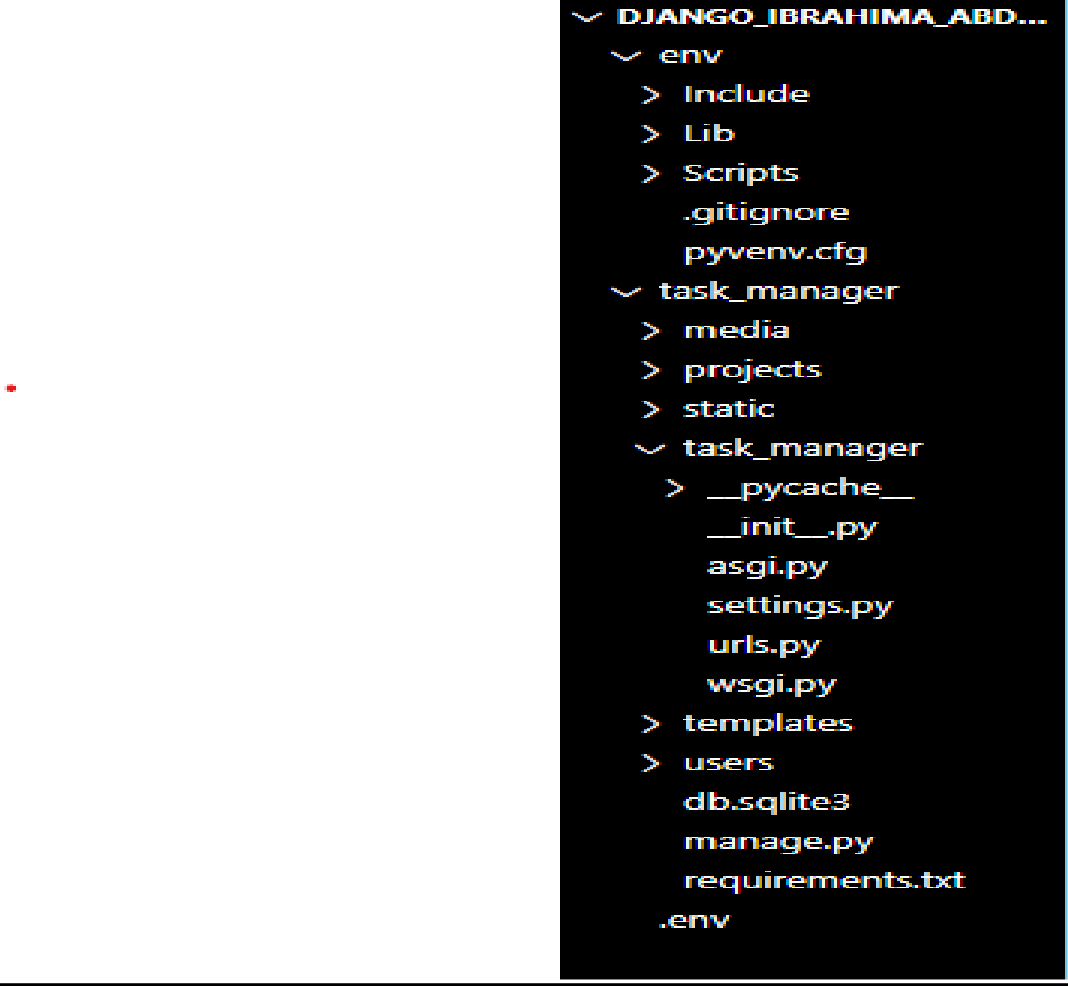
Documentation de Task\_Manager - Django Project réalisé par Ibrahima Abdoul BA

**Structure du projet**

****

**Description**

Task Manager est une application web permettant la gestion des tâches et des utilisateurs. Ce projet Django est structuré en deux applications principales :

**users** : Gestion des utilisateurs (authentification, profils, etc.).

**projects** : Gestion des projets et des tâches associées.

1. **Les users**

|  |  |  |
| --- | --- | --- |
| **UTILISATEURS** | | |
| Noms d’utilisateurs | Mots de passe | Rôles |
| BrahimKhalil | Khalilb@h12 | super utilisateur |
| SambaEden | Khalilb@h03 | simple |
| BrahimKhalilbah | Khalilb@h92 | simple |
| SambaSy | s@mb@2125 | simple |

1. **Les projects**

|  |  |
| --- | --- |
| **Utilisateurs(users)** | **Projets(projects)** |
| BrahimKhalil | Système de réservation de restaurants,  Application de gestion de bibliothèque |
| BrahimKhalilbah | aucun |
| SambaEden | MonProjet, Application de gestion de patients, Plateforme E-commerce pour produits  Technologiques |
| SambaSy | Projet initial |

**Les routes**

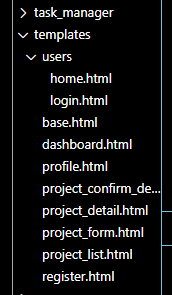
* 1. **Application users**

|  |  |
| --- | --- |
| Routes | Pages |
| /login | connexion |
| /register | inscription |
| /profile | Voir profile user connecté |
| /dashboard | La page Dashboard |
| Home | La page d’accueil par défaut |

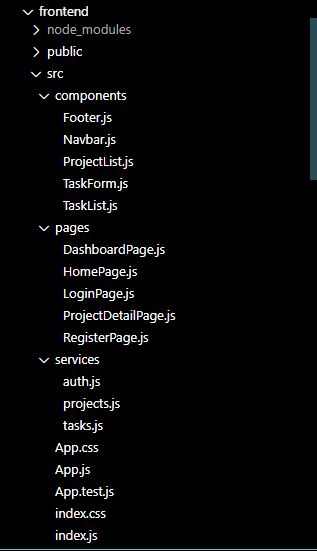
* 1. **Application projects**

|  |  |
| --- | --- |
| Routes | Pages |
| /projects/create | Création d’un projet |
| /projects/listproj | Liste des projets |
| /projects/create | Voir détails du projet |
| /projects/pk/update | Modifier un projet |
| /projects/pk/delete | Supprimer un projet |

**Templates**

****

**Partie Frontend**

****

**Structure du Frontend**

**1. Dossiers Principaux**

* **node\_modules/** : Contient toutes les dépendances installées pour le projet.
* **public/** : Contient les fichiers publics comme index.html, qui sert de point d'entrée pour l'application.
* **src/** : Contient le code source de l'application React.

**2. Structure du Dossier src/**

Le dossier src/ est organisé en plusieurs sous-dossiers et fichiers pour une meilleure modularité et maintenabilité.

**2.1. Composants (components/)**

Ce dossier contient les composants réutilisables de l'application :

* **Footer.js** : Composant pour le pied de page de l'application.
* **Navbar.js** : Composant pour la barre de navigation.
* **ProjectList.js** : Composant pour afficher la liste des projets.
* **TaskForm.js** : Composant pour le formulaire de création ou de modification de tâches.
* **TaskList.js** : Composant pour afficher la liste des tâches.

**2.2. Pages (pages/)**

Ce dossier contient les composants de page qui représentent les différentes vues de l'application :

* **DashboardPage.js** : Page du tableau de bord.
* **HomePage.js** : Page d'accueil.
* **LoginPage.js** : Page de connexion.
* **ProjectDetailPage.js** : Page de détails d'un projet.
* **RegisterPage.js** : Page d'inscription.

**2.3. Services (services/)**

Ce dossier contient les services pour interagir avec l'API backend :

* **auth.js** : Service pour gérer l'authentification (connexion, déconnexion, etc.).
* **projects.js** : Service pour gérer les opérations liées aux projets.
* **tasks.js** : Service pour gérer les opérations liées aux tâches.

**2.4. Fichiers Principaux**

* **App.css** : Feuille de style globale pour l'application.
* **App.js** : Composant principal de l'application, qui sert de conteneur pour les routes et les composants.
* **App.test.js** : Fichier de test pour le composant App.
* **index.css** : Feuille de style pour le point d'entrée de l'application.
* **index.js** : Point d'entrée de l'application React, où le rendu est effectué dans le DOM.

**Fonctionnalités Clés**

1. **Authentification** :
   * Gérée par le service auth.js.
   * Inclut les pages LoginPage.js et RegisterPage.js.
2. **Gestion des Projets** :
   * Affichage des projets via ProjectList.js.
   * Détails d'un projet via ProjectDetailPage.js.
3. **Gestion des Tâches** :
   * Création et modification de tâches via TaskForm.js.
   * Affichage des tâches via TaskList.js.
4. **Navigation** :
   * Barre de navigation via Navbar.js.
   * Pied de page via Footer.js.
5. **Routing** :
   * Les pages sont organisées dans le dossier pages/ et sont probablement gérées par un routeur (comme react-router-dom).

**Exemple de Flux**

1. **Connexion** :
   * L'utilisateur accède à LoginPage.js et se connecte via auth.js.
   * Après connexion, il est redirigé vers DashboardPage.js.
2. **Tableau de Bord** :
   * Le tableau de bord affiche les projets via ProjectList.js.
   * L'utilisateur peut cliquer sur un projet pour accéder à ProjectDetailPage.js.
3. **Détails du Projet** :
   * La page de détails affiche les tâches associées via TaskList.js.
   * L'utilisateur peut ajouter ou modifier des tâches via TaskForm.js.

🎯 **Objectif**

Ce projet vise à fournir une base solide pour gérer des tâches et des projets en Django, en appliquant les bonnes pratiques du Framework.