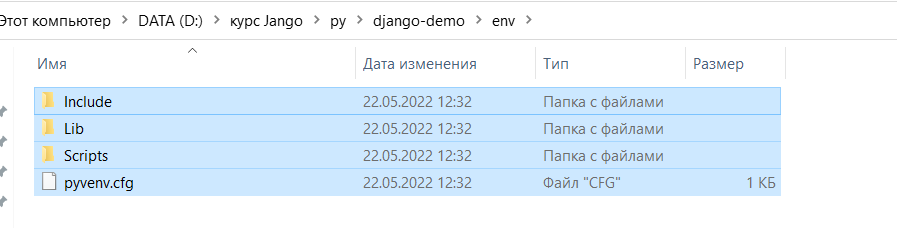
Инструкция к курсу

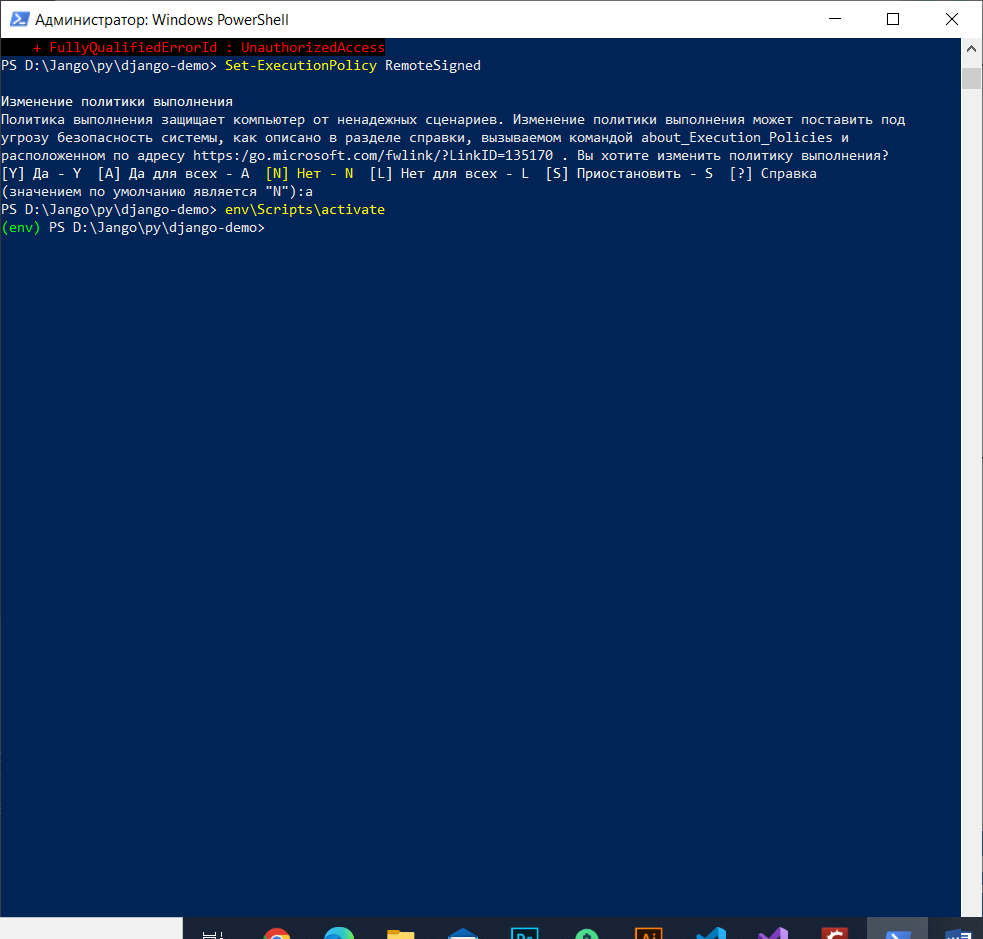
1. Создаем папку py
2. Открываем папку в PowerShell, либо командной строке (shift+пкм)
3. проверяем версию Python py -3 –V
4. проверяем установленные пакеты pip list
5. устанавливаем Django и pillow pip install django pillow
6. Установка виртуальной среды py -3 -m pip install virtualenvwrapper-win
7. В папке py создаем поддиректорию django-demo
8. Переходим в папку django-demo cd django-demo
9. Создаем в ней виртуальную среду py -3 -m venv env
10. Заходим в папку django-demo, убеждаемся, что в ней создана папка env, в которой следующее содержимое



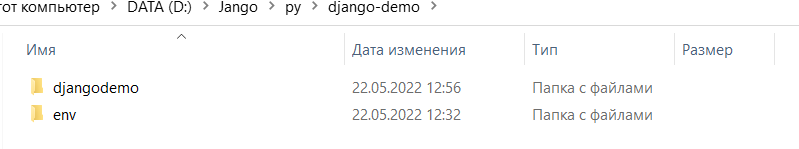
1. Активируем виртуальную среду env\Scripts\activate (deactivate — выход из текущей виртуальной среды Python)

# (если не активируется: - Открываем терминал PowerShell от админа. - Вставляем и запускаем - Set-ExecutionPolicy RemoteSigned - На вопрос отвечаем - A)

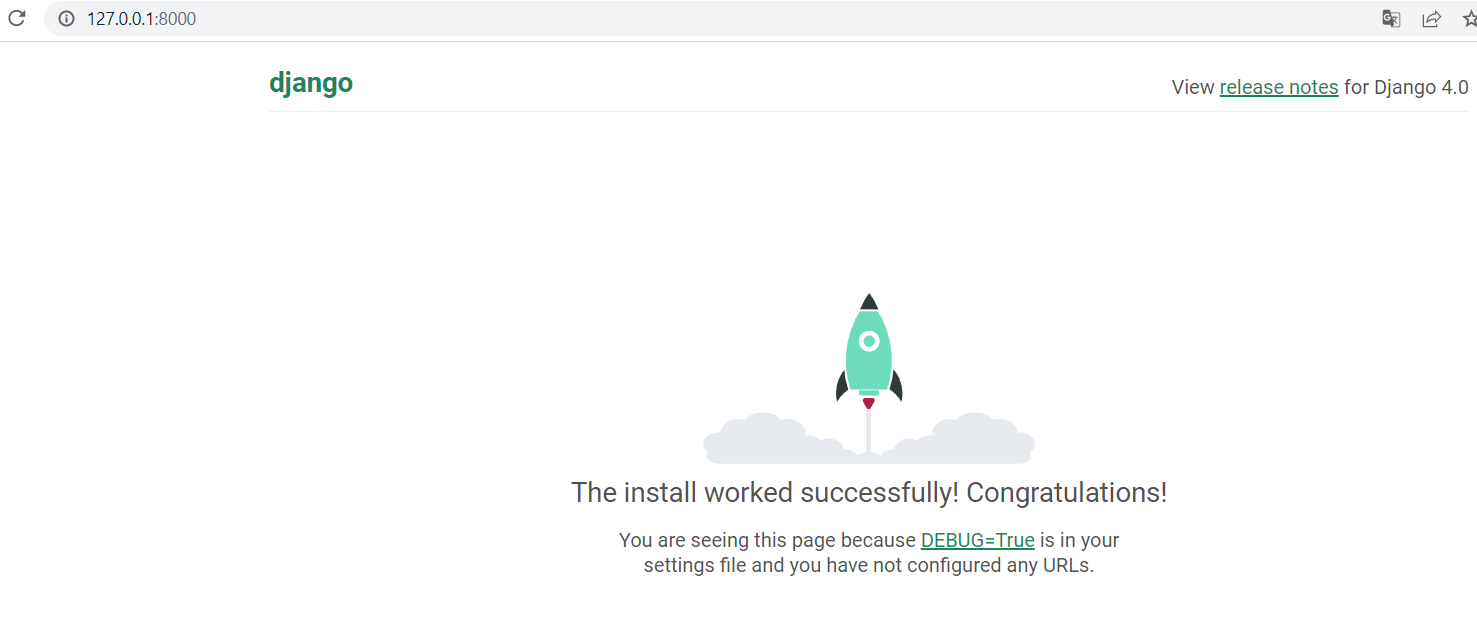
1. Убеждаемся, что среда запущена



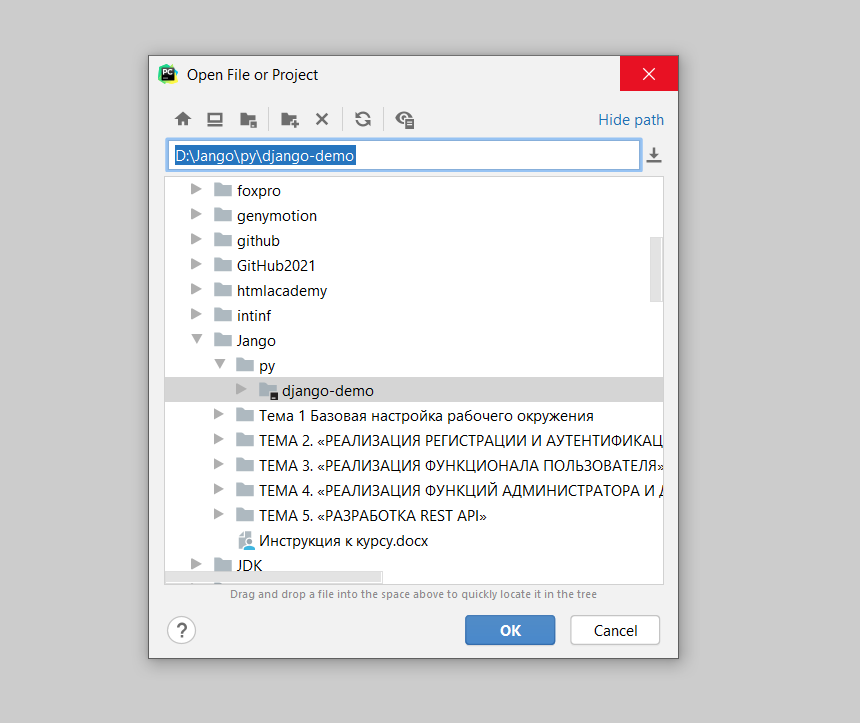
1. Далее снова проверяем pip list, обновляем его python -m pip install --upgrade pip
2. затем в виртуальную среду снова устанавливаем pip install django pillow
3. Проверяем работу django-admin
4. создаем проект django-admin startproject demoproject
5. Заходим в папку, убеждаемся, что проект создан



1. Переходим в созданную папку cd demoproject
2. Запускаем сервер разработки py -3 manage.py runserver
3. Копируем ip-адрес, убеждаемся, что django запущен



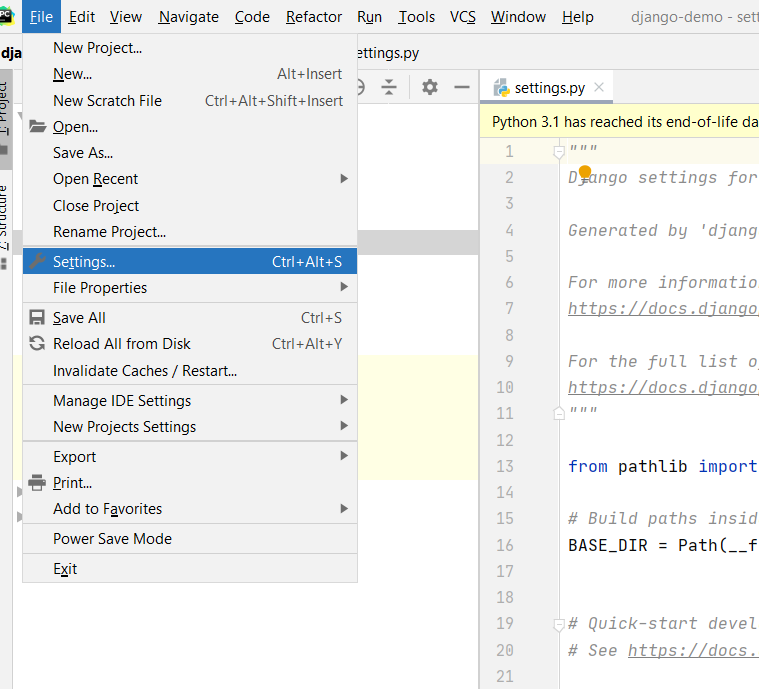
1. В настройках проекта (settings.py) разрешить проксирование заголовков ● USE\_X\_FORWARDED\_HOST = True
2. Quit the server with CTRL-BREAK.
3. Открываем папку django-demo в PyCharm



1. Отмечаем папку django-demo как Sources Root

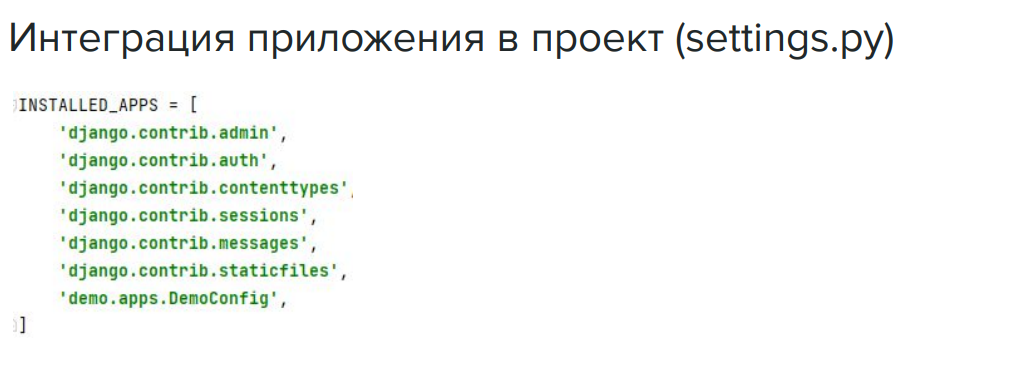


1. В настройках добавляем установленные пакеты

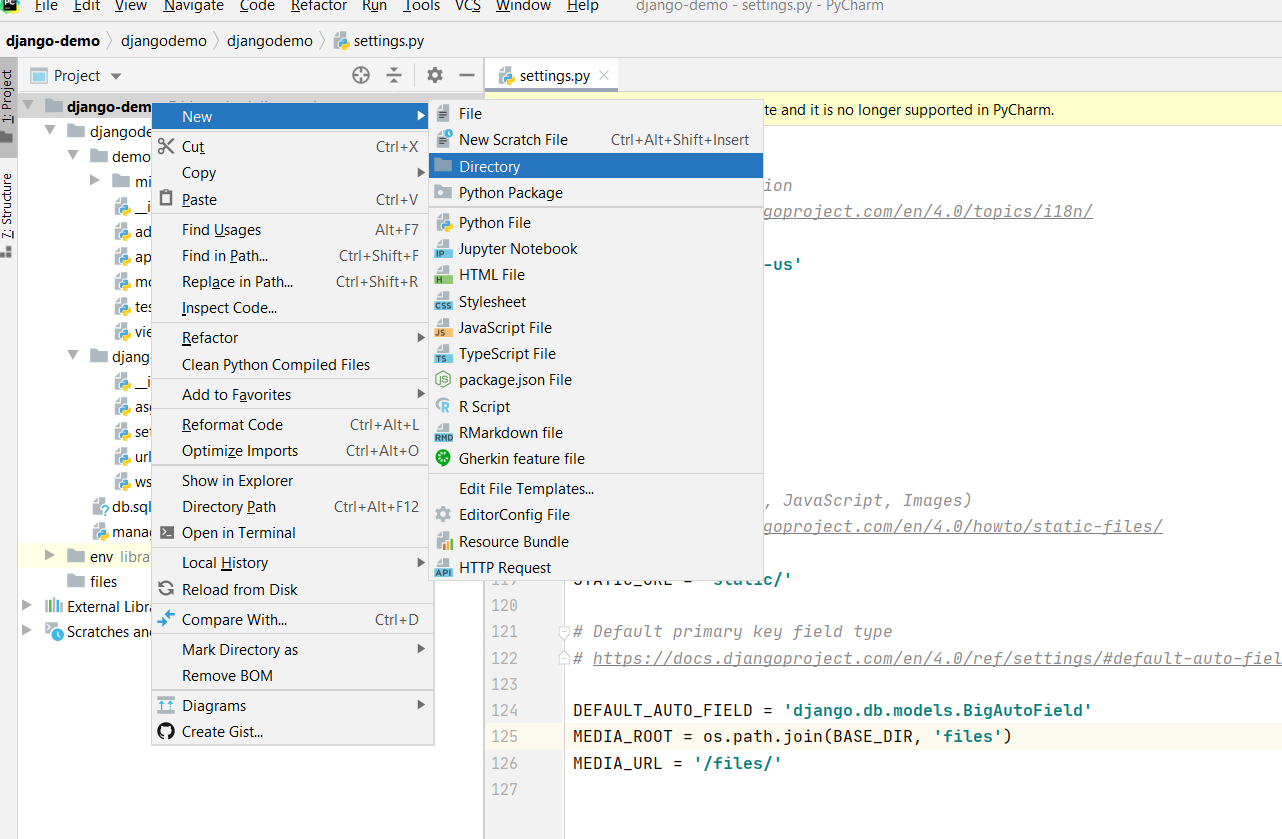




1. Затем создаем новое приложение python manage.py startapp demo
2. И интегрируем приложение в проект



1. Создаем папку для файлов проекта files

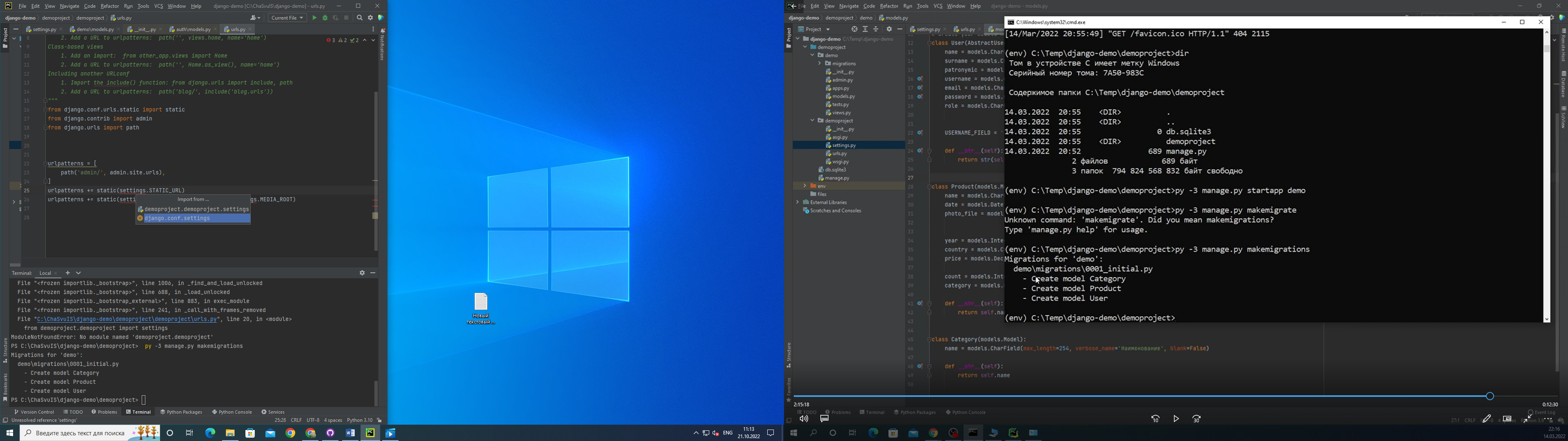


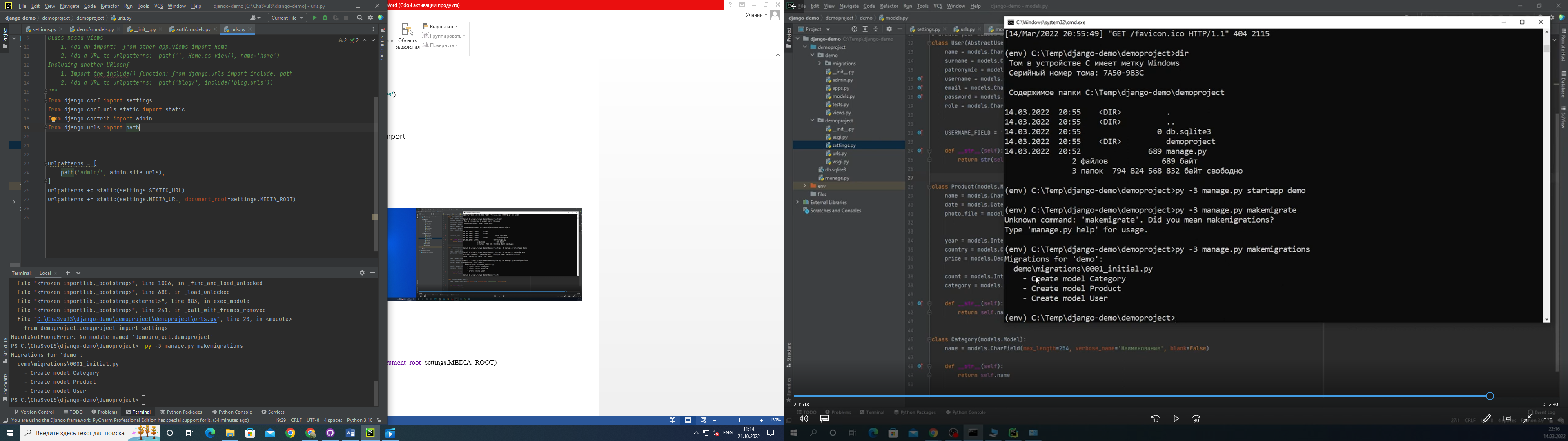
1. Прописываем доступ к папке

MEDIA\_ROOT = os.path.join(BASE\_DIR, **'files'**)  
MEDIA\_URL = **'/files/'**

AUTH\_USER\_MODEL = **'demo.User'**

1. Добавить библиотеку ALT+Enter Import
2. CTRL+ALT+L выравнивает код
3. (urls.py)

from django.conf.urls.static import static  
from django.contrib import admin  
from django.urls import path, include  
  
from demo import urls  


  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**''**, include(**'demo.urls'**)),  
]  
  
urlpatterns += static(settings.STATIC\_URL)  
urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

1. Создаем базу данных (models.py)

from django.db import models  
  
*# Create your models here.*from django.contrib.auth.models import AbstractUser  
from django.core.validators import FileExtensionValidator  
from django.db import models  
from django.utils.crypto import get\_random\_string  
  
  
def get\_name\_file(instance, filename):  
 return **'/'**.join([get\_random\_string(length=5) + **'\_'** + filename])  
  
*# Create your models here.*class User(AbstractUser):  
 name =models.CharField(max\_length=254, verbose\_name=**'Имя'**, blank=False)  
 surname = models.CharField(max\_length=254, verbose\_name=**'Фамилия'**, blank=False)  
 patronymic = models.CharField(max\_length=254, verbose\_name=**'Отчество'**, blank=True)  
 username = models.CharField(max\_length=254, verbose\_name=**'Логин'**, unique=True, blank=False)  
 email = models.CharField(max\_length=254, verbose\_name=**'Почта'**, unique=True, blank=False)  
 password = models.CharField(max\_length=254, verbose\_name=**'Пароль'**, blank=False)  
 role = models.CharField(max\_length=254, verbose\_name=**'Роль'**,  
 choices=((**'admin'**, **'Администратор'**), (**'user'**, **'Пользователь'**)), default=**'user'**)  
  
 USERNAME\_FIELD = **'username'** def \_\_str\_\_(self):  
 return str(self.name) +**" "**+ str(self.surname)  
  
class Product(models.Model):  
 name = models.CharField(max\_length=254, verbose\_name=**'Имя'**, blank=False)  
 date = models.DateTimeField(verbose\_name=**'Дата добавления'**, auto\_now\_add=True)  
 photo\_file = models.ImageField(max\_length=254, upload\_to=get\_name\_file,  
 blank=True, null=True,  
 validators=[FileExtensionValidator(allowed\_extensions=[**'png'**, **'jpg'**, **'jpeg'**])])  
 year = models.IntegerField(verbose\_name=**'Год производства'**, blank=True)  
 country = models.CharField(max\_length=254, verbose\_name=**'Страна производства'**, blank=True)  
 price = models.DecimalField(verbose\_name=**'Стоимость'**, max\_digits=10, decimal\_places=2, blank=False,  
 default=0.00)  
 count = models.IntegerField(verbose\_name=**'Количество'**, blank=False, default=0)  
 category = models.ForeignKey(**'Category'**, verbose\_name=**'Категория'**, on\_delete=models.CASCADE)  
  
 def \_\_str\_\_(self):  
 return str(self.name)  
  
class Category (models.Model):  
 name = models.CharField(max\_length=254, verbose\_name=**'Наименование'**, blank=False)  
  
 def \_\_str\_\_(self):  
 return str(self.name)  
  
class Order(models.Model):  
 STATUS\_CHOISES = [  
 (**'new'**, **'Новый'**),  
 (**'confirmed'**, **'Подтвержденный'**),  
 (**'canceled'**, **'Отмененный'**)  
 ]  
 date = models.DateTimeField(verbose\_name=**'Дата заказа'**, auto\_now\_add=True)  
 status = models.CharField(max\_length=254, verbose\_name=**'Статус'**,  
 choices=STATUS\_CHOISES,  
 default=**'new'**)  
 user = models.ForeignKey(User, verbose\_name=**'Пользователь'**, on\_delete=models.CASCADE)  
 rejection\_reason = models.TextField(verbose\_name=**'Причина отказа'**, blank=True)  
 products = models.ManyToManyField(Product, through=**'ItemInOrder'**, related\_name=**'orders'**)  
  
  
class ItemInOrder(models.Model):  
 order = models.ForeignKey(Order, verbose\_name=**'Заказ'**, on\_delete=models.CASCADE)  
 product = models.ForeignKey(Product, verbose\_name=**'Товар'**, on\_delete=models.CASCADE)  
 count = models.IntegerField(verbose\_name=**'Количество'**, blank=False, default=0)  
 price = models.DecimalField(verbose\_name=**'Стоимость'**, max\_digits=10, decimal\_places=2, blank=False,  
 default=0.00)  
  
  
class Cart(models.Model):  
 user = models.ForeignKey(User, verbose\_name=**'Пользователь'**, on\_delete=models.CASCADE)  
 product = models.ForeignKey(Product, verbose\_name=**'Товар'**, on\_delete=models.CASCADE)  
 count = models.IntegerField(verbose\_name=**'Количество'**, blank=False, default=0)  
  
 def \_\_str\_\_(self):  
 return self.product.name + **' - '** + str(self.count)

1. Выполняем в PowerShell миграцию базы данных py manage.py makemigrations

затем

py manage.py migrate

1. создается суперпользователь   
   (env) PS D:\django2\django-demo\demoproject> py -3 manage.py createsuperuser

Логин: admin

Почта: admin@admin.ru

Password:

Password (again):

The password is too similar to the Логин.

This password is too short. It must contain at least 8 characters.

This password is too common.

Bypass password validation and create user anyway? [y/N]: y

Запускаем сервер py -3 manage.py runserver

Запускаем админ панель по ссылке <http://127.0.0.1:8000/admin>,

(admin.py)

from django.contrib import admin  
  
*# Register your models here.*from demo.models import \*  
  
admin.site.register(User)  
admin.site.register(Product)  
admin.site.register(Category)  
admin.site.register(Order)  
admin.site.register(ItemInOrder)  
admin.site.register(Cart)

1. создание шаблона
2. в папке demoproject (верхней) создаем папку templates. В папке templates создаем файл base.html

<!doctype html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 {% block title %}<title>Интернет-магазин</title> {% endblock %}  
 {% load static %}  
 <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
 <link rel="stylesheet" href="{% static 'css/main.css' %}">  
</head>  
<body>  
<nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top">  
 <a class="navbar-brand" href="#">Интернет-магазин</a>  
 <div class="collapse navbar-collapse" id="navbarsExamoleDefault">  
 {% block sidebar %}  
 <ul class="navbar-nav mr-auto">  
 <li class="nav-item">  
 <a class="nav-link" href="#">Главная</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="#">Каталог</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="#">О нас</a>  
 </li>  
 </ul>  
 {% endblock %}  
 </div>  
</nav>  
<main role="main" class="container">  
 <div class="starter-template">  
 {% block content %}  
 {% endblock %}  
 </div>  
</main>  
</body>  
</html>

1. В папке demo создаем папку static, в папке static создаем папку css, в папку css кладем файл bootstrap.min.css и создаем файл main.css
2. В файл main.css добавляем код

body {  
 padding-top: 5rem;  
}  
.starter-template {  
 padding: 3rem 1.5rem;  
}

1. В папке templates создаем папку registration, в папке registration создаем файл login.html

{% extends "base.html" %}  
{% block content %}  
 <h1>Аутентификация</h1>  
 {% if form.errors %}  
 <div class="alert-danger">  
 <p class="p-1">Неправильный логин или пароль</p>  
 </div>  
 {% endif %}  
 {% if user.is.authenticated %}  
 <p>Вы уже авторизованы <a href="{% url 'logout' %}">Выйти</a></p>  
 {% else %}  
 <form method="post" action="{% url 'login' %}">  
 {% cstrf\_token %}  
 <table>  
 <tr>  
 <td>{{ form.username.label\_tag }}</td>  
 <td>{{ form.username }}</td>  
 </tr>  
 <tr>  
 <td>{{ form.username.label\_tag }}</td>  
 <td>{{ form.username }}</td>  
 </tr>  
 </table>  
 <input type="submit" value="Войти">  
 <input type="hidden" name="text" value="{{ next }}">  
 </form>  
 {% endif %}  
{% endblock %}

1. В папке Demo Создать urls.py

Код в demo/urls.py \*

from django.urls import path

from django.contrib.auth import views as auth\_views

from demo import views

urlpatterns = [

path('login/', auth\_views.LoginView.as\_view(), name='login'),

path('logout/', auth\_views.LogoutView.as\_view(), name='logout'),

]

Изменения settings.py

\*

TEMPLATES = [

{

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

'DIRS': [], // Нужно так 'DIRS': [BASE\_DIR / 'templates'],

'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',

],

},

},

]

\*

login.html удалил 13 строчку {% cstrf\_token %}