



# **BÀI TẬP**

## **CHUYÊN ĐỀ**

---

### **PHÁT TRIỂN ỨNG DỤNG WEB PYTHON VỚI DJANGO**



# BÀI 1: Tổng quan về xây dựng ƯD web



Mục tiêu chính: Ôn tập các kiến thức cơ bản về

- HTML, CSS, thiết kế một trang web cơ bản, bố cục web
- Bootstrap

## 1.1. Sử dụng các tag cơ bản

### GIỚI THIỆU HONDA Ô TÔ



Honda City

Cùng Honda City mới với những công nghệ vượt trội, hãy bứt phá lập thêm chiến công trên con đường danh vọng.



Honda Civic

Phong cách thiết kế khí động học cũng giúp hoàn thiện hiệu suất nhiên liệu và sự tĩnh lặng của cabin.



Honda Accord

Ngay từ giây đầu tiên cầm lái, phần khích ngay lập tức ủa đến với bạn. Tăng tốc mạnh mẽ hay ôm cua êm ái.



Honda CR\_V

Ngay từ giây đầu tiên cầm lái, phần khích ngay lập tức ủa đến với bạn. Tăng tốc mạnh mẽ hay ôm cua êm ái.

✓ **Mục tiêu:**

- Sử dụng các tag html cơ bản và các thuộc tính cơ bản để tạo nội dung trên

✓ **Yêu cầu thiết kế:**

<b>Stt</b>	<b>Đối tượng</b>	<b>Yêu cầu</b>
1	Tiêu đề	– Thể hiện dòng chữ " <b>GIỚI THIỆU HONDA Ô TÔ</b> "
2	Nội dung trang web	– Sử dụng tag <div> <img> <h3><p> để tạo như hình trên

✓ **Hướng dẫn:**

<b>Stt</b>	<b>Đối tượng</b>	<b>Code</b>
1	Tiêu đề	<h2 align="center" style="line-height: 1.1;font-size: 50px; color: #B3B3BB;letter-spacing: -0.6px;">GIỚI THIỆU HONDA Ô TÔ</h2>
2	Nội dung trang web	



Honda City

Cùng Honda City mới với  
những công nghệ vượt trội, hãy  
bứt phá lập thêm chiến công  
trên con đường danh vọng.

```
<div style="text-align: center; position: relative">

<div style="position: relative; min-height: 1px; padding: 10px; width: 20%; float: left ; background-color: #EFEFF1; margin: 15px">
    <div class="position: relative; margin-bottom: 30px;">
        <div class="border: 1px solid #E6E6EA; min-height: 280px; padding: 20px">
            
            <h3>Honda City</h3>
            <p style=" font-size: 15px; font-weight: 300; line-height: 22px; color: #848484;">Cùng Honda City mới với những công nghệ vượt trội, hãy bứt phá lập thêm chiến công trên con đường danh vọng.</p>
        </div>
    </div>
</div>

...
</div>
```

## 1.2. Tạo Form Liên hệ

- ✓ **Yêu cầu: tạo form liên hệ như sau:**



**Điện tử**

.VN

Trang chủ Điện thoại Máy tính bảng Liên hệ

# Liên hệ

Họ tên  
họ tên..

Điện thoại  
điện thoại..

Email  
email..

Nội dung

Liên hệ

Đại Học KHTN - Trung Tâm Tin Học - Ngành Lập Trình

✓ **Mục tiêu:**

- Sử dụng Form vào các điều khiển trên form

✓ **Yêu cầu thiết kế:**

Số	Đối tượng	Yêu cầu
1	Nội dung trang web	<ul style="list-style-type: none"> <li>+ Tạo form</li> <li>+ Sử dụng điều khiển Input text, textarea, Input submit để tạo form</li> </ul>

### 1.3. Giới thiệu template mẫu số 1

- ✓ **Yêu cầu: Tìm hiểu và hiệu chỉnh một số chi tiết trên template mẫu số 1**
- Cho template mẫu số 1:



RESTO

WELCOME SPECIALTIES MENU RESERVATION EVENTS CONTACT

Dine  
WITH US

A free bootstrap website template by ProBootstrap.com

• • •

Discover

OUR STORE

\*\*\*

Voluptatibus querat laboriosam fugit non Ut consequatur animi  
est molestiae enim a voluptate totam natus modi debitis dicta  
impedit voluptatum quod sapiente illo saepe explicabo quisquam  
perferendis labore et illum suscipit

ABOUT US



## Discover OUR SPECIALTIES

### BAKED POTATO PIZZA

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$20.99

### ROAST BEEF (4 STICKS)

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$32.99

### SALTED FRIED CHICKEN

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$19.99

### TUNA ROAST SOURCE

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$25.99

### IMPORTED SALMON STEAK

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$28.99

### FRIED POTATOES WITH GARLIC

Laboriosam pariatur modi praesentium deleniti molestiae officiis atque numquam.

\$22.99

## Discover OUR MENU



**BAKED POTATO PIZZA**  
Crab / Potatoes / Rice

\$22.99



**BAKED POTATO PIZZA**  
Crab / Potatoes / Rice

\$22.99



**TUNA ROAST SOURCE**  
Crab / Potatoes / Rice

\$22.99



**FRIED POTATOES WITH GARLIC**  
Crab / Potatoes / Rice

\$22.99



**ROAST BEEF (4 STICKS)**  
Crab / Potatoes / Rice

\$22.99



**SALTED FRIED CHICKEN**  
Crab / Potatoes / Rice

\$22.99



**SALTED FRIED CHICKEN**  
Crab / Potatoes / Rice

\$22.99



**TUNA ROAST SOURCE**  
Crab / Potatoes / Rice

\$22.99



## Contact LET'S CHAT

Voluptatibus querat laboriosam fugit non Ut consequatur animi  
est molestiae enim a voluptate totam natus modi debitis dicta  
impedit voluptatum quod sapiente illo saepe explicabo quisquam  
perferendis labore et illum suscipit

Your Name

Your Email

Your Message

SEND MESSAGE

### LOCATIONS

198 West 21th Street, Suite 721  
New York NY 10016

198 West 21th Street, Suite 721  
New York NY 10016

### OPEN HOURS

Monday - Thursday  
5:30pm - 10:00pm

Friday - Sunday  
5:30pm - 10:00pm

Available for Catering  
Email or Call Us

© 2017 ProBootstrap:Resto. All Rights Reserved. Images by GraphicBurger & Unsplash



- Hãy hiệu chỉnh lại một số nội dung trên template mẫu số 1, kết quả như sau:
  - Chính logo và slider

TTTH

GIỚI THIỆU LẬP TRÌNH VIÊN CHUYÊN ĐỀ ĐĂNG KÝ TIN TỨC LIÊN HỆ

Welcome

COMPUTER SCIENCE CENTER

Với hàng trăm bài học chất lượng cao, đáp ứng mọi yêu cầu

Trung tâm tin học KHTN csc.edu.vn

...



- Chính phần giới thiệu

# TRUNG TÂM TIN HỌC

\* \* \*

Trung tâm tin học là đơn vị thuộc trường Đại Học Khoa Học Tự Nhiên TP.HCM. Được thành lập từ năm 1986, đến nay đã hơn 30 năm với trên một triệu học viên tham gia khóa học, hơn 20 cơ sở khắp cả nước và có trên 100 chương trình đào tạo CNTT

## ABOUT US



- Chính phần footer

<b>ĐỊA CHỈ</b>	137E Nguyễn Chí Thanh Quận 10 TP. Hồ Chí Minh	227 Nguyễn Văn Cừ Quận 10 TP. Hồ Chí Minh	<b>GIỜ LÀM VIỆC</b>	Thứ Hai - Chủ Nhật 7:30 - 21:00	Thứ Hai - Chủ Nhật 7:30 - 21:00	P: (028) 38 351056 E: tuvan@csc.hcmus.edu.vn
----------------	--	--	---------------------	------------------------------------	------------------------------------	--

© 2017 ProBootstrap:Resto. All Rights Reserved. Images by GraphicBurger & Unsplash

[Twitter](#) [Facebook](#) [Instagram](#)

### ✓ Hướng dẫn

- Dùng trang web mẫu được cung cấp, tạo ra một bộ logo mới bất kỳ có kích thước tương tự như logo cũ và copy vào thư mục img.
- Mở tập tin css/style.css: tìm đến khai báo .probootstrap-navbar .navbar-brand và khai báo @media only screen and để chỉnh lại thuộc tính background

## 1.4. Giới thiệu template mẫu số 2

- ✓ **Yêu cầu: Tìm hiểu về bố cục và nội dung của template mẫu số 2**
  - Cho template mẫu số 2



The image shows a website's homepage. At the top right is a navigation bar with links for Home, About, Blog, Team, and Contact. A teal ribbon badge on the left side contains the word "Office". The main background image is a woman with long blonde hair wearing a large straw hat and dark sunglasses, looking towards the camera. Below her image, the text "MULTI-PURPOSE TEMPLATE" is displayed in white, followed by "PORTFOLIO \ CORPORATE \ CREATIVE" and three small colored circles (orange, grey, blue). The bottom section features a white header with the text "Our Mission" and "Things we provide in Office". Below this are three dark boxes containing icons for Android, Apple, and a thumbs-up.

## *Trang home (index.html)*

## *Trang about (about.html)*



The screenshot shows a website header with a navigation bar: Home, About, **Blog**, Team, Contact. A teal ribbon icon labeled 'Office' is on the left. The main content area features a large image of a modern bedroom with lime green walls, a white bed, and a large window overlooking a landscape. Below the image, the title 'ALICE IN WONDERLAND, PART DOS' is displayed with a document icon. To the right is a sign-in form with fields for 'Username or Email' and 'Password'.

Trang blog (blog.html)

The screenshot shows a website header with a navigation bar: Home, About, Blog, **Team**, Contact. A teal ribbon icon labeled 'Office' is on the left. The main content area features a large image of a conference room with people silhouetted against a city skyline at night. Below the image, the heading 'Who We Are' is displayed. The page lists team members in two columns:

Profile	Name	Role	Team
	John Doe	Web Developer , Technext	John Doe
	Jane Ley	Software Developer , Technext	Android Developer , Technext
	Chris Way	iOS Developer , Technext	Chris Way
	Jany Doe		Christopher

Trang team (team.html)



Find us at:

Johns, Smith & Associates  
Via Ludovisi 39-45, Rome, 54267, Italy  
Phone: 1.800.245.356  
Fax: 1.800.245.357  
Email: [hello@LawOffice.org](mailto:hello@LawOffice.org)

NAME \*

E-MAIL \*

Social Link

Trang contact (contact.html)

✓ **Hướng dẫn**

- Hãy cho biết bố cục của template này có các thành phần chính nào?



## BÀI 2: Tổng quan Django



*Mục tiêu chính:*

- *Cài đặt Django*
- *Tạo project web Django*
- *Tạo các web app cho project web*

### 2.1. Cài đặt Django

✓ **Yêu cầu: Cài đặt Django**

```

Administrator: C:\Windows\System32\cmd.exe - pip install django
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:>python --version
Python 3.6.2

C:>pip install django
Collecting django
  Downloading Django-1.11.5-py2.py3-none-any.whl (6.9MB)
    100% |████████████████████████████████| 7.0MB 148kB/s
Collecting pytz (from django)
  Downloading pytz-2017.2-py2.py3-none-any.whl (484kB)
    100% |████████████████████████████████| 491kB 726kB/s
Installing collected packages: django, pytz
Successfully installed django-1.11.5 pytz-2017.2

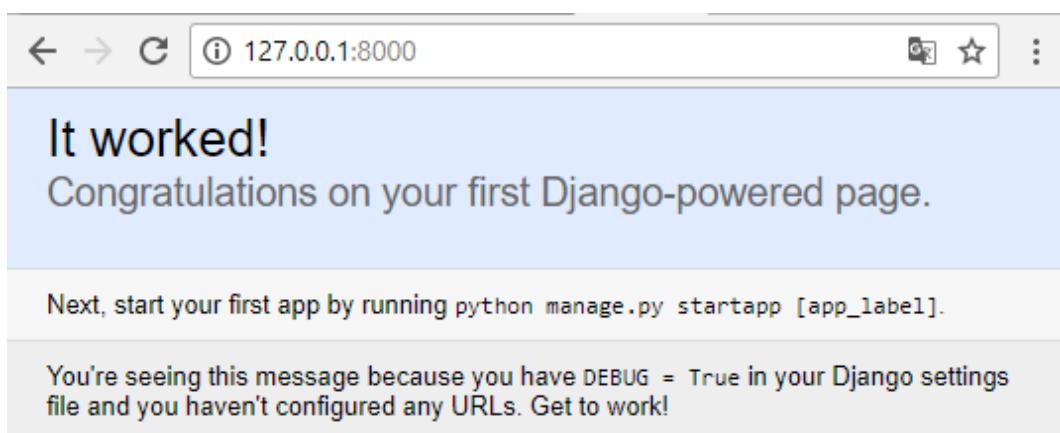
```

✓ **Hướng dẫn**

- Tìm đến nơi lưu trữ thư mục Python đã cài đặt
- Mở command line
- Gõ pip install django

### 2.2. Tạo project web

✓ **Yêu cầu: Tạo project web có tên là webbuoi2, sau đó chạy thử và xem kết quả**



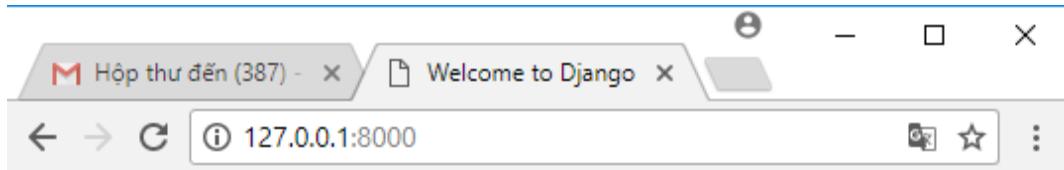
✓ **Hướng dẫn**

- Vào File > New > Others > PyDev > PyDev Django Project > đặt tên “webbuoi2” > Finish.

## 2.3. Tạo web app trong web project

✓ **Yêu cầu: Tạo web app có tên là appone trong webbuoi2, với những yêu cầu như sau:**

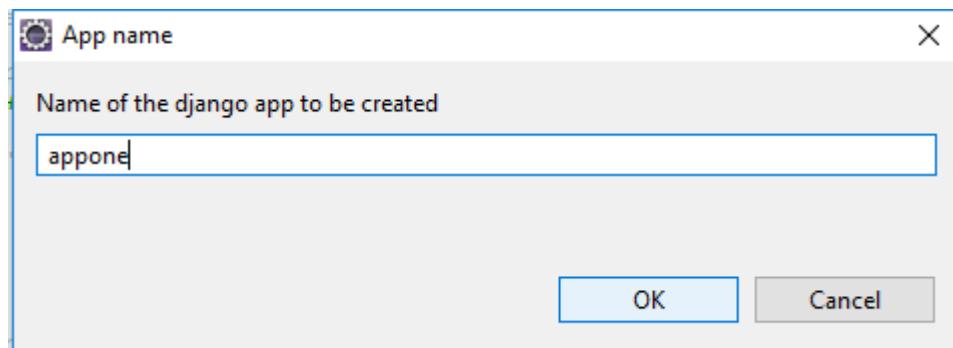
- Tạo appone
- Đăng ký appone trong INSTALL\_APPS
- Tạo view index trả về nội dung “Chào mừng bạn đến với Phát triển ứng dụng web Python với Django”
- Link view này vào urls.py
- Chạy thử website và xem kết quả



*Chào mừng bạn đến với Phát triển ứng dụng web Python với Django*

✓ **Hướng dẫn**

- Tạo appone: chọn webbuoi2 > click phải > chọn Django > chọn Create application (PyCharm: Tools > Run manage.py task > startapp ten\_app; )



- Đăng ký appone trong INSTALL\_APPS của settings.py trong webbuoi2

```
INSTALLED_APPS = [
    ...
    'appone',
]
```

- Tạo view index trong views.py của appone trả về nội dung “Chào mừng bạn đến với Phát triển ứng dụng web Python với Django”

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



```
# Create your views here.  
  
def index(request):  
    return HttpResponseRedirect("<em>Chào mừng bạn đến với Phát triển ứng dụng web  
Python với Django</em>")
```

- Link view này vào urls.py của webbuoi2

```
from django.conf.urls import url  
from django.contrib import admin  
from appone import views  
  
urlpatterns = [  
    url(r'^admin/', admin.site.urls),  
    url(r'^$',views.index,name='index'),  
]
```

- Chạy thử website và xem kết quả

# BÀI 3: Mapping URLs, Template, Static File, View

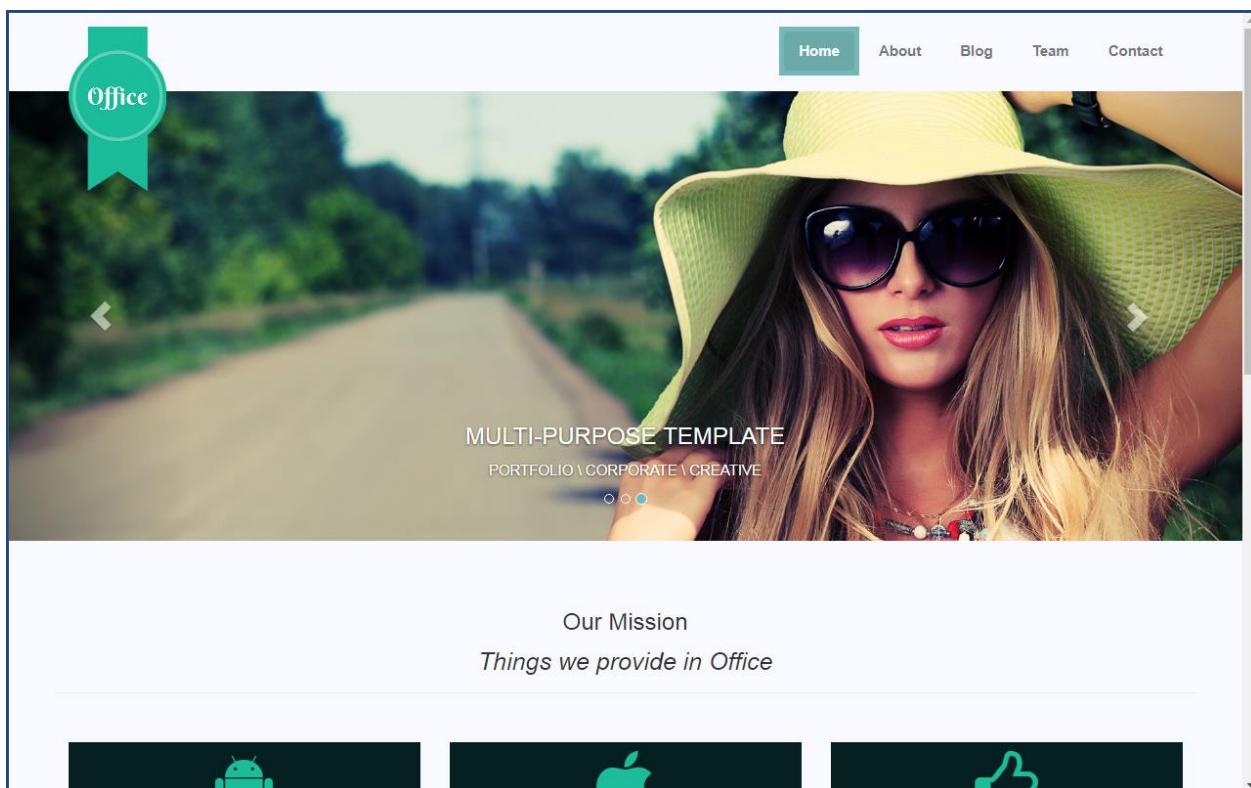


Mục tiêu chính: trang bị kiến thức và kỹ năng

- *Mapping URLs, Template*
- *Static File, View*

## 3.1. Xây dựng website khung chuẩn Django từ template mẫu được cung cấp

- ✓ **Yêu cầu:** Áp dụng kiến thức mapping URLs, Template, Static File và view để xây dựng website khung chuẩn Django từ template mẫu được cung cấp Office-master:



Trang home (*index.html*)



**About our Firm :**

Curabitur vulputate, ligula lacinia scelris que tempor, lacus lacus ornare ante, ac egeistas est urna sit amet arcu. Class apte nt taeraciti sociosqu ad litora torquent.

[Our Firm's History](#)  
[Philosophy](#)  
[Media / In the Press](#)

**Our Firm's History**

Curabitur vulputate, ligula lacinia scelrisque tempor, lacus lacus ornare ante, ac egeistas est urna sit amet arcu. Class apte nt taeraciti sociosqu ad litora torquent. Hibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi euismod magna ac lorem rutrum elementum. Donec viverra auctor lobortis. Pellentesque eu est a nulla placerat dignissim. Morbi a dolor sit amet ipsum enim in.

**2002 - LOREM IPSUM DOLOR**

Nunc eu ullamcorper orci. Quisque eget odio ac lectus vestibulum faucibus eget in metus. In pellentesque faucibus. Nunc eu ullamcorper orci. Quisque eget odio ac lectus vestibulum faucibus eget in metus. In pellentesque faucibus.

**2007 - VIVAMUS HENDERIT**

Trang about (about.html)

**ALICE IN WONDERLAND, PART DOS**

file:///D:/Phuong document/Python Web/ngay01102017/Office-master/blog.html

**Sign in**

**Username or Email**

**Password (forgot password?)**

Trang blog (blog.html)



The screenshot shows a team page with a banner featuring silhouettes of people against a city skyline. Below the banner, the text "Who We Are" is displayed. The page lists six team members in two columns:

Profile Picture	Name	Role	Company
	John Doe	Web Developer	Technext
	Jane Ley	Software Developer	Technext
	John Doe	Android Developer	Technext
	Chris Way	iOS Developer	Technext
	Jany Doe		
	Christopher		

Trang team (team.html)

The screenshot shows a contact page with a map of Ho Chi Minh City, Vietnam, highlighting the office location. Below the map, there is contact information and a message submission form.

**Contact Information:**

Trung Tâm Tin Học -  
ĐH Khoa Học Tự Nhiên  
227 Nguyễn Văn Linh, Phường 5, Quận 9, Hồ Chí Minh, Việt Nam  
4.7 ★★★★★  
[Xem bản đồ lớn hơn](#)

**Find us at:**

Johns, Smith & Associates  
Via Ludovisi 39-45, Rome, 54267, Italy  
Phone: 1.800.245.356  
Fax: 1.800.245.357  
Email: [hello@LawOffice.org](mailto:hello@LawOffice.org)

**Social Link**

**MESSAGE \***

Trang contact (contact.html)



## ✓ Hướng dẫn

- Bước 1: Tạo project có tên là myoffice => chạy thử
- Bước 2: Tạo web app có tên là officemaster trong myoffice
- Bước 3: Đăng ký officemaster trong settings.py của myoffice tại INSTALL\_APPS

```
INSTALLED_APPS = [
    'django.contrib.admin',
    ...
    'officemaster',
]
```

- Bước 4: Tạo các thư mục static (chứa các hình ảnh, css, js... tĩnh của myoffice)
- Bước 5: Tạo thư mục templates (chứa các trang html của myoffice)
- Bước 6: Tạo thư mục officemaster nằm trong thư mục templates (chứa các trang html của officemaster)
- Bước 7: Đăng ký static và templates trong settings.py của myoffice

```
TEMPLATE_DIR = os.path.join(BASE_DIR, "templates")
STATIC_DIR = os.path.join(BASE_DIR, "static")
...
TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [TEMPLATE_DIR, ],
    ...
},
]
...
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    STATIC_DIR,
]
```

- Bước 8: copy thư mục asset và thư mục js của template vào thư mục static
- Bước 9: copy tất cả các tập tin html của template vào templates/officemaster
- Bước 10: tạo view index() dẫn đến trang index.html trong views.py của officemaster

```
def index(request):
    return render(request, "officemaster/index.html")
```

- Bước 11: tạo url gọi view index() vừa viết của officemaster trong urls.py của myoffice

```
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^admin/', admin.site.urls),
]
```



- Bước 12: chạy thử website => sẽ thấy hiển thị trang index.html nhưng chưa theo định dạng chuẩn và chưa hiển thị hình ảnh, hiệu ứng
- Bước 13: chỉnh sửa trang index.html theo đúng chuẩn Django, có sử dụng static. => chạy thử để xem kết quả

```
{% load staticfiles %}
<html lang="en">
  <head>

    <!-- meta data & title -->
    <meta charset="utf-8">
    <title>Office</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- Favicon and touch icons -->
    <link rel="shortcut icon" href="{% static 'assets/ico/favicon.ico' %}">
    <link rel="apple-touch-icon-precomposed" sizes="144x144" href="{% static 'assets/ico/apple-touch-icon-144-precomposed.png' %}">
    <link rel="apple-touch-icon-precomposed" sizes="114x114" href="{% static 'assets/ico/apple-touch-icon-114-precomposed.png' %}">
    <link rel="apple-touch-icon-precomposed" sizes="72x72" href="{% static 'assets/ico/apple-touch-icon-72-precomposed.png' %}">
    <link rel="apple-touch-icon-precomposed" href="{% static 'assets/ico/apple-touch-icon-57-precomposed.png' %}">
    ...
  </head>
```

- Bước 14: lần lượt tạo các view khác của officemaster như about, blog, team, contact trong views.py của officemaster.

```
def about(request):
    return render(request, "officemaster/about.html")

def blog(request):
    return render(request, "officemaster/blog.html")

def contact(request):
    return render(request, "officemaster/contact.html")

def team(request):
```



```
return render(request, "officemaster/team.html")
```

- Bước 15: tạo các url cho các view này tại urls.py của officemaster

```
from django.conf.urls import url
from officemaster import views

urlpatterns = [
    url(r'^$',views.index,name= 'index'),
    url(r'^about.html/$',views.about,name= 'about'),
    url(r'^blog.html/$',views.blog,name= 'blog'),
    url(r'^contact.html/$',views.contact,name= 'contact'),
    url(r'^team.html/$',views.team,name= 'team'),
]
```

- Bước 16: include các urls.py của officemaster vào urls.py của project myoffice

```
...
from django.conf.urls import include

urlpatterns = [
    url(r'^$',views.index,name= 'index'),
    url(r'^admin/', admin.site.urls),
    url(r'^officemaster/ ',include('officemaster.urls')),
]
```

- Bước 17: chạy thử và xem kết quả
- Bước 18: chỉnh sửa trang about.html, blog.html, static.html theo đúng chuẩn Django, có sử dụng static. => chạy thử để xem kết quả

## BÀI 4: Database và Model



**Mục tiêu chính:**

- Xây dựng các model
- Sử dụng admin để thêm/ xóa/ cập nhật dữ liệu

### 4.1. Xây dựng website – Phần 1

- ✓ **Yêu cầu:** Xây dựng website – Phần 1 như sau:

*Trang chủ (mainpage.html)*

**Danh sách các trang web**

1. Tên trang: Facebook
  - ID: 1
  - Thuộc Topic: Social NetworkName: Facebook
  - URL: <http://facebook.com>
2. Tên trang: Phim hoạt hình
  - ID: 3
  - Thuộc Topic: Social NetworkName: Phim hoạt hình
  - URL: <http://www.disneyinternational.com/>



Trang "Các trang web" webpages.html

The screenshot shows a website with a header featuring the word "django" on a monitor icon, a magnifying glass over a globe, and the logo of the University of Technology and Education (TTU) with the text "TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC". Below the header is a navigation bar with links: Trang chủ, Các trang web, Các nội dung, Các khóa học, Đăng ký, Quản trị. The main content area is titled "Danh sách nội dung" and lists two items:

1. Tên : Cars 3
  - ID 3
  - Thuộc nhóm Webpage: Phim hoạt hình
  - Nội dung: Cars 3 bắt đầu được công chiếu từ ngày 11/8/2017
2. Tên : Chủ đề quan tâm
  - ID 2
  - Thuộc nhóm Webpage: Facebook
  - Nội dung: Những chủ đề được nhiều người quan tâm

Trang "Các nội dung" contents.html

✓ **Hướng dẫn**

- Bước 1: tạo các model cần thiết trong models.py của first\_app , sau đó migrate các model này

```
from django.db import models
from django.contrib.auth.models import User
# Create your models here.
# tao Topic (bảng topic có id và name)
class Topic(models.Model):
    top_name = models.CharField(max_length=264,unique=True)

    def __str__(self):
        return self.top_name
```



```
#tạo Webpage (bảng webpage có id, name, url, khóa ngoại là topic)
class Webpage(models.Model):
    topic = models.ForeignKey(Topic, on_delete=models.PROTECT)
    name = models.CharField(max_length=264, unique=True)
    url = models.URLField(unique=True)

    def __str__(self):
        return self.name

#tạo Content (bảng content có id, name, content, image, khóa ngoại là webpage)
class Content(models.Model):
    webpage = models.ForeignKey(Webpage, on_delete=models.PROTECT)
    name = models.CharField(max_length=264, unique=True)
    content = models.CharField(max_length=264, unique=True)
    image = models.ImageField(upload_to = "images/")

    def __str__(self):
        return str(self.name)
```

- Chú ý: vì cần phải upload hình ảnh của ứng dụng nên cần tạo thư mục media, sau đó tạo images trong media và khai báo trong settings.py và urls.py của project (nếu chưa thực hiện)

settings.py

```
MEDIA_ROOT = os.path.join(BASE_DIR, "media")
MEDIA_URL = '/media/'
STATICFILES_DIRS = [
    STATIC_DIR,
    MEDIA_ROOT,
]
```

urls.py

```
urlpatterns = [
    ...
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

- Bước 2: đăng ký các model này trong admin.py của webapp

```
from django.contrib import admin

# Register your models here.
```



```
from first_app.models import Topic, Webpage, Content
```

```
admin.site.register(Topic)
admin.site.register(Webpage)
admin.site.register(Content)
```

- Bước 3: vào Admin Interface để thêm dữ liệu cần thiết thiết (cần phải tạo tài khoản Admin khi muốn làm việc với Admin Interface)

```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

D:\FirstDjangoWeb\src>python manage.py createsuperuser
Username <leave blank to use 'hu'>: admin
Email address: tubirona@gmail.com
Password:
Password <again>:
Error: Your passwords didn't match.
Password:
Password <again>:
This password is too common.
This password is entirely numeric.
Password:
Password <again>:
Superuser created successfully.

D:\FirstDjangoWeb\src>
```

Tạo tài khoản admin

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin/login/?next=/admin/`. The page title is "Django administration". It contains two input fields: "Username" with the value "phuong13021982" and "Password" with several dots. A blue "Log in" button is at the bottom.

Đăng nhập



Django administration

WELCOME ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > First\_App > Topics

Select topic to change

Action: ----- Go 0 of 2 selected

TOPIC  
 News  
 Social Network

2 topics

*Thêm dữ liệu vào các bảng*

- Bước 4: tạo các view cần thiết

```
def mainpage(request):
    return render(request, 'first_app/mainpage.html')

def webpages(request):
    web_list = Webpage.objects.order_by('name')
    web_dict = { "webpages":web_list }
    return render(request, 'first_app/webpages.html', context=web_dict)

def contents(request):
    content_list = Content.objects.order_by('name')
    content_dict = { "contents":content_list }
    return render(request, 'first_app/contents.html', context=content_dict)
```

- Bước 5: tạo các URL cần thiết trong first\_app/urls.py

```
urlpatterns = [
    url(r'^$', views.mainpage, name='mainpage'),
    url(r'^webpages.html/$', views.webpages, name='webpages'),
    url(r'^contents.html/$', views.contents, name='contents'),
]
```

- Bước 6: tạo các template HTML cần thiết

- mainpage.html

```
{% load staticfiles %}
<html>
<head>
    <meta charset="utf-8">
```



```

<title>HOME</title>
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
      integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
      crossorigin="anonymous">

</head>
<body>
    <div>
        
    </div>
    <nav class="navbar navbar-default navbar-static-top">
        <div class="container">
            {% url 'index' %}
            <ul class="nav navbar-nav">
                <li><a class="navbar-brand" href="{% url 'mainpage' %}">Trang chủ</a></li>
                <li><a class="navbar-link" href="{% url 'webpage' %}">Các trang web</a></li>
                <li><a class="navbar-link" href="{% url 'content' %}">Các nội dung</a></li>
                <li><a class="navbar-link" href="{% url 'homepage' %}">Các khóa học</a></li>
                <li><a class="navbar-link" href="{% url 'register' %}">Đăng ký</a></li>
                <li><a class="navbar-link" href="{% url 'admin:index' %}">Quản trị</a></li>
            </ul>
        </div>
    </nav>
    <h2>Chào mừng bạn đến với website Học Web Django</h2>
    <h3>Nơi bạn có thể xem thông tin về các trang web, nội dung, khóa học</h3>
</body>
</html>

```

- webpages.html

```

...
<h1>Danh sách trang web</h1>
{% if webpages%}
<ol>
    {% for web in webpages %}
        <li>Tên trang: {{web.name}}</li>
        <ul>

```



```

<li>ID: {{web.pk}} </li>
<li>Thuộc Topic: {{web.topic}}Name: {{web.name}}</li>
<li>URL: {{web.url}}</li>
</ul>
{% endfor %}
</ol>

{% endif %}

...

```

▪ contents.html

```

...
{% load staticfiles %}
    <h1>Danh sách nội dung</h1>
    {% if contents%}
<ol>
    {% for content in contents %}
        <li>Tên : {{content.name}}</li>
        <ul>
            <li> ID {{content.pk}} </li>
            <li>Thuộc nhóm Webpage: {{content.webpage}}</li>
            <li>Nội dung: {{content.content}}</li>
        </ul>
        <p>
            
        </p>
        {% endfor %}
    </ol>
    {% endif %}
...

```

- Bước 7: chạy thử và xem kết quả

## 4.2. Xây dựng website – Phần 2

- ✓ **Yêu cầu: Xây dựng website – Phần 2 như sau:**



Trang chủ Các trang web Các nội dung Các khóa học Đăng ký Quản trị

Chào mừng bạn đến với website Học Web Django  
Nơi bạn có thể xem thông tin về các trang web, nội dung, khóa học

*Trang chủ (mainpage.html) – đã làm từ phần 1 đã làm link qua trang "Các khóa học"*

TTTH GIỚI THIỆU LẬP TRÌNH VIÊN CHUYÊN ĐỀ ĐĂNG KÝ TIN TỨC LIÊN HỆ

*Study and Learning*  
**WITH US**  
Với hàng trăm khóa học phù hợp mọi yêu cầu

...

*About us*  
**TRUNG TÂM TIN HỌC**

*Trang "Các khóa học" homepage.html*



TTTH

GIỚI THIỆU LẬP TRÌNH VIÊN CHUYÊN ĐỀ ĐĂNG KÝ TIN TỨC LIÊN HỆ

## LẬP TRÌNH VIÊN MỚI NHẤT

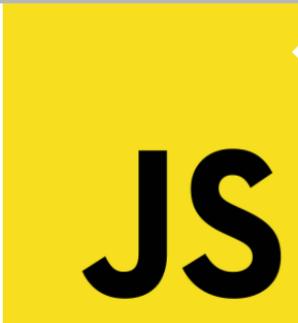


### LẬP TRÌNH VIÊN .NET

"Lập trình viên .Net" do TTTH  
cấp

6400000 vnđ

XEM THÊM

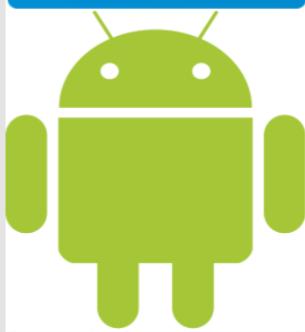


### LẬP TRÌNH VIÊN JAVASCRIPT

"Lập trình viên JavaScript" do  
TTTH cấp

6400000 vnđ

XEM THÊM



### LẬP TRÌNH VIÊN ANDROID

"Lập trình viên Android" do  
TTTH cấp

6400000 vnđ

XEM THÊM



### LẬP TRÌNH VIÊN PHP

"Lập trình viên Web PHP" do  
TTTH cấp

6400000 vnđ

XEM THÊM



### LẬP TRÌNH VIÊN JAVA

"Lập trình viên Java" do TTTH  
cấp

6400000 vnđ



### LẬP TRÌNH VIÊN IOS

"Lập trình viên iOS" do TTTH  
cấp

6400000 vnđ

Phần lập trình viên



TTTH

GIỚI THIỆU LẬP TRÌNH VIÊN CHUYÊN ĐỀ ĐĂNG KÝ TIN TỨC LIÊN HỆ

## Discover CÁC LỚP CHUYÊN ĐỀ



LẬP TRÌNH PYTHON CƠ BẢN

"Lập trình Python Cơ bản" do TTTH cấp

23000000 vnd

vnd



PHÁT TRIỂN UD WEB PYTHON VỚI DJANGO 6000000 vnd

"Phát triển ứng dụng Web Python với Django" do TTTH cấp

XEM THÊM



LẬP TRÌNH PYTHON NÂNG CAO

"Lập trình Python Nâng Cao" do TTTH cấp

2700000 vnd



TRUY VẤN CSDL VỚI SQL SERVER 2014 1600000 vnd

"Truy vấn CSDL với SQL Server 2014" do TTTH cấp

XEM THÊM

*Phần Chuyên đề*

Trang chủ   Các trang web   Các nội dung   Các khóa học   Đăng ký   Quản trị

## Phát triển UD Web Python với Django

Nhóm Chuyên đề

Học phí: 2800000 vnđ  
Thời lượng: 48 giờ

Nội dung: ☐ Trang bị cho học viên (HV) những kiến thức từ cơ bản đến nâng cao và những kỹ năng cần thiết về cách phát triển website dựa trên Django Framework, một trong những Python Web Framework phổ biến nhất hiện nay, để học viên có thể xây dựng và triển khai những website hoàn chỉnh, hiện đại, đáp ứng nhu cầu thực tế một cách nhanh chóng và linh động. ☐ Cung cấp cho HV kỹ thuật tùy biến và hiệu chỉnh template cho phù hợp với yêu cầu của website. Rèn luyện và phát triển kỹ năng tổ chức, xây dựng và triển khai ứng dụng web.

Chứng chỉ "Phát triển ứng dụng Web Python với Django" do TTHH cấp

python<sup>TM</sup>

Khi chọn "Xem thêm" của phần "Lập trình viên" hay "Chuyên đề" => chuyển sang trang "Nội dung khóa học" course\_detail.html

✓ **Hướng dẫn**

- Bước 1: tạo các model cần thiết trong first\_app/models.py, sau đó migrate các model này

```
class Category(models.Model):
    name = models.CharField(max_length=264,unique=True)
    description = models.TextField()

    def __str__(self):
        return self.name

class Course(models.Model):
    category = models.ForeignKey(Category, on_delete=models.PROTECT)
    name = models.CharField(max_length=264,unique=True)
    duration = models.IntegerField()
```



```

fee = models.IntegerField()
description = models.TextField()
certificateion = models.CharField(max_length=500)
image = models.ImageField(upload_to = "images/")

def __str__(self):
    return str(self.name)

```

- Bước 2: đăng ký các model này trong admin.py của first\_app

```

admin.site.register(Category)
admin.site.register(Course)

```

- Bước 3: vào Admin Interface để thêm dữ liệu cần thiết (cần phải tạo tài khoản Admin khi muốn làm việc với Admin Interface)

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > First\_App > Categories

Select category to change

Action: ----- Go 0 of 2 selected

<input type="checkbox"/>	CATEGORY
<input type="checkbox"/>	Chuyên đề
<input type="checkbox"/>	Lập trình viên

2 categories

- Bước 4: tạo các view cần thiết

```

def homepages(request):
    course_list_ltv = Course.objects.filter(category = 1).order_by('name')
    part = int(len(course_list_ltv)/2)
    lst1 = []
    for i in range(0, part, 1):
        lst1.append(course_list_ltv[i])
    lst2 = []
    for i in range(part, len(course_list_ltv), 1):
        lst2.append(course_list_ltv[i])

    course_list_cd = Course.objects.filter(category = 2).order_by('name')
    part2 = int(len(course_list_cd)/2)
    lst3 = []
    for i in range(0, part2, 1):

```

```

lst3.append(course_list_cd[i])
lst4 = []
for i in range(part2, len(course_list_cd), 1):
    lst4.append(course_list_cd[i])

course_dict = { "courses":course_list_ltv, "courses_cd": course_list_cd,\n
                "course1":lst1, "course2":lst2, "course3":lst3, "course4":lst4}
return render(request, "first_app/homepage.html", context = course_dict)

def course_detail(request, id=None):
    course = Course.objects.get(pk=id)
    return render(request, "first_app/course_detail.html", context ={ "course":course})

```

- Bước 5: tạo các URL cần thiết

```

urlpatterns = [
    ...
    url(r'^homepage.html/$',views.webpages,name='homepages'),
    url(r'^homepages/course_detail/(\d+)/$', views.course_detail, name='course_detail'),
]

```

- Bước 6: tạo các template HTML cần thiết

- homapege.html

```

...
<!-- probootstrap-bg-white -->
<section class="probootstrap-section">
    <div class="container">
        <div class="row">
            <div class="probootstrap-cell-retro">
                <!-- duyet cac phan tu trong course row -->
                {% if courses %}
                    <div class="half">
                        {% for value in course1 %}
                            <div class="probootstrap-cell probootstrap-animate" data-animate-effect="fadeIn">
                                <div class="image" style="background-image: url('{{value.image.url}}');background-size: 100% 100%;">
                                    </div>
                                <div class="text text-center">
                                    <h3>{{value.name}}</h3>
                                    <p>{{value.certificateion}}</p>
                                    <p class="price">{{value.fee}} vnđ</p>
                                    <p> <a href="{% url 'first_app:course_detail' value.pk %}" class='btn btn-primary' style="float: center;">Xem thêm</a></p>

```

```
        </div>
    </div>

    {% endfor %}

    <div class="half">
        {% for value in course2 %}
            <div class="probootstrap-cell probootstrap-animate" data-
animate-effect="fadeIn">
                <div class="image" style="background-image:
url('{{value.image.url}}');background-size: 100% 100%;">
                    </div>
                <div class="text text-center">
                    <h3>{{value.name}}</h3>
                    <p>{{value.certificatetion}}</p>
                    <p class="price">{{value.fee}} vnđ</p>
                    <a href="{% url 'first_app:course_detail' value.pk %}">Xem thêm</a>
                </div>
            </div>
        {% endfor %}
    </div>
    {% endif %}

    ...
    <section class="probootstrap-section probootstrap-bg-white">
        <div class="container">
            {% if courses_cd %}
                <div class="row">
                    <div class="col-md-6">
                        <ul class="menus">
                            {% for value in course3 %}
                                <li>
                                    <figure class="image">
class="img-fluid" alt="Free Bootstrap Template by
ProBootstrap.com"></figure>
                                    <div class="text">
                                        <span class="price">{{value.fee}} vnđ</span>
                                        <h3>{{value.name}}</h3>
                                        <p>{{value.certificatetion}}</p>
                                        <a href="{% url 'first_app:course_detail' value.pk %}">Xem thêm</a>
                                    </div>
                                </li>
                            {% endfor %}
                        </ul>
                    </div>
                    <div class="col-md-6">
                        <ul class="menus">
                            {% for value in course4 %}
                                <li>
                                    <figure class="image">>/figure>
    <div class="text">
        <span class="price">{{value.fee}} vnđ</span>
        <h3>{{value.name}}</h3>
        <p>{{value.certificateion}}</p>
        <a href="{% url 'first_app:course_detail' value.pk %}">Xem thêm</a>
    </div>
</li>
{% endfor %}

</ul>
</div>
</div>
{% endif %}
</div>
</section>

...

```

- course\_detail.html

```

...
{% load staticfiles %}
<div class="container">
{% if course%}
    <h1>{{course.name}}</h1>
    <h3>Nhóm {{course.category}}</h3>
    <p>Học phí: {{course.fee}} vnđ</p>
    <p>Thời lượng: {{course.duration}} giờ</p>
    <p style="color: blue;">Nội dung: {{course.description}}</p>
    <h3 style="color: red;">Chứng chỉ
    {{course.certIFICATEION}}</h3>
    <div>  </div>
{% endif %}
</div>
...

```

- Bước 7: chạy thử và xem kết quả

# BÀI 5: Forms



**Mục tiêu chính:**

- Xây dựng các form cần thiết cho website

## 5.1. Form Contact

- ✓ Yêu cầu: Thực hiện form Contact trong trang contact.html như sau:



Địa chỉ:  
Trung Tâm tin Học ĐH KHTN  
227 Nguyễn Văn Cừ P9 Q5  
Phone: (028) 8351056  
Email: tuvan@csc.hcmus.edu.vn

Social Link

Call us      Like us      Follow us

Name

Phone

Email

Message

**Gửi liên hệ**

Trang contact.html



Office

Trung Tâm Tin Học ĐH Khoa Học Tự Nhiên  
227 Nguyễn Văn Cử P9 Q5  
4,7 ★★★★☆  
Xem bản đồ lớn hơn

Chỉ đường  
Lưu

Home About Blog Team Contact

Địa chỉ:  
Trung Tâm tin Học ĐH KHTN  
227 Nguyễn Văn Cử P9 Q5  
Phone: (028) 8351056  
Email: tuvan@csc.hcmus.edu.vn

Cảm ơn bạn đã gửi ý kiến/ thông tin liên hệ

Social Link

Skype Call us Facebook Like us Twitter Follow us

Liên hệ:

Trung tâm tin học ĐH KHTN TP HCM  
Phone: (028) 8351056

Links

About ( Who we are )  
Services ( What we do )

Socialize

g+ linkedin twitter f

Sau khi người dùng nhập đầy đủ thông tin và gửi đi thì sẽ nhận được thông báo cảm ơn. Và nội dung sẽ được ghi nhận trong CSDL.

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Officemaster > Contacts > Trần Thị Thanh Thảo

Change contact

HISTORY

Name: Trần Thị Thanh Thảo

Phone: 0913100100

Email: thanhthao@gmail.com

Message: Vui lòng tư vấn cho tôi các sản phẩm của công ty

Delete Save and add another Save and continue editing SAVE



## ✓ Hướng dẫn

- Sử dụng lại project myoffice đã làm ở bài 3 (Mapping URLs, Template, Static File, View) để làm tiếp.
- Bước 1: trong officemaster/models.py tạo model Contact. Đăng ký model này với admin. Sau đó migrate officemaster để ghi nhận sự thay đổi này.

```
class Contact(models.Model):
    name = models.CharField(max_length = 264, blank = False)
    phone = models.CharField(max_length = 20)
    email = models.EmailField(blank = False)
    message = models.TextField()

    def __str__(self):
        return self.name

class Meta:
    db_table= u'Contact'
```

- Bước 2: trong officemaster /forms.py tạo form ContactForm:

```
from django import forms
from .models import Contact

class FormContact(forms.ModelForm):
    name = forms.CharField(max_length=100)
    phone = forms.CharField(max_length=20)
    email = forms.EmailField()
    message = forms.Textarea()

class Meta:
    model = Contact
    fields = [ 'name', 'phone', 'email', 'message']
```

- Bước 3: trong officemaster/views.py cập nhật lại view contact() đã viết ở bài 3:

```
def contact(request):
    registered = False
    if request.method == "POST":
        form_contact = forms.FormContact(data = request.POST)
        if form_contact.is_valid():
            register = form_contact.save()
```



```

register.save()

registered = True
else:
    print(form_contact.errors)
else:

    form_contact = forms.FormContact()
return render(request, "officemaster/contact.html", {'contact_form':form_contact,
'registered': registered})

```

- Bước 4: chỉnh lại trang contact.html với phần form đã viết:

```

...
<div class="col-lg-5">
    <div class="feedback-form">
        <div id="contact-response"></div>
        {% if registered %}
            <h3>Cảm ơn bạn đã gửi ý kiến/ thông tin liên
hệ</h3>
        {% else %}
            <table>
                <form method="post" id="contact-form">
                    <fieldset>
                        {% csrf_token %}
                        {% for field in contact_form%}
                            <tr><td>
                                <div class="form-group form-group-
fullname">
                                    <label class="control-label"
for="fullname">{{field.label}}</label>
                                    </div></td>
                                <td>
                                    <div>
                                        {{field}}
                                    </div></td>
                                {% endfor%}
                            <tr><td colspan="2" align="center">
                                <input type="submit" class="btn btn-
primary" value="Gửi Liên hệ">
                            </td></tr>
                        </fieldset>
                    </form>
                </table>
            {% endif%}
        </div>
    </div>
...

```



- Bước 5: chạy thử và xem kết quả

## 5.2. Form Đăng ký khóa học

- ✓ **Yêu cầu: Thực hiện form đăng ký khóa học như sau:**

- Từ trang homepage.html người dùng chọn 1 khóa học để xem chi tiết => chuyển sang trang course\_detail.html

homepage.html

Tên khóa học	Giá	Mô tả
LẬP TRÌNH PYTHON CƠ BẢN	23000000 vnd	"Lập trình Python Cơ bản" do TTH cung cấp
XEM THÊM		
LẬP TRÌNH PYTHON NÂNG CAO	2700000 vnd	"Lập trình Python Nâng Cao" do TTH cung cấp
XEM THÊM		
PHÁT TRIỂN UD WEB PYTHON VỚI DJANGO	6000000 vnd	"Phát triển ứng dụng Web Python với Django" do TTH cung cấp
XEM THÊM		
TRUY VẤN CSDL VỚI SQL SERVER 2014	1600000 vnd	"Truy vấn CSDL với SQL Server 2014" do TTH cung cấp
XEM THÊM		

Trang chủ   Các trang web   Các nội dung   Các khóa học   Đăng ký   Quản trị

## Lập trình Python Nâng Cao

Nhóm Chuyên đề

Học phí: 2700000 vnd

Thời lượng: 48 giờ

Nội dung: Khóa học cung cấp cho học viên (HV) những kiến thức và kỹ năng cần thiết để có thể xây dựng các loại ứng dụng tương tác với CSDL, desktop GUI (Graphical User Interface) ... Rèn luyện và phát triển kỹ năng tổ chức và xây dựng chương trình theo hướng đối tượng, có truy xuất tập tin XML, JSON, cơ sở dữ liệu SQLite, có xử lý đa luồng (multi-thread), áp dụng debugging & testing để tìm và bắt lỗi, nâng cao chất lượng phần mềm ... Xây dựng cơ sở vững chắc trong ngôn ngữ lập trình Python tạo tiền đề cho việc học kiến thức nâng cao như lập trình ứng dụng Desktop, Web, Game và phát triển nghề nghiệp.

Chứng chỉ "Lập trình Python Nâng Cao" do TTH cung cấp

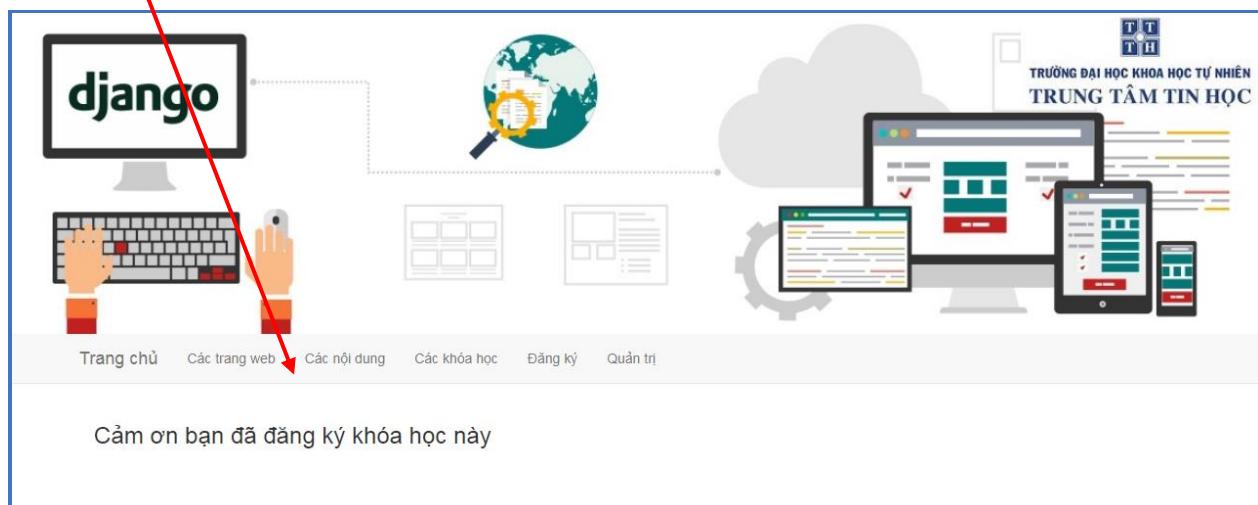
Đăng ký khóa học

Trang course\_detail.html

- Từ trang course\_detail.html, người dùng chọn "Đăng ký khóa học" => chuyển sang trang course\_registration.html

- Từ trang course\_registration.html, người dùng điền thông tin và nhấn "Đăng ký" để hoàn thành việc đăng ký khóa học.

The screenshot shows a Django-powered web application interface. At the top, there's a navigation bar with icons for a computer monitor labeled 'django', a magnifying glass over a globe, a cloud, and a mobile device. Below the navigation bar is a menu with links: Trang chủ, Các trang web, Các nội dung, Các khóa học, Đăng ký, Đăng xuất, and Quản trị. The main content area has a heading 'Vui lòng điền vào form đăng ký'. Below it is a form with fields: Course (Lập trình viên Android), Name (Nguyễn Thị Thanh Duyên), Email (thanhduyen@gmail.com), Phone (0989741258), and Date (October 17, 2017). A blue 'Đăng ký' button is at the bottom right of the form. A red arrow points from the 'Đăng ký' button to the second screenshot below.



Trang course\_registration.html

- Vào admin => Registrations để xem kết quả



Django administration

WELCOME ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > First\_App > Registrations > Nguyễn Thị Thanh Duyên

Change registration

HISTORY

Course: Lập trình viên Android

Name: Nguyễn Thị Thanh Duyên

Phone: 0989741258

Email: thanhduyen@gmail.com

Date: 2017-10-17 Today |

Note: You are 7 hours ahead of server time.

**Delete** **Save and add another** **Save and continue editing** **SAVE**

## ✓ Hướng dẫn

- Bước 1: trong first\_app/models.py tạo model Registration. Đăng ký model này với admin. Sau đó migrate first\_app để ghi nhận sự thay đổi này.

```
class Registration(models.Model):
    course = models.CharField(max_length=264, default="")
    name = models.CharField(max_length=264, blank=False)
    phone = models.CharField(max_length=20)
    email = models.EmailField()
    date = models.DateField()

    def __str__(self):
        return str(self.name)
```

- Bước 2: trong first\_app/forms.py tạo form RegistrationForm:

```
class RegistrationForm(forms.ModelForm):
    course = forms.CharField(max_length=264, required = True)
    name = forms.CharField(max_length=264, required = True)
    phone = forms.CharField(max_length=20)
    email = forms.EmailField()
    date = forms.DateField(widget=forms.SelectDateWidget(),
initial=datetime.date.today())

    class Meta():
        model = Registration
        fields = ('course', 'name', 'email', 'phone', 'date')
```

- Bước 3: trong first\_app/views.py tạo view course\_register:



```

def course_register(request):
    registered = False
    if request.method == "POST":
        form_register = forms.RegistrationForm(data = request.POST)
        if form_register.is_valid():
            register = form_register.save()
            register.save()
            registered = True
    else:
        print(form_register.errors)
    else:
        form_register = forms.RegistrationForm(initial={ 'course': request.GET.get( "khoa" ) })

    return render(request, "first_app/course_registration.html",
{'register_form':form_register, 'registered': registered})

```

- Bước 4: trong first\_app/urls, tạo url mới dành cho course\_register:

```

urlpatterns = [
    ...
    url(r'^course_registration/$',views.course_register,name='course_registration'),
]

```

- Bước 5: chỉnh lại trang course\_detail.html có thêm form “Đăng ký khóa học” dẫn qua trang course\_registration

```

...
<p>
    <form action="{% url 'first_app:course_registration' %}"
method="GET">
        {% csrf_token %}
        <input type="hidden" name="khoa" value="{{course.name}}">
        <input type="submit" name="c_submit" id="c_submit"
value="Đăng ký khóa học" class="btn btn-primary btn-lg">
    </form>
</p>
...

```

- Bước 6: viết trang course\_registration.html

```

{% extends "first_app/base.html" %}
{% load staticfiles %}

```



```
{% block body_block %}

<div class= "container">

    {% if registered %}

        <h3>Cảm ơn bạn đã đăng ký khóa học này</h3>

    {% else %}

        <h3>Vui lòng điền vào form đăng ký</h3>
        <table>
            <form method= "POST">
                {% csrf_token %}
                {% for field in register_form %}
                    <tr><td>{{field.label}} </td>
                    <td>{{field}}</td></tr>
                {% endfor %}
                {% for field in profile_form %}
                    <tr><td>{{field.label}} </td>
                    <td>{{field}}</td></tr>
                {% endfor %}

                <tr><td colspan= 2>
                    <p><input type="submit" class="btn btn-primary" value="Đăng ký"></p>
                    </td></tr>
                </form>
            </table>
        </div>
    {% endif %}
</div>

{% endblock %}
```

- Bước 7: chạy thử và xem kết quả

# BÀI 6: Template



*Mục tiêu chính:*

- Áp dụng cách đặt URL mới
- Sử dụng template inheritance

## 6.1. Xây dựng website ShopOnline

- ✓ Yêu cầu: Xây dựng website shoponline chuyên bán tivi – phần 1

Tivi	Kích thước	Mô tả	Giá	Link
Asanzo 50 inch	50"	Tivi Asanzo 4K màn hình phẳng 50 inch 9000000 vnd	9000000 vnd	<a href="#">Xem thêm</a>
Panasonic 40 inch	40"	Tivi Panasonic màn hình phẳng 40 inch 6590000 vnd	6590000 vnd	<a href="#">Xem thêm</a>
Samsung 32 inch	32"	Tivi Samsung thế hệ mới 32 inch 5999000 vnd	5999000 vnd	<a href="#">Xem thêm</a>
Sharp 45 inch	45"	Tivi Sharp màn hình cong siêu mỏng 7490000 vnd	7490000 vnd	<a href="#">Xem thêm</a>
	48"			
	55"			
	55"			
	32"			

Trang chủ (`index.html`)



ABC Shop Online

Home

**Sony 55 inch**  
Tivi Sony LED màn hình phẳng 55 inch  
20590000 vnđ

Copyright © Your Website 2017

### Trang Chi tiết sản phẩm (product\_detail.html)

#### ✓ Hướng dẫn

- Bước 1: tạo project web Django có tên là shoponline
- Bước 2: tạo thư mục static, media, templates và khai báo các thư mục này trong settings.py của project

```

...
TEMPLATE_DIR = os.path.join(BASE_DIR, "templates")
STATIC_DIR = os.path.join(BASE_DIR, "static")
MEDIA_ROOT = os.path.join(BASE_DIR, "media")

...
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_DIR],
        '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',
            ],
        },
    },
]
...
STATIC_URL = '/static/'

```



```
MEDIA_URL = '/media/'
STATICFILES_DIRS = [
    STATIC_DIR,
    MEDIA_ROOT,
]
```

- Khai báo sử dụng media trong urls.py của project

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', views.index, name='index'),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

- Bước 3: tạo web app có tên là mystore => đăng ký trong INSTALL\_APPS của settings.py
- Bước 4: tạo class Product trong models.py của mystore , sau đó migrate model này

```
from django.db import models

# Create your models here.

class Product(models.Model):
    name = models.CharField(max_length=264, unique=True)
    fee = models.IntegerField()
    description = models.TextField()
    image = models.ImageField(upload_to = "images/")

    def __str__(self):
        return str(self.name)
```

- Bước 5: đăng ký model này trong admin.py của webapp

```
from django.contrib import admin
from mystore.models import Product

# Register your models here.
admin.site.register(Product)
```

- Bước 6: vào Admin Interface để thêm dữ liệu cần thiết thiết (cần phải tạo tài khoản Admin khi muốn làm việc với Admin Interface)



<http://phuong13021982.pythonanywhere.com/admin/mystore/product/>

Django administration

WELCOME, ADMIN | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Mystore > Products

Select product to change

Action: ----- Go 0 of 8 selected

- PRODUCT
- TCL 55 inch
- Sony 48 inch
- Sharp 45 inch
- Samsung 32 inch
- Sony 55 inch
- Panasonic 40 inch
- Asanzo 50 inch
- Toshiba 32 inch

8 products

- Bước 7: tạo các view cần thiết

```
# Create your views here.

def index(request):
    pro_list = Product.objects.order_by('name')
    pro_dict = { "products":pro_list }
    return render(request, "mystore/index.html", context=pro_dict)

def product_detail(request, id=None):
    product = Product.objects.get(pk=id)
    return render(request, "mystore/product_detail.html", context= { 'product': product })
```

- Bước 8: tạo các URL cần thiết trong mystore/urls.py

```
app_name = "mystore"

urlpatterns = [
    url(r'^$',views.index,name='index'),
    url(r'^product_detail/(|d+)$', views.product_detail, name='product_detail'),
]
```

- Bước 9: đăng ký mystore.urls vào urls.py của web project

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$',views.index,name='index'),
    url(r'^mystore/$', include('mystore.urls')),
```



] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

– Bước 10: tạo các template HTML cần thiết

- Tạo trang base.html là trang khung, sẽ được kế thừa ở các trang khác (dựa vào template mẫu được cung cấp)

```
<!DOCTYPE html>
{% load staticfiles %}
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <meta name="description" content="">
        <meta name="author" content="">

        <title>Shop Online</title>
        <!-- Bootstrap core CSS -->
        <link href="{% static 'vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
        <!-- Custom styles for this template -->
        <link href="{% static 'css/heroic-features.css' %}" rel="stylesheet">
    </head>

    <body>

        <!-- Navigation -->
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
            <div class="container">
                <a class="navbar-brand" href="#">ABC Shop Online</a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarResponsive">
                    <ul class="navbar-nav ml-auto">
                        <li class="nav-item active">
                            <a class="nav-link" href="{% url 'mystore:index' %}>Home
                                <span class="sr-only">(current)</span>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </body>

```



```
</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'cart:cart_detail'%}" style="background-image: url({% static "images/card.png" %})">Cart</a>
</li>
</ul>
</div>
</div>
</nav>

<!-- Page Content --&gt;
&lt;div class="container"&gt;

    &lt;!-- Jumbotron Header --&gt;
    &lt;header class="jumbotron my-4"&gt;
        &lt;h1 class="display-3"&gt;ABC &lt;/h1&gt;
        &lt;p class="lead"&gt;All things you need are here! &lt;/p&gt;
        &lt;a href="#" class="btn btn-primary btn-lg"&gt;Contact!&lt;/a&gt;
    &lt;/header&gt;
&lt;/div&gt;

&lt;div class="container"&gt;
    {% block body_block %}
    &lt;!-- Anything outsite of this will be inherited if you extend! --&gt;
    {% endblock %}
&lt;/div&gt;

&lt;!-- Footer --&gt;
&lt;footer class="py-5 bg-dark"&gt;
    &lt;div class="container"&gt;
        &lt;p class="m-0 text-center text-white"&gt;Copyright &amp;copy; Your Website 2017&lt;/p&gt;
    &lt;/div&gt;
    &lt;!-- /.container --&gt;
&lt;/footer&gt;</pre>
```



```
<!-- Bootstrap core JavaScript -->
<script src="{% static 'vendor/jquery/jquery.min.js' %}"></script>
<script src="{% static 'vendor/popper/popper.min.js' %}"></script>
<script src="{% static 'vendor/bootstrap/js/bootstrap.min.js' %}">
</script>
</body>
</html>
```

- Tạo trang index.html là trang chính, có kế thừa base.html

```
<!DOCTYPE html>
{% extends "mystore/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <!-- Page Features -->
    <div class="row text-center">
        {% if products %}
            {% for value in products %}
                <div class="col-lg-3 col-md-6 mb-4">
                    <div class="card">
                        
                        <div class="card-body">
                            <h4 class="card-title">{{value.name}}</h4>
                            <p class="card-text">{{value.description}}<br/>{{value.fee}} vnd</p>
                        </div>
                    <div>
                        <a href="{% url 'mystore:product_detail' value.pk%}" class="btn btn-primary" style="float: center;"> Xem thêm </a>
                    </div>
                </div>
            {% endfor %}
            {% endif%}
        </div>
    {% endblock %}
```

- Tạo trang product\_detail.html, là trang chi tiết sản phẩm, có kế thừa base.html

```
<!DOCTYPE html>
{% extends "mystore/base.html" %}
```

```

{% block body_block %}
    {% load staticfiles %}

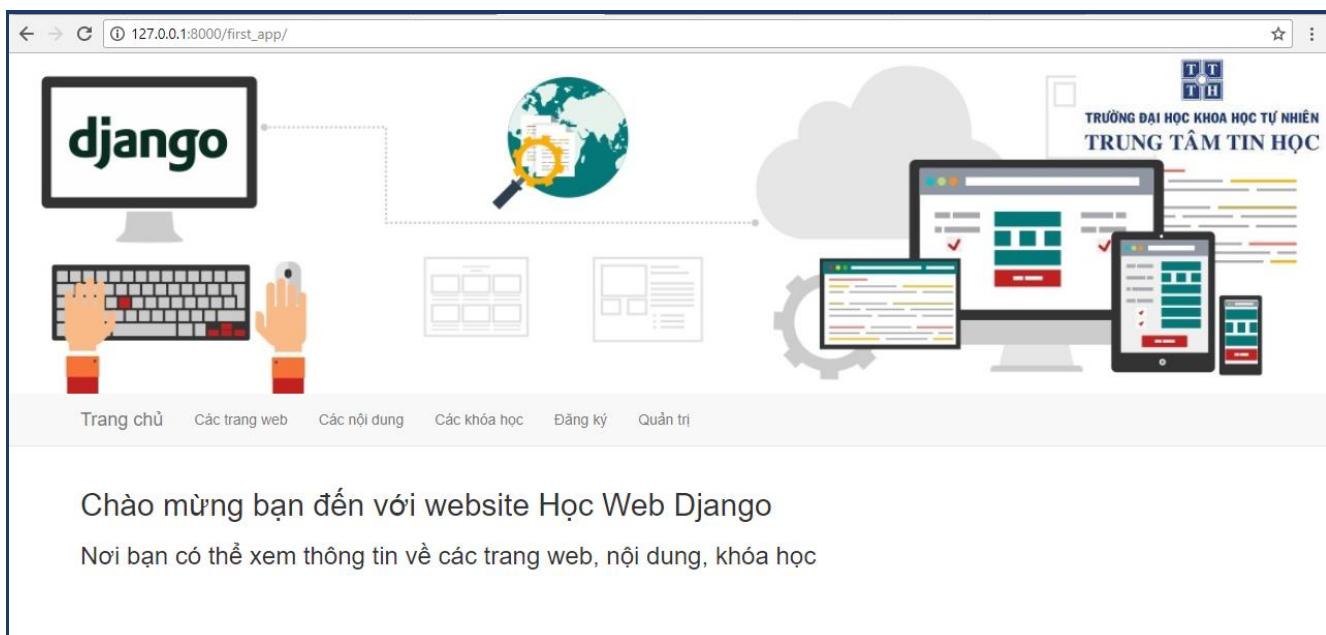
    <!-- Page Features -->
    <div class= "row text-center">
        {% if product %}
            <div class= "container">
                <div class= "card">
                    <img alt="{{product.name}}" width="350" height="350"/>
                    <div class= "card-body">
                        <h4 class= "card-title">{{product.name}}</h4>
                        <p class= "card-text">{{product.description}}<br/>{{product.fee}} vnđ</p>
                    </div>
                </div>
            </div>
        {% endif %}
    </div>
    {% endblock %}

```

- Bước 10: chạy thử và xem kết quả

## 6.2. Hiệu chỉnh website trong bài Database - Models

- ✓ **Yêu cầu: Hãy hiệu chỉnh lại website trong bài Database – Models có sử dụng inheritance:**





## ✓ Hướng dẫn

- Bước 1: vào urls.py của first\_app, chỉnh lại URL theo cách mới:

```
app_name = "first_app"

urlpatterns = [
    url(r'^$', views.mainpage, name='mainpage'),
    url(r'^webpages/$', views.webpages, name='webpages'),
    url(r'^contents/$', views.contents, name='contents'),
    url(r'^homepages/$', views.homepages, name='homepages'),
    url(r'^homepages/course_detail/(\d+)/$', views.course_detail, name='course_detail'),
]
```

- Bước 2: trong templates/first\_app, tạo base.html chứa những phần hiển thị chung của website như header, menu

```
<!DOCTYPE html>
{% load staticfiles %}
<html>
<head>
    <meta charset="utf-8">
    <title>HOME</title>
    <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
        integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
        crossorigin="anonymous">
</head>
<body>
    <div>
        
    </div>
    <nav class="navbar navbar-default navbar-static-top">
        <div class="container">
            <ul class="nav navbar-nav">
                <li><a class="navbar-brand" href="{% url 'first_app:mainpage' %}">Trang chủ</a></li>
                <li><a class="navbar-link" href="{% url 'first_app:webpages' %}">Các trang web</a></li>
                <li><a class="navbar-link" href="{% url 'first_app:contents' %}">Contents</a></li>
            </ul>
        </div>
    </nav>
</body>
```



```
%}">Các nội dung</a></li>
    <li><a class="navbar-link" href="{% url
'first_app:homepages' %}">Các khóa học</a></li>
    <li><a class="navbar-link" href="{% url 'first_app:register'
%}">Đăng ký</a></li>
    <li><a class="navbar-link" href="{% url 'admin:index'
%}">Quản trị</a></li>
</ul>
</div>
</nav>
<div class="container">
    {% block body_block %}
    <!-- Anything outside of this will be inherited if you extend! -->
    {% endblock %}
</div>

</body>
</html>
```

- Bước 3: chỉnh lại các trang html bằng cách kế thừa trang base.html
  - Trang mainpage.html

```
<!DOCTYPE html>
{% extends "first_app/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <h2>Chào mừng bạn đến với website Học Web Django</h2>
    <h3>Nơi bạn có thể xem thông tin về các trang web, nội dung, khóa học</h3>
{% endblock %}
```

- Trang webpages.html

```
<!DOCTYPE html>
{% extends "first_app/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <h1>Danh sách trang web</h1>
    {% if webpages%}
        <ol>
            {% for web in webpages %}
```

```

<li>Tên trang: {{web.name}}</li>
<ul>
    <li>ID: {{web.pk}} </li>
    <li>Thuộc Topic: {{web.topic}}Name: {{web.name}}</li>
    <li>URL: {{web.url}}</li>
</ul>
{% endfor %}
</ol>
{% endif %}
{% endblock %}

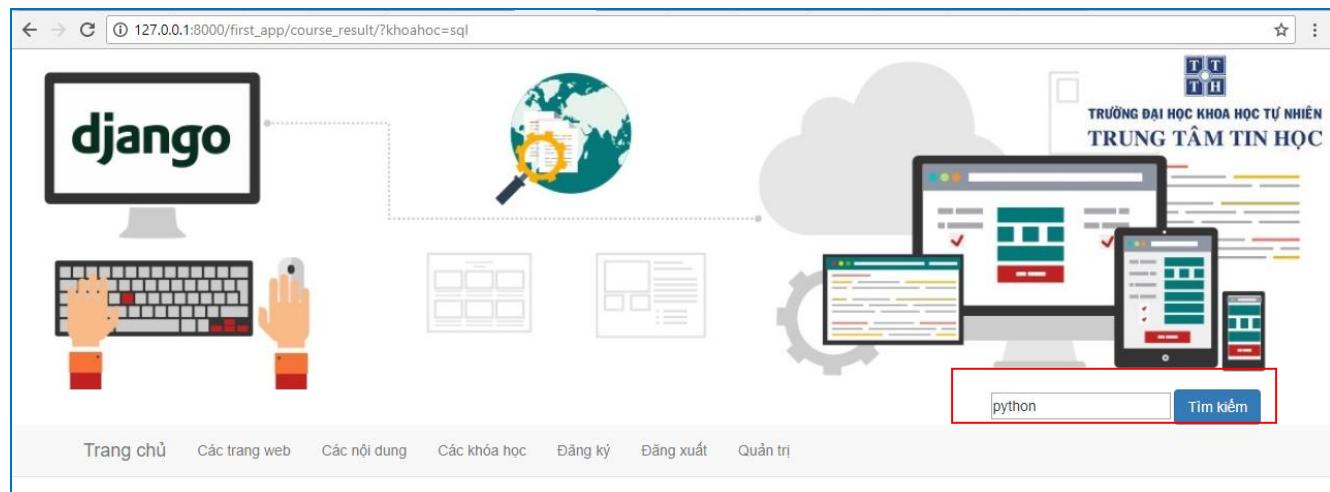
```

- Làm tương tự cho các trang còn lại

### 6.3. Form Tìm kiếm khóa học

✓ **Yêu cầu: Thực hiện form tìm kiếm khóa học như sau:**

- Người dùng ở một trang bất kỳ trong website gõ vào tên khóa học cần tìm (tìm tương đối)



*Trang bất kỳ*

- Hiển thị kết quả tương ứng



Nhập tên khóa học cần tìm

Trang chủ Các trang web Các nội dung Các khóa học Đăng ký Đăng xuất Quản trị

## Tìm thấy 3 khóa học

### Lập trình Python Cơ bản

Nhóm Chuyên đề

Học phí: 2300000 vnđ

Thời lượng: 36 giờ

[Xem thêm](#)

### Lập trình Python Nâng Cao

Nhóm Chuyên đề

Học phí: 2700000 vnđ

Thời lượng: 48 giờ

[Xem thêm](#)

### Phát triển UD Web Python với Django

Nhóm Chuyên đề

Học phí: 2800000 vnđ

Thời lượng: 48 giờ

[Xem thêm](#)

*Trang course\_result.html*



Nhập tên khóa học cần tìm

Trang chủ Các trang web Các nội dung Các khóa học Đăng ký Đăng xuất Quản trị

## Lập trình Python Cơ bản

Nhóm Chuyên đề

Học phí: 2300000 vnđ

Thời lượng: 36 giờ

Nội dung: Khóa học cung cấp cho học viên (HV) những kiến thức nền tảng và những kỹ năng cần thiết để có thể xây dựng các ứng dụng đơn giản bằng Python – một ngôn ngữ lập trình cấp cao, thông dịch, hướng đối tượng và đa mục đích. Rèn luyện và phát triển kỹ năng lập trình, tư duy logic. Xây dựng nền tảng cơ bản vững chắc trong ngôn ngữ lập trình Python tạo tiền đề cho việc học các kiến thức lập trình nâng cao, lập trình Web và phát triển nghề nghiệp.

Chứng chỉ "Lập trình Python Cơ bản" do TTTH cấp



[Đăng ký khóa học](#)



Chọn "Xem thêm" => chuyển sang trang *course.detail.html*

## ✓ Hướng dẫn

- Bước 1: trong base.html của templates, bổ sung thêm form tìm kiếm ở phía phải trên của menu:

```
<div class = "container" align= "right">
    <form action="{% url 'first_app:course_result' %}" method="GET">
        <input type="text" name="khoa_hoc" placeholder="Nhập tên khóa học cần
tìm">
        <input type="submit" class="btn btn-primary" value="Tìm kiếm">
    </form>
</div>
```

- Bước 2: viết view course\_result() trong views.py của web app để thực hiện việc tìm kiếm và trả kết quả về

```
def course_result(request):
    course_list = Course.objects.order_by('name')
    print(course_list)
    course_result = []
    count = 0
    if request.method == "GET":
        khoahoc = request.GET.get("khoa_hoc")
        for course in course_list:
            if khoahoc.lower() in course.name.lower():
                course_result.append(course)
        if len(course_result) > 0:
            count = len(course_result)

    return render(request, "first_app/course_result.html", context = {"course_result":course_result, "count":count})
```

- Bước 3: bổ sung thêm url trong urls.py của webapp

```
url(r'^course_result/$',views.course_result, name='course_result'),
```

- Bước 4: tạo trang course\_result.html trong templates để hiển thị kết quả và dẫn link qua trang course\_detail.html để xem chi tiết một khóa học khi người dùng chọn "Xem thêm"

```
<!DOCTYPE html>
{% extends "first_app/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <div class="container">
```



```

{%- if count%}
    <ol><h3><font color= "red">Tìm thấy {{count}} khóa học</font></h3></ol>
{%- endif %}
</div>

<div class= "container">
{%- if course_result%}
    <ol>
        {% for course in course_result %}
            <h3>{{course.name}}</h3>
            <h4>Nhóm {{course.category}}</h4>
            <p>Học phí: {{course.fee}} vnđ</p>
            <p>Thời lượng: {{course.duration}} giờ</p>
            <p><a href="{% url 'first_app:course_detail' course.pk %}" class='btn btn-primary' style="float: center;" >
                Xem thêm</a>
            </p>
        {% endfor %}
    </ol>
{%- endif %}
</div>
{%- endblock %}

```

- Bước 5: chạy thử và xem kết quả

#### 6.4. Hiệu chỉnh website trong bài 3 - Mapping URLs, Template, Static File, View

- ✓ **Yêu cầu: Hãy hiệu chỉnh lại website myoffice trong bài 3 - Mapping URLs, Template, Static File, View có sử dụng inheritance:**



*Trang chủ (index.html)*

✓ **Hướng dẫn**

- Làm tương tự như bài 6.2

# BÀI 7: Cookie và Session



Mục tiêu chính:

- Áp dụng cookie và session trong web app
- Thực hành tạo và sử dụng giỏ hàng

## 7.1. Sử dụng cookie để hiển thị lần đăng nhập cuối

- ✓ Yêu cầu: tạo cookie để hiển thị lần đăng nhập cuối

Trang chủ

Trang readcookie.html

- ✓ Hướng dẫn

- Trong web app second\_app, mở module views.py. Bổ sung thêm cookie vào function index():

```
def index(request):
    date1 = datetime.now()
    value = request.COOKIES.get('last_visit', date1.strftime('%d-%m-%Y %H:%M:%S'))
    response = render(request, "second_app/index.html", {'last_visit':value})
    response.set_cookie('last_visit', date1.strftime('%d-%m-%Y %H:%M:%S'))
    return response
```

- Viết function read\_cookie() để lấy nội dung có trong cookie và hiển thị

```
def read_cookie(request):
```



```
value = request.COOKIES.get('last_visit')
text = "<h2>The last time I come here is %s</h1>" % value
return HttpResponse(text)
```

- Trong urls.py của second\_app, khai báo thêm đường dẫn mới tương ứng với read\_cookie:

```
from django.conf.urls import url
from second_app import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
    ...
    url(r'^readcookie.html/$', views.read_cookie, name = 'readcookie'),
]
```

- Thêm giá trị cookie vào trang index.html trong templates\second\_app

```
<div class="container">
    <div class="jumbotron">
        <div>
            </div>
            <h1 class="text-danger">Welcome to HOMEPAGE</h1>
            <h3>Last time I come here {{last_visit}} </h3>
            <h3><a href="second_app/readcookie.html">Read
Cookie</a></h3>
        </div>
    </div>
```

## 7.2. Sử dụng session để xây dựng Giỏ hàng

- ✓ **Yêu cầu:** Xây dựng giỏ hàng trong web app như sau:
  - Ở trang chi tiết sản phẩm, người dùng có thể chọn số lượng và thêm hàng vào giỏ



ABC Shop Online

Home

Sony 55 inch  
Tivi Sony LED màn hình phẳng 55 inch  
20590000 vnd  
Số lượng: 1 ▾  
Thêm vào giỏ hàng

Copyright © Your Website 2017

- Ở trang chi tiết giỏ hàng người dùng có thể cập nhật giỏ hàng hoặc xóa hàng trong giỏ hàng

ABC Shop Online

Home

### Thông tin giỏ hàng

Sản phẩm	Tên sản phẩm	Số lượng	Đơn giá (vnd)	Thành tiền (vnd)	Xóa
	Asanzo 50 inch	2 ▾ Cập nhật	900000	1800000	Xóa
	Sony 55 inch	1 ▾ Cập nhật	20590000	20590000	Xóa

✓ **Hướng dẫn**



- Tiếp tục làm với web project shoponline đã xây dựng ở bài Template
- Bước 1: Tạo web app cart để chứa những gì liên quan đến giỏ hàng => đăng ký cart trong INSTALL\_APPS của settings.py của shoponline
- Bước 2: Trong settings.py của shoponlie:
  - Khai báo cart trong INSTALL\_APPS
  - Khai báo biến CART\_SESSION\_ID = 'cart'
- Bước 3: Trong web app cart tạo module cart.py để chứa class Cart với các phương thức như : \_\_init\_\_(), add(), save(), remove, \_\_iter\_\_(), \_\_len\_\_(), get\_total\_fee(self)

```

from django.conf import settings
from mystore.models import Product

class Cart(object):
    def __init__(self, request):
        """
        initialize the session
        """
        self.session = request.session
        cart = self.session.get(settings.CART_SESSION_ID)

        if not cart:
            cart = self.session[settings.CART_SESSION_ID] = {}
        self.cart = cart

    def add(self, product, quantity= 1, update_quantity = False):
        """
        add the product to the card or update its quantity
        """
        product_id = str(product.pk)
        if product_id not in self.cart:
            self.cart[product_id] = { 'quantity':0, 'fee': str(product.fee)}
        if update_quantity:
            self.cart[product_id]['quantity']= quantity
        else:
            self.cart[product_id]['quantity'] += quantity
        self.save()

    def save(self):

```

```

self.session[settings.CART_SESSION_ID] = self.cart
self.session.modified = True

def remove(self, product):
    product_id = str(product.pk)
    if product_id in self.cart:
        del self.cart[product_id]
        self.save()

def __iter__(self):
    """
    iterate over the items in card and get the product from the database
    """
    product_ids = self.cart.keys()
    products = Product.objects.filter(id__in=product_ids)
    for product in products:
        self.cart[str(product.pk)][product] = product
    for item in self.cart.values():
        item['fee'] = item['fee']
        item['total_fee'] = int(item['fee']) * item['quantity']
        yield item

def __len__(self):
    """
    count all items in the cart
    """
    return sum(int(item['quantity']) for item in self.cart.values())

def get_total_fee(self):
    """
    get total fee of product
    """
    return sum((int(item['fee'])) * int(item['quantity']) for item in self.cart.values())

```

- Bước 4: trong web app cart, tạo module forms.py trong đó có CartAddProductForm để cho người dùng có thể nhập số lượng

```
from django import forms
```



```
PRODUCT_QUANTITY_CHOICES = [(i, str(i)) for i in range(1, 10)]
```

```
class CardAddProductForm(forms.Form):
    quantity = forms.TypedChoiceField(label= 'Số lượng', choices =
PRODUCT_QUANTITY_CHOICES, coerce=int)
    update = forms.BooleanField(required = False, widget = forms.HiddenInput)
```

- Bước 5: trong web app cart, mở views.py và tạo các view cần thiết cho cart như: cart\_add(), cart\_remove(), cart\_detail()

```
from django.shortcuts import render, redirect, get_object_or_404
from django.views.decorators.http import require_POST
from mystore.models import Product
from .cart import Cart
from .forms import CardAddProductForm

# Create your views here.

@require_POST
def cart_add(request, product_id=None):
    cart = Cart(request)
    product = get_object_or_404(Product, id = product_id)
    form = CardAddProductForm(request.POST)
    if form.is_valid():
        cd = form.cleaned_data
        cart.add(product = product, quantity = cd[ 'quantity'], update_quantity = cd[ 'update'])
        print(cart)
    return redirect( 'cart:cart_detail')

def cart_remove(request, product_id=None):
    cart = Cart(request)
    product = get_object_or_404(Product, id = product_id)
    cart.remove(product)
    return redirect( 'cart:cart_detail')

def cart_detail(request):
    cart = Cart(request)
```



```

for item in cart:
    item[ 'update_quantity_form' ] = CardAddProductForm(initial = { 'quantity': item[ 'quantity' ], 'update': True } )
return render(request, 'cart/cart_detail.html', { 'cart':cart})

```

- Bước 6: tạo url cần thiết trong urls.py của web app cart:

```

from django.conf.urls import url
from cart import views
from django.conf.urls.static import static
from django.conf import settings
app_name = "cart"

urlpatterns = [
    url(r'^cart_add/(|d+)/$',views.cart_add,name= 'cart_add'),
    url(r'^cart_remove/(|d+)/$',views.cart_remove,name= 'cart_remove'),
    url(r'^cart_detail/$',views.cart_detail,name= 'cart_detail'),
]

```

- Bước 7: đưa cart.urls vào urls.py của shoponline

```

from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin
from mystore import views
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^cart/', include( 'cart.urls' )),
    url(r'^$',views.index,name= 'index'),
    url(r'^mystore/', include('mystore.urls')),

]
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

- Bước 8: trong templates/mystore/product\_details.html, bổ sung form cho người dùng thêm hàng vào giỏ:

```

<!DOCTYPE html>
{% extends "mystore/base.html" %}
{% block body_block %}

```



```

    {% load staticfiles %}

    <!-- Page Features -->

    <div class= "row text-center">

        {% if product %}

            <div class= "container">

                <div class= "card">

                    <div class= "card-body">
                        <h4 class= "card-title">{{product.name}}</h4>
                        <p class= "card-text">{{product.description}}<br/>{{product.fee}} vnd</p>
                    </div>

                <div>

                    <form action="{% url 'cart:cart_add' product.pk%}"
                    method="post">

                        {{cart_product_form}}
                        {% csrf_token %}
                        <div class= "card-footer">
                            <input type="submit" value = "Thêm vào giỏ hàng"
                            class= "btn btn-primary">
                        </div>
                    </form>

                </div>
                </div>
            </div>
        {% endif%}
    </div>
    {% endblock %}

```

- Chú ý: cần phải cập nhật lại view product\_detail.html có thêm phần form CardAddProduct

```

def product_detail(request, id=None):
    product = Product.objects.get(pk=id)
    cart_product_form = forms.CardAddProductForm()
    return render(request, 'mystore/product_detail.html', context= {'product': product,
    'cart_product_form': cart_product_form})

```

- Bước 9: tạo thư mục cart trong template => tạo cart\_detail.html để hiển thị giỏ hàng

```
{% extends "mystore/base.html" %}
```



```
{% block body_block %}

    {% load staticfiles %}

    <div class="container"></div>
    <h1>Thông tin giỏ hàng</h1>
    <table class="table">

        <thead class="danger">
            <th>Sản phẩm</th>
            <th>Tên sản phẩm</th>
            <th>Số lượng</th>
            <th>Đơn giá (vnđ)</th>
            <th>Thành tiền (vnđ)</th>
            <th>Xóa</th>
        </thead>
        <tbody>

            {% for item in cart %}

                <tr class="success">
                    <td>
                        <a href="{% url 'mystore:product_detail' item.product.pk%}">
                            {{item.update_quantity_form.quantity}}
                            {{item.update_quantity_form.update}}
                            <input type = "submit" value = "Cập nhật"/>
                            {% csrf_token %}
                        </form>
                    </td>
                    <td>{{item.fee}}</td>
                    <td>{{item.total_fee}}</td>
                    <td><a href="{% url 'cart:cart_remove' item.product.pk%}">Xóa</a></td>
                </tr>
            {% endfor %}
```



```

</tbody>
</table>
</div>
{% endblock %}

```

- Bước 10: chỉnh/ bổ sung các đường dẫn cần thiết trong base.html của template/mystore/base.html

```

<div class="collapse navbar-collapse" id="navbarResponsive">
    <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
            <a class="nav-link" href="{% url 'mystore:index' %}">Home
                <span class="sr-only">(current)</span>
            </a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{% url 'cart:cart_detail' %}" style="background-image: url({% static "images/card.png" %})">Cart</a>
        </li>
    </ul>
</div>

```

### 7.3. Tạo context processor cho cart hiện hành

- ✓ **Yêu cầu:** Xây dựng giỏ hàng trong web app như sau: thực hiện việc tạo context processor cho cart hiện hành hiển thị tổng số sản phẩm và tổng giá:

The screenshot shows a Django-based e-commerce website. At the top, there's a header bar with the text "ABC Shop Online", a "Home" link, a shopping cart icon, and a message indicating 4 items in the cart totaling \$31369000 vnđ. Below the header is a large, colorful "ONLINE SHOPPING" banner featuring a cursor icon pointing at the word "ONLINE". To the left of the banner is a promotional image of two shopping bags, one purple and one orange, with icons like a laptop, a calculator, and a shopping cart. A blue button labeled "Contact!" is visible on the bag. Below the banner, there are four product cards displayed in a grid:

- Asanzo 50 inch**: Tivi Asanzo 4K màn hình phẳng 50 inch 9000000 vnđ. Includes a "Xem thêm" button.
- Panasonic 40 inch**: Tivi Panasonic màn hình phẳng 40 inch 6590000 vnđ. Includes a "Xem thêm" button.
- Samsung 32 inch**: Tivi Samsung thế hệ mới 32 inch 5999000 vnđ. Includes a "Xem thêm" button.
- Sharp 45 inch**: Tivi Sharp màn hình cong siêu mỏng 7490000 vnđ. Includes a "Xem thêm" button.

## ✓ Hướng dẫn

- Bước 1: tạo context\_processors.py có phương thức cart(request) trong cart app

```
from cart.cart import Cart

def cart(request):
    return {'cart' : Cart(request)}
```

- Bước 2: khai báo context\_processors.cart trong settings.py

```
TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [TEMPLATE_DIR,],
    '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',
            'cart.context_processors.cart',
        ],
    },
}]
```



- Bước 3: hiển thị thông tin thông báo trong giỏ hàng hiện có bao nhiêu sản phẩm và tổng số tiền là bao nhiêu: cập nhật base.html

```
<li class="nav-item">
    <a class="nav-link" href="{% url 'cart:cart_detail'%}"> </a>
</li>
<li class="cart">
    {% if cart %}
        <a class="nav-link" href="{% url 'cart:cart_detail'%}">
            Giỏ hàng có {{cart|length}} sản phẩm , VNĐ: {{cart.get_total_fee}}</a>
    {% else %}
        Giỏ hàng đang trống
    {% endif %}
</li>
```

- Bước 4: chạy thử và xem kết quả

## 7.4. Người dùng đặt hàng

- ✓ Yêu cầu: cho phép người dùng đặt hàng như sau:

The screenshot shows a shopping cart page from 'ABC Shop Online'. At the top, there's a navigation bar with 'Home' and a shopping cart icon showing 'Giỏ hàng có 2 sản phẩm, \$ 23198000 vnd'. Below the header, there's a large 'ONLINE SHOPPING' logo with a red cursor icon pointing at the 'O'.

**Thông tin giỏ hàng**

Sản phẩm	Tên sản phẩm	Số lượng	Đơn giá (vnđ)	Thành tiền (vnđ)	Xóa
	Sony 48 inch	2 ▾ Cập nhật	11599000	23198000	Xóa

Tổng: 23198000 vnđ

[Tiếp tục mua hàng](#) [Thanh toán](#)

Copyright © Your Website 2017

Trang cart\_detail.html



ABC Shop Online

Home Giỏ hàng có 2 sản phẩm , \$ 23198000 vnd

ONLINE SHOPPING

Khách hàng

Chi tiết đơn hàng

Copyright © Your Website 2017

### Trang thông tin thanh toán (create.html)

ABC Shop Online

Home Giỏ hàng đang trống

ONLINE SHOPPING

Đặt hàng thành công

Đơn đặt hàng của bạn đã đặt thành công. Mã đơn hàng 12

Copyright © Your Website 2017

### Trang thông báo đặt hàng xong (created.html)

✓ **Hướng dẫn:**

- Bước 1: tạo web app orders cho phép người dùng đặt hàng => đăng ký vào INSTALL\_APPS trong settings.py của project



- Bước 2: trong models.py của orders tạo 2 model mới là Order và OrderItem => migration

```
from django.db import models
from mystore.models import Product

# Create your models here.
class Order(models.Model):
    first_name = models.CharField(max_length = 50)
    last_name = models.CharField(max_length = 50)
    email = models.EmailField()
    address = models.CharField(max_length = 250)
    postal_code = models.CharField(max_length = 20)
    city = models.CharField(max_length = 100)
    created = models.DateTimeField(auto_now = True)
    updated = models.DateTimeField(auto_now = True)
    paid = models.BooleanField(default = False)

    class Meta:
        ordering = ('-created',)

    def __str__(self):
        return 'Order{}'.format(self.id)

    def get_total_cost(self):
        return sum(item.get_cost() for item in self.items.all())

class OrderItem(models.Model):
    order = models.ForeignKey(Order, related_name= 'items')
    product = models.ForeignKey(Product, related_name= 'items')
    price = models.IntegerField()
    quantity = models.PositiveIntegerField(default = 1)

    def __str__(self):
        return '{}'.format(self.id)

    def get_cost(self):
        return self.price * self.quantity
```

- Bước 3: đăng ký với admin.py

```
from django.contrib import admin
from orders.models import Order, OrderItem

# Register your models here.
class OrderItemInline(admin.TabularInline):
    model = OrderItem
    raw_id_fields = ['product']

class OrderAdmin(admin.ModelAdmin):
    list_display = ['id', 'first_name', 'last_name', 'email', 'address',
    'postal_code', 'city', 'created', 'updated', 'paid']
    list_filter = ['paid', 'created', 'updated']
    inlines = [OrderItemInline]

admin.site.register(Order, OrderAdmin)
```

- Bước 4: kiểm tra kết quả trong admin



[127.0.0.1:8000/admin/orders/order/add/](http://127.0.0.1:8000/admin/orders/order/add/)

Django administration

WELCOME, KTPHUONG. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Orders > Orders > Add order

Add order

First name:

Last name:

Email:

Address:

Postal code:

City:

Paid

**ORDER ITEMS**

PRODUCT	PRICE	QUANTITY	DELETE?
<input type="text"/> Q	<input type="text"/>	<input type="text" value="1"/>	<input type="button" value="Delete"/>
<input type="text"/> Q	<input type="text"/>	<input type="text" value="1"/>	<input type="button" value="Delete"/>
<input type="text"/> Q	<input type="text"/>	<input type="text" value="1"/>	<input type="button" value="Delete"/>
<a href="#">+ Add another Order item</a>			

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

- Bước 5: tạo forms.py trong orders app

```
from django import forms
from orders.models import Order

class OrderCreateForm(forms.ModelForm):
    class Meta:
        model = Order
        fields = ['first_name', 'last_name', 'email',
                  'address', 'postal_code', 'city']
```

- Bước 6: vào cart.py của cart app, bổ sung phương thức clear cho Cart sau khi người dùng đã đặt hàng xong:

```
def clear(self):
    self.cart = {}
    self.save()
```

- Bước 7: viết view order\_create trong views.py

```
from django.shortcuts import render
from orders.models import OrderItem
from orders.forms import OrderCreateForm
from cart.cart import Cart

# Create your views here.
def order_create(request):
    cart = Cart(request)
    form = OrderCreateForm()
```

```

if request.method == 'POST':
    print("I'm here")
    form = OrderCreateForm(request.POST)
    if form.is_valid():
        order = form.save()
        for item in cart:
            OrderItem.objects.create(order = order, product =
item['product'], price = item['fee'], quantity = item['quantity'])
        #clear the cart
        cart.clear()

        return render(request, 'orders/created.html', context = {'order' :
order})
    else:
        form = OrderCreateForm()

return render(request, 'orders/create.html', context={'form':form})

```

- Bước 8: tạo urls.py trong orders app và cập nhật url cho order\_create vừa viết ở bước 7

```

from django.conf.urls import url
from orders import views

urlpatterns = [
    url(r'^create/$', views.order_create, name = 'order_create'),
]

```

- Bước 9: đăng ký urls của orders trong urls.py của project

```
url(r'^orders/$', include('orders.urls', namespace="orders")),
```

- Bước 10: hiệu chỉnh lại cart\_detail.html có thêm link dẫn qua trang thanh toán

```

...
<tr class='total'>
    <td colspan="4"></td>
    <td class="num">Tổng: {{cart.get_total_fee}} vnd</td>
</tr>
</tbody>
</table>
<p class="text-right">
    <a href="{% url 'mystore:index' %}" class="button Light">Tiếp tục mua
hàng</a>
    <a href="{% url 'orders:order_create'%}" class="btn btn-primary">Thanh
toán</a>
</p>
</div>
{% endblock %}

```

- Bước 11: trong thư mục templates tạo thư mục orders => tạo trang create.html trong thư mục templates/orders cho phép người dùng nhập thông tin và xác nhận việc đặt hàng:

```

{% extends "mystore/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <div class="container"></div>
    <table align="center">
        <tr>
            <td valign="top" width='50%'>
                <h1>Khách hàng</h1>
                <form action"." method="post">

```

```

        {% for field in form %}
        <tr>
            <td>{{field.label}} </td>
            <td>{{field}}</td>
        </tr>
        {% endfor %}
        <tr><td colspan="2" align="center">
            <p><input type = "submit" value = "Đặt hàng" class="btn
btn-primary"></p></td>></tr>
        </table>
        {% csrf_token %}
        </form>
    </td>

    <td valign="top">
        <h1>Chi tiết đơn hàng</h1>
        <ul>
            {% for item in cart %}
            <li>
                {{item.quantity}} tivi x
                {{item.product.name}} = <span>{{item.total_fee}}</span>
            </li>
            {% endfor %}
        </ul>
        <ul>Tổng: {{cart.get_total_fee}} vnd</ul>
    </td>
</tr>
</table>
</div>
{% endblock %}

```

- Bước 12: tạo trang created.html trong templates/orders xác nhận đặt hàng thành công:

```

{% extends "mystore/base.html" %}
{% block body_block %}
    <div class="container">
        <h1>Đặt hàng thành công</h1>
        <p>Đơn đặt hàng của bạn đã đặt thành công. Mã đơn hàng
        <strong>{{order.id}}</strong></p>
    </div>
{% endblock %}

```

- Chạy thử và xem kết quả

## BÀI 8: User Authentication



**Mục tiêu chính:**

- *User Model*
- *Registration*
- *Login - logout*

### 8.1. Xây dựng form đăng ký Register

- ✓ **Yêu cầu: Đăng ký một tài khoản mới vào website**
  - Người dùng chọn “Đăng ký” => vào form đăng ký.

- Người dùng điền thông tin hợp lệ => “Đăng ký” => thông báo đăng ký thành công

- Một tài khoản mới được tạo, có thể kiểm tra trong admin



127.0.0.1:8000/admin/auth/user/5/change/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Authentication and Authorization > Users > xuanxuan

Change user

Username: xuanxuan  
Required: 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password: pbkdf2\_sha256 iterations: 36000 salt: srGPCU\*\*\*\*\* hash: 7UvZmg\*\*\*\*\*  
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

**Personal info**

First name:

Last name:

Email address: xuanxuan@gmail.com

## ✓ Hướng dẫn

- Bước 1: tạo model UserProfileInfo trong models.py của web app => migrate web app

```
from django.contrib.auth.models import User
```

```
#Tạo class UserProfileInfor
class UserProfileInfo(models.Model):
    #create relationship from this class to User
    user = models.OneToOneField(User)
    #add any more attribute you want
    portfolio = models.URLField(blank=True)
    picture = models.ImageField(upload_to = "images/", blank=True)

    def __str__(self):
        return self.user.username
```

- Bước 2: Tạo các form cần thiết cho việc đăng ký trong forms.py của web app

```
class UserForm(forms.ModelForm):
    password = forms.CharField(widget = forms.PasswordInput())
```

```
class Meta():
    model = User
    fields = ('username', 'email', 'password')
```

```
class UserProfileInfoForm(forms.ModelForm):
    #porfolio = forms.URLField(required = False)
```



```
picture = forms.ImageField(required = False)
```

```
class Meta():
    model = UserProfileInfo
    fields = ('portfolio', 'picture')
```

- Bước 3: tạo view register trong views.py của web app

```
def register(request):
    registered = False
    if request.method == "POST":
        form_user = forms.UserForm(data = request.POST)
        form_por = forms.UserProfileInfoForm(data = request.POST)
        if form_user.is_valid() and form_por.is_valid():
            user = form_user.save()
            user.set_password(user.password)
            user.save()

            profile = form_por.save(commit = False)
            profile.user = user

            print(request.FILES)
            if 'picture' in request.FILES:
                profile.picture = request.FILES['picture']

            profile.save()

            registered = True
    else:
        print(form_user.errors, form_por.errors)
    else:
        form_user = forms.UserForm()
        form_por = forms.UserProfileInfoForm()

    return render(request, "first_app/registration.html", { 'user_form':form_user,
                                                          'profile_form': form_por,
                                                          'registered': registered})
```

- Bước 4: cập nhật url trong urls.py của web app



```
url(r'^register/$', views.register, name='register'),
```

- Bước 5: tạo trang registration.html trong template

```
{% extends "first_app/base.html" %}

{% load staticfiles %}

{% block body_block %}

<div class="container">

    {% if registered %}

        <h3>Bạn đã đăng ký thành công.</h3>

    {% else %}

        <h3>Điền vào form đăng ký</h3>

        <form method="POST" enctype="multipart/form-data">

            <table>

                {% for field in user_form %}

                    <tr><td>{{field.label}} </td>
                    <td>{{field}}</td></tr>

                {% endfor %}

                {% for field1 in profile_form %}

                    <tr><td>{{field1.label}} </td>
                    <td>{{field1}}</td></tr>

                {% endfor %}

                {% csrf_token %}

                <tr><td colspan='2' align="center">
                <input type="submit" class="btn btn-primary" value="Đăng ký">
                </td></tr>

            </table>

        </form>

    </div>

    {% endif %}

</div>

{% endblock %}
```

- Bước 7: Bổ sung link Đăng ký trong base.html của template

```
<li><a class="navbar-Link" href="{% url 'first_app:register' %}">Đăng
ký</a></li>
```

- Bước 8: chạy thử và kiểm tra kết quả

## 8.2. Xây dựng form đăng nhập



✓ **Yêu cầu: Thực hiện việc đăng nhập – đăng xuất khỏi ứng dụng**

- Người dùng nhấn “Đăng nhập” trên menu => điền thông tin => xem kết quả đăng nhập.

Trang chủ    Các trang web    Các nội dung    Các khóa học    Đăng ký    **Đăng nhập**    Quản trị

**Đăng nhập**

Username:

Password:

**Submit**

Trang chủ    Các trang web    Các nội dung    Các khóa học    Đăng ký    **Đăng xuất**    Quản trị

Chào bạn xuanxuan

- Sau khi “Đăng nhập” người dùng sẽ thấy “Đăng xuất” xuất hiện trên menu => đăng xuất => thông báo kết quả “Đăng xuất”

Trang chủ    Các trang web    Các nội dung    Các khóa học    Đăng ký    **Đăng nhập**    Quản trị

Bạn đã đăng xuất. Vui lòng chọn 'Đăng nhập'

✓ **Hướng dẫn:**

- Bước 1: tạo các view cần thiết cho việc login và logout trong views.py của web app

```
from django.contrib.auth.decorators import login_required
from django.contrib.auth import authenticate, login, logout

def user_login(request):
    if request.method == "POST":
        username = request.POST.get("username")
        password = request.POST.get("password")
        user = authenticate(username=username, password=password)
        if user:
```



```

if user.is_active:
    login(request, user)
    result = "Chào bạn " + username
    return render(request, "first_app/mainpage.html", { "result":result})
else:
    print("Không đăng nhập được")
    print( "Username: {} and password: {}".format(username, password))
    login_result = "Username và password không hợp lệ"
    return render(request, "first_app/login.html", { "login_result": login_result})
else:
    return render(request, "first_app/login.html")

@login_required
def user_logout(request):
    logout(request)
    result = "Bạn đã đăng xuất. Vui lòng chọn 'Đăng nhập'"
    return render(request, "first_app/mainpage.html", { "result":result})

```

- Bước 2: cập nhật các url trong urls.py của web app

```

urlpatterns = [
    ...
    url(r'^logout/$',views.user_logout,name= 'logout'),
    url(r'^login/$',views.user_login,name= 'login'),
]

```

- Bước 3: tạo trang login.html trong template

```

<!DOCTYPE html>
{% extends "first_app/base.html" %}
{% block body_block %}
    {% load staticfiles %}
    <div class= "container">
        <h1>Đăng nhập</h1>
        <form method="POST" action="{% url 'first_app:login' %}">
            <table>
                <tr><td>
                    <label for= "username">Username:</label></td>
                    <td><input type= "text" name= "username" placeholder= "Enter
username">
```



```

</td></tr>
<tr><td>
<lable for= "password">Password:</lable></td>
<td><input type= "password" name= "password" placeholder= "Enter
password"></td></tr>
<tr><td colspan= "2" align= "center">
<input type="submit" class="btn btn-primary" value="Submit">
</td></tr>
</table>
{% csrf_token %}
</form>
</div>
<div class= "container"><font
color= "red"></font>{{login_result}}</font></div>
{% endblock %}

```

- Bước 4: cập nhật các đường dẫn trong base.html

```

<nav class="navbar navbar-default navbar-static-top">
    <div class="container">
        <ul class="nav navbar-nav">
            <li><a class="navbar-brand" href="{% url
'first_app:mainpage' %}">Trang chủ</a></li>
            <li><a class= "navbar-link" href= "{% url
'first_app:webpages' %}">Các trang web</a></li>
            <li><a class= "navbar-link" href= "{% url 'first_app:contents'
%}">Các nội dung</a></li>
            <li><a class="navbar-link" href= "{% url
'first_app:homepages' %}">Các khóa học</a></li>
            <li><a class="navbar-link" href= "{% url 'first_app:register'
%}">Đăng ký</a></li>
            {% if user.is_authenticated %}
                <li><a class= "navbar-link" href= "{% url
'first_app:logout' %}">Đăng xuất</a></li>
            {% else %}
                <li><a class= "navbar-link" href= "{% url
'first_app:login' %}">Đăng nhập</a></li>
            {% endif %}
            <li><a class= "navbar-link" href= "{% url 'admin:index'
%}">Quản trị</a></li>
        </ul>

```



```
</div>
</nav>
<div class= "container">
<h4><font color= "red">{{result}}</font></h4>
</div>
```

- Bước 5: chạy thử và kiểm tra kết quả

## BÀI 9: Triển khai website



*Mục tiêu chính:*

- Đưa website lên Internet (sử dụng Python Anywhere) và xuất bản.

### 9.1. Triển khai website lên Internet

- ✓ **Yêu cầu:** Hãy thực hiện công việc đưa website lên Internet như sau:

- ✓ **Hướng dẫn**

- Vào <https://www.pythonanywhere.com/> đăng ký một tài khoản
- Đăng nhập
- Tạo một webapp
- Lần lượt tạo cây thư mục và upload các file đã viết trong project lên theo đúng thứ tự và đầy đủ
- Reload web app
- Chạy thử và kiểm tra
- Xuất bản



# BÀI 10: Django web service

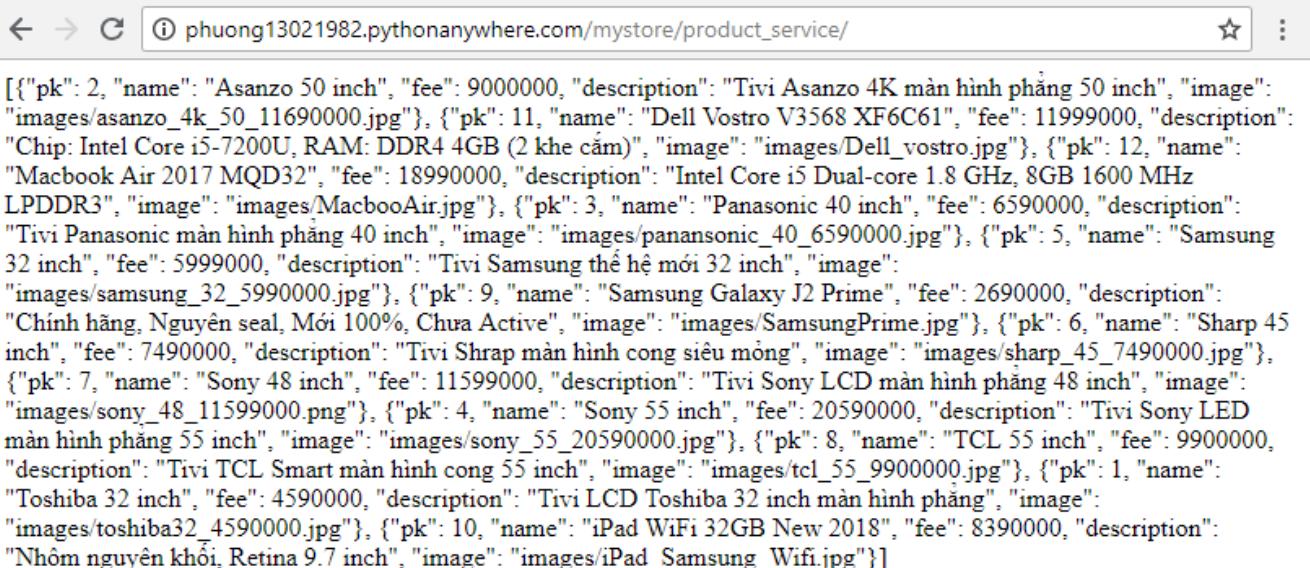


Mục tiêu chính:

- Xây dựng web service đơn giản
- Xây dựng web service với django rest framework

## 10.1. Xây dựng web service đơn giản

- ✓ **Yêu cầu: Hãy xây dựng web service cung cấp danh sách các sản phẩm đang có trong website shoponline:**

A screenshot of a web browser window. The address bar shows the URL: "phuong13021982.pythonanywhere.com/mystore/product\_service/". The page content displays a JSON array of product data. The JSON data is as follows:

```
[{"pk": 2, "name": "Asanzo 50 inch", "fee": 9000000, "description": "Tivi Asanzo 4K màn hình phẳng 50 inch", "image": "images/asanzo_4k_50_11690000.jpg"}, {"pk": 11, "name": "Dell Vostro V3568 XF6C61", "fee": 11999000, "description": "Chip: Intel Core i5-7200U, RAM: DDR4 4GB (2 khe cắm)", "image": "images/Dell_vostro.jpg"}, {"pk": 12, "name": "Macbook Air 2017 MQD32", "fee": 18990000, "description": "Intel Core i5 Dual-core 1.8 GHz, 8GB 1600 MHz LPDDR3", "image": "images/MacbooAir.jpg"}, {"pk": 3, "name": "Panasonic 40 inch", "fee": 6590000, "description": "Tivi Panasonic màn hình phẳng 40 inch", "image": "images/panasonic_40_6590000.jpg"}, {"pk": 5, "name": "Samsung 32 inch", "fee": 5999000, "description": "Tivi Samsung thế hệ mới 32 inch", "image": "images/samsung_32_5990000.jpg"}, {"pk": 9, "name": "Samsung Galaxy J2 Prime", "fee": 2690000, "description": "Chính hãng, Nguyên seal, Mới 100%, Chưa Active", "image": "images/SamsungPrime.jpg"}, {"pk": 6, "name": "Sharp 45 inch", "fee": 7490000, "description": "Tivi Sharp màn hình cong siêu mỏng", "image": "images/sharp_45_7490000.jpg"}, {"pk": 7, "name": "Sony 48 inch", "fee": 11599000, "description": "Tivi Sony LCD màn hình phẳng 48 inch", "image": "images/sony_48_11599000.png"}, {"pk": 4, "name": "Sony 55 inch", "fee": 20590000, "description": "Tivi Sony LED màn hình phẳng 55 inch", "image": "images/sony_55_20590000.jpg"}, {"pk": 8, "name": "TCL 55 inch", "fee": 9900000, "description": "Tivi TCL Smart màn hình cong 55 inch", "image": "images/tcl_55_9900000.jpg"}, {"pk": 1, "name": "Toshiba 32 inch", "fee": 4590000, "description": "Tivi LCD Toshiba 32 inch màn hình phẳng", "image": "images/toshiba32_4590000.jpg"}, {"pk": 10, "name": "iPad WiFi 32GB New 2018", "fee": 8390000, "description": "Nhôm nguyên khối, Retina 9.7 inch", "image": "images/iPad_Samsung_Wifi.jpg"}]
```

Web service cung cấp danh sách sản phẩm theo định dạng JSON

The screenshot shows a web browser displaying a shopping website. The header reads "ABC Shop Online" and "Home". The main banner features two bags labeled "ABC" and "All things you need are here!", with icons for a laptop, a cart, and a calculator. To the right is a large "ONLINE SHOPPING" logo with a red arrow pointing up. Below the banner are four product cards for different types of televisions:

- Asanzo 50 inch**: 50" screen, Tivi Asanzo 4K màn hình phẳng 50 inch 9000000 vnd. Button: Xem
- Panasonic 40 inch**: 40" screen, Tivi Panasonic màn hình phẳng 40 inch 6590000 vnd. Button: Xem
- Samsung 32 inch**: 32" screen, Tivi Samsung thế hệ mới 32 inch 5999000 vnd. Button: Xem
- Sharp 45 inch**: 45" screen, Tivi Sharp màn hình cong siêu mỏng 7490000 vnd. Button: Xem

*Đọc nội dung trên web service và hiển thị danh sách sản phẩm*

### ✓ Hướng dẫn

- Bước 1: tạo view product\_service trả về HttpResponse chứa chuỗi JSON trong views.py của mystore

```
from mystore.models import Product
import json
def product_service(request):
    pro_list = Product.objects.order_by('name')
    result_list = list(pro_list.values('pk', 'name', 'fee', 'description', 'image'))
    return JsonResponse(json.dumps(result_list, ensure_ascii=False).encode('utf8'))
```

- Bước 2: khai báo url của view product\_service vừa tạo trong urls.py của mystore:

```
urlpatterns = [
    ...
    url(r'^product_service/$', views.product_service, name='product_service'),
]
```



- Bước 3: chạy thử và kiểm tra
- Bước 4: tạo view readfromservice trong views.py của mystore để đọc và lấy nội dung JSON của service => chuyển thành danh sách product và gửi kết quả này đến trang readfromservice.html

```

import json
import urllib.request
def read_json_from_internet_unicode(url1):
    DEFAULT_ENCODING = 'utf-8'
    urlResponse = urllib.request.urlopen(url1)
    if hasattr(urlResponse.headers, 'get_content_charset'):
        encoding = urlResponse.headers.get_content_charset(DEFAULT_ENCODING)
    else:
        encoding = urlResponse.headers.getparam('charset') or
    DEFAULT_ENCODING
    noi_dung = json.loads(urlResponse.read().decode(encoding))
    return noi_dung

def readfromservice(request):
    url1 = "http://phuong13021982.pythonanywhere.com/mystore/product_service/"
    pro_list = read_json_from_internet_unicode(url1)
    pro_dict = { "products":pro_list}
    return render(request, "mystore/readfromservice.html", context=pro_dict)

```

- Bước 5: khai báo url của view readfromservice vừa tạo trong urls.py của mystore

```

urlpatterns = [
    ...
    url(r'^readfromservice/$',views.readfromservice,name='readfromservice'),
]

```

- Bước 6: copy trang index.html đã viết trong templates/mystore và đổi tên thành readfromservice.html, chỉ chỉnh lại đường dẫn hình ảnh

```



```

- Bước 7: chạy thử và xem kết quả

## 10.2. Xây dựng web service với djangorestframework

- ✓ **Yêu cầu: Hãy xây dựng lại web service cho bài 10.1 theo cách sử dụng djangorestframework:**



[127.0.0.1:8000/router/?format=api](http://127.0.0.1:8000/router/?format=api)

## Django REST framework

# Api Root

The default basic root view for DefaultRouter

**GET /router/?format=api**

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "users": "http://127.0.0.1:8000/router/users/?format=api",
    "products": "http://127.0.0.1:8000/router/products/?format=api"
}
```

[127.0.0.1:8000/router/products/?format=api](http://127.0.0.1:8000/router/products/?format=api)

## Django REST framework

### Product List

**GET /router/products/?format=api**

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[{"name": "Toshiba 32 inch", "fee": 4590000, "description": "Tivi LCD Toshiba 32 inch màn hình phẳng", "image": "http://127.0.0.1:8000/media/images/toshiba32_4590000.jpg"}, {"name": "Asanzo 50 inch", "fee": 900000, "description": "Tivi Asanzo 4K màn hình phẳng 50 inch", "image": "http://127.0.0.1:8000/media/images/asanzo_4k_50_11690000.jpg"}, {"name": "Panasonic 40 inch", "fee": 6590000, "description": "Tivi Panasonic màn hình phẳng 40 inch", "image": "http://127.0.0.1:8000/media/images/panasonic_40_6590000.jpg"}, {"name": "Sony 55 inch", "fee": 20590000, "description": "Tivi Sony LED màn hình phẳng 55 inch", "image": "http://127.0.0.1:8000/media/images/sony_55_20590000.jpg"}, {"name": "Samsung 32 inch", "fee": 5990000, "description": "Tivi Samsung thế hệ mới 32 inch", "image": "http://127.0.0.1:8000/media/images/samsung_32_5990000.jpg"}, {"name": "Sharp 45 inch", "fee": 7490000, "description": "Tivi Sharp màn hình cong siêu mỏng", "image": "http://127.0.0.1:8000/media/images/sharp_45_7490000.jpg"}, {"name": "Sony 48 inch", "fee": 11599000, "description": "Tivi Sony LCD màn hình phẳng 48 inch", "image": "http://127.0.0.1:8000/media/images/sony_48_11599000.jpg"}]
```

Web service cung cấp danh sách sản phẩm theo định dạng JSON



*Đọc nội dung trên web service và hiển thị danh sách sản phẩm*

✓ **Hướng dẫn**

- Làm tương tự như hướng dẫn trong slide
- Chạy thử và xem kết quả



# Mục lục

<b>BÀI 1: Tổng quan về xây dựng ỨD web .....</b>	<b>2</b>
1.1.    Sử dụng các tag cơ bản .....	2
1.2.    Tạo Form Liên hệ .....	3
1.3.    Giới thiệu template mẫu số 1 .....	4
1.4.    Giới thiệu template mẫu số 2 .....	8
<b>BÀI 2: Tổng quan Django .....</b>	<b>12</b>
2.1.    Cài đặt Django .....	12
2.2.    Tạo project web .....	12
2.3.    Tạo web app trong web project .....	13
<b>BÀI 3: Mapping URLs, Template, Static File, View .....</b>	<b>15</b>
3.1.    Xây dựng website khung chuẩn Django từ template mẫu được cung cấp.....	15
<b>BÀI 4: Database và Model .....</b>	<b>21</b>
4.1.    Xây dựng website – Phần 1 .....	21
4.2.    Xây dựng website – Phần 2 .....	27
<b>BÀI 5: Forms .....</b>	<b>36</b>
5.1.    Form Contact .....	36
5.2.    Form Đăng ký khóa học .....	40
<b>BÀI 6: Template .....</b>	<b>45</b>
6.1.    Xây dựng website ShopOnline.....	45
6.2.    Hiệu chỉnh website trong bài Database - Models.....	52
6.3.    Form Tìm kiếm khóa học.....	55
6.4.    Hiệu chỉnh website trong bài 3 - Mapping URLs, Template, Static File, View.....	58
<b>BÀI 7: Cookie và Session .....</b>	<b>60</b>
7.1.    Sử dụng cookie để hiển thị lần đăng nhập cuối .....	60
7.2.    Sử dụng session để xây dựng Giỏ hàng .....	61
<b>BÀI 8: User Authentication.....</b>	<b>77</b>
8.1.    Xây dựng form đăng ký Register .....	77
8.2.    Xây dựng form đăng nhập .....	80
<b>BÀI 9: Triển khai website .....</b>	<b>85</b>
9.1.    Triển khai website lên Internet .....	85
<b>BÀI 10: Django web service .....</b>	<b>86</b>



10.1. Xây dựng web service đơn giản .....	86
10.2. Xây dựng web service với djangorestframework .....	88
<b>Mục lục.....</b>	<b>91</b>