

Les taches effectuées:

- \*Création du projet avec la commande: `django-admin startproject InventoryApp`

- \*Le fichier `requirements.txt` va nous permettre d'installer les bibliotheques dont on aura besoin sur un autre serveur grace a la commande : `pip install -r requirements.txt`.

- \*Pour avoir un aperçu de notre application en lance le serveur en local avec `python manage.py runserver` puis on copie l'adresse pour aller sur un navigateur.

- \*Connecter notre application a une base de données postgresql `Inventory_db` (user : `Inventory_user`).

- \*création d'un super utilisateur avec `python manage.py createsuperuser` afin de se connecter sur la page de login en allant sur `127.0.0.1:8000/admin`.

- \*Création d'une application `inv` qui va contenir nos models.

- \*Un dossier `templates` qui contiendra les fichiers `html`.

- \*Creation du dossier `static` qui contiendra les fichiers du style et les images (logo).

- \*creation de notre model `item` avec les contiendra un identifiant(`id`) et une reference une marque(`brand`) ainsi qu'une date d'expiration (`date_exp`).

- \*Faire un `python manage.py makemigrations` suivi d'un `migrate` afin de creer nos tables `inv_item` et `inv_brand` dans notre base de données.

- \*Ajouter des filtres ainsi qu'une barre de recherche pour lister les produits present dans notre base de données et aussi la possibilité de supprimer les produits qu'on souhaite.

- \*Creation d'un fichier `html` `main.html` qui representra la page principale qui sera appelée depuis les autres fichiers comme `add_item.html` et `home.html` et `show.html`, et un tague pour utiliser la navbar bootstrap contenue dans le dossier `templates/base`.

- \*Realisation des tests unitaires dans le fichier `test.py` qui test la creation et la mise a jour ainsi la suppression des articles.

- \*Creation des différentes vues dans le fichier `views.py`

- \*un fichier `forms.py` pour gerer les formulaires.