateurs, inscriptions)	4 jours
Développement Frontend	HTML/CSS/JS (Affichage dynamique, AJAX pour l'inscription et les likes)	3 jours
Tests et Débogage	Vérification des fonctionnalités et corrections	2 jours
Mise en ligne	Hébergement et derniers ajustements	1 jour

Total : 12 jours environ

8. Diagramme :

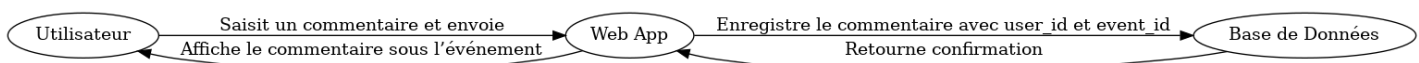
8.1 Diagramme de sequence Inscription :



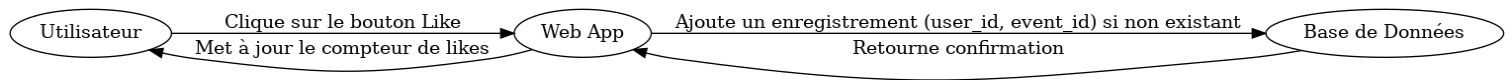
8.2 Diagramme de sequence Évènement :



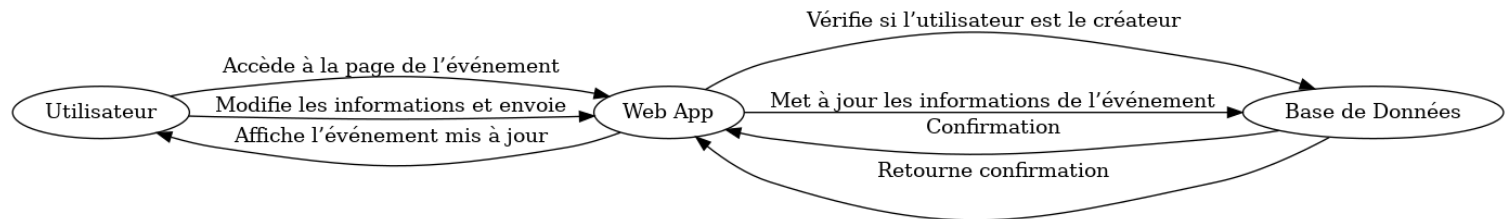
8.3 Diagramme de séquence ajout de commentaire :



8.4 Diagramme de séquence ajout de like :



8.5 Diagramme de séquence Modification d'un événement :



8.6 Diagramme de séquence d'inscription à un événement :



8.7 Diagramme de la base de données :

