# Application web Attisport :

## Compétence 1.1 :

<u>Exploiter des référentiels, normes et standards adoptés par le prestataire informatique :</u>

Eco-conception :

Amélioration du code pour qu'il soit plus éco-responsable grâce aux bonnes pratiques définit par le CNUMR.

### Utiliser le chargement paresseux

```
<img src="{{ asset("images/svg/down-arrow-thick.svg") }}" loading="lazy">
```

### Choisir un format de données adapté

```
#[ORM\Column(type: Types::SMALLINT)]
private ?int $team_2_score = null;

#[ORM\Column(type: Types::SMALLINT, nullable: true)]
private ?int $team_1_halftime_score = null;

#[ORM\Column(type: Types::SMALLINT, nullable: true)]
private ?int $team_2_halftime_score = null;
```

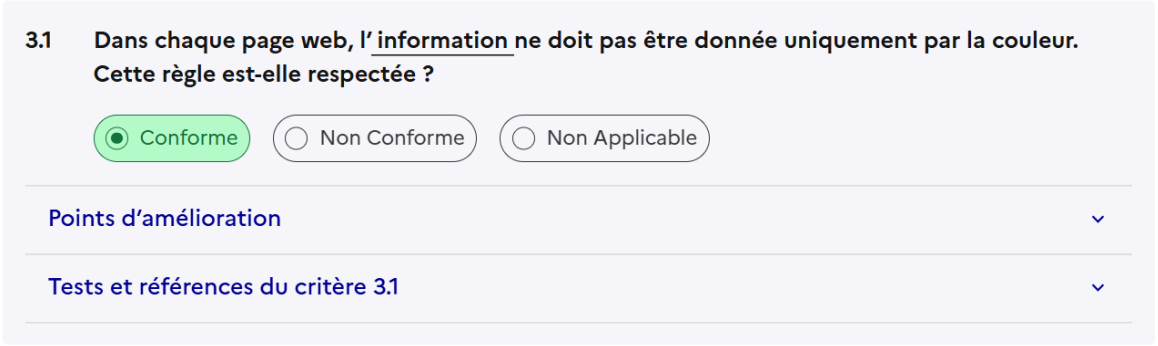
## Utiliser les notations CSS abrégées

```
margin-top: 10px;
margin-left: 5px;
```

```
margin: 0 auto 0 200px;
```

Accessibilité :

Réalisation d'un audit d'accessibilité, puis amélioration du code.





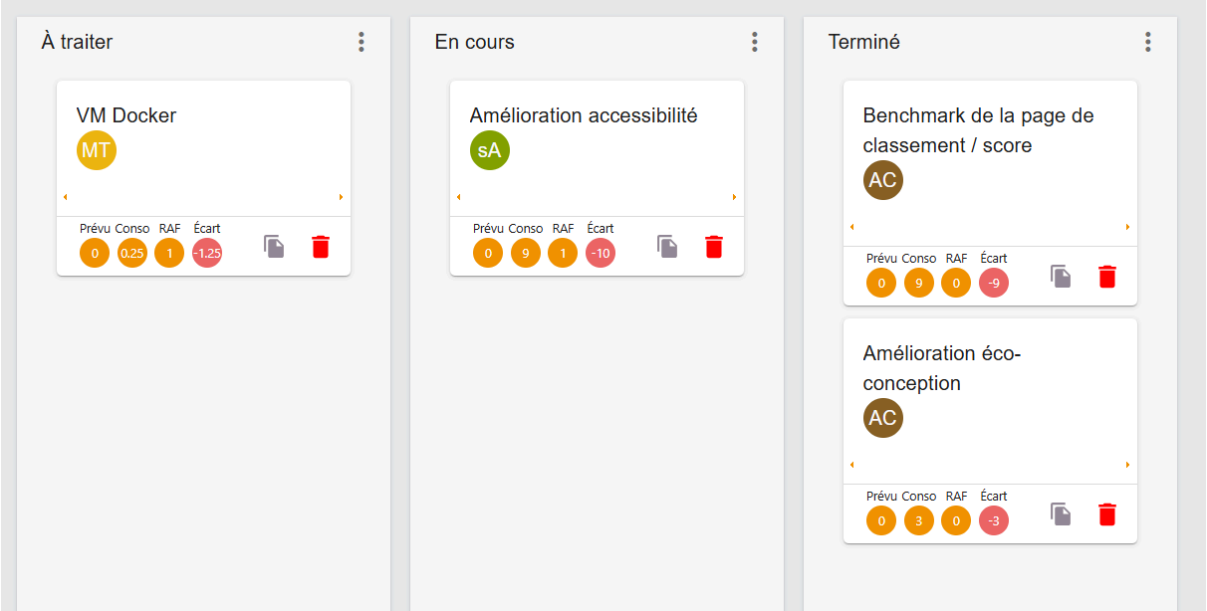
```
<div class="header-container">
    <ul role="presentation">
        <li>
            <a href="{{ path('app_matches'
        </li>
```

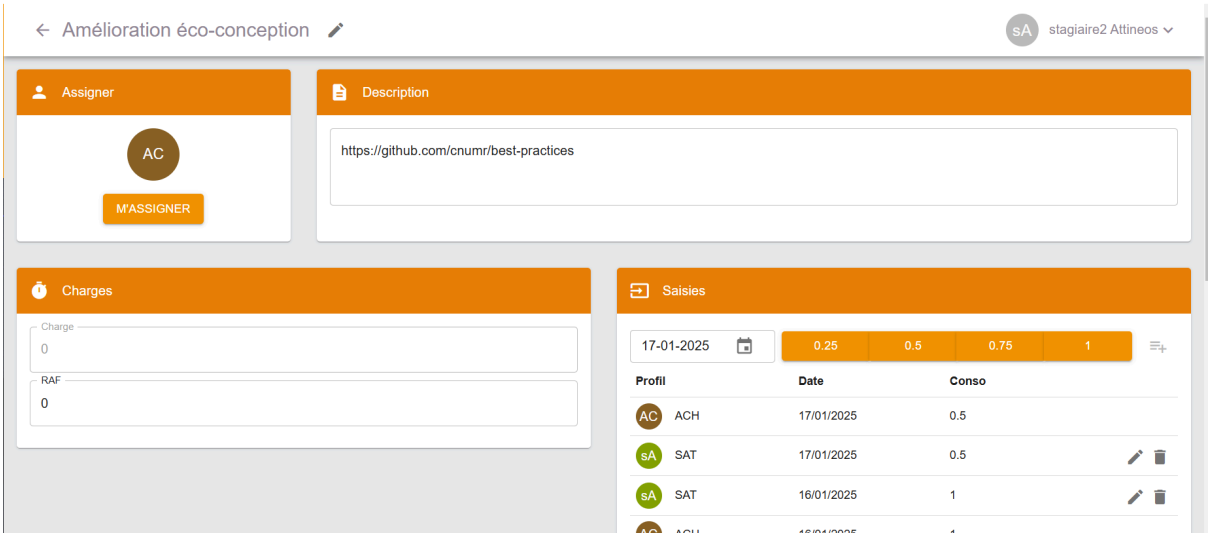
(pour que les technologie de lecteur d'écran ignore la signification du ul li)

## Compétence 1.4 :

<u>Planifier les activités :</u>

Utilisation de l'application Attiprod, développée par l'entreprise.




On avait également des réunions chaque semaine.

## Compétence 1.5 :

Accompagner les utilisateurs dans la mise en place d'un service :

Fichier readme

# Attisport V3

## Installation

- Installer Docker au préalable (si vous êtes sous windows, installer WSL2 : https://docs.docker.com/desktop/windows/wsl/)
- Git clone attisport
- Dans le fichier host (C:\Windows\System32\drivers\etc), ajouter ces lignes (il faut etre admin pour modifier le fichier) :

```
127.0.0.1 attisport.test
127.0.0.1 admin.attisport.test
```

- Copier le fichier .env.example de la racine du projet en .env et ajuster les valeurs au besoin
- Dans le back, copier le fichier .env.example et le nommer .env en remplaçant au besoin les valeurs par celles renseignées précédemment
- Dans le front-new, copier .env.developement en .env

## Pour lancer le projet

Lander l'environnement Docker :

```
docker login docker.attineos.cloud:1443 # entrez les identifiants Gitlab
docker compose up -d
```

### Initialisation BDD

Faire exécuter les commandes suivantes par le container PHP

```
docker exec -it attisport-api-php php artisan migrate
docker exec -it attisport-api-php php artisan db:seed
```

## Mails

Un docker maildev est inclu dans l'architecture de dev.
L'interface web est accessible à l'adresse suivante : http://localhost:1080

## Urls Dev

- Front : http://localhost:3000
- Back : http://attineos.test:8080
- Maildev : http://localhost:1080

# Documentation

## Symfony6.3Local

### Commencer

Pour vous permettre de démarrer facilement avec GitLab, voici une liste des prochaines étapes recommandées.

Vous êtes déjà un pro ? Modifiez simplement ce README.md et faites-en votre propre. Vous voulez vous faciliter la tâche ? Utilisez le modèle en bas !

### Ajoutez vos fichiers

- [ ] Créer ou télécharger des fichiers
- [ ] Ajoutez des fichiers à l'aide de la ligne de commande ou envoyez un référentiel Git existant avec la commande suivante :

```
cd existing_repo
git remote add origin https://gitlab.attineos.cloud/a.chevrel/symfony6.3local.git
git branch -M main
git push -uf origin main
```

### Intégrez-vous à vos outils

- [ ] Configurer les intégrations de projets

### Collaborez avec votre équipe

- [ ] Inviter les membres de l'équipe et les collaborateurs
- [ ] Créer une nouvelle demande de fusion

### Collaborez avec votre équipe

- [ ] Inviter les membres de l'équipe et les collaborateurs
- [ ] Créer une nouvelle demande de fusion
- [ ] Fermer automatiquement les problèmes à partir des demandes de fusion
- [ ] Activer les approbations de demandes de fusion
- [ ] Définir la fusion automatique

### Tester et déployer

Utilisez l'intégration continue intégrée dans GitLab.

- [ ] Démarrer avec GitLab CI/CD
- [ ] Analysez votre code pour détecter les vulnérabilités connues avec les tests de sécurité des applications statiques (SAST)
- [ ] Déployer sur Kubernetes, Amazon EC2 ou Amazon ECS à l'aide d'Auto Deploy
- [ ] Utilisez des déploiements basés sur l'extraction pour une meilleure gestion de Kubernetes
- [ ] Mettre en place des environnements protégés

## Modification de ce fichier README

Lorsque vous êtes prêt à personnaliser ce fichier README, il vous suffit de le modifier et d'utiliser le modèle pratique ci-dessous (ou n'hésitez pas à le structurer comme vous le souhaitez - ce n'est qu'un point de départ !). Merci à makeareadme.com pour ce modèle.

### Suggestions pour un bon README

Chaque projet est différent, alors réfléchissez à celles de ces sections qui s'appliquent au vôtre. Les sections utilisées dans le modèle sont des suggestions pour la plupart des projets open source. Gardez également à l'esprit que même si un fichier README peut être trop long et détaillé, il vaut mieux qu'il soit trop long que trop court. Si vous

des projets open source. Gardez également à l'esprit que même si un fichier README peut être trop long et détaillé, il vaut mieux qu'il soit trop long que trop court. Si vous pensez que votre fichier README est trop long, envisagez d'utiliser une autre forme de documentation plutôt que de supprimer des informations.

## Nom

Choisissez un nom explicite pour votre projet.

## Description

Faites savoir aux gens ce que votre projet peut faire spécifiquement. Fournissez un contexte et ajoutez un lien vers toute référence que les visiteurs pourraient ne pas connaître. Une liste de fonctionnalités ou une sous-section Contexte peut également être ajoutée ici. S'il existe des alternatives à votre projet, c'est un bon endroit pour énumérer les facteurs de différenciation.

## Insignes

Sur certains fichiers README, vous pouvez voir de petites images qui transmettent des métadonnées, comme si tous les tests du projet ont réussi ou non. Vous pouvez utiliser Shields pour en ajouter à votre fichier README. De nombreux services proposent également des instructions pour ajouter un badge.

## Visuels

Selon ce que vous faites, il peut être judicieux d'inclure des captures d'écran ou même une vidéo (vous verrez souvent des GIF plutôt que des vidéos réelles). Des outils comme ttygif peuvent vous aider, mais consultez Asciinema pour une méthode plus sophistiquée.

## Installation

Au sein d'un écosystème particulier, il peut exister une manière courante d'installer des éléments, comme l'utilisation de Yarn, NuGet ou Homebrew. Cependant, envisagez la possibilité que la personne qui lit votre fichier README soit novice et souhaite obtenir plus de conseils. La liste des étapes spécifiques permet d'éliminer toute ambiguïté et d'amener les gens à utiliser votre projet le plus rapidement possible. S'il ne s'exécute que dans un contexte spécifique, comme une version de langage de programmation ou