#### o Documentation technique de votre application

## Réflexions initiales technologiques sur le sujet

Dès le début du projet, j'ai choisi de m'appuyer sur des technologies modernes et robustes pour développer la plateforme **Esportify**, afin de garantir à la fois une architecture claire, une bonne évolutivité et un rendu visuel de qualité.

#### Symfony (PHP) :

J'ai retenu Symfony comme framework principal car il est reconnu pour sa solidité, son respect du modèle MVC et sa forte communauté. Il facilite la gestion des routes, des entités, de la sécurité et propose un ensemble d'outils intégrés qui accélèrent le développement.

## Doctrine ORM :

Doctrine m'a permis de modéliser et gérer facilement la base de données via des entités PHP, sans écrire manuellement de requêtes SQL complexes. Cela apporte aussi plus de cohérence et de maintenabilité au projet.

## • Doctrine MongoDB ODM:

J'utilise le bundle Doctrine MongoDB ODM, qui permet de gérer facilement la persistance des documents MongoDB depuis Symfony avec un mapping similaire à Doctrine ORM.

# • Twig:

Pour la partie frontend, j'ai choisi Twig, moteur de template intégré à Symfony, qui permet de structurer proprement les vues et de réutiliser des morceaux de code HTML.

#### Bootstrap et FontAwesome :

Pour la partie design et ergonomie, j'ai opté pour Bootstrap (grilles responsives, composants prêts à l'emploi) et FontAwesome (icônes modernes et facilement intégrables). Cela m'a permis de rendre l'application agréable à utiliser sur desktop comme sur mobile, sans repartir de zéro en CSS.

#### • JavaScript (vanilla et AJAX) :

Pour certaines fonctionnalités interactives comme l'ajout/suppression de favoris, le rafraîchissement dynamique des listes d'événements ou l'affichage de messages de confirmation, j'ai intégré du JavaScript natif et des requêtes AJAX. Cela améliore l'expérience utilisateur sans recharger la page.

### Git :

Pour le versionnage et le suivi du projet, j'ai utilisé Git, ce qui m'a permis de revenir en arrière, corriger des erreurs, et tester des fonctionnalités sans risquer de casser l'application principale.

Ces choix technologiques m'ont semblé les plus pertinents pour construire une plateforme fiable, claire, et qui reste évolutive

• Configuration de votre environnement de travail

Pour développer et exécuter le projet **Esportify**, j'ai mis en place un environnement technique complet permettant de travailler efficacement, tester localement et préparer le déploiement.

## Technologies et versions utilisées

• **PHP**: version PHP 8.3.7 (cli)

• **Symfony**: Symfony 7.3.0

• Composer : pour la gestion des dépendances PHP

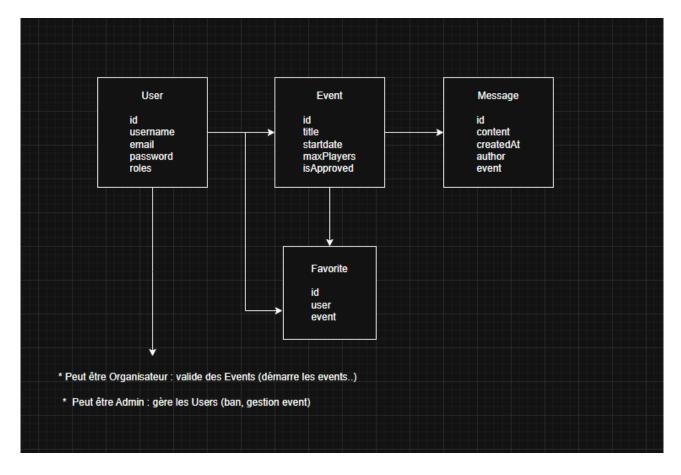
• Bootstrap et FontAwesome : pour le design et les icônes

• MySQL : pour la base de données relationnelle

## Outils utilisés pendant le développement

- **IDE**: Visual Studio Code, pour l'écriture du code PHP, Twig et JS
- Terminal intégré pour lancer les commandes Symfony, Composer
- phpMyAdmin ou ligne de commande pour visualiser et gérer la base de données
- Git pour versionner le projet et gérer les différentes versions du code

# Représentation (diagramme de classes simplifié)



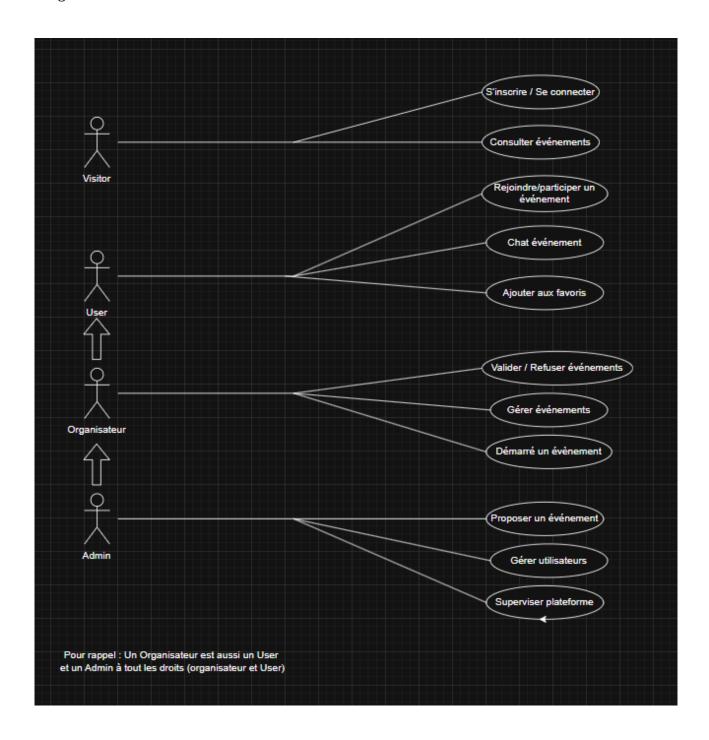
L'entité User possède un attribut roles qui peut contenir :

- ROLE\_USER
- ROLE\_ORGANISATEUR
- ROLE\_ADMIN

Cette approche permet de ne pas dupliquer les données utilisateur tout en donnant des droits différents.

• Diagramme d'utilisation ainsi que le diagramme de séquence

# Diagramme d'utilisation :



# Diagramme de séquence :

« Un utilisateur connecté s'inscrit à un événement depuis la page de l'événement. »

