

## Configuration de l'environnement de travail

Pour Les besoins du projet, j'ai configuré et utiliser les technologies suivantes :

### - Visual studio Code

Cet IDE reconnu me permet de travail avec des extensions qui facilite le travail

### Initialisation du projet

Pour configurer mon projet, j'utilise Symfony coté back end qui me permettra d'avoir le squelette de mon projet avec toute les dépendances nécessaires ainsi que le serveur de mon projet.

Pour le Front end, React et Bootstrap me faciliteront le travail question style et structure

### - Structure des dossiers

Les fichiers ou je vais travailler le plus souvent :

**Arcadia/src/** : contient tout les fichiers sources

Il contient :

**Controller** : Classe servant à traiter les requêtes et renvoyer les réponses

**Entity** : Objet gérer par un ORM qui fera le pont entre symfony et Ma base de donnée (SQL)

**Datafixtures** : Permet de remplir ma base de donnée avec des données de test avec des valeurs réaliste.

**Form** : Abstraction pour gérer les formulaires pour La vue

**Template** : Vue qui sert pour l'interaction utilisateur

**Service** : Différents class réutilisable pour éviter les répétitions

**Arcadia/src/** :

**Styles** : Les fichiers css de mon projet

**Uploads** : Les images de différents entités

**Livrables** : Les différents document pdf pour l'utilisation et construction du projet

### Base de données :

Dans le fichier «.env» du projet symfony, je configure la base de données avec «Mysql». Qui est complé avec XAMPP qui me permet de gérer un serveur me donnant accès à phpMyAdmin.

## **Pattern**

Utilisant Symfony, je serais évidemment avec le design MVC qui me permettra de travailler sur le projet de manière adaptée :

Model = Les entités pour les données

View = Affichage de l'interface

Controller = Interaction entre le Model et Views

## **Depot git**

après avoir initialiser le squelette de mon projet symfony, je configure le dépôt git avec la commande init.

Selon sur quel partie je travail, une branche git sera créer (feature/entity, ect) ce qui permet une organisation de la modification travail ainsi que de ne pas infecter le dépôt principal si une des features a besoin de correction qui pourrait infecter la racine du projet.