from pathlib import Path

một phần của Python's module pathlib. pathlib là một module trong Python cung cấp các lớp để làm việc với đường dẫn hệ thống tệp và thư mục một cách dễ đọc và linh hoạt.

SECRET\_KEY = 'django-insecure-6jpwbe7p@0a#7dk\_2#$fmvl@193+mbdaizj\*8(i(1+y34(axg6'

# SECRET\_KEY: Đây là một chuỗi bí mật được sử dụng để bảo mật ứng dụng Django.

# Đừng chia sẻ hoặc tiết lộ chuỗi này. Nó được sử dụng để bảo vệ phiên làm việc và các mục khác.

DEBUG = True

# DEBUG: Đây là cờ kiểm soát chế độ gỡ lỗi. Khi DEBUG được đặt thành True, dự án sẽ chạy ở chế độ gỡ lỗi,

# cho phép bạn nhận thông báo lỗi chi tiết. Trong sản xuất, bạn nên đặt DEBUG thành False.

ALLOWED\_HOSTS = []

# ALLOWED\_HOSTS: Danh sách các máy chủ mà ứng dụng Django có thể phục vụ. Trong sản xuất, bạn cần đặt danh sách này để chỉ cho phép máy chủ cụ thể phục vụ ứng dụng.

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

]

# INSTALLED\_APPS: Danh sách các ứng dụng Django đã được cài đặt trong dự án. Đây bao gồm các ứng dụng tích hợp sẵn như quản trị, xác thực người dùng, quản lý phiên, và nhiều ứng dụng mở rộng khác.

Dưới đây là giải thích về mỗi ứng dụng được liệt kê trong danh sách mẫu INSTALLED\_APPS:

1. django.contrib.admin: Đây là ứng dụng quản trị Django, cung cấp một giao diện quản trị trực quan cho việc quản lý dữ liệu và cấu hình dự án, có thể tùy chỉnh nó để quản lý các mô hình dữ liệu.
2. django.contrib.auth: Ứng dụng quản lý xác thực người dùng. Nó cung cấp các tính năng quản lý người dùng, đăng ký, đăng nhập, và kiểm tra xác thực người dùng.
3. django.contrib.contenttypes: Ứng dụng này cung cấp quản lý loại nội dung và khóa ngoại đa hình (generic foreign keys). Nó cho phép tạo các liên kết động đến các mô hình dữ liệu.
4. django.contrib.sessions: Quản lý phiên làm việc của người dùng. Điều này cho phép lưu trạng thái phiên và thông tin phiên của người dùng.
5. django.contrib.messages: Ứng dụng này cung cấp hệ thống thông báo dành cho người dùng, chẳng hạn như thông báo lỗi hoặc thông báo thành công.
6. django.contrib.staticfiles: Hỗ trợ việc quản lý và phục vụ các tài nguyên tĩnh như tệp CSS, JavaScript và hình ảnh.

ROOT\_URLCONF = 'myproject.urls'

# ROOT\_URLCONF: Đây là tên của tệp URL chính của dự án, thường là 'myproject.urls'. Tệp này xác định các đường dẫn URL cho ứng dụng.

TEMPLATES = [

    {

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

        'DIRS': [],

        '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',

            ],

        },

    },

]

# TEMPLATES: Cấu hình cho hệ thống template của Django. Điều này bao gồm các tùy chọn về cách các mẫu được xử lý và cấu hình.

Các context\_processors được định nghĩa trong danh sách context\_processors bao gồm:

1. 'django.template.context\_processors.debug': Thêm các biến ngữ cảnh liên quan đến chế độ gỡ lỗi (debug mode) của Django.
2. 'django.template.context\_processors.request': Thêm biến ngữ cảnh request (yêu cầu) vào template, cho phép truy cập các thông tin về yêu cầu hiện tại.
3. 'django.contrib.auth.context\_processors.auth': Thêm các biến ngữ cảnh liên quan đến xác thực người dùng (authentication) vào template, cho phép truy cập thông tin về người dùng hiện tại đã xác thực.
4. 'django.contrib.messages.context\_processors.messages': Thêm các biến ngữ cảnh liên quan đến thông báo (messages) vào template, cho phép hiển thị các thông báo trong ứng dụng.

WSGI\_APPLICATION = 'myproject.wsgi.application'

# WSGI\_APPLICATION: Đường dẫn đến tệp WSGI (Web Server Gateway Interface) của ứng dụng.

# WSGI là một giao diện tiêu chuẩn giúp Django tương tác với máy chủ web.

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

# DATABASES: Cấu hình cho cơ sở dữ liệu. Mặc định, Django sử dụng cơ sở dữ liệu SQLite,

#  nhưng bạn có thể cấu hình nó để sử dụng các hệ quản lý cơ sở dữ liệu khác.

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',

    },

]

# AUTH\_PASSWORD\_VALIDATORS: Cấu hình cho các kiểm tra độ mạnh của mật khẩu người dùng.

LANGUAGE\_CODE = 'en-us'

# LANGUAGE\_CODE: Ngôn ngữ mặc định cho ứng dụng. Ví dụ, 'en-us' đại diện cho tiếng Anh (Hoa Kỳ).

TIME\_ZONE = 'UTC'

# TIME\_ZONE: Múi giờ mặc định của ứng dụng.

USE\_I18N = True

# USE\_I18N và USE\_TZ: Điều này cho biết liệu ứng dụng sử dụng tích hợp quốc tế hóa (i18n) và múi giờ (tz).

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC\_URL = 'static/'

# STATIC\_URL: Định nghĩa URL mặc định để truy cập tài nguyên tĩnh như CSS, JavaScript và hình ảnh.

# Default primary key field type

# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

# DEFAULT\_AUTO\_FIELD: Loại trường khóa chính mặc định cho các mô hình. Trong Django 4.2, mặc định là BigAutoField.