

# DOCUMENTATION TECHNIQUE - ESPORTIFY

## Introduction technique du projet

Esportify est une application web de gestion d'événements esport permettant aux organisateurs de créer et gérer des tournois, et aux joueurs de s'y inscrire et suivre leurs activités.

- Projet full stack: PHP / MySQL / HTML / CSS
- Fonctionnalités principales : authentification, gestion des événements, inscription, tableaux de bord, chat
- Déploiement avec Render et Docker

## Choix technologiques

Partie	Technologie	Justification
Backend	PHP 8.x	Langage robuste, facile à héberger
Base de données	MySQL	SQL standard, supporté par Docker
Frontend	HTML/CSS	Responsive, rapide à intégrer
Versionning	Git + GitHub	Collaboration, bonnes pratiques
Environnement	Docker	Portabilité et déploiement local facile

## Architecture du projet

- /assets/ – Contient des ressources (Images, mais aussi les polices d'écritures utilisés).
- /CSS/ – Contient les fichiers de style CSS.
- /Doc/ – Répertoire de documentation.
- /pages/ – Contient les différentes pages de votre application :
  - /pages/API/ – Probablement les points d'entrée pour les API.

- /pages/auth/ – Pages liées à l'authentification (connexion, inscription, etc.).
- /pages/event/ – Pages spécifiques aux événements.
- /pages/404.html – Page d'erreur 404.
- /pages/about.html – Page "À propos".
- /pages/config.php – Fichier de configuration PHP.
- /pages/contact.html – Page de contact.
- /pages/home.html – Page d'accueil HTML.
- /pages/init\_db.php – Script d'initialisation de la base de données.
- /Scripts/ – Contient divers scripts :
  - /Scripts/DropDownMenu/ – Scripts liés aux menus déroulants.
  - /Scripts/Routeur/ – Scripts de routage.
  - /Scripts/Swiper/ – Scripts liés à Swiper JS (Pour images déroulants).
  - /Scripts/script.js – Fichier JavaScript principal.
- /.env – Fichier d'environnement.
- /.gitignore – Fichier de configuration Git pour ignorer certains fichiers/répertoires.
- /.htaccess – Fichier de configuration du serveur Apache.
- /docker-compose.yml – Fichier de configuration Docker Compose.
- /Dockerfile – Fichier Docker pour la construction de l'image.
- /esports\_db.sql – Fichier de script SQL pour la base de données.
- /index.php – Page d'accueil principale de l'application (probablement le point d'entrée).

Table	Champs	Description
users	id, username, email, password, role, created_at	Stocke les utilisateurs du site (joueurs, organisateurs, admin).
events	id, title, description, start_time, end_date, player_count, created_by, status, can_start_from, is_suspended	Stocke les événements e-sport créés par les organisateurs.
participations	user_id, event_id, status, joined	Gère les inscriptions aux événements, avec le statut (accepté, refusé...).
favorites	user_id, event_id	Permet à un joueur de marquer un événement comme favori.
scores	id, user_id, event_id, score, created_at	Historique des scores obtenus par un joueur à un événement.

## Configuration de l'environnement

- Docker + Docker Compose
  - o Service PHP
  - o Service MySQL
- Base de données initialisée avec esports\_db.sql
- Visual Studio Code comme IDE

## Sécurité

Mesure	Implémentation
Hachage mot de passe	password_hash() et password_verify()
Contrôle des accès	Vérification du rôle dans chaque page
Prévention SQL Injection	Requêtes préparées PDO
Sessions sécurisées	session_start(), régénération d'ID
Validation des formulaires	Frontend + Backend
Prévention failles XSS (à implémenter)	.replace(/&/g, "&amp;") .replace(/</g, "&lt;") .replace(/>/g, "&gt;") .replace(/"/g, "&quot;") .replace(/'/g, "&#039;")

## Déploiement

- **Docker** pour la conteneurisation de l'application (PHP, Apache, MariaDB, PostgreSQL).
- **Render.com** comme plateforme d'hébergement cloud (alternative à Heroku, simple pour projets PHP/MySQL).