""" 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)