1️⃣ Documentation Technique

La **documentation technique** est dédiée aux équipes de développement et à toutes les parties prenantes impliquées dans le projet. Elle décrit les choix technologiques, l'architecture du système, les API utilisées, ainsi que la configuration et l'intégration des différents composants.

1.1 Introduction

Cette section présente un aperçu global du projet et de ses objectifs, ainsi que des technologies choisies pour sa réalisation.

- **Nom du projet :** Portfolio Dynamique
- **Technologies utilisées :**
    - Frontend : React.js
    - Backend : Symfony
    - Base de données : MySQL
    - Communication entre le frontend et le backend : Axios (pour les requêtes API)
    - Outils : Git (versioning).

1.2 Architecture du projet

Une vue d'ensemble de l'architecture logicielle et de l'organisation du code.

- **Frontend (React.js) :**
    - Structure des composants (projets, popup, navbar, etc.).
    - Utilisation de `useState` et `useEffect` pour la gestion des états et la récupération des données via l'API Symfony.
    - Gestion du routing et de la navigation fluide entre les pages du portfolio.
- **Backend (Symfony) :**
    - API RESTful permettant la gestion des projets via des requêtes GET, POST, PUT, DELETE.

    - Gestion de la base de données MySQL avec deux tables principales : `projects` et `project_images`.
    - Sécurité de l'API avec gestion des permissions et des rôles.

1.3 API

Description des différentes routes API et des actions associées.

Méthode	Route	Description
GET	/api/projects	Récupère la liste des projets.
POST	/api/projects	Ajoute un nouveau projet.
PUT	/api/projects/{id}	Met à jour un projet existant.
DELETE	/api/projects/{id}	Supprime un projet existant.

1.4 Base de données

Structure de la base de données et de ses tables.

- **Table projects** : Stocke les informations sur les projets.
    - `id` : Identifiant du projet.
    - `title` : Titre du projet.
    - `description` : Description du projet.
    - `functionalities` : Liste des fonctionnalités du projet.
- **Table project_images** : Stocke les images associées aux projets.
    - `id` : Identifiant de l'image.
    - `project_id` : Référence au projet associé.
    - `image_url` : URL de l'image.

1.5 Sécurité et déploiement

- Mise en place de mécanismes de sécurité (validation des entrées, gestion des erreurs).

- Déploiement de l'API Symfony avec un serveur de développement local (http://127.0.0.1:8000).

- Utilisation de Docker pour simplifier le déploiement en environnement de production.