

Une documentation technique :

Lors du développement de mon application, dans un premier temps j'ai analysé le sujet sur lequel nous étions évalué (les avantages et les contraintes). J'ai préféré utiliser symfony en tant que framework car il s'adapte à toute situation et ce qui est bien la partie front et back end sont très adaptable par rapport a d'autre frameworks comme react

1-Technologies choisies :

La partie front end : • HTML 5 • CSS 3 • Bootstrap • JavaScript

Outils transverse : Mamp

La partie back end : • Symfony • MySQL

2-La configuration de l'environnement de travail :

J'ai démarré mon projet directement sur l'éditeur de texte Visual studio code qui s'adapte très bien à ce qui est demandé.

Démarrage d'un projet avec le framework symfony :

taper dans la ligne de commande `symfony new --full symfony medaiv2`

ensuite ajouter les annotations avec `composer require symfony/orm-pack/composer require --dev et symfony/maker:bundle.`

Ensuite se connecter au serveur distant symfony par le biais de mamp avec la commande `symfony server:start.`

Par la suite création de la base de donnée avec la ligne de commande `php bin/console doctrine database:create.`

Par la suite, j'ai créé un nouveau repository via github pour ce projet afin d'effectuer régulièrement des commits et push afin de sauvegarder mon travail tout au long de cette évaluation.

J'ai démarré par la partie front en minimisant les écrans et par la suite en transverse j'étais sur la partie back end.

Les diagrammes de classe, de séquence ou de cas d'utilisation sont disponibles en format PDF dans mon dossier ECF transmis.

Par la suite, j'ai codé en me documentant lorsque j'étais perdu ou en posant des questions dans le forum discord.

3- La liste des bonnes pratiques de sécurité mises en place sur cette application, avec la justification pour chacune d'entre elles :

```
public function __construct(UrlGeneratorInterface $urlGenerator,  
    UserPasswordEncoderInterface $passwordEncoder)  
{  
    $this->urlGenerator = $urlGenerator;  
}
```

Suite à la commande `php bin/console make:auth`, symfony nous a créé une rubrique pour la sécurisation du site au moment de l'authentification lors de la connexion.

-

-User password encoder interface est une fonction afin de crypter le mot de passe