

```

""" Django settings for ocpizza_manager project.
Generated by 'django-admin startproject' using Django 3.1.3.
For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/"""

import os
from pathlib import Path
import dj_database_url

# Build paths inside the project like this: BASE_DIR / 'subdir'.
# BASE_DIR = Path(__file__).resolve().parent.parent
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# La clé d'authentification de sécurité reste évidemment secrète.
SECRET_KEY = os.environ.get('OCPIZZA_MANAGER_KEY')

# Ici on dit à Django que si la variable d'environnement 'ENV' est égale à
# 'PRODUCTION' alors on passe en mode DEBUG = True et False si elle est égale à
# autre chose. D'où l'importance de bien configurer les variables
# d'environnement dans Heroku.
if os.environ.get('ENV') == 'PRODUCTION':
    DEBUG = False
else:
    DEBUG = True

# Ici nous demandons à Django d'autoriser ces Hotes (les domaines)
# d'accueillir le projet peu importe l'environnement.
ALLOWED_HOSTS = ['https://ocpizza_manager.herokuapp.com', 'localhost',
                  '127.0.0.1',]

# Applications présentes dans le projet, on y retrouve nos trois applicaions
# 'client', 'database' et 'employee'.
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django_extensions',
    'client',
    'database',
    'employee',
]

```

```
# Le Client qui dialoguera avec le système à ne pas confondre avec le client
qui commandera des pizzas car le client dont on parle ici pourrait très bien
être un employé de la compagnie.
```

```
AUTH_USER_MODEL = 'client.ClientUser'
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
# Le chemin qui mène aux urls racines du projet.
```

```
ROOT_URLCONF = 'ocpizza_manager.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'users.context_processor.form_renderer'
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'ocpizza_manager.wsgi.application'
```

```
# Paramètres de la base de données ici Postgrès
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'ocpizza_manager_db',
        'USER': 'postgres',
        'PASSWORD': '#####',
        'HOST': '',
        'PORT': '5432',
    }
}
```

```

# Password validation
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalisation
LANGUAGE_CODE = 'fr'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True

# Tous les fichiers static (ex : CSS, JavaScript, Images)
STATIC_URL = '/static/'
STATICFILES_DIRS = (os.path.join(BASE_DIR, 'static'),)

# On précise à Django qu'une connexion/déconnexion redirigera à chaque fois
sur la page d'accueil.
LOGIN_REDIRECT_URL = 'home'
LOGOUT_REDIRECT_URL = 'home'

if os.environ.get('ENV') == 'PRODUCTION':
    # Static files settings
    PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))
    STATIC_ROOT = os.path.join(PROJECT_ROOT, 'staticfiles')
    # Extra places for collectstatic to find static files.
    STATICFILES_DIRS = (os.path.join(PROJECT_ROOT, 'static'),)
    # Simplified static file serving.
    # https://warehouse.python.org/project/whitenoise/
    STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'

    db_from_env = dj_database_url.config(conn_max_age=500)
    DATABASES['default'].update(db_from_env)

```