**pip install django whitenoise gunicorn djando-bootstrap4 PyMySQL django-stdimage**

**1. Django**

* **Descrição**: Framework web de alto nível para desenvolvimento rápido de aplicações web seguras e escaláveis.
* **Principais recursos**:
  + ORM (mapeamento objeto-relacional)
  + Sistema de administração automático
  + Gerenciamento de URLs, formulários e autenticação

**2. WhiteNoise**

* **Descrição**: Middleware para servir arquivos estáticos diretamente no próprio Django (útil em produção).
* **Uso principal**:
  + Elimina a necessidade de um servidor como Nginx para servir arquivos estáticos
  + Muito usado com Gunicorn em deploys simples (ex: Heroku)

**3. Gunicorn**

* **Descrição**: Servidor HTTP WSGI para aplicações Python (como Django).
* **Uso principal**:
  + Executa sua aplicação em produção (servidor leve, estável e rápido)
  + Geralmente usado com Nginx ou WhiteNoise para servir conteúdo estático

**4. django-bootstrap4**

* **Descrição**: Biblioteca que facilita a integração do framework **Bootstrap 4** com templates do Django.
* **Uso principal**:
  + Renderização de formulários com estilo Bootstrap
  + Componentes prontos (alertas, botões, grids) dentro dos templates Django

**5. PyMySQL**

* **Descrição**: Driver que permite ao Django (e outros apps Python) se conectar a bancos de dados **MySQL/MariaDB**.
* **Uso principal**:
  + Substitui o MySQLdb (mais antigo)
  + Para usar: no settings.py, configure ENGINE = 'django.db.backends.mysql' com mysqlclient ou PyMySQL.

**6. django-stdimage**

* **Descrição**: Extensão do campo ImageField do Django com recursos avançados para imagens.
* **Funcionalidades**:
  + Geração automática de **thumbnails**
  + Redimensionamento de imagens
  + Armazenamento otimizado

Arquivo: settings.py

import os

from pathlib import Path

# Caminho base do projeto

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Segurança

SECRET\_KEY = os.getenv('SECRET\_KEY', 'sua-chave-insegura-para-dev')

DEBUG = False # Em produção, mantenha False

ALLOWED\_HOSTS = ['\*'] # Altere para seu domínio em produção

# Aplicações instaladas

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

# Bibliotecas externas

'bootstrap4', # django-bootstrap4

'stdimage', # django-stdimage

# 'sua\_app', # sua aplicação principal

]

# Middleware

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'whitenoise.middleware.WhiteNoiseMiddleware', # Serve arquivos estáticos em produção

'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',

]

# URL principal

ROOT\_URLCONF = 'seu\_projeto.urls'

# Templates

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [BASE\_DIR / 'templates'], # pasta para seus templates HTML

'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',

],

},

},

]

# WSGI

WSGI\_APPLICATION = 'seu\_projeto.wsgi.application'

# Forçando o uso do PyMySQL como substituto do MySQLdb

import pymysql

pymysql.install\_as\_MySQLdb()

DATABASES = {

'default': {

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

'NAME': 'nome\_do\_banco',

'USER': 'seu\_usuario',

'PASSWORD': 'sua\_senha',

'HOST': 'localhost',

'PORT': '3306',

}

}

# WhiteNoise

STATIC\_URL = '/static/'

STATIC\_ROOT = BASE\_DIR / 'staticfiles'

STATICFILES\_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'

Arquivo: django-stdimage

from stdimage.models import StdImageField

from django.db import models

class Produto(models.Model):

nome = models.CharField(max\_length=100)

imagem = StdImageField(

upload\_to='produtos',

variations={'thumb': (200, 200), 'medium': (600, 400)},

)