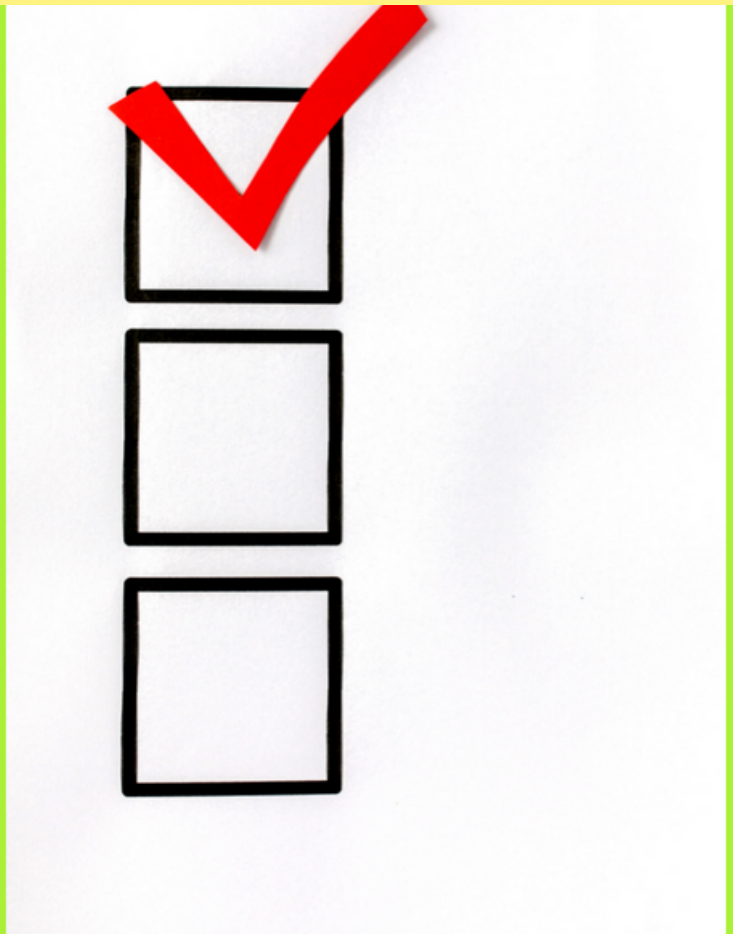
PROJET-8

DOCUMENTATION TECHNIQUE

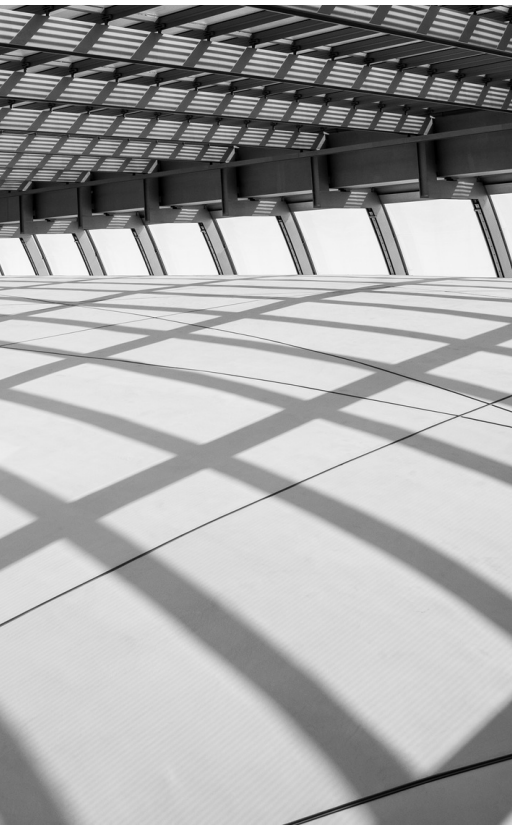
TO DO LIST



2022

LA MISE EN PLACE DU PROJET

Le projet TodoList est une application permettant de gérer ses tâches quotidiennes. Ce projet a été développé à ce jour dans l'idée d'un futur développement grâce à une levée de fonds. Il utilise le framework symfony.



A- L'initialisation du projet

Description de l'initialisation du projet- Les débuts du projet et les potentiels d'évolution.

B- L'organisation du projet

Organisation globale du projet Comment gérer les modifications et évolutions du projet.

C- Les tests unitaires et fonctionnels

INITIALISATION

A- L'initialisation du projet

Clonez le projet ici:

git@github.com:marinejourdan/PROJET8-Todolist.git

- Realisation technique du blog

Visual studio code-éditeur de texte (html/php/css)
p**hpmyadmin:** serveur de gestion de base de données
G**it/Github:** Gestionnaire de versions

- Installation du projet

Installez git et récupérez le projet de Github
git@github.com:marinejourdan/PROJET8-Todolist.git

Paramêtrer le projet avec vos propres données (.env).
-exécutez les migrations (3 versions à éxecuter avec chargement du jeu de données dans les migrations)

- composer.json

Au sein de votre composer.json à la racine du projet, vous aurez accès aux versions actuelles des bundles et services de votre projet.
Il vous faudra faire évoluer le projet à partir des mises à jour de ce composer.

- console

Accès à la console au sein du dossier bin au sein du projet.

- security.yml

Accès à la partie sécurité du projet : connexions et rôles définis.

```
 access_control:

 - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
 - { path: ^/logout, roles: ROLE_USER }
 - { path: ^/tasks/.*, roles: ROLE_USER }
 - { path: ^/tasks, roles: IS_AUTHENTICATED_ANONYMOUSLY }
 - { path: ^/users, roles: ROLE_ADMIN }


role_hierarchy:
 ROLE_ADMIN: ROLE_USER
```

LES POINTS PRINCIPAUX DU PROJET

Au sein du projet vous aurez accès à plusieurs dossiers:

- **Cas utilisation :** Cas pratique pour l'utilisateur du projet
- **les diagrammes de séquences :** échanges client/serveur et base de données
- **modèles de données:** modèle conceptuel et physique de données/ organisation des relations basées sur le modèle UML.
- **config:** Outils de configuration du projet (security) affichage des pages en twig
- **bin:** console
- **src:** data fixtures- controller - entités- formulaires
- **var:** cache-logs-sessions
- **vendor:** bundles
- **web**:css, font, images...
- **tests**: à effectuer après chaque modification
- **migrations**: à effectuer lors de l'initialisation du projet

LES ETAPES DE GESTION DU PROJET

Notre organisation de travail se déroule en étape, chaque mof=ification entraîne des points obligatoires pourpermettre une évolution securisée du projet et un travail en équipe.

1- Demande de modification du projet

2- réflexion et (re)écriture des tests fonctionnels en adéquation avec la modification à apporter (attention, certains tests fonctionnels seront à modifier si les fonctions évoluent)

3- modification à apporter dans la fonction

4- passage en revue en ligne de commande des tests complets unitaires et fonctionnels

5- correction des éventuels bugs lors des tests

6- enregistrement de la couverture de code

7- Validation avec le chef de projet

Attention ! Chaque modification doit être suivie pas à pas sur github à l'aide des commits détaillant les étapes et d'une pull request accessible au chef de projet.