

Configuration de l'environnement de travail

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

- Visual studio Code

Cet IDE reconnu me permet de travail avec des extensions qui facilitent 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 Coté Front end

- Structure des dossiers

Les fichiers avec lesquels je vais travailler le plus souvent sont :

Arcadia/src/ : contient tous 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 données (SQL)

Datafixtures : Permet de remplir ma base de données avec de fausses données aux valeurs réalistes pour tester les bases de données.

Form : Abstraction pour gérer les formulaires pour la vue

Template : Vue utilisée pour l'interaction avec l'utilisateur

Service : Différentes classes réutilisablse pour éviter les répétitions

Arcadia/Public/ :

Styles : Les fichiers css de mon projet

Uploads : Les images de différentes 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». J'utilise également XAMPP qui permet de gérer un serveur local pour la bdd.

Pattern

Utilisant Symfony, je m'appuierai évidemment sur le désigne MVC (Model-View-Controller) qui me facilitera le travail de mon projet de manière appropriée :

Model = Les entités du projet

View = Affichage de l'interface

Controller = Interaction entre le Model et Views

Depot git

A la suite de l'installation du squelette de mon projet Symfony, qui est la branche « main » (principale), j'effectue la configuration du dépôt Git en utilisant la commande init. Selon la fonctionnalité sur laquelle je travaille, une branche git sera créée (par exemple : Feature ou Entity) et fusionnée avec la branche de développement pour des modifications éventuelles. Ceci me permet d'organiser le travail de modification tout en évitant d'affecter le dépôt principal si une fonctionnalité nécessite des ajustements qui pourraient compromettre la racine du projet. **Selon sur quelle fonctionnalité je travaille, une branche git sera créée (feature/entity par exemple) qui sera merge sur la branche de développement pour d'éventuelle correction ce qui permet une organisation de la modification travail ainsi que de ne pas infecter le dépôt principal si une des fonctionnalités besoin de correction ce qui pourrait infecter la racine du projet.**

Une fois les corrections de la branche effectuées, je vais procéder à la fusion avec la branche « main », qui servira de branche principale pour la production de l'hébergeur du projet. **Dès que les corrections de la de développement sont faites, je merge sur la branche «main» qui sera la branche pour la production sur L'hébergeur du projet.**