**shop**

**Layout.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>**{{** title }} - My Django Application</title>

{% load staticfiles %}

<link rel="stylesheet" type="text/css" href="{% static 'app/content/bootstrap.min.css' %}" />

<link rel="stylesheet" type="text/css" href="{% static 'app/content/site.css' %}" />

<script src="{% static 'app/scripts/modernizr-2.6.2.js' %}"></script>

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#">

<!--<img src=" static

/app/content/Cinema\_D.png" width="50" height="50" alt="logo" /> </a>-->

</div>

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li><a href="{% url 'home' %}">Home</a></li>

<li><a href="{% url 'about' %}">About</a></li>

<li><a href="{% url 'contact' %}">Contact</a></li>

<li><a href="{% url 'base\_view' %}">Каталог</a></li>

{% if not request.get\_full\_path == '/base\_view' %}

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown"

role="button" aria-haspopup="true"

aria-expanded="false">Категории <span class="caret"></span>

</a>

<ul class="dropdown-menu">

{% for category in categories %}

<li><a href="**{{** category.get\_absolute\_url }}">**{{** category.name }}</a></li>

{% endfor %}

</ul>

</li>

{% endif %}

{% if not request.user.is\_authenticated %}

<li>

<p class="navbar-text">Здравствуйте, гость!</p>

</li>

<li>

<a href="{% url 'login\_view' %}">Войти</a>

</li>

<li>

<a href="{% url 'registration' %}">Регистрация</a>

</li>

{% else %}

<li>

<p class="navbar-text">Здравствуйте, **{{** request.user.username }}!</p>

</li>

<li>

<a href="{% url 'account' %}">Личный кабинет</a>

</li>

{% endif %}

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="{% url 'cart' %}">Корзина

<span class="label label-default"> <span class="bange" id="cart\_count">{% if not request.session.total %}

0 {% else %}**{{** request.session.total }} {% endif %}</span></span><span class="label label-default"></a></li>

</ul>

{% include 'app/loginpartial.html' %}

</div>

</div>

</div>

<div class="container body-content">

{% block content %}{% endblock %}

<hr/>

<footer>

<p>&copy; **{{** year }} - My Django Application</p>

</footer>

</div>

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script src="{% static 'app/scripts/bootstrap.js' %}"></script>

<script src="{% static 'app/scripts/respond.js' %}"></script>

{% block scripts %}{% endblock %}

</body>

</html>

**Base.html**

{% extends "app/layout.html" %}

{% load static %}

{% block content %}

<div class="container">

<div class="row">

<div id="testCarousel" class="carousel slide" data-ride="carousel">

<!-- Слайды карусели -->

<div class="carousel-inner">

{% for product in products %}

{% if forloop.first %}

<!-- Слайд 1 -->

<div class="item active">

<img class="slide-image" src="**{{** product.image.url}}" alt="Alternate Text"

style="padding-left:260px;height:200px;" />

<p class="text-center"><a href="**{{** product.image.url}}">

<button class="btn btn-primary">

**{{** product.title}} </button></a></p>

</div>

{% else %}

<!-- Слайд 2 -->

<div class="item">

<img class="slide-image" src="**{{** product.image.url}}" alt="Alternate Text"

style="padding-left:260px;height:200px;" />

<p class="text-center"><a href="**{{** product.image.url}}">

<button class="btn btn-primary">

**{{** product.title}} </button></a></p>

</div>

{% endif %}

{% endfor %}

</div>

<!-- Навигация карусели (следующий или предыдущий слайд) -->

<!-- Кнопка, переход на предыдущий слайд с помощью атрибута data-slide="prev" -->

<a class="left carousel-control" href="#testCarousel"

role="button" data-slide="prev" style="background:transparent;" >

<span class="glyphicon glyphicon-chevron-left" style="color:black;" ></span>

</a>

<!-- Кнопка, переход на следующий слайд с помощью атрибута data-slide="next" -->

<a class="right carousel-control" href="#testCarousel"

role="button" data-slide="next" style="background:transparent;" >

<span class="glyphicon glyphicon-chevron-right" style="color:black;" ></span>

</a>

</div>

<div class="col-md-2">

<div class="list-group">

{% for category in categories %}

<a href="**{{** category.get\_absolute\_url }}" class="list-group-item">

**{{** category.name }}</a>

{% endfor %}

</div>

</div>

</div>

<br />

<br />

<div class="row">

{% for product in products %}

<div class="col-sm-3 col-lg-3 col-md-3">

<div class="thumbnail">

<img src="**{{** product.image.url }}" alt="" style='width:70%;

height:200px;'/>

<div class="caption">

<h5 class="text-center"style="font-size:20px"><strong><a href="**{{** product.get\_absolute\_url }}">

**{{** product.title }}</a></strong></h5>

<h4 class="text-center" style="font-size:25px">

<strong>**{{** product.price }} руб.</strong></h4>

{% if request.user.is\_authenticated %}

<p class="text-center"><a href="#" class="add\_to\_cart"

data-slug="**{{**product.slug}}" ><button class="btn btn-danger">Добавить в корзину</button></a></p>

{% else %}

<small class="text-center">

Заказывать товар могут только авторизированные пользователи.

Пожалуйста, <a href="#">войдите</a> или <a href="#">зарегистрируйтесь</a>.

</small>

{% endif %}

</div>

</div>

</div>

{% endfor %}

</div>

</div>

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script>

$(document).ready(function()

{

$('.add\_to\_cart').on('click', function (e)

{

e.preventDefault()

product\_slug = $(this).attr('data-slug')

data =

{

product\_slug:product\_slug

}

$.ajax({

type: "GET",

url: '{% url "add\_to\_cart" %}',

data: data,

success: function (data)

{

$("#cart\_count").html(data.cart\_total)

}

})

})

})

</script>

{% endblock %}

**Product.html**

{% extends "app/layout.html" %}

{% load static %}

{% block content %}

<br />

<div class="container">

<div class="row">

<div class="col-sm-4">

<img src="**{{** product.image.url }}" alt="" style='width:70%; height:200px;'/>

</div>

<div class="col-sm-8">

<h3 class="text-center">**{{** product.title }} </h3>

<p>{% lorem %}</p>

<p>{% lorem %}</p>

<hr />

<p style="font-size:25px">Цена :<strong >**{{** product.price }} руб.</strong></p>

{% if request.user.is\_authenticated %}

<a href="#"

class="add\_to\_cart" data-slug="**{{** product.slug}}">

<button class="btn btn-danger">Добавить в корзину</button></a>

{% else %}

<p class="text-center">

Заказывать товар могут только авторизированные пользователи.

Пожалуйста, <a href="{% url 'login\_view' %}">войдите</a> или <a href="{% url 'registration' %}">зарегистрируйтесь</a>.

</p>

{% endif %}

</div>

</div>

</div>

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script>

$(document).ready(function()

{

$('.add\_to\_cart').on('click', function (e)

{

e.preventDefault()

product\_slug = $(this).attr('data-slug')

data =

{

product\_slug:product\_slug

}

$.ajax({

type: "GET",

url: '{% url "add\_to\_cart" %}',

data: data,

success: function (data)

{

$("#cart\_count").html(data.cart\_total)

}

})

})

})

</script>

{% endblock %}

**Order.html**

{% extends "app/layout.html" %}

{% load static %}

{% load crispy\_forms\_tags %}

{% block content %}

<form action="{% url "make\_order" %}" method="post"enctype="multipart/form-data">

{% csrf\_token %}

<div class="col-sm-4">**{{** form.name|as\_crispy\_field}}</div>

<div class="col-sm-4">**{{** form.last\_name|as\_crispy\_field}}</div>

<div class="col-sm-4">**{{** form.phone|as\_crispy\_field}}</div>

**{{** form.buying\_type|as\_crispy\_field}}

**{{** form.address|as\_crispy\_field}}

**{{** form.date|as\_crispy\_field}}

**{{** form.comments|as\_crispy\_field}}

<input type="submit" class="btn btn-success" value="Оформить заказ" />

</form>

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script>

$(document).ready(function () {

$('#div\_id\_address').css('display','none')

$('#id\_buying\_type').on('click', function () {

buying\_type = $(this).val()

if (buying\_type == 'delivery') {

$('#div\_id\_address').css('display', 'block')

}

else {

$('#div\_id\_address').css('display', 'none')

}

})

$('#id\_date\_month').css('width','100px')

$('#id\_date\_month').css('display','inline')

$('#id\_date\_year').css('width', '100px')

$('#id\_date\_year').css('display','inline')

$('#id\_date\_day').css('width', '50px')

$('#id\_date\_day').css('display','inline')

})

</script>

{% endblock %}

**Cart.html**

{% extends "app/layout.html" %}

{% load static %}

{% block content %}

<h3 class="text-center">Ваша корзина</h3>

<br />

{% if cart.items.count > 0 %}

<div class="my-cart">

<table class="table">

<tr>

<td class="text-center"><strong>Товар</strong></td>

<td class="text-center" ><strong>Количество</strong></td>

<td class="text-center"><strong>Цена</strong></td>

<td class="text-center"></td>

</tr>

{% for item in cart.items.all %}

<tr class="cart-item-**{{** item.product.id}}">

<td class="text-center">**{{** item.product.title }}</td>

<td class="text-center">

<form action="/" method="get">

<input type="number" name="qty" class="cart-item-qty"

data-id="**{{** item.id }}" value="**{{** item.qty }}" min="1" style="width:60px;"/>

</form>

</td>

<td class="text-center" id="cart-item-total-**{{** item.id }}">**{{** item.item\_total }} руб.</td>

<td class="text-center"> <a href="#"

class="remove\_from\_cart" data-slug="**{{** item.product.slug }}"

data-id="**{{** item.product.id }}">

<button class="btn btn-default">Удалить из корзины</button></a></td>

</tr>

{% endfor %}

<tr>

<td></td>

<td class="text-center"><strong>Итого :</strong></td>

<td class="text-center" id="cart-total-price"><strong>**{{** cart.cart\_total}} руб.</td>

<td class="text-center"></td>

</tr>

</table>

<br />

<a href="{% url 'checkout' %}" class="pull-right"><button class="btn btn-danger">Предварительный заказ</button></a>

</div>

<h3 class="cart-empty"></h3>

{% else %}

<h3 class="text-center">Ваша корзина пуста</h3>

{% endif %}

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script>

$(document).ready(function()

{

$('.cart-empty').css('display','none')

$('.remove\_from\_cart').on('click', function (e)

{

e.preventDefault()

product\_slug = $(this).attr('data-slug')

item\_product\_id=$(this).attr('data-id')

data =

{

product\_slug:product\_slug

}

$.ajax({

type: "GET",

url: '{% url "remove\_from\_cart" %}',

data: data,

success: function (data)

{

$("#cart\_count").html(data.cart\_total)

$('.cart-item-'+item\_product\_id).css('display','none')

$('#cart-total-price').html('<strong>'+parseFloat(data.cart\_total\_price).toFixed(2)+'руб.</strong>')

if(parseInt(data.cart\_total) == 0)

{

$('.my-cart').css('display','none')

$('.cart-empty').css('display','block')

$('.cart-empty').html('<h3 class="text-center">Ваша корзина пуста</h3>')

}

}

})

})

})

$(document).ready(function ()

{

$('.cart-item-qty').on('click', function ()

{

qty=$(this).val()

item\_id=$(this).attr('data-id')

data =

{

qty:qty,

item\_id:item\_id

}

$.ajax({

type: "GET",

url: '{% url "change\_item\_qty" %}',

data: data,

success: function(data)

{

$('#cart-item-total-'+ item\_id).html(parseFloat(data.item\_total).toFixed(2)+'руб.')

$('#cart-total-price').html('<strong>'+ parseFloat(data.cart\_total\_price).toFixed(2)+'руб.</strong>')

}

})

})

})

</script>

{% endblock %}

**Category.html**

{% extends "app/layout.html" %}

{% load static %}

{% block content %}

<div class="container">

<div class="row">

<div class="col-sm-12">

<h2>**{{** category.name}}</h2>

<hr />

{% for product in products\_of\_category%}

<div class="col-sm-4">

<img src="**{{** product.image.url }}" alt="" style='width:70%; height:250px;'/>

<h4 class="text-center">**{{** product.title }}</h4>

<p class="text-center" style="font-size:25px">Цена :<strong >**{{** product.price }}</strong></p>

{% if request.user.is\_authenticated %}

<p class="text-center"><a href="#" class="add\_to\_cart" data-slug="**{{** product.slug}}">

<button class="btn btn-danger">Добавить в корзину</button></a></p>

{% else %}

<p class="text-center">

Заказывать товар могут только авторизированные пользователи.

Пожалуйста, <a href="#">войдите</a> или <a href="#">зарегистрируйтесь</a>.

</p>

{% endif %}

</div>

{% endfor %}

</div>

</div>

</div>

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

<script>

$(document).ready(function()

{

$('.add\_to\_cart').on('click', function (e)

{

e.preventDefault()

product\_slug = $(this).attr('data-slug')

data =

{

product\_slug:product\_slug

}

$.ajax({

type: "GET",

url: '{% url "add\_to\_cart" %}',

data: data,

success: function (data)

{

$("#cart\_count").html(data.cart\_total)

}

})

})

})

</script>

{% endblock %}

**Checkout.html**

{% extends "app/layout.html" %}

{% block content %}

<h3 class="text-center">Ваш предварительный заказ</h3>

<br />

<table class="table">

<tr>

<td class="text-center"><strong>Товар</strong></td>

<td class="text-center" ><strong>Количество</strong></td>

<td class="text-center"><strong>Цена</strong></td>

<td class="text-center"></td>

</tr>

{% for item in cart.items.all %}

<tr>

<td class="text-center">**{{** item.product.title }}</td>

<td class="text-center">**{{** item.qty }}</td>

<td class="text-center">**{{** item.item\_total }} руб.</td>

<td > </td>

</tr>

{% endfor %}

<tr>

<td></td>

<td class="text-center"><strong>Итого :</strong></td>

<td class="text-center"><strong>**{{** cart.cart\_total}} руб.</td>

<td class="text-center"></td>

</tr>

</table>

<a href="{% url "order\_create" %}" class="pull-right"><button class="btn btn-success">Оформить заказ</button></a>

{% endblock %}

**Account.html**

{% extends "app/layout.html" %}

{% block content %}

<h3 class="text-center">Личный кабинет пользователя **{{** request.user }}</h3>

<hr />

<h4 class="text-center">Ваши заказы</h4>

<table class="table">

<tr style="background:lightblue;">

<td class="text-center">Номер заказа</td>

<td class="text-center">Дата заказа</td>

<td class="text-center">Статус заказа</td>

<td class="text-center">Стоимость заказа</td>

<td class="text-center">Детали заказа</td>

</tr>

{% for item in order %}

<tr>

<td class="text-center" style="padding-top:10px;padding-bottom:20px;">**{{** item.id }}</td>

<td class="text-center" style="padding-top:10px;padding-bottom:20px;">**{{** item.date }}</td>

<td class="text-center" style="padding-top:10px;padding-bottom:20px;">**{{** item.status }}</td>

<td class="text-center" style="padding-top:10px;padding-bottom:20px;">**{{** item.total }} руб.</td>

<td class="text-center" style="padding-top:10px;padding-bottom:20px;"><button type="button"

class="btn btn-primary btn-md" data-toggle="modal" data-target="#myModal-**{{** item.id }}">

Детали заказа</button>

</td>

</tr>

{% endfor %}

</table>

<!-- Button trigger modal -->

<!-- Modal -->

{% for item in order %}

<div class="modal fade" id="myModal-**{{** item.id }}" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h4 class="modal-title" id="myModalLabel">Заказ №**{{** item.id }}</h4>

</div>

<div class="modal-body">

<h3>Товар</h3>

<table class="table">

{% for order\_item in item.items.all %}

{% for ord\_item in order\_item.items.all %}

<tr>

<td>**{{** ord\_item.product.title}}</td>

<td>**{{** ord\_item.qty }} шт.</td>

<td>**{{** ord\_item.item\_total}} руб.</td>

</tr>

{% endfor %}

<tr>

<td></td>

<td>Итого</td>

<td><strong>**{{** order\_item.cart\_total}} руб.</strong></td>

</tr>

{% endfor %}

</table>

<h3>Заказчик</h3>

<p>Имя:**{{** item.first\_name }}</p>

<p>Фамилия:**{{** item.last\_name }}</p>

<p>Телефон:**{{** item.phone }}</p>

<h3>Тип доставки</h3>

<p>{% if item.buying\_type == 'self' %}Самовывоз{% else %}Доставка{% endif %}</p>

<p>{% if item.buying\_type == 'delivery' %}**{{** item.address }}{% endif %}</p>

<h3>Комментарий к заказу</h3>

<p>**{{** item.comments }}</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Закрыть</button>

</div>

</div>

</div>

</div>

{% endfor %}

{% endblock %}

**Registration.html**

{% extends "app/layout.html" %}

{% load static %}

{% load crispy\_forms\_tags %}

{% block content %}

<h3 class="text-center">Регистрация нового пользователя</h3>

<hr />

<div class="col-sm-8 col-sm-offset-2">

<form action="{% url 'registration' %}" method="post" enctype="multipart/form-data">

{% csrf\_token %}

**{{**form|crispy}}

<br />

<input type="submit" class="btn btn-success btn-block" value="Зарегистрироваться" />

</form>

</div>

{% endblock %}

**Thank\_you.html**

{% extends "app/layout.html" %}

{% block content %}

<div class="col-sm-8 col-sm-offset-2">

<h3 class="text-center">Спасибо за Ваш заказ!

Он принят в обработку.В ближайшее время менеджер с Вами свяжется.</h3>

<p class="text-center"><a href="{% url 'base\_view'%}">

<button class="btn btn-default">Вернуться к покупкам</button></a>

</p>

<hr />

</div>

{% endblock %}

**LoginUser.html**

{% extends "app/layout.html" %}

{% load static %}

{% load crispy\_forms\_tags %}

{% block content %}

<h3 class="text-center">Авторизация</h3>

<hr />

<div class="col-sm-8 col-sm-offset-2">

<form action="{% url 'login\_view' %}" method="post" enctype="multipart/form-data">

{% csrf\_token %}

**{{**form|crispy}}

<br />

<input type="submit" class="btn btn-success btn-block" value="Войти в систему" />

</form>

</div>

{% endblock %}

**Form.py**

#-\*- coding: utf-8 -\*-

"""

Definition of forms.

"""

from django.contrib.auth.models import User

from django.utils import timezone

from django import forms

from django.contrib.auth.forms import AuthenticationForm

from django.utils.translation import ugettext\_lazy as \_

class BootstrapAuthenticationForm(AuthenticationForm):

"""Authentication form which uses boostrap CSS."""

username = forms.CharField(max\_length=254,

widget=forms.TextInput({

'class': 'form-control',

'placeholder': 'User name'}))

password = forms.CharField(label=\_("Password"),

widget=forms.PasswordInput({

'class': 'form-control',

'placeholder':'Password'}))

class OrderForm(forms.Form):

name=forms.CharField()

last\_name=forms.CharField(required=False)

phone=forms.CharField()

buying\_type=forms.ChoiceField(widget=forms.Select(),

choices=([("self","Самовывоз"),("delivery","Доставка")]))

date=forms.DateField(widget=forms.SelectDateWidget(),initial=timezone.now())

address=forms.CharField(required=False)

comments=forms.CharField(widget=forms.Textarea,required=False)

def \_\_init\_\_(self,\*args,\*\*kwargs):

super(OrderForm,self).\_\_init\_\_(\*args,\*\*kwargs)

self.fields['name'].label='Имя'

self.fields['last\_name'].label='Фамилия'

self.fields['phone'].label='Контактный телефон'

self.fields['phone'].help\_text='Пожалуйста, указывайте реальный номер телефона, по которому с Вами можно связаться'

self.fields['buying\_type'].label='Способ получения'

self.fields['address'].label='Адрес доставки'

self.fields['address'].help\_text='\*Обязательно указывайте город!'

self.fields['comments'].label='Комментарий к заказу'

self.fields['date'].label='Дата доставки'

self.fields['date'].help\_text='Доставка производится на следующий день после оформления заказа. Менеджер с Вами предварительно свяжется.'

class RegistrationForm(forms.ModelForm):

password=forms.CharField(widget=forms.PasswordInput)

password\_check=forms.CharField(widget=forms.PasswordInput)

class Meta:

model=User

fields=[

'username',

'password',

'password\_check',

'first\_name',

'last\_name',

'email'

]

def \_\_init\_\_(self,\*args,\*\*kwargs):

super(RegistrationForm,self).\_\_init\_\_(\*args,\*\*kwargs)

self.fields['username'].label='Логин'

self.fields['password'].label='Пароль'

self.fields['password'].help\_text='Придумайте пароль'

self.fields['password\_check'].label='Повторите пароль'

self.fields['first\_name'].label='Имя'

self.fields['last\_name'].label='Фамилия'

self.fields['email'].label='Ваша почта'

self.fields['email'].help\_text='Пожалуйста, указывайте реальный адрес'

def clean(self):

username=self.cleaned\_data['username']

password=self.cleaned\_data['password']

password\_check=self.cleaned\_data['password\_check']

email=self.cleaned\_data['email']

if User.objects.filter(username=username).exists():#условия True

raise forms.ValidationError('Пользователь с данным логином уже зарегистрирован в системе!')

if User.objects.filter(email=email).exists():#условия True

raise forms.ValidationError('Пользователь с данным почтовым адресом зарегистрирован!')

if password != password\_check:

raise forms.ValidationError('Ваши пароли не совпадают! Попробуйте снова!')

class LoginForm(forms.Form):

username=forms.CharField()

password=forms.CharField(widget=forms.PasswordInput)

def \_\_init\_\_(self,\*args,\*\*kwargs):

super(LoginForm,self).\_\_init\_\_(\*args,\*\*kwargs)

self.fields['username'].label='Логин'

self.fields['password'].label='Пароль'

def clean(self):

username=self.cleaned\_data['username']

password=self.cleaned\_data['password']

if not User.objects.filter(username=username).exists():#условия True

raise forms.ValidationError('Пользователь с данным логином не зарегистрирован в системе!')

user=User.objects.get(username=username)

if user and not user.check\_password(password):

raise forms.ValidationError('Неверный пароль!')

**Models.py**

"""

Definition of models.

"""

from django.conf import settings

from django.core.urlresolvers import reverse

import site

from decimal import Decimal

from transliterate import translit # переводит кириллицу

from django.utils.text import slugify #превращает поле в тип slug

from django.db.models.signals import pre\_save

from django.db import models

from django.contrib import admin #добавили использование административного модуля

# Create your models here.

class Category(models.Model):

name=models.CharField(max\_length=120)

slug=models.SlugField(blank=True)

def \_\_str\_\_(self):

return self.name

def get\_absolute\_url(self):

return reverse ('category\_detail',kwargs={'category\_slug':self.slug})

def pre\_save\_category\_slug(sender,instance,\*args,\*\*kwargs):

if not instance.slug:

slug=slugify(translit(instance.name,reversed=True))# заполняет латиницей slug

instance.slug=slug

pre\_save.connect(pre\_save\_category\_slug,sender=Category)

admin.site.register(Category)

class Brand(models.Model):

name=models.CharField(max\_length=120)

def \_\_str\_\_(self):

return self.name

admin.site.register(Brand)

#Чтобы корректно сохранять изображения

def image\_folder(instance,filename):

filename=instance.slug + '.'+ filename.split('.')[1]

return "{0}/{1}".format(instance.slug,filename)

#Чтобы при снятие галочки в admin ->available изчезал продукт с каталога

class ProductManager(models.Manager):

def all(self, \*args, \*\*kwargs):

return super(ProductManager,self).get\_queryset().filter(available=True)

class Product(models.Model):

category=models.ForeignKey(Category)

brand=models.ForeignKey(Brand)

title=models.CharField(max\_length=120)

slug=models.SlugField()

description=models.TextField()

image=models.ImageField(upload\_to=image\_folder)

price=models.DecimalField(max\_digits=9,decimal\_places=2)

available=models.BooleanField(default=True)

objects=ProductManager()

def \_\_str\_\_(self):

return self.title

def get\_absolute\_url(self):

return reverse ('product\_detail',kwargs={'product\_slug':self.slug})

admin.site.register(Product)

# Создаем корзину

#модель продукта

class CartItem(models.Model):

product=models.ForeignKey(Product)#продукт

qty=models.PositiveIntegerField(default=1)#кол-во yне может быть отрицательным поэтому PositiveIntegerField

item\_total=models.DecimalField(max\_digits=9, decimal\_places=2, default=0.0)#для изменения в корзине кол-во

def \_\_str\_\_(self):

return "Cart item for product {0}". format(self.product.title)

admin.site.register(CartItem)

#модель корзины

class Cart(models.Model):

items=models.ManyToManyField(CartItem,blank=True)#Мы можем добавить много в корзину CartItem

cart\_total=models.DecimalField(max\_digits=9, decimal\_places=2, default=0.0)

def \_\_str\_\_(self):

return str(self.id)# возращает id корзины

def add\_to\_cart(self,product\_slug):

cart=self

product=Product.objects.get(slug=product\_slug)

new\_item, \_=CartItem.objects.get\_or\_create(product=product,item\_total=product.price)

if new\_item not in cart.items.all():

cart.items.add(new\_item)

cart.save()

def remove\_from\_cart(self,product\_slug):

cart=self

product=Product.objects.get(slug=product\_slug)

for cart\_item in cart.items.all():

if cart\_item.product == product:

cart.items.remove(cart\_item)

cart.save()

def change\_qty(self,qty,item\_id):

cart=self

cart\_item=CartItem.objects.get(id=int(item\_id))

cart\_item.qty=int(qty)

cart\_item.item\_total = int(qty) \* Decimal(cart\_item.product.price)

cart\_item.save()

new\_cart\_total=0.00

for item in cart.items.all():

new\_cart\_total+=float(item.item\_total)

cart.cart\_total=new\_cart\_total

cart.save()

ORDER\_STATUS\_CHOICES=(('Принят в обработку','Принят в обработку'),

('Выполняется','Выполняется'),

('Оплачен','Оплачен')

)

admin.site.register(Cart)

class Order(models.Model):

user=models.ForeignKey(settings.AUTH\_USER\_MODEL)

items=models.ManyToManyField(Cart)

total=models.DecimalField(max\_digits=9, decimal\_places=2,default=0.00)

first\_name=models.CharField(max\_length=200)

last\_name=models.CharField(max\_length=200)

phone=models.CharField(max\_length=20)

address=models.CharField(max\_length=255)

buying\_type=models.CharField(max\_length=40, choices=(('Самовывоз','Самовывоз'),('Доставка','Доставка')),default='Самовывоз')

date=models.DateTimeField(auto\_now\_add=True)

comments=models.TextField()

status=models.CharField(max\_length=100,choices=ORDER\_STATUS\_CHOICES,

default=ORDER\_STATUS\_CHOICES[0][0])

def \_\_str\_\_(self):

return "Заказ №{0}".format(str(self.id))

admin.site.register(Order)

**App/urls.py**

"""

Definition of urls for eldorado.

"""

from app.views import login\_view

from app.views import registration\_view

from app.views import account\_view

from django.views.generic import TemplateView

from app.views import make\_order\_view

from app.views import order\_create\_view

from app.views import checkout\_view

from app.views import change\_item\_qty

from app.views import remove\_from\_cart\_view

from app.views import add\_to\_cart\_view

from app import views

from app.views import cart\_view

from app.views import category\_view,product\_view

from django.conf.urls.static import static

from django.contrib.staticfiles.urls import staticfiles\_urlpatterns

from django.conf import settings

from django.conf.urls import include

from datetime import datetime

from django.conf.urls import url

import django.contrib.auth.views

import app.forms

import app.views

# Uncomment the next lines to enable the admin:

#from django.contrib import admin

#admin.autodiscover()

urlpatterns = [

# Examples:

url(r'^$', app.views.home, name='home'),

url(r'^contact$', app.views.contact, name='contact'),

url(r'^about$', app.views.about, name='about'),

url(r'^base\_view$', app.views.base\_view, name='base\_view'),

# url(r'^login/$',

# django.contrib.auth.views.login,

# {

# 'template\_name': 'app/login.html',

# 'authentication\_form': app.forms.BootstrapAuthenticationForm,

# 'extra\_context':

# {

# 'title': 'Log in',

# 'year': datetime.now().year,

# }

# },

# name='login'),

# url(r'^logout$',

# django.contrib.auth.views.logout,

# {

# 'next\_page': '/',

# },

# name='logout'),

# Uncomment the admin/doc line below to enable admin documentation:

# # url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

# url(r'^category/(?P<category\_slug>[-\w]+)/$','app.views.category\_view', name='category\_detail'),

#url(r'^product/(?P<product\_slug>[-\w]+)/$', 'app.views.product\_view', name='product\_detail'),

# Uncomment the next line to enable the admin:

# url(r'^admin/', include(admin.site.urls)),

url(r'^category/(?P<category\_slug>[-\w]+)/$',category\_view, name='category\_detail'),

url(r'^product/(?P<product\_slug>[-\w]+)/$', product\_view, name='product\_detail'),

url(r'^add\_to\_cart/$', add\_to\_cart\_view, name='add\_to\_cart'),

url(r'^remove\_from\_cart/$', remove\_from\_cart\_view, name='remove\_from\_cart'),

url(r'^change\_item\_qty/$', change\_item\_qty, name='change\_item\_qty'),

url(r'^cart/$',cart\_view,name='cart'),

url(r'^checkout/$',checkout\_view,name='checkout'),

url(r'^order/$',order\_create\_view,name='order\_create'),

url(r'^make\_order/$',make\_order\_view,name='make\_order'),

url(r'^thank\_you/$',TemplateView.as\_view(template\_name='app/thank\_you.html'),name='thank\_you'),

url(r'^account/$',account\_view,name='account'),

url(r'^registration/$',registration\_view,name='registration'),

url(r'^login/$',login\_view,name='login\_view'),

]

if settings.DEBUG:

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

urlpatterns += staticfiles\_urlpatterns()

#if settings.DEBUG:

#urlpatterns += static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)

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

**Eldorado/urls.py**

"""

Definition of urls for eldorado.

"""

from django.conf.urls.static import static

from django.contrib.staticfiles.urls import staticfiles\_urlpatterns

from django.conf import settings

from datetime import datetime

from django.conf.urls import url

import django.contrib.auth.views

import app.forms

import app.views

# Uncomment the next lines to enable the admin:

from django.conf.urls import include

from django.contrib import admin

admin.autodiscover()

urlpatterns = [

# Examples:

url(r'^',include('app.urls')),

# Uncomment the admin/doc line below to enable admin documentation:

url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

# Uncomment the next line to enable the admin:

url(r'^admin/', include(admin.site.urls)),

]

if settings.DEBUG:

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

urlpatterns += staticfiles\_urlpatterns()

#if settings.DEBUG:

#urlpatterns += static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)

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

**Views.py**

"""

Definition of views.

"""

from django.contrib.auth import login,authenticate

#from django.urls import reverse

from django.core.urlresolvers import reverse

from django.http import JsonResponse

from django.http import HttpResponse

from django.http import HttpResponseRedirect

from app.models import Category,Product,CartItem

from app.models import Cart

from django.shortcuts import render

from django.http import HttpRequest

from django.template import RequestContext

from datetime import datetime

from decimal import Decimal

from app.forms import OrderForm

from app.models import Order

from app.forms import RegistrationForm

from app.forms import LoginForm

def home(request):

"""Renders the home page."""

assert isinstance(request, HttpRequest)

return render(

request,

'app/index.html',

{

'title':'Home Page',

'year':datetime.now().year,

}

)

def contact(request):

"""Renders the contact page."""

assert isinstance(request, HttpRequest)

return render(

request,

'app/contact.html',

{

'title':'Contact',

'message':'Your contact page.',

'year':datetime.now().year,

}

)

def about(request):

"""Renders the about page."""

assert isinstance(request, HttpRequest)

return render(

request,

'app/about.html',

{

'title':'About',

'message':'Your application description page.',

'year':datetime.now().year,

}

)

def base\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

categories=Category.objects.all()

products=Product.objects.all()

context={

'categories':categories,

'products':products,

}

return render(request,'app/base.html',context)

def product\_view(request,product\_slug):

try:

cart\_id=request.session['cart\_id']

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id

cart=Cart.objects.get(id=cart\_id)

product=Product.objects.get(slug=product\_slug)

categories=Category.objects.all()

context={

'product':product,

'categories':categories,

'cart':cart,

}

return render(request,'app/product.html',context)

def category\_view(request,category\_slug):

category=Category.objects.get(slug=category\_slug)

products\_of\_category=Product.objects.filter(category=category)

context={

'category':category,

'products\_of\_category': products\_of\_category

}

return render(request,'app/category.html',context)

def cart\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

context={

'cart':cart

}

return render(request,'app/cart.html',context)

def add\_to\_cart\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

product\_slug=request.GET.get('product\_slug')

product=Product.objects.get(slug=product\_slug)

cart.add\_to\_cart(product.slug)

new\_cart\_total=0.00

for item in cart.items.all():

new\_cart\_total+=float(item.item\_total)

cart.cart\_total=new\_cart\_total

cart.save()

return JsonResponse({'cart\_total':cart.items.count(),'cart\_total\_price':cart.cart\_total})

def remove\_from\_cart\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

product\_slug=request.GET.get('product\_slug')

product=Product.objects.get(slug=product\_slug)

cart.remove\_from\_cart(product.slug)

new\_cart\_total=0.00

for item in cart.items.all():

new\_cart\_total+=float(item.item\_total)

cart.cart\_total=new\_cart\_total

cart.save()

return JsonResponse({'cart\_total':cart.items.count(),'cart\_total\_price':cart.cart\_total})

def change\_item\_qty(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

qty=request.GET.get('qty')

item\_id=request.GET.get('item\_id')

cart.change\_qty(qty, item\_id)

cart\_item=CartItem.objects.get(id=int(item\_id))

cart\_item.qty=int(qty)

cart\_item.item\_total = int(qty) \* Decimal(cart\_item.product.price)

cart\_item.save()

return JsonResponse({'cart\_total':cart.items.count(),

'item\_total':cart\_item.item\_total,

'cart\_total\_price':cart.cart\_total})

def checkout\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

context={

'cart':cart

}

return render(request,'app/checkout.html',context)

def order\_create\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

form=OrderForm(request.POST or None)

context={

'form':form,

'cart':cart

}

return render(request,'app/order.html',context)

def make\_order\_view(request):

try:

cart\_id=request.session['cart\_id']# Получение значения сессии при помощи ключа(то есть, 'cart\_id').

cart=Cart.objects.get(id=cart\_id)

request.session['total']=cart.items.count()

except:

cart=Cart()

cart.save()

cart\_id=cart.id

request.session['cart\_id']=cart\_id# Передача значения в сессию

cart=Cart.objects.get(id=cart\_id)

form=OrderForm(request.POST or None)

if form.is\_valid():

name=form.cleaned\_data['name']

last\_name=form.cleaned\_data['last\_name']

phone=form.cleaned\_data['phone']

buying\_type=form.cleaned\_data['buying\_type']

address=form.cleaned\_data['address']

comments=form.cleaned\_data['comments']

new\_order=Order()

new\_order.user=request.user

new\_order.save()

new\_order.items.add(cart)

new\_order.first\_name=name

new\_order.last\_name=last\_name

new\_order.phone=phone

new\_order.address=address

new\_order.buying\_type=buying\_type

new\_order.comments=comments

new\_order.total=cart.cart\_total

new\_order.save()

del request.session['cart\_id']

del request.session['total']

return HttpResponseRedirect(reverse('thank\_you'))

def account\_view(request):

order=Order.objects.filter(user=request.user).order\_by('-id')#.order\_by('-id') более новый заказ на верхних строчках

context={

'order':order,

}

return render(request,'app/account.html',context)

def registration\_view(request):

form=RegistrationForm(request.POST or None)

if form.is\_valid():

new\_user=form.save(commit=False)

username=form.cleaned\_data['username']

password=form.cleaned\_data['password']

email=form.cleaned\_data['email']

first\_name=form.cleaned\_data['first\_name']

last\_name=form.cleaned\_data['last\_name']

new\_user.username=username

new\_user.set\_password=(password)

new\_user.first\_name=first\_name

new\_user.last\_name=last\_name

new\_user.email=email

new\_user.save()

login\_user=authenticate(username=username,password=password)

if login\_user:

login(request,login\_user)

return HttpResponseRedirect(reverse('base\_view'))

context={

'form':form,

}

return render(request,'app/registration.html',context)

def login\_view(request):

form=LoginForm(request.POST or None)

if form.is\_valid():

username=form.cleaned\_data['username']

password=form.cleaned\_data['password']

login\_user=authenticate(username=username,password=password)

if login\_user:

login(request,login\_user)

return HttpResponseRedirect(reverse('base\_view'))

context={

'form':form

}

return render(request,'app/LoginUser.html',context)

**Setting.py**

"""

Django settings for eldorado project.

Generated by 'django-admin startproject' using Django 1.9.1.

For more information on this file, see

https://docs.djangoproject.com/en/1.9/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/1.9/ref/settings/

"""

import os

import posixpath

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/1.9/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '57f19813-ad76-47f7-bf4c-7cd7e9d24222'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'app',#созданное приложение

# Add your apps here to enable them

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'crispy\_forms',

]

MIDDLEWARE\_CLASSES = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.auth.middleware.SessionAuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'eldorado.urls'

TEMPLATES = [

{

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

'DIRS': [],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

'django.template.context\_processors.media',

],

},

},

]

WSGI\_APPLICATION = 'eldorado.wsgi.application'

# Database

# https://docs.djangoproject.com/en/1.9/ref/settings/#databases

DATABASES = {

'default': {

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

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

# Password validation

# https://docs.djangoproject.com/en/1.9/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/1.9/topics/i18n/

LANGUAGE\_CODE = 'ru-ru'

TIME\_ZONE = 'Europe/Moscow'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/1.9/howto/static-files/

#общая папка для статических файлов

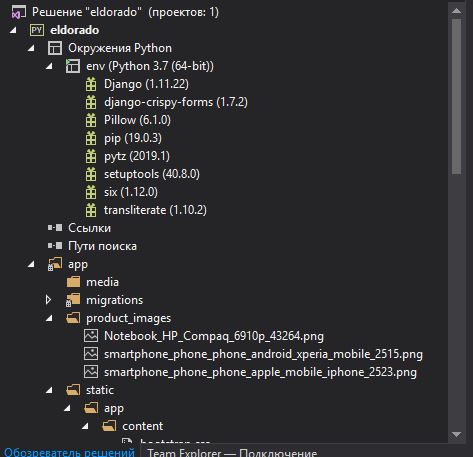
STATIC\_URL = '/static/'

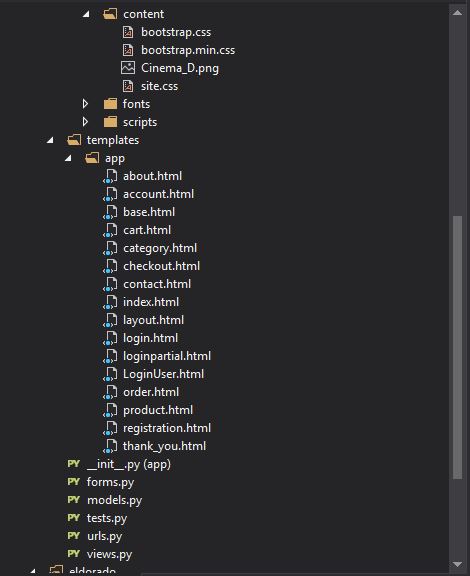
STATIC\_ROOT = posixpath.join(\*(BASE\_DIR.split(os.path.sep) + ['static']))

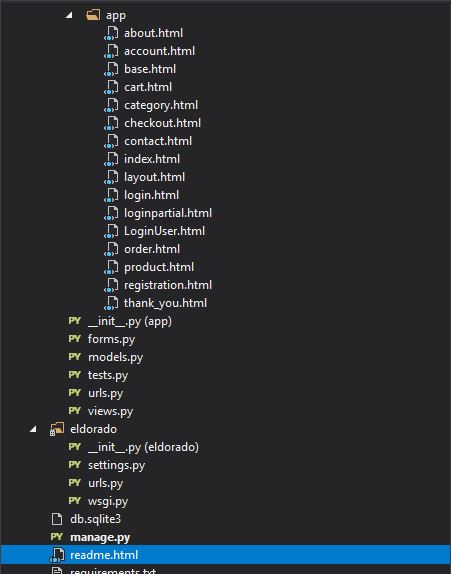
MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

MEDIA\_URL = '/media/'

CRISPY\_TAMPLATE\_PACK='bootstrap3'

****

****

****

**Style.CSS**

body {

padding-top: 70px;

padding-bottom: 20px;

}

/\* Set padding to keep content from hitting the edges \*/

.body-content {

padding-left: 15px;

padding-right: 15px;

}

/\* Set width on the form input elements since they're 100% wide by default \*/

input,

select,

textarea {

max-width: 280px;

}

/\* styles for validation helpers \*/

.field-validation-error {

color: #b94a48;

}

.field-validation-valid {

display: none;

}

input.input-validation-error {

border: 1px solid #b94a48;

}

input[type="checkbox"].input-validation-error {

border: 0 none;

}

.validation-summary-errors {

color: #b94a48;

}

.validation-summary-valid {

display: none;

}

#testCarousel {

width: 800px;

height: 200px;

position: absolute;

left:40%;

margin-left: -250px; /\*половина ширины блока, который центрируем \*/

}

/\*.item img {

width: 250px;

height:50px;

padding: 40px 10px;

padding-left: -16px;

}\*/

.carousel-item {

transition: transform 2s ease, opacity .5s ease-out

}

/\*.carousel .item {

background: #333;

text-align: center;

height: 300px !important;

}

.carousel h2 {

color: #fff;

margin: 0;

padding-top: 50px;

font-size: 48px;

}

.carousel p {

color: #666;

font-weight: bold;

font-size: 16px;

margin-top: 30px;

margin-bottom: 40px;

}\*/

**Index.html**

{% extends "app/layout.html" %}

{% block content %}

<div class="jumbotron">

<h1>Django</h1>

<p class="lead">Django is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.</p>

<p><a href="https://www.djangoproject.com/" class="btn btn-primary btn-large">Learn more &raquo;</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>Getting started</h2>

<p>

Django gives you a powerful, patterns-based way to build dynamic websites that

enables a clean separation of concerns and gives you full control over markup

for enjoyable, agile development.

</p>

<p><a class="btn btn-default" href="http://www.djangobook.com/en/2.0/index.html">Learn more &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Get more libraries</h2>

<p>The Python Package Index is a repository of software for the Python programming language.</p>

<p><a class="btn btn-default" href="https://pypi.org/">Learn more &raquo;</a></p>

</div>

<div class="col-md-4">

<h2>Microsoft Azure</h2>

<p>You can easily publish to Microsoft Azure using Visual Studio. Find out how you can host your application using a free trial today.</p>

<p><a class="btn btn-default" href="http://azure.microsoft.com">Learn more &raquo;</a></p>

</div>

</div>

{% endblock %}