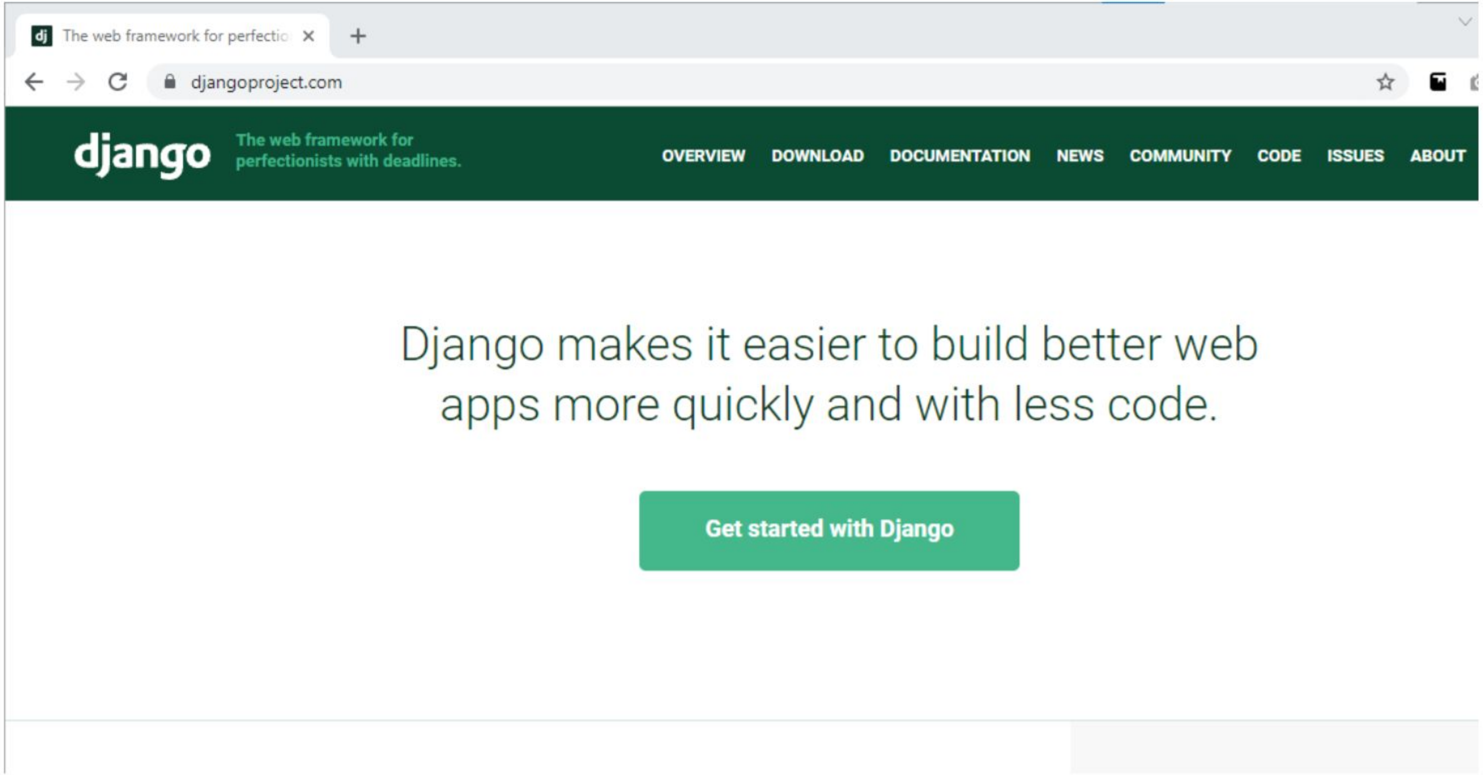
Get to Know Django Better

Django - Python web framework


The web framework for perfectio
djangoproject.com
django
The web framework for perfectionists with deadlines.
OVERVIEW
DOWNLOAD
DOCUMENTATION
NEWS
COMMUNITY
CODE
ISSUES
ABOUT

Django makes it easier to build better web apps more quickly and with less code.

Django features

Data-driven via connection with database

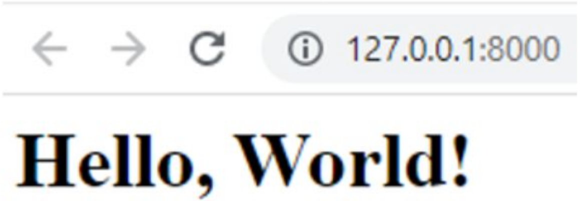
Built-in Object-Relational Mapper (ORM)

Built-in administration portal

Built-in development environment

django-admin startproject

Docker exercises used provided, pre-generated Django project



127.0.0.1:8000

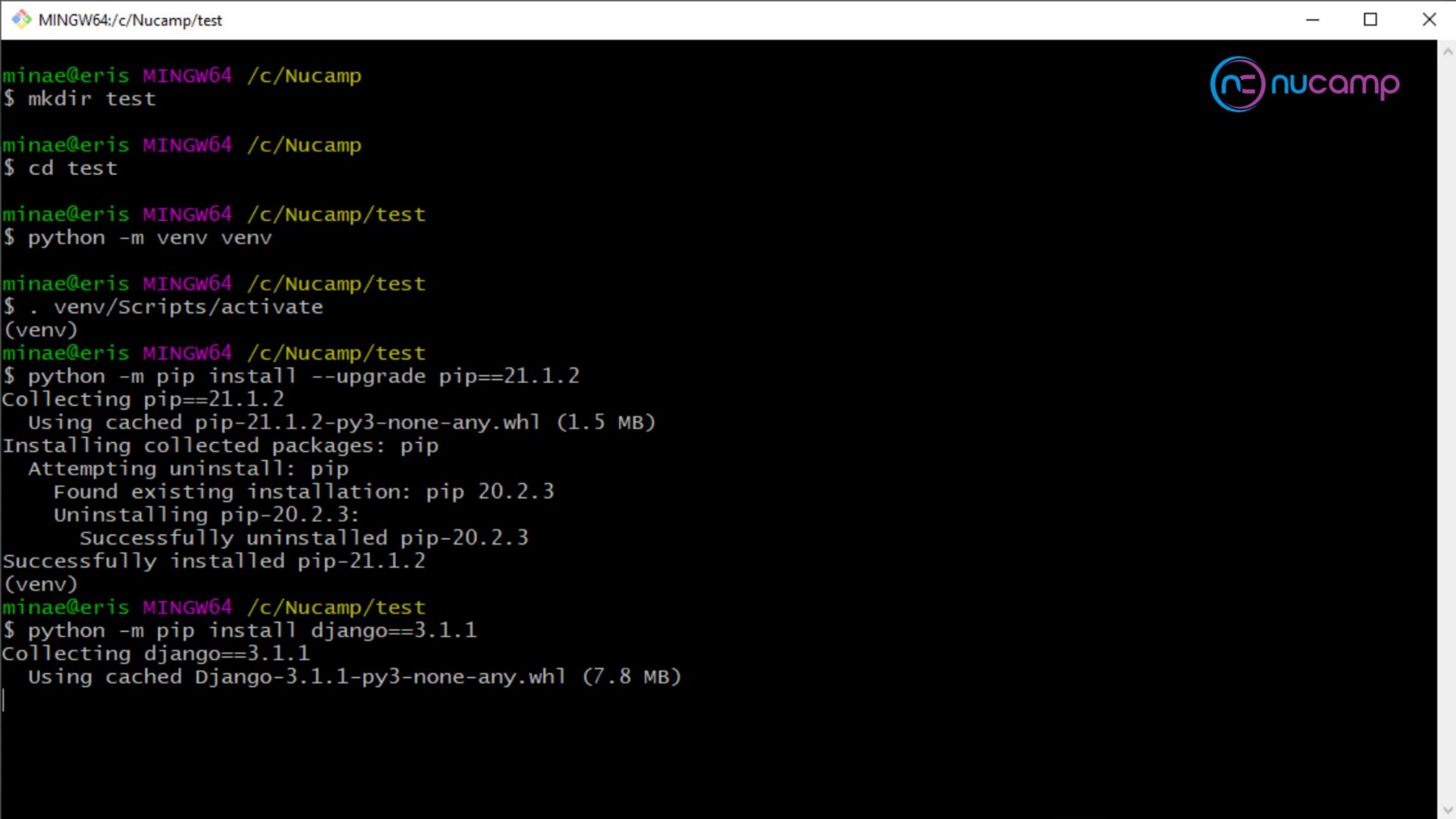
Hello, World!

Workshop: You will generate the Django project yourself

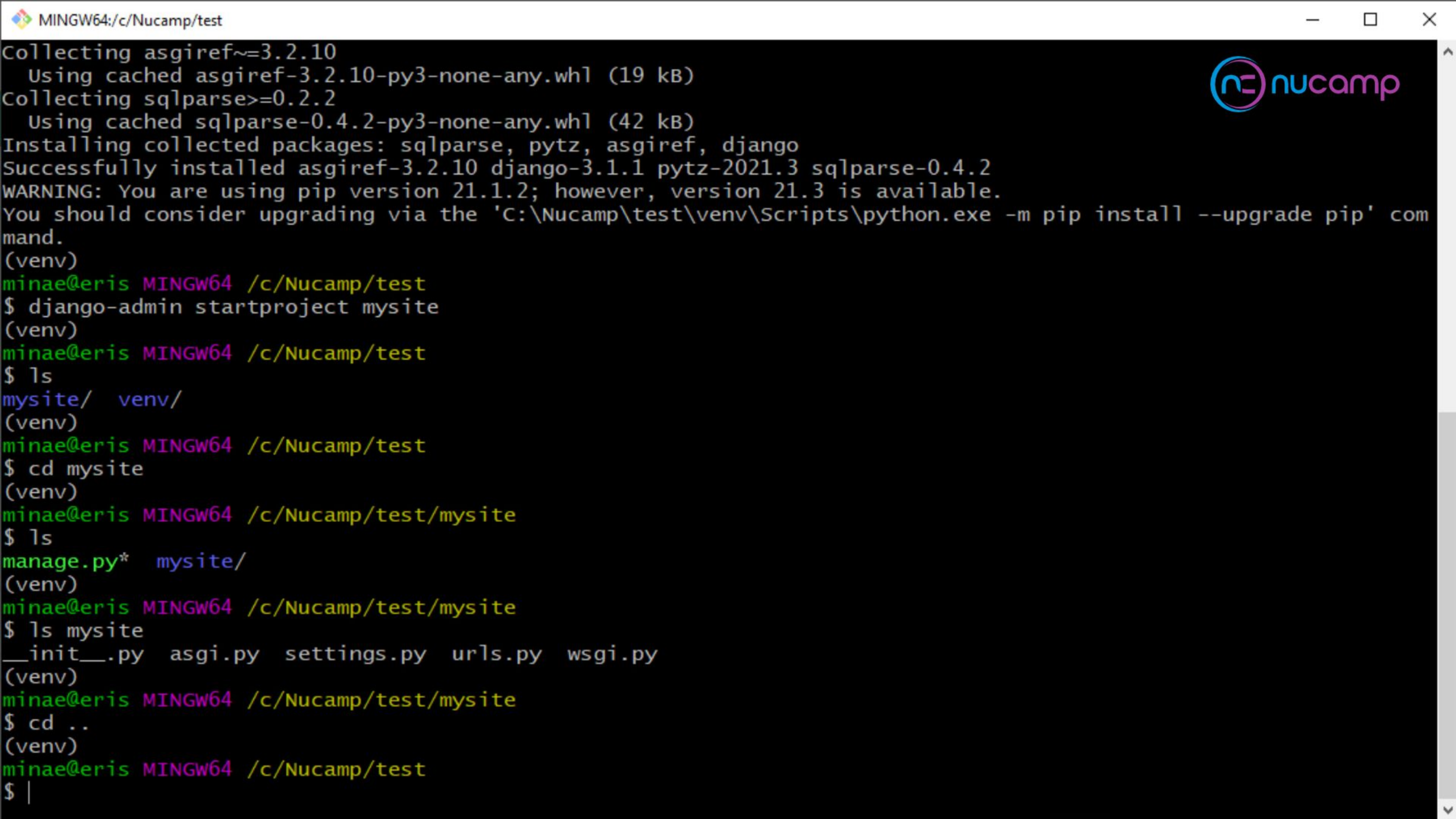
Use **django-admin** CLI tool to scaffold out starter files

Syntax:

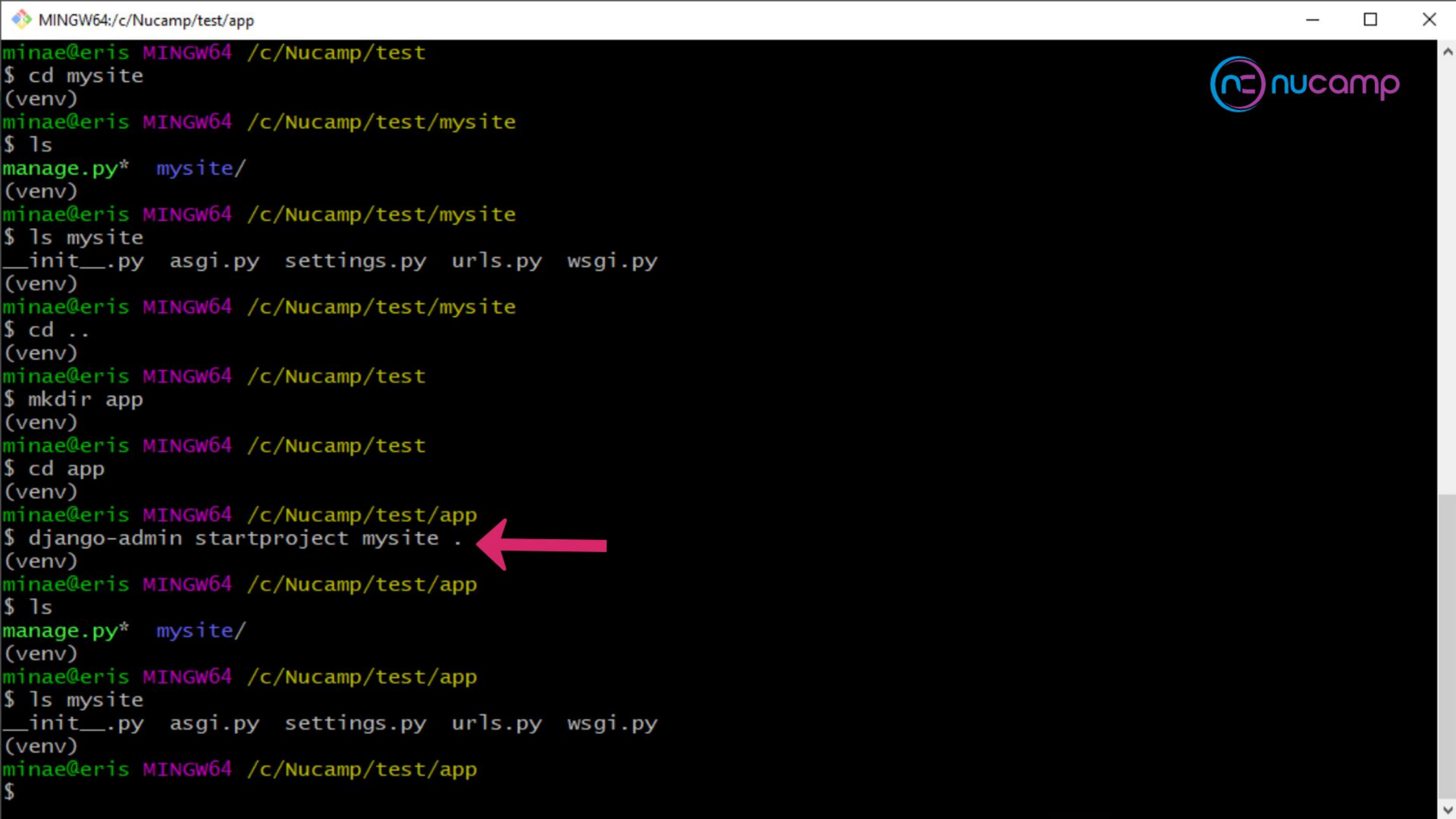
django-admin startproject *mysite* (starts new project folder named mysite)
django-admin startproject *mysite* . (uses current folder as project folder)



```
minae@eris MINGW64 /c/Nucamp
$ mkdir test

minae@eris MINGW64 /c/Nucamp
$ cd test

minae@eris MINGW64 /c/Nucamp/test
$ python -m venv venv

minae@eris MINGW64 /c/Nucamp/test
$ . venv/Scripts/activate
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ python -m pip install --upgrade pip==21.1.2
Collecting pip==21.1.2
  Using cached pip-21.1.2-py3-none-any.whl (1.5 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.2.3
    Uninstalling pip-20.2.3:
      Successfully uninstalled pip-20.2.3
Successfully installed pip-21.1.2
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ python -m pip install django==3.1.1
Collecting django==3.1.1
  Using cached Django-3.1.1-py3-none-any.whl (7.8 MB)
```



```
Collecting asgiref~=3.2.10
  Using cached asgiref-3.2.10-py3-none-any.whl (19 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.2.10 django-3.1.1 pytz-2021.3 sqlparse-0.4.2
WARNING: You are using pip version 21.1.2; however, version 21.3 is available.
You should consider upgrading via the 'C:\Nucamp\test\venv\Scripts\python.exe -m pip install --upgrade pip' com
mand.
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ django-admin startproject mysite
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ ls
mysite/  venv/
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ cd mysite
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ ls
manage.py*  mysite/
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ ls mysite
__init__.py  asgi.py  settings.py  urls.py  wsgi.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ cd ..
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ |
```



MINGW64:/c/Nucamp/test/app

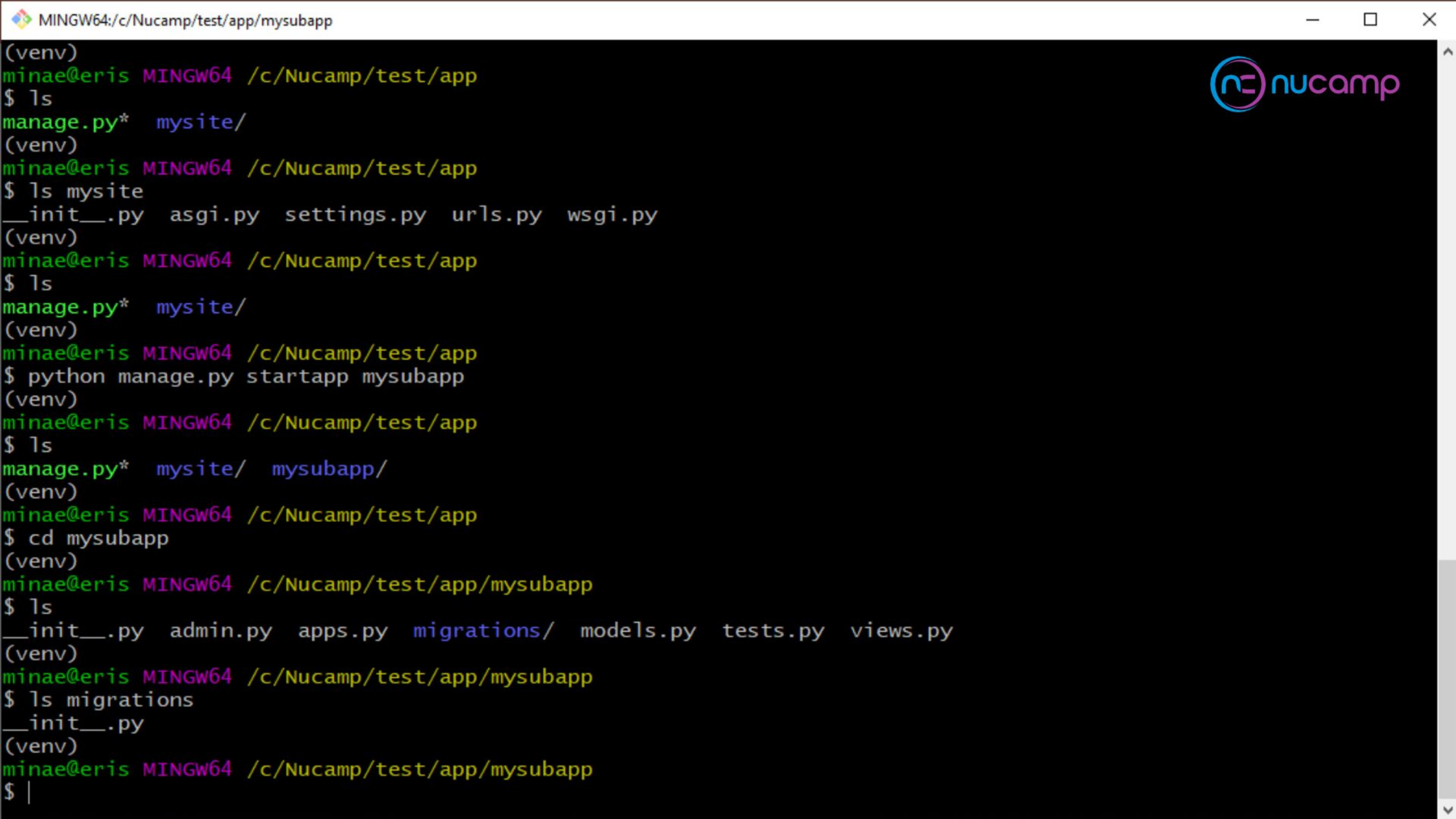
```
minae@eris MINGW64 /c/Nucamp/test
$ cd mysite
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ ls
manage.py*  mysite/
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ ls mysite
__init__.py  asgi.py  settings.py  urls.py  wsgi.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/mysite
$ cd ..
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ mkdir app
(venv)
minae@eris MINGW64 /c/Nucamp/test
$ cd app
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ django-admin startproject mysite .
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls
manage.py*  mysite/
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls mysite
__init__.py  asgi.py  settings.py  urls.py  wsgi.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$
```

Add sub-apps

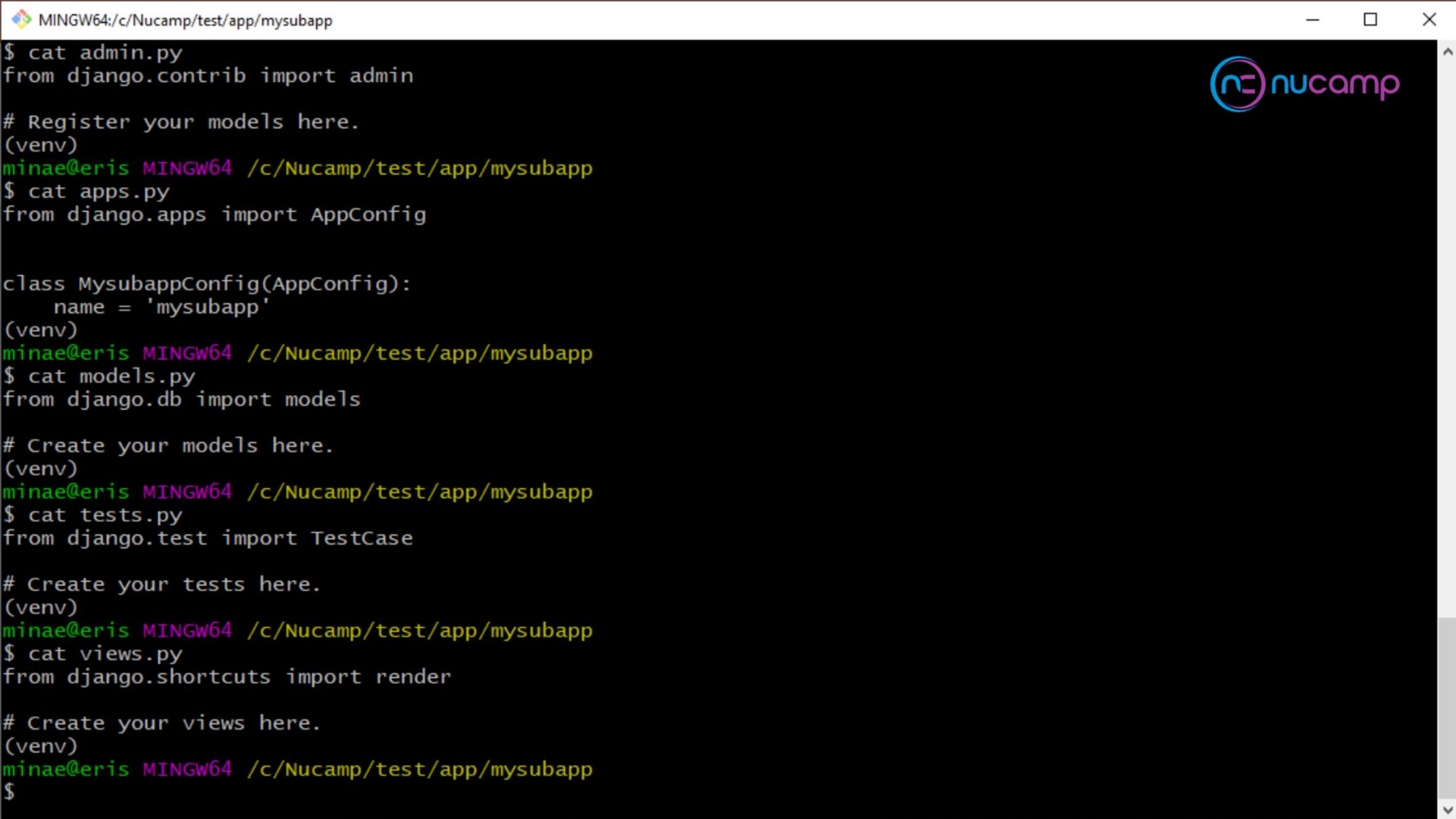
From project root folder (same location as manage.py), use:
python manage.py startapp *mysubapp*

New ***mysubapp*** folder will be created inside project root folder
Starter files will be generated inside it

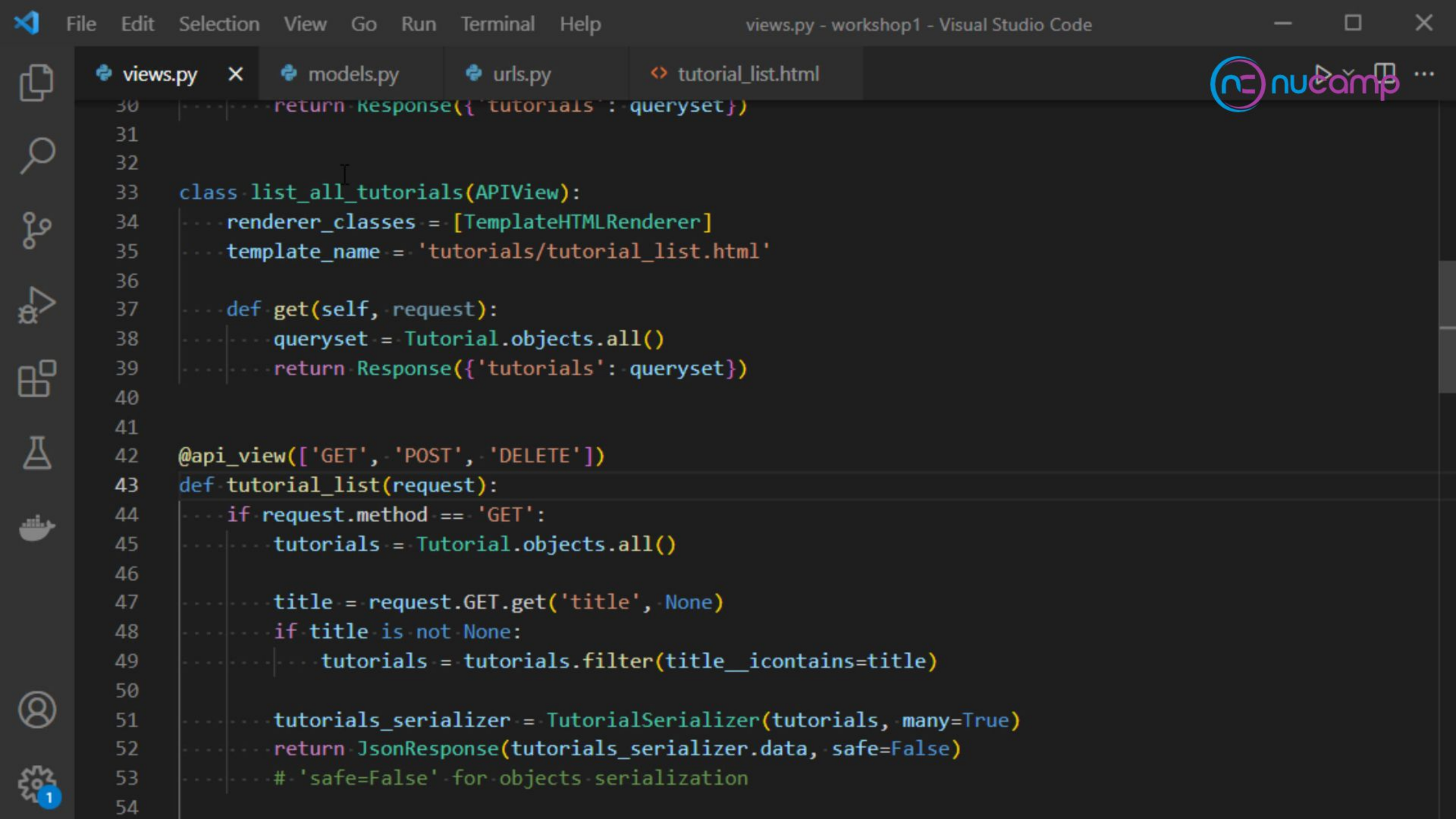
Must add sub-app to main app's **settings.py INSTALLED_APPS** list



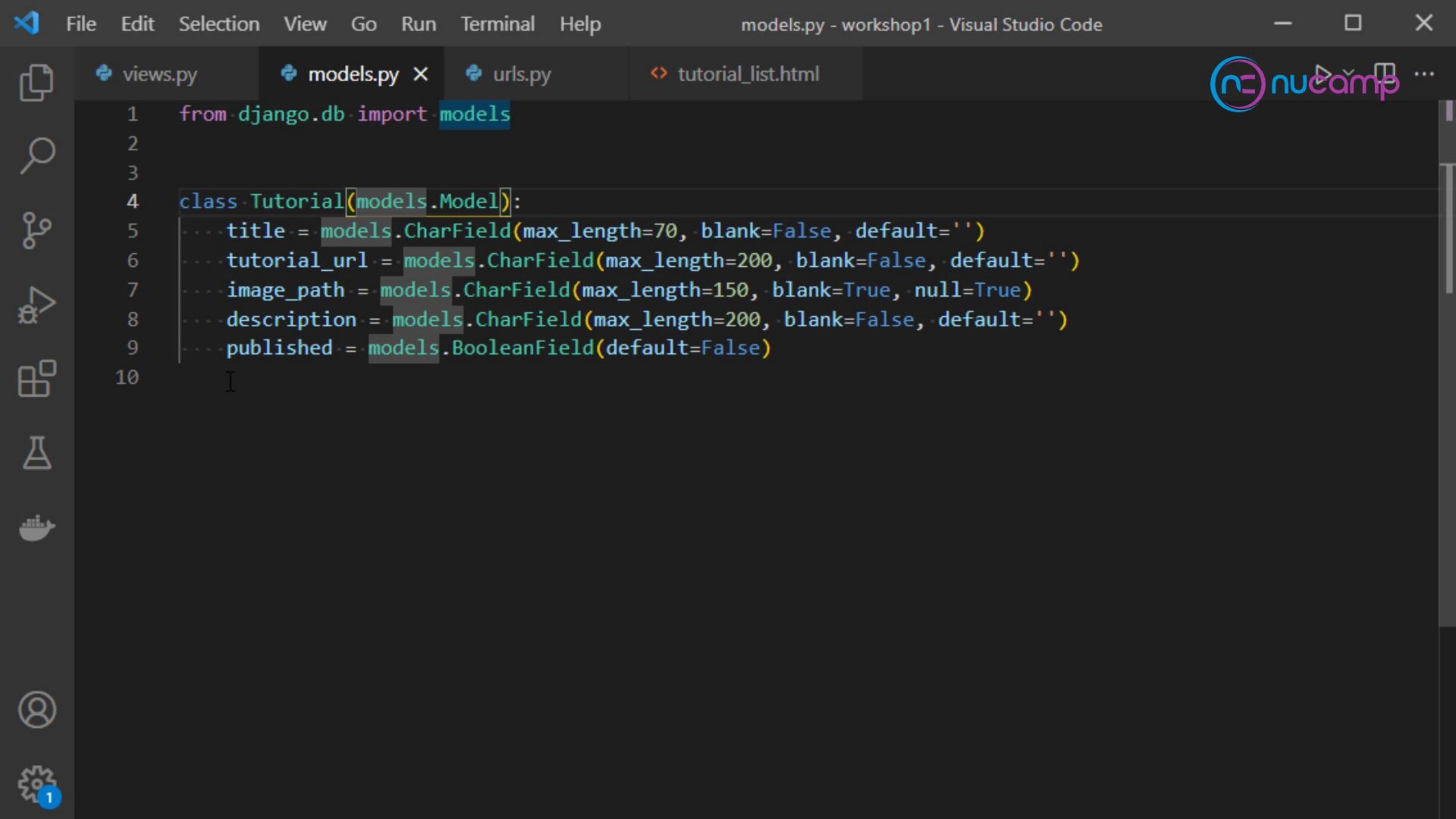
```
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls
manage.py*  mysite/
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls mysite
__init__.py  asgi.py  settings.py  urls.py  wsgi.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls
manage.py*  mysite/
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ python manage.py startapp mysubapp
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ ls
manage.py*  mysite/  mysubapp/
(venv)
minae@eris MINGW64 /c/Nucamp/test/app
$ cd mysubapp
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ ls
__init__.py  admin.py  apps.py  migrations/  models.py  tests.py  views.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ ls migrations
__init__.py
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ 
```



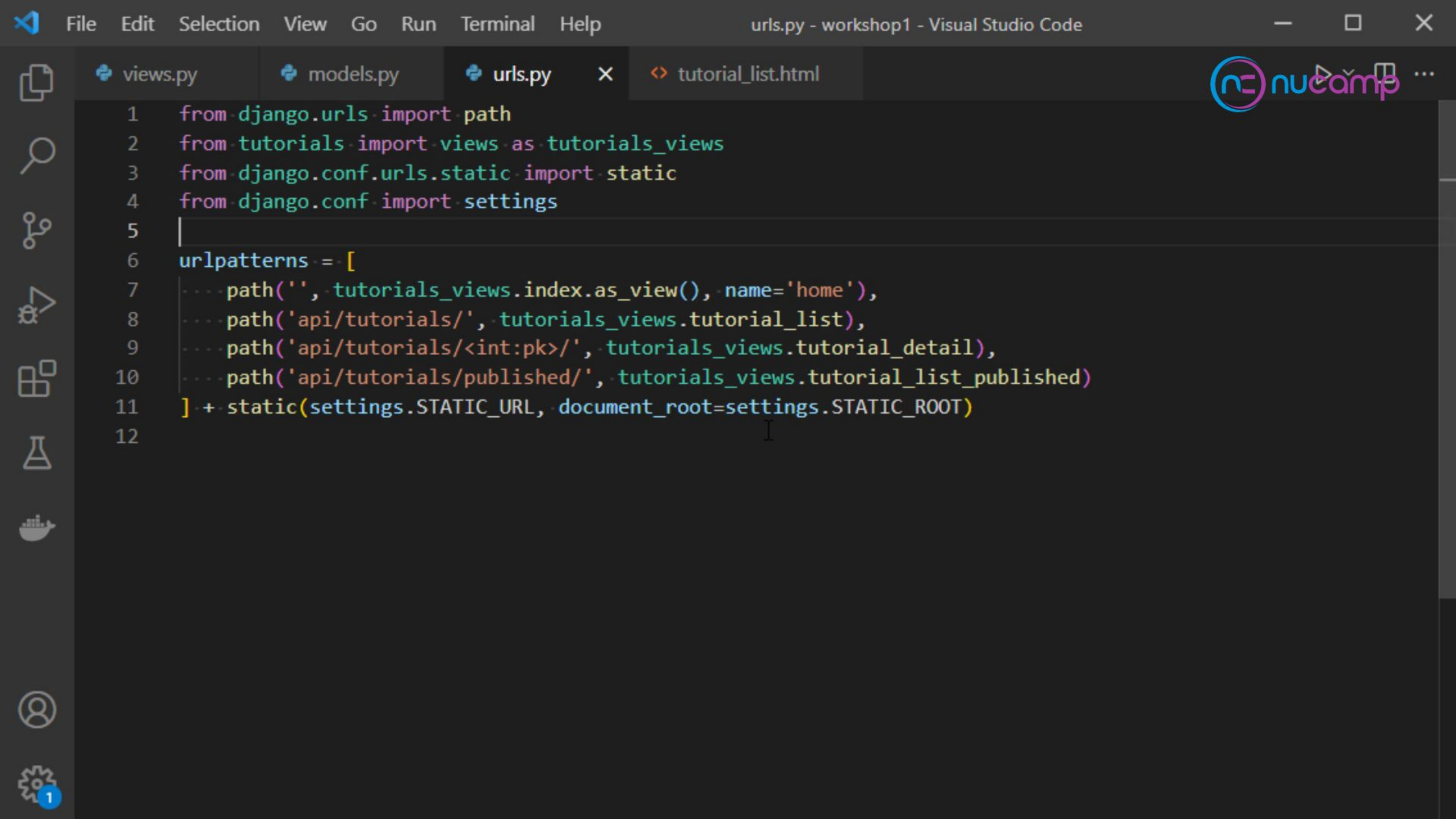
```
$ cat admin.py
from django.contrib import admin

# Register your models here.
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ cat apps.py
from django.apps import AppConfig


class MysubappConfig(AppConfig):
    name = 'mysubapp'
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ cat models.py
from django.db import models

# Create your models here.
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ cat tests.py
from django.test import TestCase

# Create your tests here.
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$ cat views.py
from django.shortcuts import render

# Create your views here.
(venv)
minae@eris MINGW64 /c/Nucamp/test/app/mysubapp
$
```



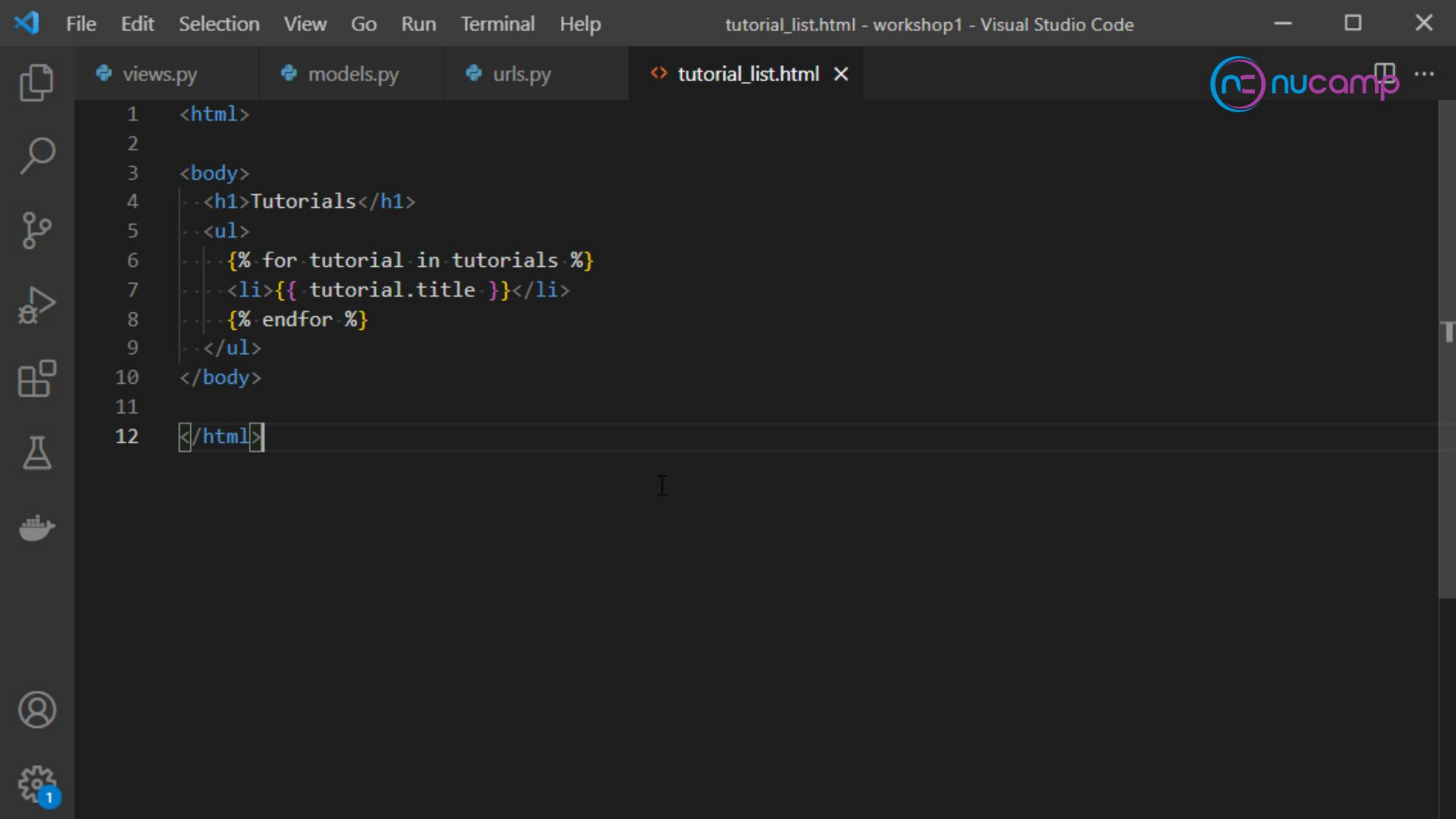
```python
        return Response({'tutorials': queryset})


class list_all_tutorials(APIView):
    renderer_classes = [TemplateHTMLRenderer]
    template_name = 'tutorials/tutorial_list.html'

    def get(self, request):
        queryset = Tutorial.objects.all()
        return Response({'tutorials': queryset})


@api_view(['GET', 'POST', 'DELETE'])
def tutorial_list(request):
    if request.method == 'GET':
        tutorials = Tutorial.objects.all()

        title = request.GET.get('title', None)
        if title is not None:
            tutorials = tutorials.filter(title__icontains=title)

        tutorials_serializer = TutorialSerializer(tutorials, many=True)
        return JsonResponse(tutorials_serializer.data, safe=False)
        # 'safe=False' for objects serialization

```



```python
from django.db import models


class Tutorial(models.Model):
    title = models.CharField(max_length=70, blank=False, default='')
    tutorial_url = models.CharField(max_length=200, blank=False, default='')
    image_path = models.CharField(max_length=150, blank=True, null=True)
    description = models.CharField(max_length=200, blank=False, default='')
    published = models.BooleanField(default=False)

```



```python
from django.urls import path
from tutorials import views as tutorials_views
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('', tutorials_views.index.as_view(), name='home'),
    path('api/tutorials/', tutorials_views.tutorial_list),
    path('api/tutorials/<int:pk>/', tutorials_views.tutorial_detail),
    path('api/tutorials/published/', tutorials_views.tutorial_list_published)
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```



File Edit Selection View Go Run Terminal Help
tutorial_list.html - workshop1 - Visual Studio Code
views.py
models.py
urls.py
tutorial_list.html
nucamp

```
<html>

<body>
  <h1>Tutorials</h1>
  <ul>
    {% for tutorial in tutorials %}
    <li>{{ tutorial.title }}</li>
    {% endfor %}
  </ul>
</body>

</html>
```