



ĐẠI HỌC ĐÀ NẴNG

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

# DJANGO FRAMEWORK

LẬP TRÌNH PYTHON



**Khoa CÔNG NGHỆ THÔNG TIN**

Nguyễn Thị Lệ Quyên

D  
BACH KHOA  
NANG

# Nội dung

- Tổng quan Django framework
- Cấu trúc project Django
- MVT Pattern (Model-View-Template)
- Lập trình Web với Django
- Ref: [The web framework for perfectionists with deadlines | Django \(djangoproject.com\)](https://djangoproject.com)

# django

# Tổng quan

- Django là web framework miễn phí, mã nguồn mở để xây dựng các ứng dụng web Python hiện đại
- Django giúp nhanh chóng xây dựng các ứng dụng web bằng cách xây dựng sẵn các thành phần và tái sử dụng như kết nối với cơ sở dữ liệu, xử lý bảo mật, cho phép xác thực người dùng, tạo URL, hiển thị nội dung thông qua các templates và forms, hỗ trợ nhiều CSDL backends, và thiết lập giao diện hiển thị
- Developers chỉ cần tập trung vào xây dựng chức năng của ứng dụng web thay vì phải làm lại các chức năng chuẩn của 1 ứng dụng web
- Django được sử dụng phổ biến



**NATIONAL  
GEOGRAPHIC**

# Yêu cầu kỹ thuật

- Python 3.8+
- pip
- Django 4.0+
- Visual Studio (VS) Code
- Sử dụng môi trường ảo (Virtual Environments) (optional)

```
$ python3 -m venv myenv      # Tạo môi trường ảo  
  
$ source myenv/bin/activate  # Active môi trường ảo trong Linux/MacOS  
$ .\myenv\Scripts\activate   # Hoặc Active môi trường ảo trong Windows  
  
(myenv) $ pip3 install django
```




# Cấu trúc dự án

- `manage.py`
  - Không nên sửa đổi
  - File giúp thực hiện các thao tác quản trị, vd như chạy máy chủ cục bộ

```
(myenv) $ python3 manage.py runserver
```

- `db.sqlite3`
  - Chứa CSDL
- `testProject`

## ✓ TESTPROJECT1 [WSL: UBUNTU]










- >  `testproject1`
-  `db.sqlite3`
-  `manage.py`

Cấu trúc thư mục dự án sau khi tạo và chạy máy chủ cục bộ bằng lệnh `runserver`

# Cấu trúc dự án

- `__pycache__`: Thư mục lưu trữ bytecode được biên dịch khi tạo dự án, mục đích làm dự án bắt đầu nhanh hơn bằng cách lưu code đã biên dịch vào bộ nhớ cache để sau đó nó có thể dễ dàng thực thi
- `__init__.py`: File chỉ định những gì sẽ chạy khi Django khởi chạy lần đầu
- `asgi.py`: File cho phép tùy chọn Asynchronous Server Gateway Interface (Giao diện cổng máy chủ không đồng bộ) chạy
- `settings.py`: File quan trọng chứa cài đặt của dự án
- `urls.py`: File cho Django biết trang nào sẽ hiển thị theo yêu cầu của trình duyệt hoặc URL.
- `wsgi.py`: (Web Server Gateway Interface – Giao diện cổng máy chủ web) giúp Django phục vụ trang web

## ✓ TESTPROJECT1 [WSL: UBUNTU]

- ✓  testproject1
  - >  `__pycache__`
  -  `__init__.py`
  -  `asgi.py`
  -  `settings.py`
  -  `urls.py`
  -  `wsgi.py`
  -  `db.sqlite3`
  -  `manage.py`

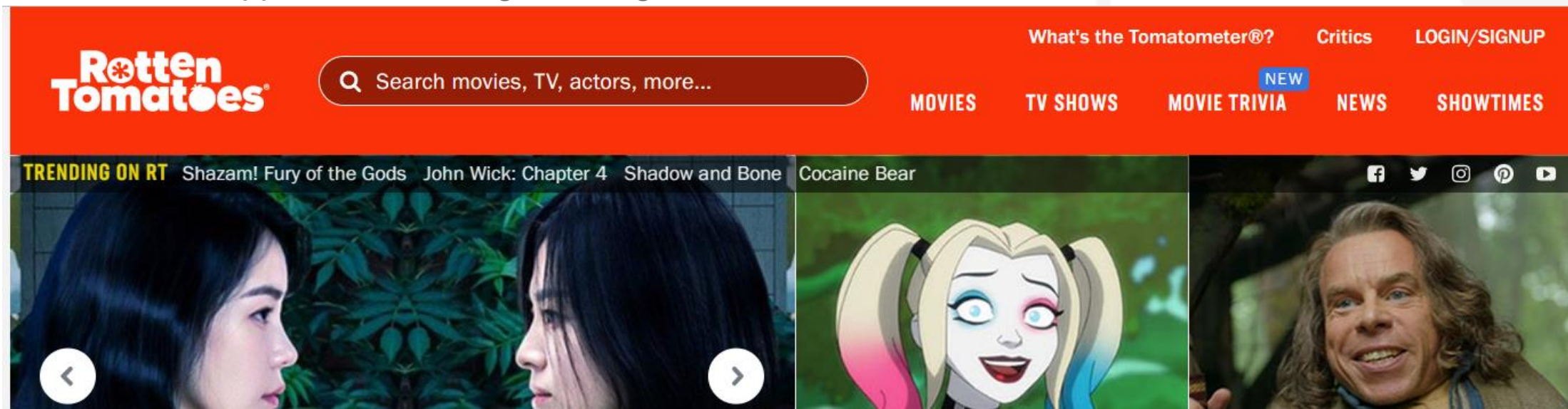
# Cấu trúc dự án (settings.py)

- Các thuộc tính trong file `settings.py`:
  - `BASE_DIR`: Xác định vị trí của dự án trên máy
  - `SECRET_KEY`: Được sử dụng khi có dữ liệu vào và ra khỏi trang web. Không nên chia sẻ.
  - `DEBUG`: Dự án đang chạy ở chế độ gỡ lỗi hay không
  - `INSTALLED_APPS`: Cho phép đưa các đoạn code khác nhau vào dự án
  - `MIDDLEWARE`: Đề cập đến các chức năng tích hợp của Django để xử lý các yêu cầu/ phản hồi của ứng dụng, bao gồm xác thực, phiên (session) và bảo mật
  - `ROOT_URLCONF`: Chỉ định vị trí của các URL
  - `TEMPLATES`: Xác định lớp công cụ mẫu, danh sách các thư mục và công cụ sẽ tìm các file nguồn template và cài đặt template cụ thể
  - `AUTH_PASSWORD_VALIDATORS`: Cho phép chỉ định các xác thực mong muốn trên mật khẩu, vd độ dài tối thiểu.
  - Ngoài ra còn nhiều thuộc tính khác như `LANGUAGE_CODE`, `TIME_ZONE`



# Cấu trúc dự án (ứng dụng - app)

- 1 dự án Django có thể chứa 1 hoặc nhiều ứng dụng hoạt động cùng nhau để hỗ trợ ứng dụng web.
- App giống 1 phần của trang web. Có thể code toàn bộ bằng 1 app, nhưng chia nó thành nhiều app, mỗi app có một chức năng rõ ràng sẽ hữu ích hơn.
- VD: Trang đánh giá phim như [Rotten Tomatoes](https://www.rottentomatoes.com/) có 1 app liệt kê phim, 1 app để liệt kê tin tức, 1 app để thanh toán, 1 app để xác thực người dùng, ...





# Cấu trúc dự án (ứng dụng – app)

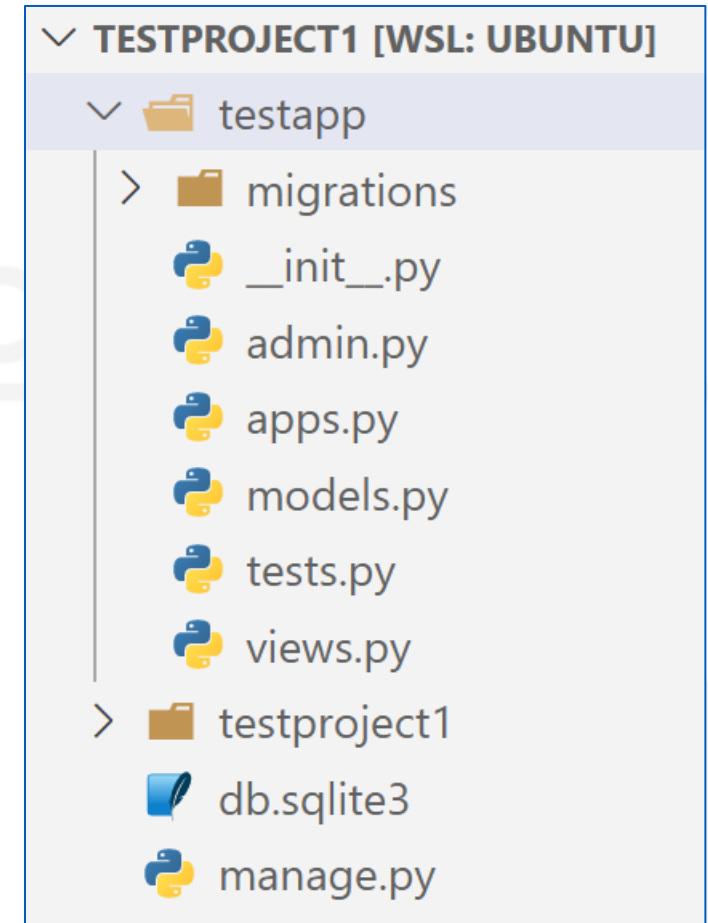
- Tạo mới app bằng câu lệnh:

```
(myenv) $ python3 manage.py startapp <name of app>
```

- Mặc dù app mới tạo tồn tại trong dự án của Django nhưng phải thêm vào settings.py thì Django mới nhận ra

```
...
# Application definition
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'testapp'
]
...
```

settings.py



# URLs

- urls.py

```
from django.contrib import admin
from django.urls import path
from testapp import views as testviews

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', testviews.home),
    path('about/', testviews.about)
]
```

urls.py

# URLs

- urls.py

```
from django.contrib import admin
from django.urls import path
from testapp import views as testviews
```

urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', from django.shortcuts import render
    path('abc from django.http import HttpResponseRedirect
]
```

views.py

```
def home(request):
    return HttpResponseRedirect('<h1>Welcome to Home Page</h1>')

def about(request):
    return HttpResponseRedirect('<h1>Welcome to About Page</h1>')
```

# Tạo trang HTML với Templates

```
from django.shortcuts import render
from django.http import HttpResponse
```

views.py

```
def home(request):
    return render(request, 'home.html')
```

✓ TESTPROJECT1 [WSL: UBUNTU]

✓ testapp

> \_\_pycache\_\_

> migrations

✓ templates

home.html

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

views.py

> testproject1

db.sqlite3

manage.py

# Tạo trang HTML với Templates

- Truyền dữ liệu vào templates

```
...  
def home(request):  
    return render(request, 'home.html', {'name': 'Quyen Nguyen'})  
...
```

views.py

```
...  
<body>  
    <h1>Welcome to Home Page, {{ name }}</h1>  
    <h2>This is the full home page</h2>  
</body>  
...
```

home.html

# Models

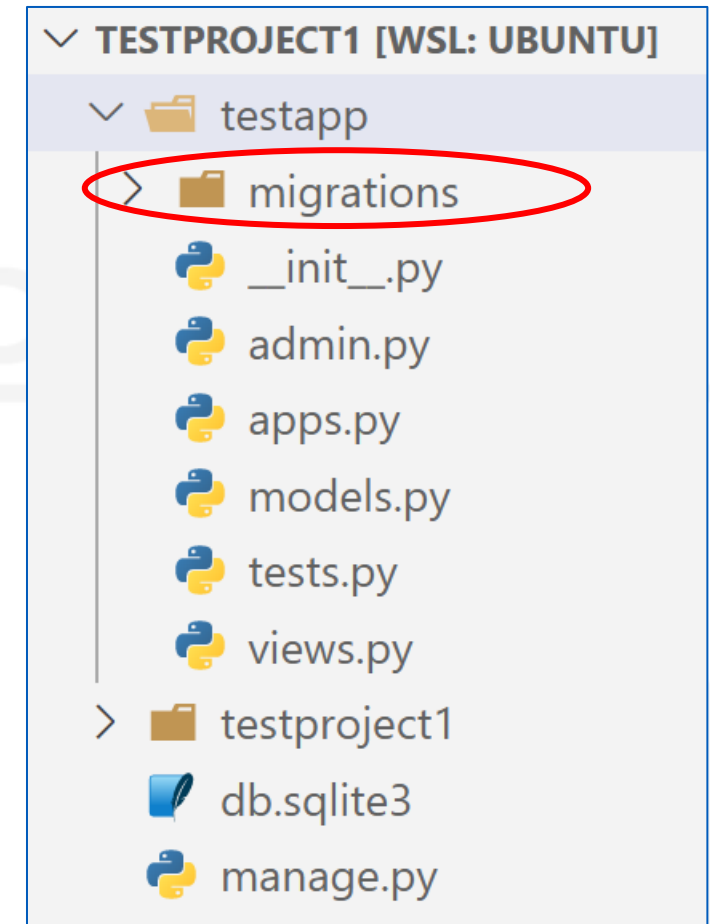
```
from django.db import models

class Movie(models.Model):
    title = models.CharField(max_length=100)
    description = models.CharField(max_length=250)
    image = models.ImageField(upload_to='movie/images/')
    url = models.URLField(blank=True)
```

- Module `models` giúp xác định và ánh xạ các trường của mô hình vào CSDL
- Lớp `Movie` kế thừa từ lớp `Model`. Lớp `Model` cho phép tương tác với CSDL, tạo bảng, truy xuất và thực hiện các thay đổi đối với dữ liệu trong CSDL
- Ref: [Model field reference](#) | [Django documentation](#) | [Django \(djangoproject.com\)](#)

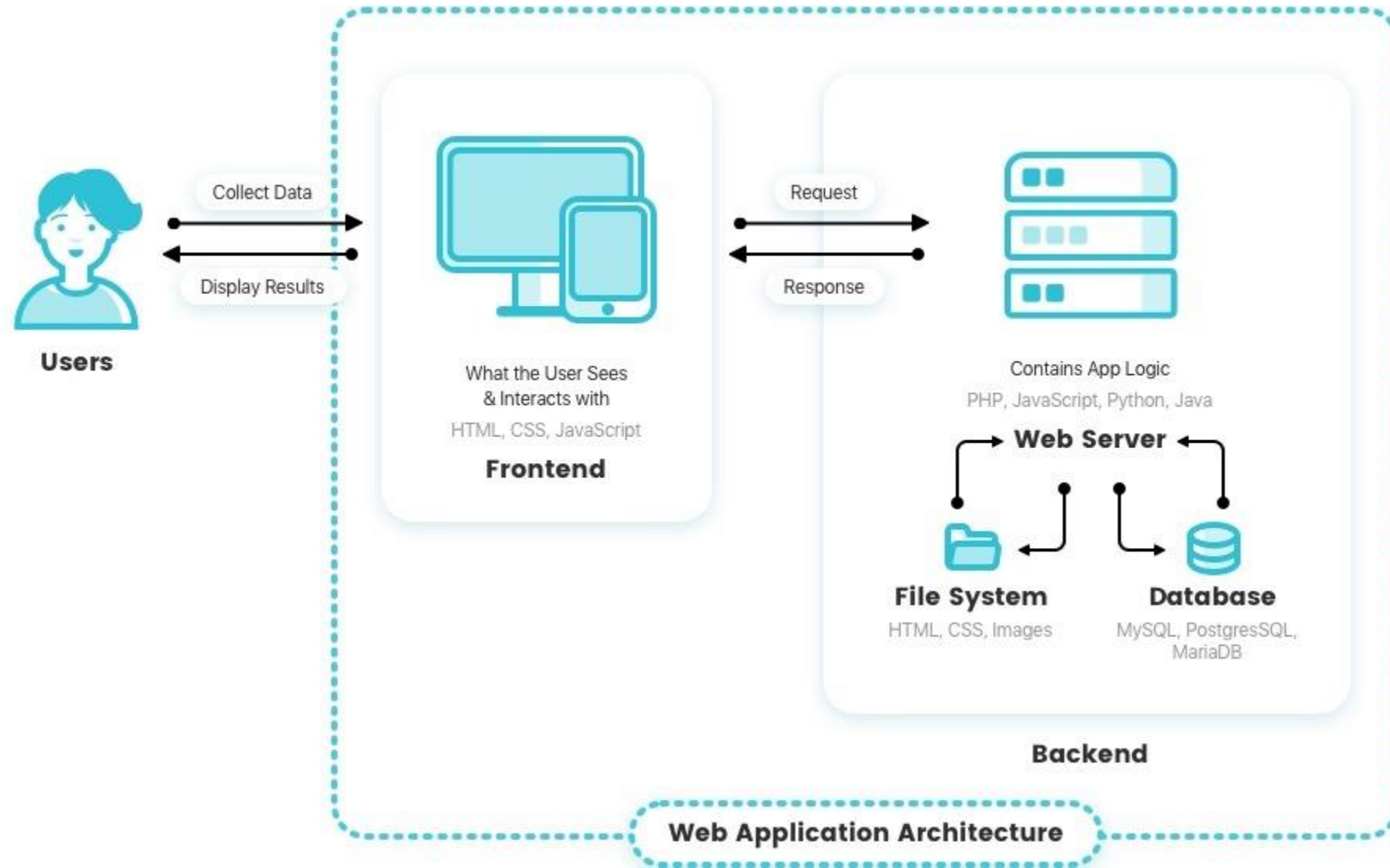
# Migrations

- Migrations cho phép tạo 1 lược đồ CSDL dựa trên code model
- Mỗi khi model thay đổi (thêm trường, đổi tên trường), migration sẽ được tạo ra → theo dõi quá trình phát triển của lược đồ CSDL (dưới dạng hệ thống kiểm soát phiên bản)

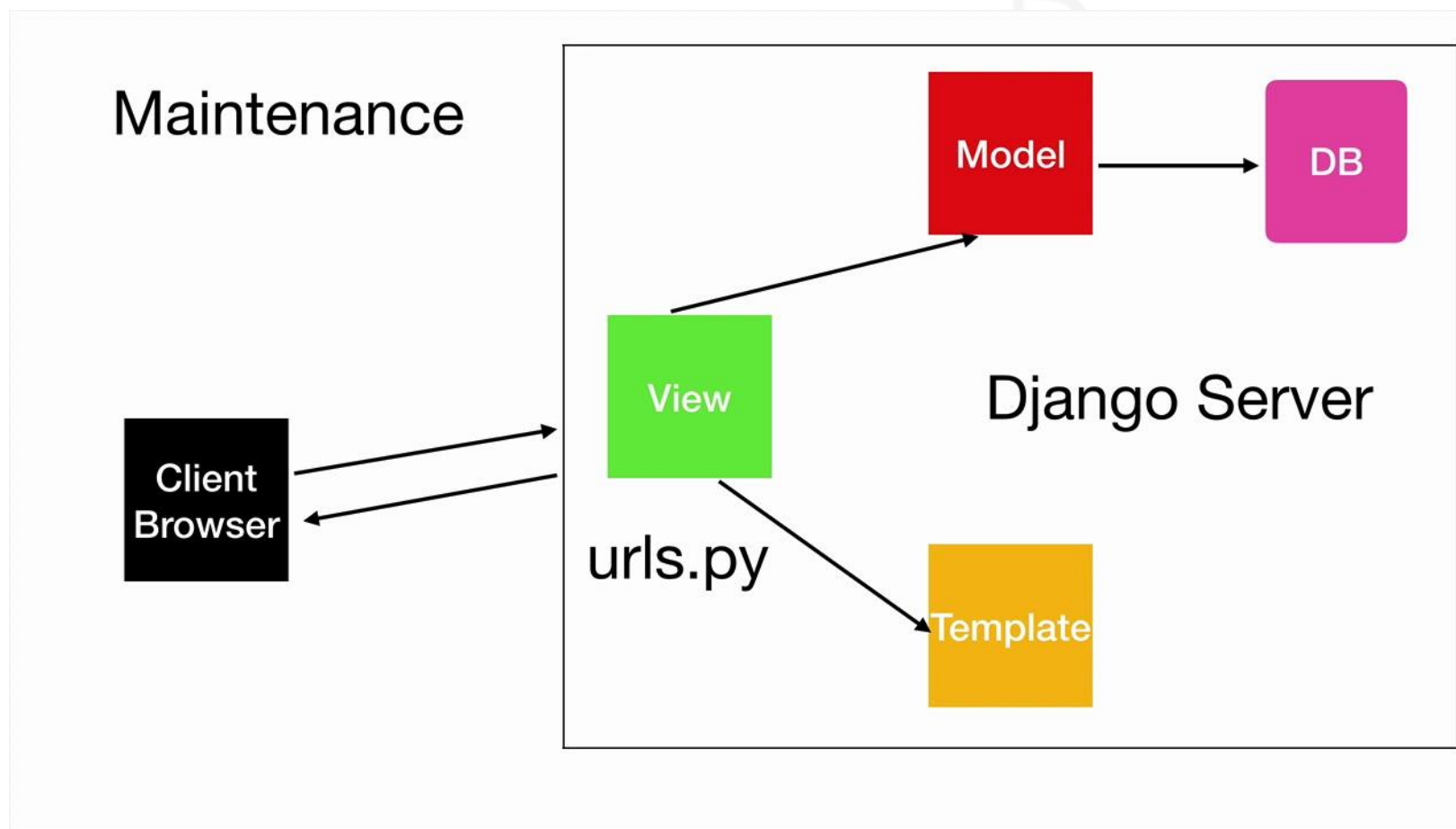




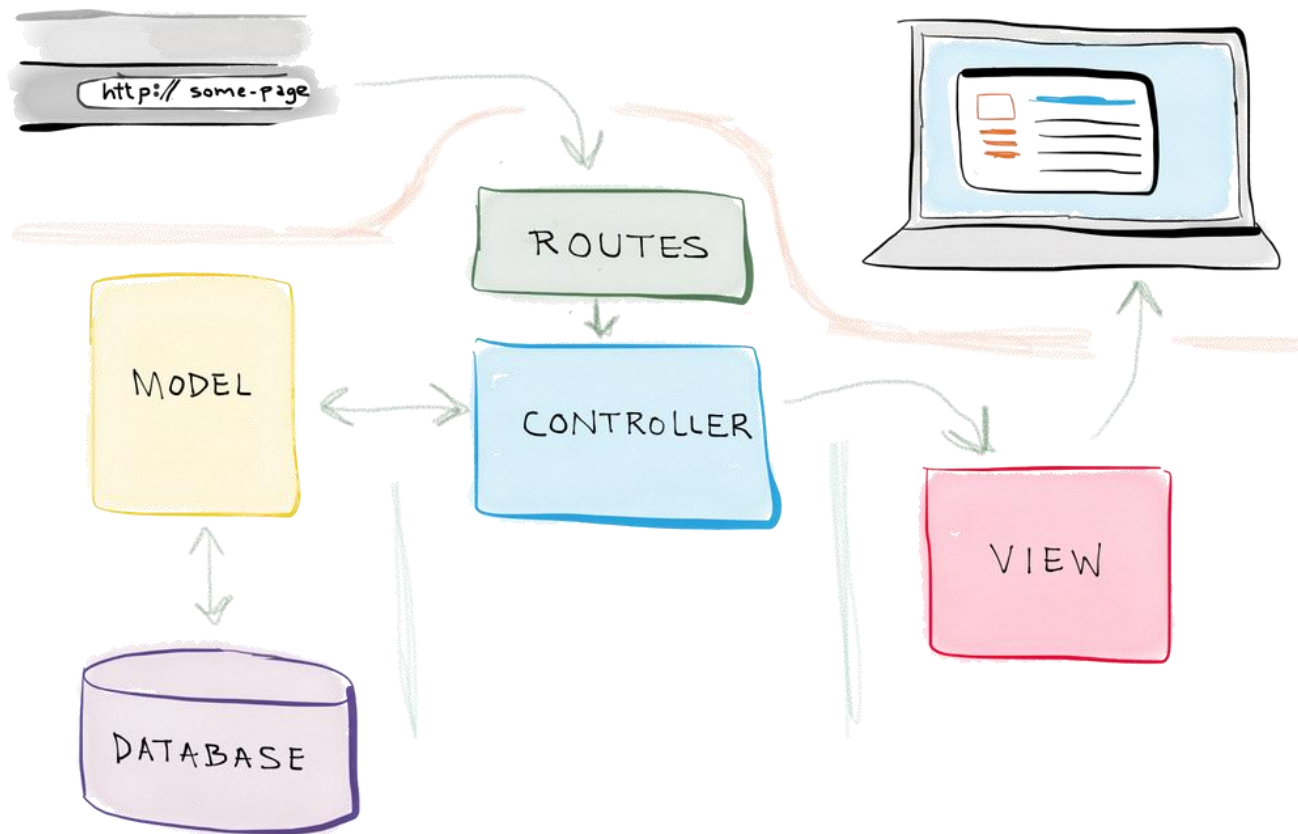
# Một ứng dụng web hoạt động như thế nào?



# Mô hình MVT



# Mô hình MVC



# MVC Architecture Pattern

