DOCUMENTATION TECHNIQUE – ESPORTIFY

Introduction technique du projet

Esportify est une application web full-stack dédiée à la gestion de tournois e-sport. Elle permet aux organisateurs de créer et gérer des événements, et aux joueurs de s'y inscrire, suivre leur progression, et interagir avec la communauté. Le projet repose sur une architecture claire et modulaire, alliant frontend dynamique et backend.

Technologies utilisées

- **Frontend :** HTML5 / CSS3 (sans framework), JavaScript vanilla
- **Backend** : PHP, MariaDB / PostgreSQL pour les données relationnelles
- **Déploiement** : Conteneurisation avec **Docker**, hébergement via **Render**

Fonctionnalités clés

- Authentification sécurisée avec gestion des rôles (joueur, organisateur, admin)
- Création, validation et participation à des événements
- Filtres dynamiques, tableaux de bord personnalisés, système de scores
- Préparation à l'intégration d'une base **NoSQL (MongoDB)** pour les suggestions et les messages

Choix technologiques

Partie	Technologie	Justification
Backend	PHP 8.x	Langage serveur, facile à intégrer, compatible avec les hébergements web.
Base de données	MySQL	Système relationnel fiable, compatible avec Docker, SQL standard.
Frontend	HTML/CSS	Créer une interface responsive rapidement
Versionning	Git + GitHub	Suivi de l'évolution du projet, facilité de collaboration, bonne organisation du code.
Environnement	Docker	Déploiement et portabilité simplifié.

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/ – Assure la connexion de l'utilisateur.
    - /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 et d'initialisation de la base de données.
    - /pages/contact.html – Page de contact.
    - /pages/home.php – Page d'accueil HTML.
- /Scripts/ – Contient divers scripts :
    - /Scripts/DropMenu/ – 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 principale qui structure le site, incluant le header, le footer et les scripts essentiels à l'interface.

Modèle de données (MCD)

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 de travail en local

- Configuration Dockerfile + docker-compose.yml ce qui me créera :
  - Un Service PHP
  - Un Service MySQL
- Base de données initialisée avec esports_db.sql
- Utilisation de Visual Studio Code comme IDE (Environnement de développement)

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, "&") .replace(/&/g, "<") .replace(/&/g, ">") .replace(/&/g, """) .replace(/&/g, "'")

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).

Diagrammes UML

Vous retrouverez tous les documents dans le dossier Doc du projet