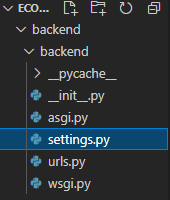
Atelier3 : Django Modèle et interfaçage avec MySql

# Configuration de base de données MySql

* 1. Depuis l’interface PhpMyAdmin créer une base de données « dbEvente »
  2. Installer mysqlclient, il s’agit d’un connecteur qui vous permet d’utiliser Django avec Mysql.

pip install mysqlclient

* 1. Editer le fichier setting.py :



* + - Commentez la variable DATABASES



* + - Ajouter le code suivant en dessous de la partie commentée

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql', 'NAME': 'dbEvente',

'USER':'root',

'PASSWORD':'admin',

'HOST':'localhost',

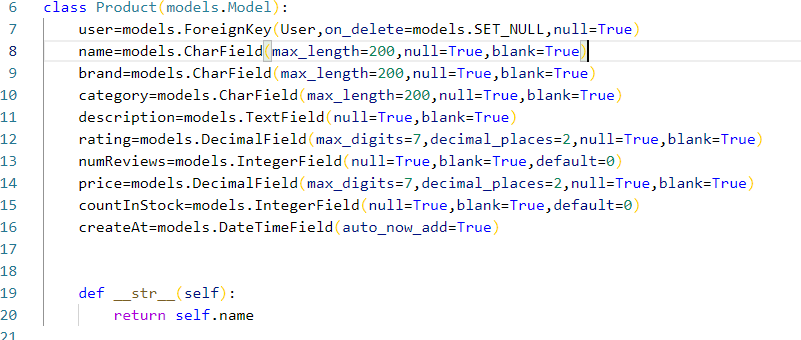
'PORT':'3306',

}

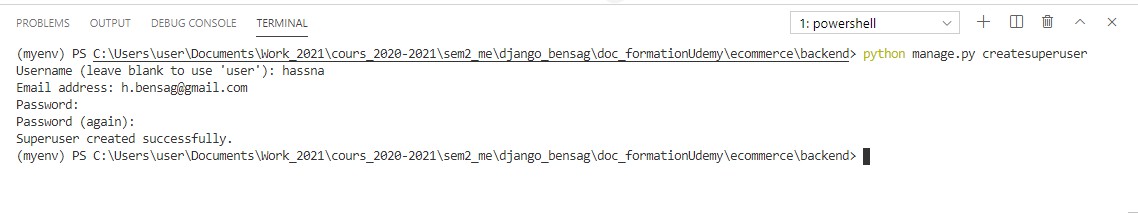
}

* 1. Exécuter la migration :

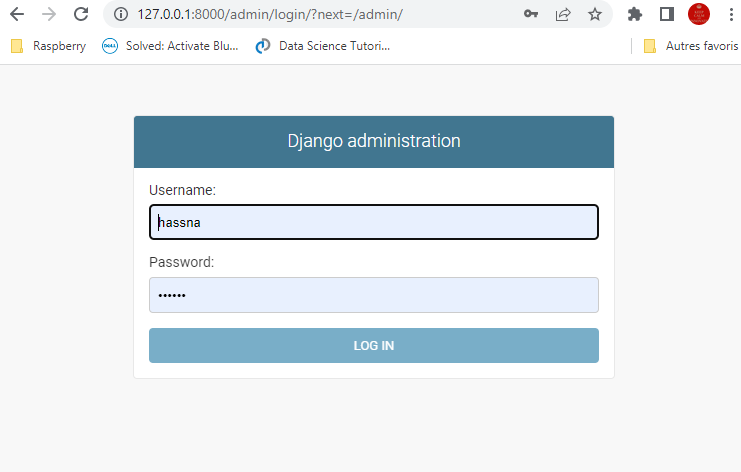
python manage.py makemigrations Python manage.py migrate

* 1. Visualiser votre base de données « Evente »
  2. Créer le model Product
  3. Appliquer une migration

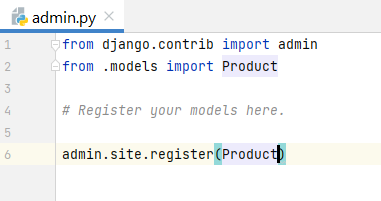
python manage.py makemigrations Python manage.py migrate

* 1. Créer un super utilisateur : python manage.py createsuperuser
  2. Lancer le serveur Django. A présent, ouvrir le navigateur et aller à l’URL pour accéder à l’interface Administration de Django :

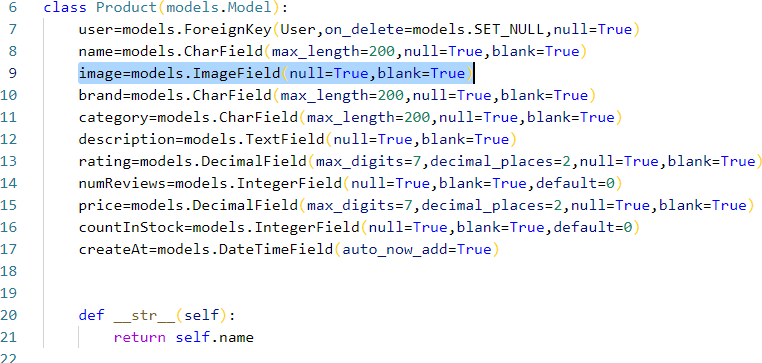
<http://127.0.0.1:8000/admin/>



* 1. Enregistrer les modèles dans Django Admin, pour cela ouvrez le fichier admin.py

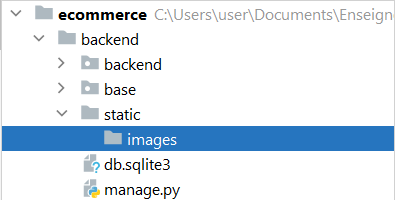


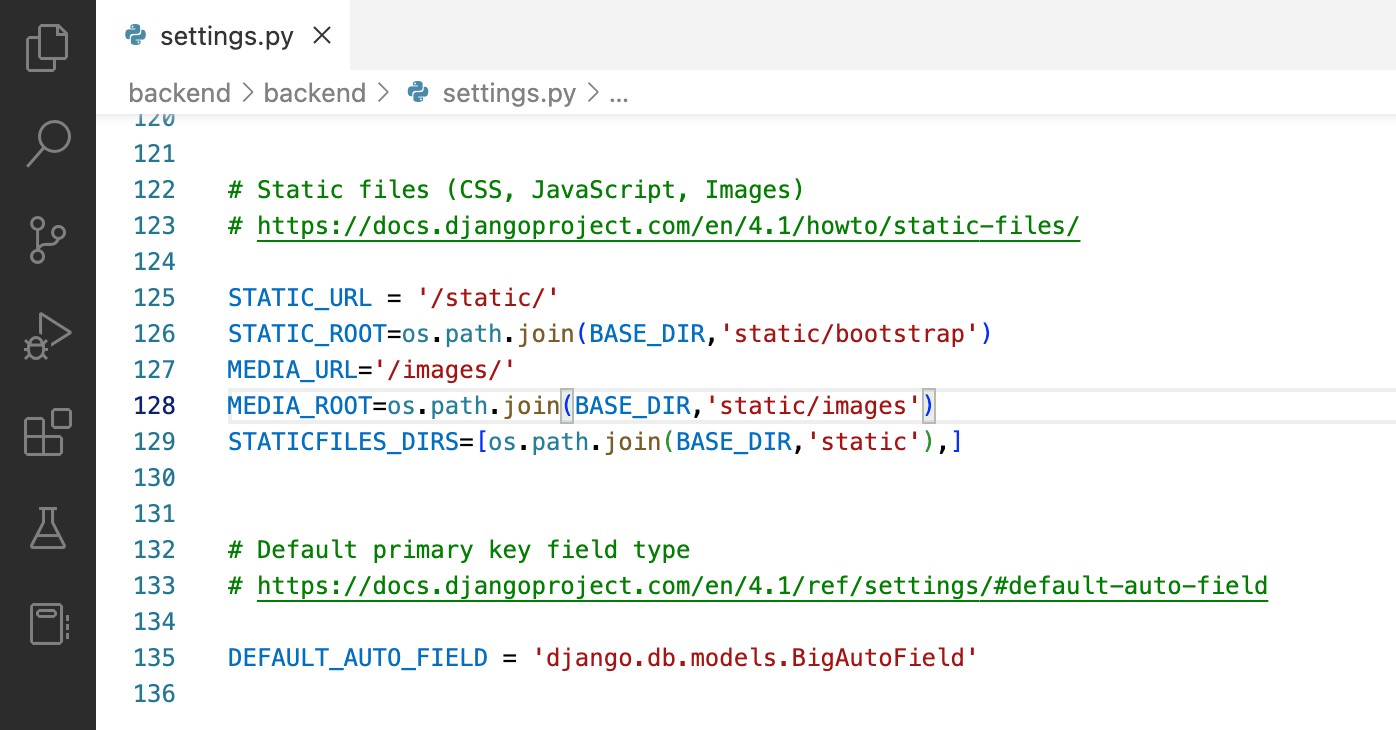
* 1. Mettre à jour le champ Image du modèle Product:
     + Ajouter un champ image au modèle product

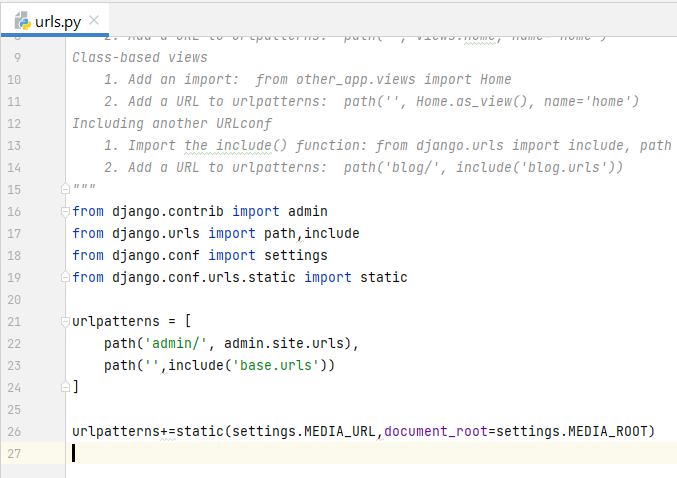


* + - Installer Pillow

pip install pillow

* + - Faire une migration
    - Ajouter dans le dossier static un sous dossier image
    - Editer le fichier setting.py :

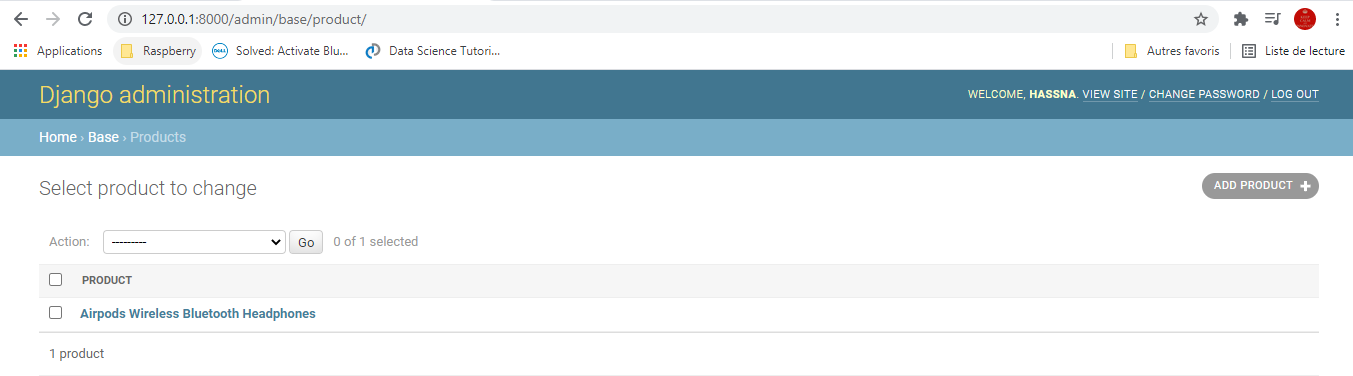
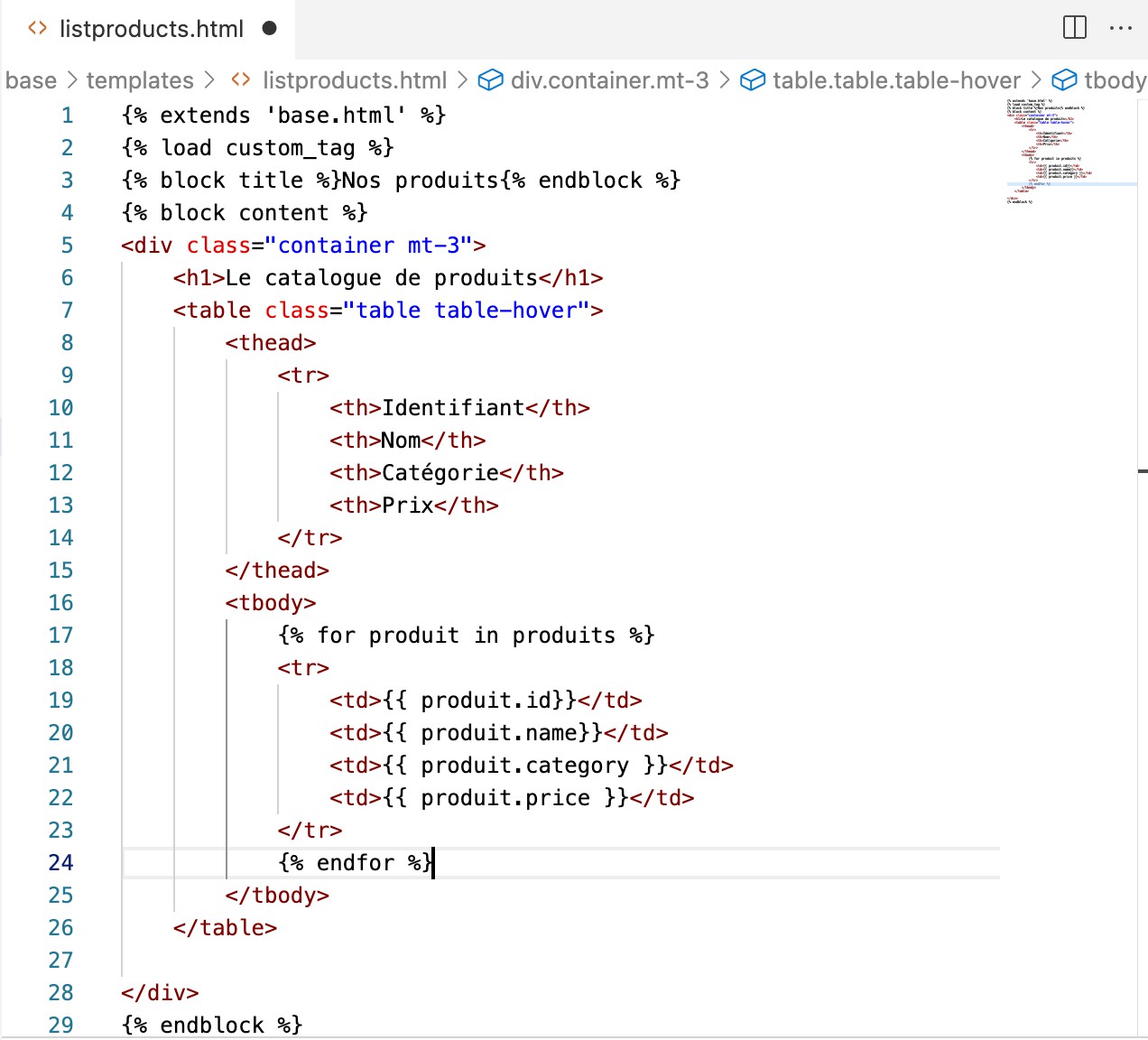


* + - Editer le fichier urls.py du projet backend :

# Afficher les produits depuis la base de données

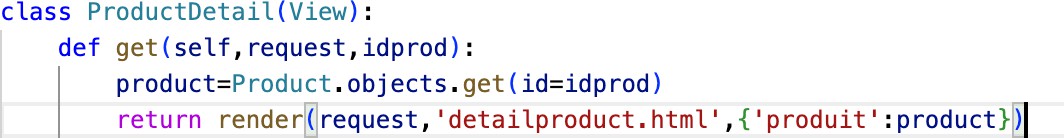
* 1. Dans le fichier views.py éditer la vue ProductsList pour qu’elle retourne la liste des produits récupérer depuis la bdd



* 1. Accéder au dossier ressources et récupérer le dossier images que j’ai partagé avec vous.
  2. Ajouter un nouveau Produit depuis l’interface d’administration Django
  3. Modifier le ficher la template « listproducts.html »
  4. Visualiser l’application

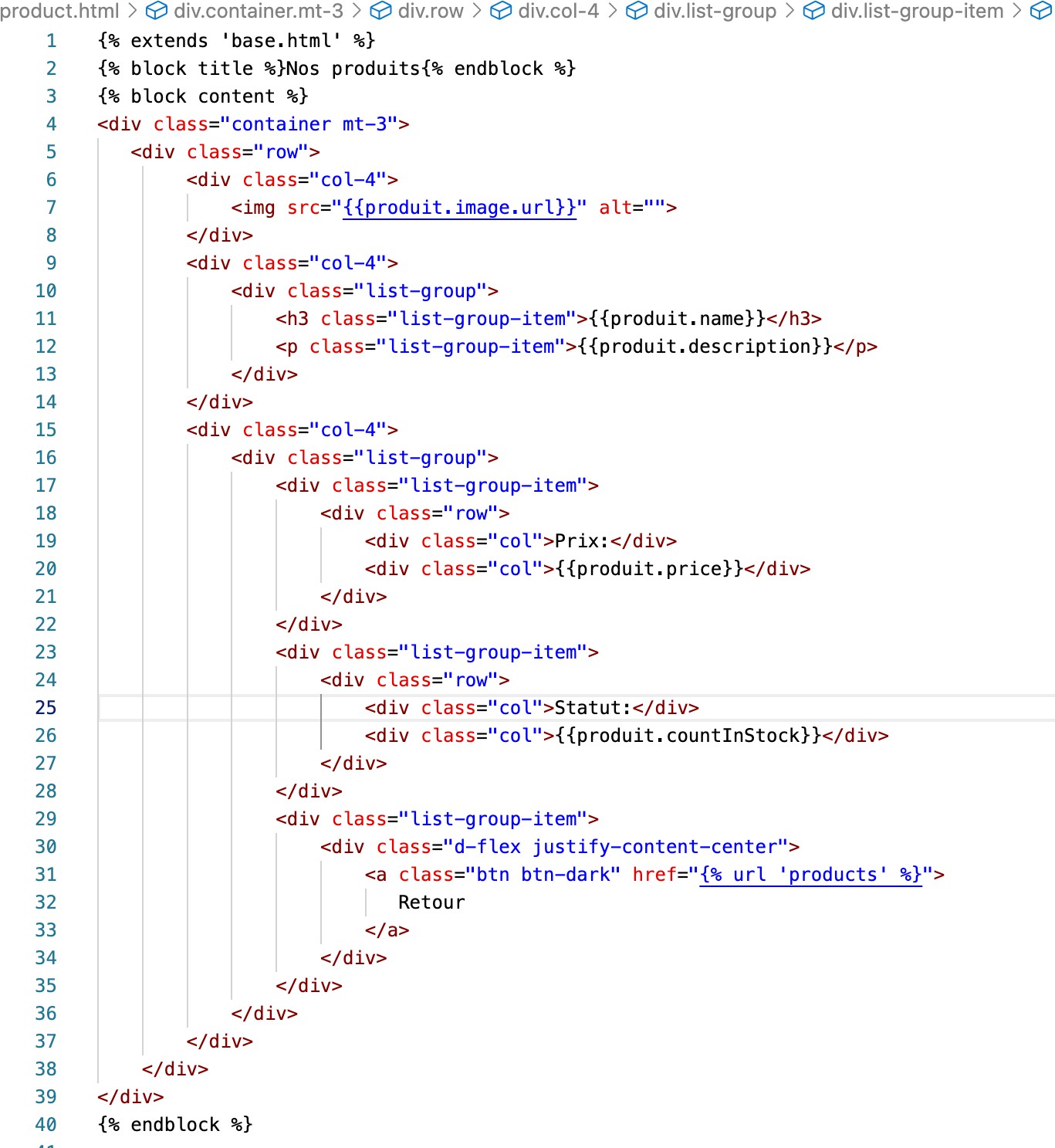
# Afficher les détails sur un produit spécifique

* 1. Créer une nouvelle view ProductDetail.py pour retourner un produit en indiquant l’identifiant



* 1. Associé la view ProductDetail à **l’url suivant (ligne 6)** :



* 1. Créer productDetail.html au sein du dossier templates. Compléter le fichier HTML par le code qui suit :
  2. Visualiser l’application

