**D O C U M E N T A T I O N**

## Description du projet :

Ce projet consiste à faire un site e-commerce avec un maximum de fonctionnalités.

Le site contient 2 espaces :

* Espace administrateur
* Espace client

## Espace Administrateur :

L’administrateur est le gestionnaire du site, il pourra ajouter, modifier ou supprimer des :

* Articles
* Catégories
* Sous-catégories

Mais aussi, voir les messages des visiteurs ou des clients.

## Espace Client :

L’utilisateur pourra s’inscrire via une page d’inscription ergonomique où ce dernier n’aura qu’à saisir son nom d’utilisateur, email & mot de passe.

## Maquettes :

On a une page d’accueil sur laquelle on affiche tous nos articles :

Une image contenant capture d’écran

Description générée automatiquement

Une page catalogue où on liste toutes nos catégories et sous catégories :

Une image contenant capture d’écran

Description générée automatiquement

Une page d’inscription qui permettra à l’utilisateur de s’enregistrer :

Une image contenant capture d’écran

Description générée automatiquement

Une page d’authentification :

Une image contenant capture d’écran

Description générée automatiquement

Une page dashboard pour l’administrateur pour la gestion du site :

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement

## Technologies Utilisées :

Pour les technologies utilisées, en ce qui concerne le **back-end** : on a utilisé le framework Symfony 4 de PHP où nos APIs sont gérées avec une librairie nommée API PLATFORM qui permettra de générer du JSON-LD avec toutes les méthodes nécessaires d’un API : GET / POST / DELETE / PUT

Pour le **front-end** : on a utilisé Angular 7 qui est une technologie développée par Google pour le front et ceci à été fait grâce à des Services, Component, Controller, Entity en Typescript et aussi du HTML

## Description de la base de données :

Dans le back, on a géré notre base de données avec un ORM (Object Relational Mapping) nommé Doctrine qui permet de refléter nos objet dans l’application en tables dans la base de données.

Et cela, grâce à des commandes telles que :

* $php bin/console make :entity
* $php bin/console doctrine:database:create
* $php bin/console make:migration
* $php bin/console doctrine:migrations:migrate
* $php bin/console doctrine:schema:update –force

### Diagramme de classe :

Une image contenant texte, carte

Description générée automatiquement

## Méthode d’organisation :

On s’est réparti les tâches, 2 pour le front et un pour le back et la documentation.

Pour le front : Services, Controller / Design

Pour le back : API Platform + Documentation + Base de données