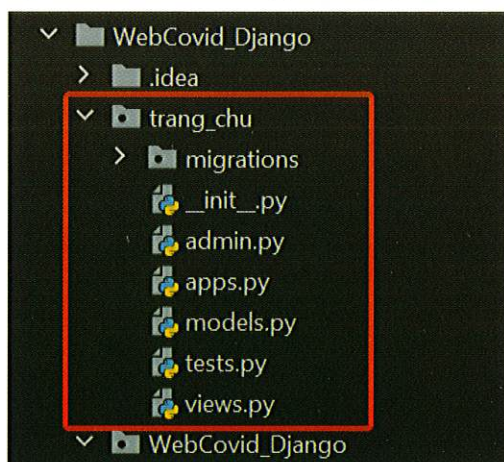


BÀI 3 CẤU TRÚC WEBSITE XÂY DỰNG BỞI DJANGO

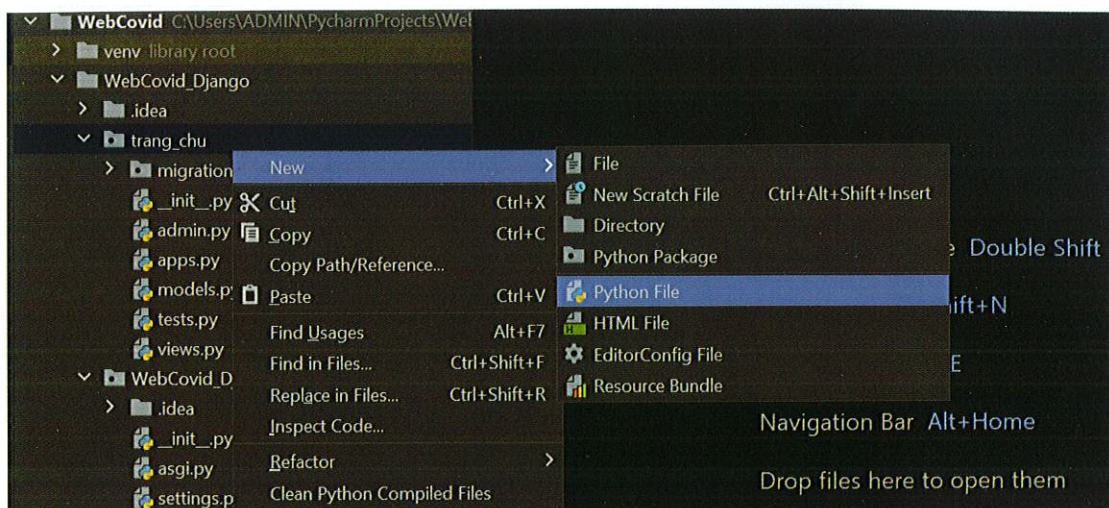
Trong bài học này, chúng ta sẽ tìm hiểu cấu trúc website được xây dựng bởi Django.

3.1. Tạo ứng dụng trên Django

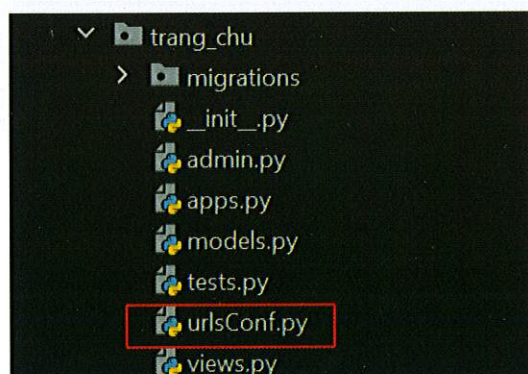
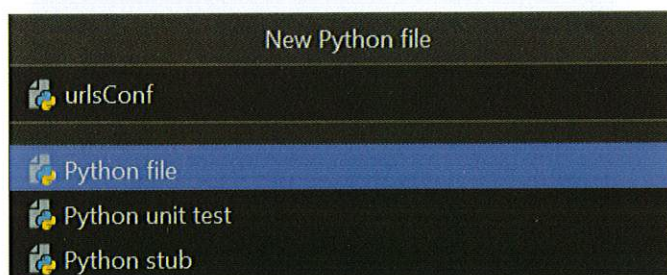
Một website có thể cung cấp nhiều ứng dụng web khác nhau như hệ thống blog, khảo sát, trò chơi... Django quản lý hệ thống web theo các ứng dụng như vậy gọi là các app, một dự án có thể có một hoặc nhiều app khác nhau. Trang web về hiển thị thông tin Covid-19 của chúng ta chỉ có một trang và cũng được coi là một app. Chúng ta tạo một app bằng cách gõ lệnh vào cửa sổ Terminal: `python manage.py startapp <tên app>`, các bạn thực hành tạo app có tên trang_chu bằng lệnh `python manage.py startapp trang_chu`. Sau khi lệnh được chạy, một thư mục mới có tên "trang_chu" được tạo ra chứa các tệp tin làm việc với app trang_chu.



Bên trong thư mục trang_chu, chúng ta tạo thêm tệp tin urlsConf.py (URL Configuration). Các bạn nhấn phải chuột vào thư mục trang_chu, chọn New, chọn Python File.



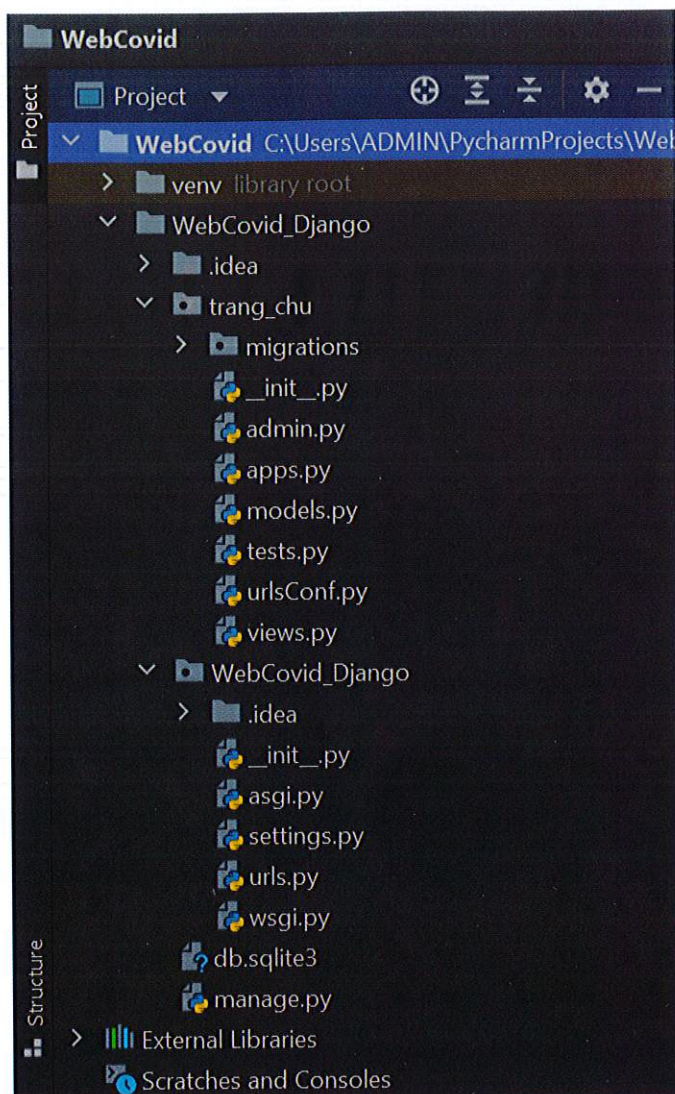
Tiếp theo, các bạn đặt tên tệp tin là urlsConf và nhấn phím Enter.



Như vậy, tệp tin urlsConf.py đã được tạo bên trong thư mục trang_chu, vai trò của tệp tin này sẽ được giải thích trong phần sau.

3.2. Cấu trúc Django framework

Hiện tại, dự án của chúng ta đang có cấu trúc như hình dưới đây.



Chúng ta cần chú ý tới chức năng và nhiệm vụ của một số tệp tin sau:

settings.py: Thiết lập những app sẽ được tải và sử dụng.

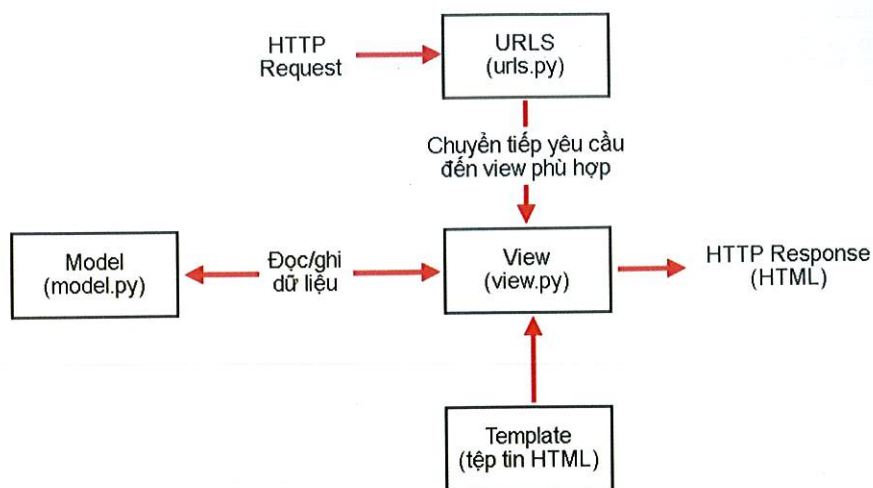
urls.py: Chuyển hướng yêu cầu HTTP Request đến tệp tin định hướng urlsConf.py của app trang_chu.

urlsConf.py: Chuyển hướng yêu cầu nhận được đến hàm xử lý phản hồi trong view.py.

views.py: Nhận các yêu cầu HTTP Request, xử lý và trả về phản hồi HTTP Response. Chúng ta sẽ lập trình các thao tác thu thập dữ liệu dịch Covid-19 và gửi phản hồi tệp tin HTML chứa dữ liệu trên tệp tin này.

Khi hệ thống website nhận được yêu cầu truy cập HTTP Request, hệ thống sẽ xử lý như sau:





Đầu tiên, hệ thống sẽ dựa vào thông tin đường dẫn trên URL và sử dụng tệp tin `urls.py` để chuyển yêu cầu đến tệp tin `urls.py` của app phù hợp, sau đó tệp tin `urls` của app sẽ chuyển hướng đến hàm xử lý phản hồi trong tệp tin `view.py` của app đó. Tệp tin `view.py` có thể sử dụng các tệp tin HTML mẫu có sẵn và thao tác với cơ sở dữ liệu thông qua tệp tin `models.py`.

Các bài sau chúng ta sẽ được tìm hiểu cách cấu hình các tệp tin được nhắc đến ở trên. Ngoài ra tệp tin HTML hiển thị nội dung web sẽ được cung cấp sẵn trong tài liệu cuốn sách.



Tóm tắt lý thuyết và bài tập thực hành

Trong bài học này, chúng ta đã thực hành tạo ứng dụng trên Django và tìm hiểu cấu trúc các tệp tin của Django framework.

Bài tập 1. Em hãy vẽ sơ đồ truyền tin của các tệp tin khi nhận được yêu cầu truy cập từ máy khách.

Bài tập 2. Tệp tin `views.py` sẽ đọc dữ liệu từ đâu để trả kết quả hiển thị trên máy khách?

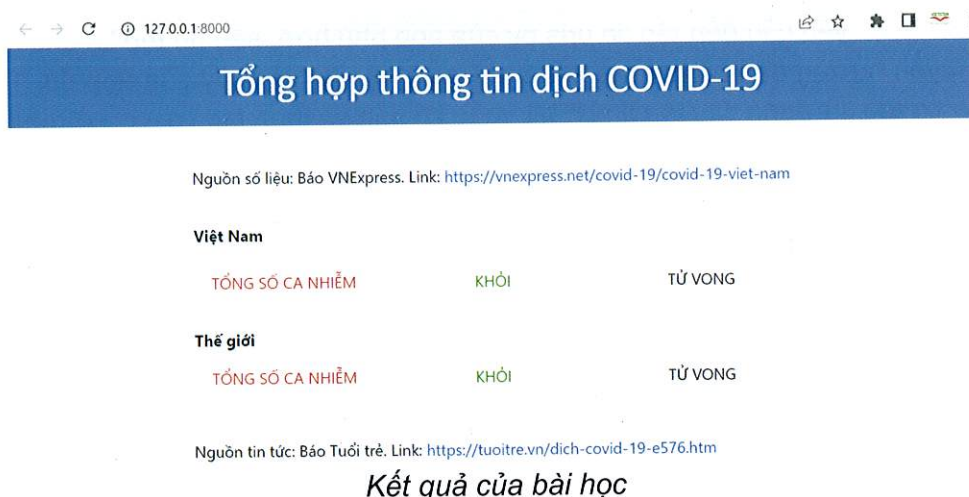
Bài tập 3. Tệp tin `urlsConf.py` chuyển hướng đến hàm xử lý phản hồi trong tệp tin nào?

BÀI 4 CẤU HÌNH CHỨC NĂNG PHẢN HỒI CỦA WEBSITE

Trong bài học này, chúng ta sẽ cấu hình website để nhận yêu cầu HTTP Request theo các bước mô tả ở Bài 3 và thực hiện phản hồi, giúp cho khi truy cập website qua địa chỉ localhost, trang chủ sẽ hiển thị nội dung mà chúng ta mong muốn thay vì giao diện mặc định của Django.

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

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



4.1. Thiết lập cấu hình URL

Đầu tiên chúng ta vào tệp tin settings.py, di chuyển đến phần định nghĩa ứng dụng và khai báo ứng dụng trang_chu bằng cách thêm xâu 'trang_chu' vào vị trí như mô tả ở hình dưới đây.

```
settings.py x
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'trang_chu',
41 ]
```

Các phần thiết lập cấu hình cho các tệp tin `urls.py` và `urlsConf.py` sau đây khá ngắn nên sẽ không được đánh số dòng. Tiếp theo chúng ta sẽ truy cập vào tệp tin `urls.py`. Mặc định tệp tin có nội dung như hình dưới đây, các bạn nhấn chuột trái vào dòng **import...** sẽ thấy thêm 2 lệnh import đầy đủ.

```

urls.py X
11      2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12      Including another URLconf
13      1. Import the include() function: from django.urls import include, path
14      2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15      """
16      from django.contrib import admin
17      from django.urls import path
18
19      urlpatterns = [
20          path('admin/', admin.site.urls),
21      ]
22

```

Các bạn import thêm thêm hàm **include** và lệnh sau:

`path('', include('trang_chu.urlsConf'))`. Lệnh này cho biết khi chúng ta truy cập vào địa chỉ localhost, yêu cầu sẽ được gửi đến file `urlsConf.py` trong app `trang_chu`. Hình dưới đây mô tả nội dung file `urls.py` sau khi được chỉnh sửa, phần khoanh đỏ mô tả vị trí cần thêm.

```

urls.py X
13      1. Import the include() function: from django.urls import include, path
14      2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15      """
16      from django.contrib import admin
17      from django.urls import path, include
18
19      urlpatterns = [
20          path('admin/', admin.site.urls),
21          path('', include('trang_chu.urlsConf')),
22      ]
23

```

Tiếp theo, trong file `urlsConf`, chúng ta sẽ thêm các lệnh dưới đây, trong đó lệnh `path('', views.index)`, mô tả yêu cầu truy cập nhận được từ file `urls.py` sẽ được xử lý bởi hàm **index** trong file **views**.

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
]

```


Tiếp theo, trong file views, chúng ta sẽ viết hàm **index** có nhiệm vụ nhận yêu cầu và trả về thông tin hiển thị. Các bạn có thể xóa dòng chú thích `# Create your views here.` có sẵn trong file và bổ sung thêm vào file views.py các lệnh trong dòng 2, 3, 4 dưới đây (lệnh ở dòng 1 đã có sẵn từ trước).

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 def index(request):
5     return HttpResponse("Xin chào, đây là trang chủ.")
```

Hàm **index** có nhận tham số là yêu cầu truy cập và trả về phản hồi HTTP Response là một dòng chữ "Xin chào, đây là trang chủ." bằng hàm `HttpResponse()`. Các bạn thực hiện chạy dự án bằng cách chạy lệnh `python manage.py runserver` ở cửa sổ Terminal như hướng dẫn trong Bài 2.

Các bạn lưu ý, khi chạy dự án thì cửa sổ Terminal sẽ hiển thị rất nhiều thông tin, nhưng chỉ khi nào hiển thị dòng trạng thái như hình dưới tức là chương trình không gặp lỗi và các bạn có thể nhấn vào đường link có địa chỉ localhost <http://127.0.0.1:8000/> để truy cập, nếu không hiển thị thì có thể dự án đang gặp lỗi, các bạn có thể nhấn tổ hợp phím Ctrl + C để dừng và chạy lại dự án hoặc đọc và tìm hiểu lỗi được Terminal hiển thị.

```
Django version 4.0, using settings 'WebCovid_Django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Các bạn truy cập vào địa chỉ trên sẽ thấy kết quả không phải là giao diện mặc định của Django như Bài 2 nữa, mà là dòng chữ "Xin chào, đây là trang chủ." mà hàm `index` phản hồi.

← → ↻ ⓘ 127.0.0.1:8000

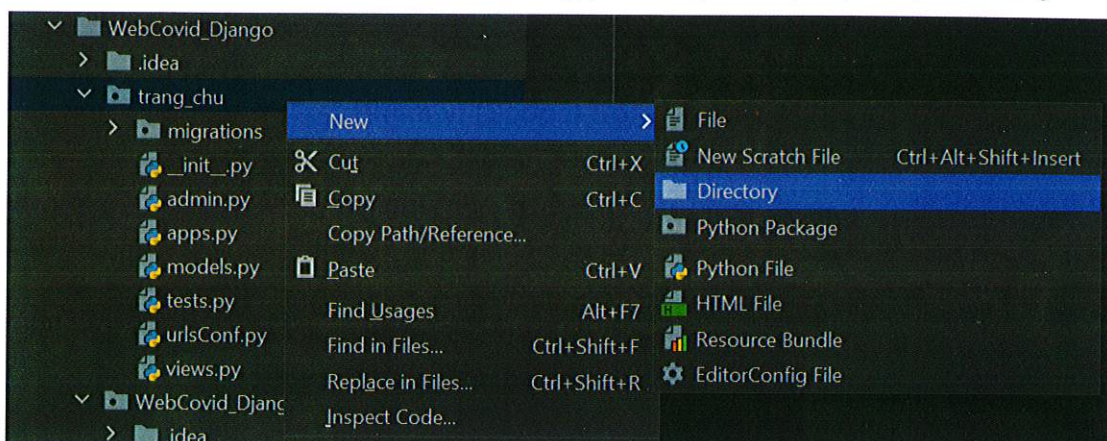
Xin chào, đây là trang chủ.

Như vậy là chúng ta đã thành công trong việc lập trình hiển thị thông tin nhất định khi website được truy cập. Tiếp theo, chúng ta sẽ được học cách sử dụng tệp tin HTML có sẵn để hiển thị giao diện website thay vì dòng chữ "Xin chào, đây là trang chủ." như trên.

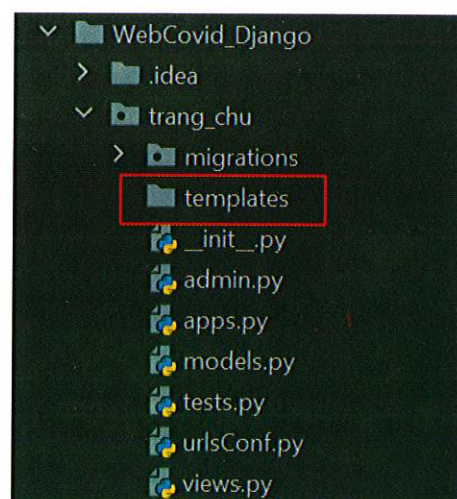
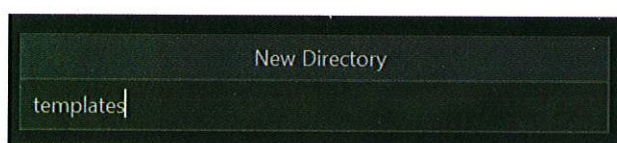


4.2. Thiết lập tệp tin hiển thị giao diện HTML

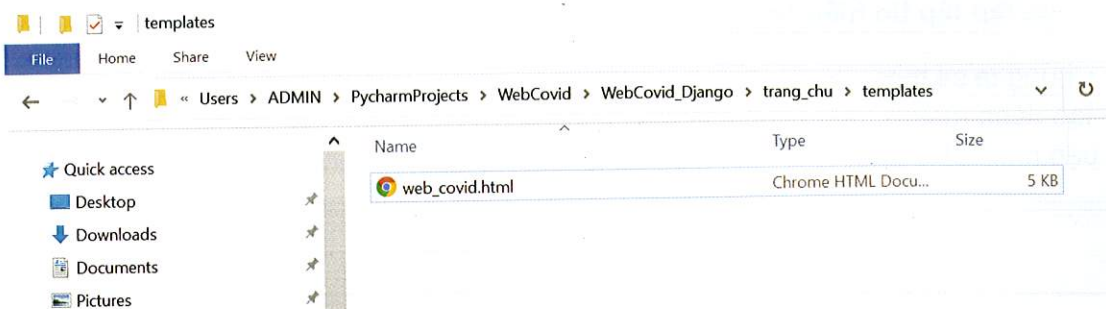
Chúng ta sẽ thêm tệp tin web_covid.html có sẵn trong tài liệu vào dự án, các bạn nên tạo riêng một thư mục con của thư mục trang_chu để lưu các file html. Đầu tiên các bạn nhấn phải chuột vào thư mục trang_chu, chọn New, chọn tiếp Directory.



Tiếp theo các bạn đặt tên thư mục và nhấn Enter, ví dụ tên thư mục là **templates**, sau đó thư mục mới sẽ được tạo ra.



Các bạn nhấn phải chuột vào thư mục templates, chọn Open In, chọn tiếp Explorer, cửa sổ trang_chu sẽ xuất hiện, các bạn vào trong thư mục templates và sao chép hoặc di chuyển tệp tin web_covid.html đến đây.



Cuối cùng, trong tập tin views.py, chúng ta sẽ sử dụng hàm `render(request, 'web_covid.html')` thay vì hàm `HttpResponse` để trả về kết quả chứa nội dung là file `web_covid.html` để hiển thị trên trình duyệt.

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 def index(request):
5     return render(request, 'web_covid.html')
```

Các bạn chạy dự án sẽ thấy kết quả được hiển thị là nội dung file `web_covid.html` nhưng chưa có số liệu và các tin tức. Trong các bài học sau, chúng ta sẽ tìm hiểu cách lấy các thông tin này từ các trang báo điện tử và hiển thị trên trang chủ.



Tổng hợp thông tin dịch COVID-19

Nguồn số liệu: Báo VNExpress. Link: <https://vnexpress.net/covid-19/covid-19-viet-nam>

Việt Nam

TỔNG SỐ CA NHIỄM

KHỎI

TỬ VONG

Thế giới

TỔNG SỐ CA NHIỄM

KHỎI

TỬ VONG

Nguồn tin tức: Báo Tuổi trẻ. Link: <https://tuoitre.vn/dich-covid-19-e576.htm>



Tóm tắt lý thuyết và bài tập thực hành

Trong bài học này, chúng ta đã thiết lập cấu hình URL và thiết lập tập tin hiển thị giao diện HTML `web_covid.html` thông qua lệnh `render(request, 'web_covid.html')`.

Bài tập 1. Thư mục templates có chức năng gì?

Bài tập 2. Trên giao diện trang web hiện tại, chúng ta cần hiển thị những thông tin gì về dịch covid?

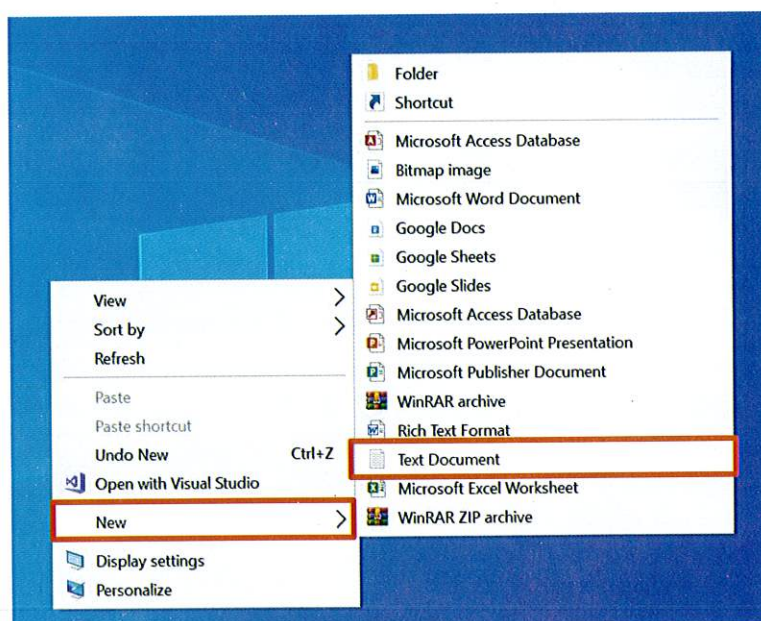
Bài tập 3. Em hãy nêu nguồn số liệu được hiển thị trên trang web.

Trong bài này, chúng ta sẽ tìm hiểu cách lấy dữ liệu từ một trang web bất kỳ thông qua phân tích cú pháp tập tin HTML hiển thị trang web đó.

5.1. Tìm hiểu về tập tin HTML

Ở mức độ cơ bản, chúng ta đã biết khi sử dụng trình duyệt để truy cập vào một trang web, trang web đó sẽ gửi trả về trình duyệt tập tin HTML, trình duyệt sẽ sử dụng tập tin HTML này để hiển thị giao diện trang web. HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, sử dụng để quy định cách trình duyệt hiển thị siêu văn bản và không phải là ngôn ngữ lập trình. HTML sử dụng các thẻ (tag) để quy định siêu văn bản sẽ được hiển thị như thế nào, các thẻ được đặt trong cặp dấu ngược nhọn < >.

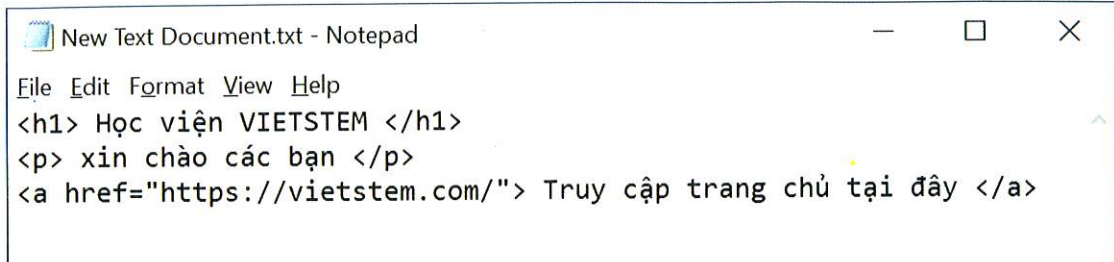
Sau đây chúng ta thực hành một ví dụ nhỏ về việc tạo tập tin HTML, phần thực hành này là không bắt buộc. Các bạn tạo một tập tin văn bản Text có phần mở rộng .txt ở màn hình Desktop bằng cách nhấn phải chuột vào màn hình Desktop, chọn New, chọn Text Document.



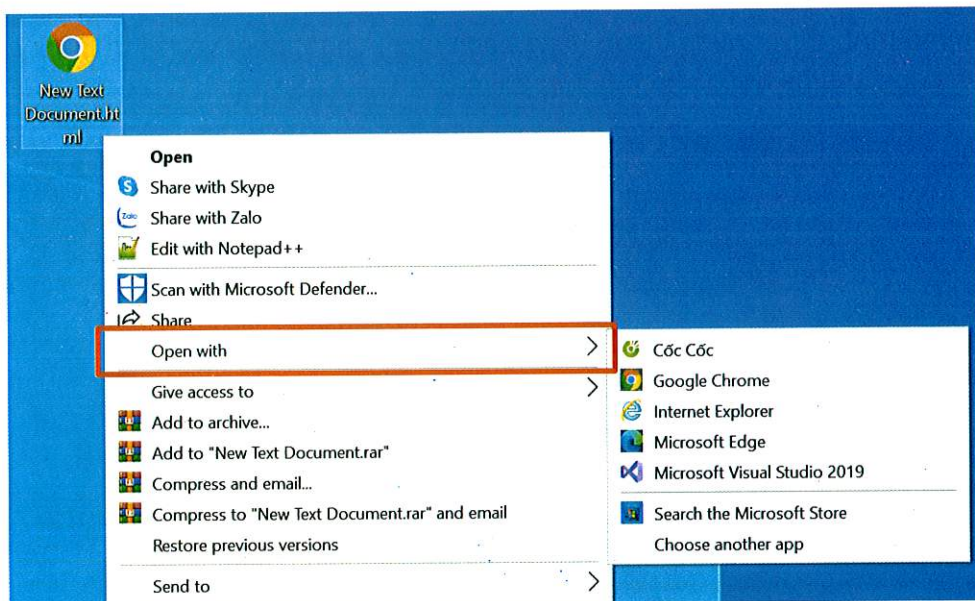
Tập tin New Text Document.txt sẽ được tạo, các bạn thêm vào tập tin đoạn mã HTML sau:

```
<h1> Học viện VIETSTEM </h1>
<p> xin chào các bạn </p>
<a href="https://vietstem.com/"> Truy cập trang chủ tại đây </a>
```





Các bạn nhấn Ctrl + S để lưu lại tập tin, sau đó đổi phần mở rộng ở tên tập tin từ .txt thành .html (toàn bộ tên tập tin được đổi từ "New Text Document.txt" thành "New Text Document.html"). Các bạn lưu ý không phải máy tính nào cũng hiển thị sẵn phần mở rộng của tập tin, nếu máy tính của mình chỉ hiển thị tên "New Text Document" chứ không hiển thị phần mở rộng .txt thì các bạn tự tìm kiếm cách hiển thị phần mở rộng và thay đổi phần mở rộng tập tin trên internet. Cuối cùng, các bạn nhấn phải chuột vào tập tin và chọn Open with, sau đó chọn một trình duyệt bất kỳ được cài đặt trên máy tính để mở.



Kết quả hiển thị trên trình duyệt:



Mỗi thẻ khác nhau sẽ hiển thị thông tin theo cách khác nhau, một thẻ có thể nằm bên trong một thẻ khác, thẻ bên ngoài được gọi là thẻ cha, thẻ nằm bên trong là thẻ con. Trong ví dụ này:

- Thẻ <h1> giúp hiển thị dòng chữ "Học viện VIETSTEM" kích thước lớn,
- Thẻ <p> hiển thị dòng chữ "xin chào các bạn" với kích thước nhỏ hơn,
- Thẻ <a> hiển thị dòng chữ "Truy cập trang chủ tại đây" với thông tin đường dẫn tới trang chủ <https://vietstem.com/>.

Các thẻ HTML có thể được lồng vào nhau. Trong các thẻ, phần văn bản (text) được gọi là nội dung, các thông tin thêm của thẻ được gọi là thuộc tính, ví dụ như `href="https://vietstem.com/"` có ý nghĩa đặt thuộc tính **href** giá trị `"https://vietstem.com/"`.

5.2. Cách trình duyệt hiển thị mã nguồn trang web

Khi truy cập một trang web, các bạn có thể quan sát tập tin mã nguồn HTML của trang web mà trình duyệt nhận được bằng cách nhấn chuột phải vào vị trí bất kỳ và chọn View page source hoặc nhấn tổ hợp phím Ctrl + U.

```

1 <!DOCTYPE html>
2 <html lang="vi" xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>VnExpress - Báo tiếng Việt nhiều người xem nhất</title> <meta name="description" content="VnExpress tin tức mới nhất - Thông
5 tin nhanh & chính xác được cập nhật hàng giờ. Đọc báo tin tức online Việt Nam & Thẻ giới nông nhất trong ngày về thể thao,
6 thời sự, pháp luật, kinh doanh,..."/>
7 <meta name="keywords" content="VnExpress, tin tức, tin the gioi, tin nhanh, tin tức viet nam, doc bao"/>
8 <meta name="news_keywords" content="VnExpress, tin tức, tin the gioi, tin nhanh, tin tức viet nam, doc bao"/>
9 <meta charset="utf-8">
10 <meta content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=5, user-scalable=1" name="viewport"/>
11 <meta http-equiv="X-UA-Compatible" content="IE=100"/>
12 <meta property="fb:app_id" content="1547540628876392"/>
13 <meta http-equiv="REFRESH" content="1800"/>
14 <meta name="apple-mobile-web-app-capable" content="yes"/>
15 <meta name="apple-mobile-web-app-title" content="Vnexpress.net"/>
16 <meta name="tt_article_id" content="1000000"/>
17 <meta name="tt_category_id" content="1000000"/>
18 <meta name="tt_site_id" content="1000000"/>
19 <meta name="tt_site_id_new" content="1000000"/>
20 <meta name="tt_list_folder" content="1000000"/>
21 <meta name="tt_list_folder_name" content="VnExpress"/>
22 <meta name="tt_page_type" content="site"/>
23 <meta name="tt_page_type_new" content="1"/>
24 <!-- add meta for pvtt3334 -->
25 <!-- end meta for pvtt -->
26 <!-- META FOR FACEBOOK -->
27 <meta property="og:site_name" content="vnexpress.net"/>
28 <meta property="og:rich_attachment" content="true"/>
29 <meta property="og:type" content="website"/>
30 <meta property="og:url" itemprop="url" content="https://vnexpress.net"/>
31 <meta property="og:image" itemprop="thumbnailUrl" content="https://s1.vnecdn.net/vnexpress/restruct/i/v520/logo_default.jpg"/>
32 <meta property="og:image:width" content="800"/>
33 <meta property="og:image:height" content="354"/>
34 <meta content="VnExpress - Báo tiếng Việt nhiều người xem nhất" itemprop="headline" property="og:title"/>
35 <meta content="VnExpress tin tức mới nhất - Thông tin nhanh & chính xác được cập nhật hàng giờ. Đọc báo tin tức online Việt Nam & Th
36 giới nông nhất trong ngày về thể thao, thời sự, pháp luật, kinh doanh,..." itemprop="description" property="og:description"/>

```

Tuy nhiên khi nhận được tập tin mã nguồn trên, trình duyệt sẽ không hiển thị trực tiếp tập tin ngay mà trải qua ba bước xử lý:

- Sửa lỗi HTML (ví dụ thẻ mở và thẻ đóng không trùng nhau)
- Chuẩn hóa HTML, giúp tối ưu quá trình tìm kiếm và xử lý văn bản hay ký tự
- Sử dụng Javascript để thao tác với các thẻ HTML

Sau khi trải qua ba bước này, giao diện trang web sẽ được hiển thị lên trình duyệt. Các bạn có thể quan sát tập tin HTML thực tế được hiển thị sau khi trình duyệt xử lý bằng cách nhấn phím F12.



Ý tưởng của việc thu thập thông tin từ các trang web chính là gửi yêu cầu truy cập đến trang web đó, nhận về tập tin mã nguồn HTML, phân tích và tìm đúng thông tin mà chúng ta cần thông qua các thẻ.



Tóm tắt lý thuyết và bài tập thực hành

Bài tập 1. Em hãy lập trình hiển thị thông tin cá nhân gồm họ tên, trường, lớp trên tệp tin html.

Bài tập 2. Lập trình để họ tên là chữ đậm.

Bài tập 3. Lập trình để dòng hiển thị tên trường là một liên kết tới trang web trường của bạn.

Bài tập 4. Em hãy thử tạo hoặc tìm kiếm một tập tin HTML khác và hiển thị tập tin HTML đó thay cho tập tin web_covid.html.