

DOCUMENTATION TECHNIQUE - ESPORTIFY

1. 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 / Bootstrap
- Fonctionnalités principales : authentification, gestion des événements, inscription, tableaux de bord, chat
- Déploiement avec Docker et GitHub

2. 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 + Bootstrap 5	Responsive, rapide à intégrer
Versionning	Git + GitHub	Collaboration, bonnes pratiques
Environnement	Docker	Portabilité et déploiement local facile

3. Architecture du projet

- /index.php – page d'accueil
- /login.php, /register.php
- /events.php, /event_details.php
- /myaccount.php, /myaccount_settings.php
- /admin/, /organizer/ – interfaces spécifiques
- /config/database.php – gestion PDO
- /assets/ – CSS + images
- /includes/ – navbar, footer, header

4. Modèle de données (ERD / MCD)

Tables principales:

- users (id, username, e-mail, password, role, created_at)
- events (id, title, description, start_datetime, end_datetime, max_players, organizer_id, is_validated)
- event_registrations (id, user_id, event_id, status)
- event_messages (id, user_id, event_id, message, created_at)

5. Configuration de l'environnement

- Docker + Docker Compose
 - Service PHP
 - Service MySQL
- Base de données initialisée avec esportify.sql
- Visual Studio Code comme IDE
- Extensions utiles: PHP Intellisense, Docker

6. 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

7. Diagrammes UML

Diagramme de cas d'utilisation

- Joueur : s'inscrire, se connecter, s'inscrire à un événement, discuter
- Organisateur : créer/modifier événement, voir inscriptions
- Admin: valider événements, supprimer utilisateurs

Diagramme de séquence (exemple : inscription à un événement)

1. Joueur clique "S'inscrire"
2. Formulaire POST vers register_event.php
3. Vérification du login, duplication
4. Insertion dans event_registrations
5. Redirection + message de succès

8. Déploiement

- Lancement via Docker : docker-compose up -d
- Import manuel de la BDD via fichier .sql
- Accès sur localhost:8082
- Déploiement potentiel (???)