1. Создание проекта в Shell

# Windows

> cd onedrive\desktop\code

> mkdir helloworld

> cd helloworld

1. Создание проекта

Shell

django-admin startproject django\_project

1. Создание виртуального окружения

Shell

# Windows

> python -m venv .venv #точка говорит о том что элемент скрытый -> Альтернативный способ – virtualenv venv

> venv\Scripts\Activate.ps1

(.venv) > python -m pip install django~=4.0.0

1. Первоначальная миграция

Shell

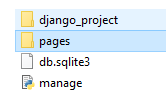
# Windows

> python manage.py migrate

1. Создание нового приложения

Shell

# Windows

(.venv) > python manage.py startapp pages 

1. Add new app in settings.py -> installed\_apps
2. pages/views.py:

# pages/views.py

from django.http import HttpResponse

def homePageView(request):

return HttpResponse("Hello, World!")

1. create a new file called urls.py

# pages/urls.py

from django.urls import path

from .views import homePageView

urlpatterns = [

path("", homePageView, name="home"),

]

1. django\_project/urls.py

# django\_project/urls.py

from django.contrib import admin

from django.urls import path, include # new

urlpatterns = [

path("admin/", admin.site.urls),

path("", include("pages.urls")), # new

]



Shell

(.venv) > git init



Shell

(.venv) > git status

1. Создать файл .gitignor и добавить в него .venv, чтобы не добавлять в репозиторий
2. pip freeze > requirements.txt
3. git add -A
4. git commit -m "initial commit"

**Templates**

1. Create pages/templates/pages/home.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
*<!-- templates/home.html -->*<h1>Homepage</h1>  
</body>  
</html>

1. update pages/views.py:

*# pages/views.py*from django.views.generic import TemplateView  
  
  
class HomePageView(TemplateView):  
 template\_name = "pages/home.html"

1. Update pages/urls.py:

*# pages/urls.py*from django.urls import path  
from .views import HomePageView  
  
urlpatterns = [  
 path("", HomePageView.as\_view(), name="home"),  
]

1. Create pages/templates/pages/about.html:
2. <!DOCTYPE html>  
   <html lang="en">  
   <head>  
    <meta charset="UTF-8">  
    <title>Title</title>  
   </head>  
   <body>  
   *<!-- templates/about.html -->*<h1>About page</h1>  
   </body>  
   </html>
3. Add to pages/views.py:

class AboutPageView(TemplateView): *# new* template\_name = "pages/about.html"

1. Update pages/urls.py:

*# pages/urls.py*from django.urls import path  
from .views import HomePageView, AboutPageView  
  
urlpatterns = [  
 path("about/", AboutPageView.as\_view(), name="about"), *# new* path("", HomePageView.as\_view(), name="home"),  
]

1. Create templates/base.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
*<!-- templates/base.html -->*<header>  
 <a href="{% url 'home' %}">Home</a> |  
 <a href="{% url 'about' %}">About</a>  
</header>  
{% block content %} {% endblock content %}  
  
</body>  
</html>

1. Update template home:

*<!-- templates/home.html -->*{% extends "base.html" %}  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
*<!-- templates/home.html -->*{% block content %}  
 <h1>Homepage</h1>  
{% endblock content %}  
  
</body>  
</html>

1. Update template about:

*<!-- templates/about.html -->*{% extends "base.html" %}  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
*<!-- templates/about.html -->*{% block content %}  
 <h1>About page</h1>  
{% endblock content %}  
</body>  
</html>

**TEST**

1. Add to pages/tests.py:

*# pages/tests.py*from django.test import SimpleTestCase  
  
  
class HomepageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/")  
 self.assertEqual(response.status\_code, 200)  
  
  
class AboutpageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/about/")  
 self.assertEqual(response.status\_code, 200)



Shell

(.venv) > python manage.py test

1. Update pages/tests.py:

*# pages/tests.py*from django.test import SimpleTestCase  
from django.urls import reverse  
  
  
class HomepageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/")  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_url\_available\_by\_name(self): *# new* response = self.client.get(reverse("home"))  
 self.assertEqual(response.status\_code, 200)  
  
  
class AboutpageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/about/")  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_url\_available\_by\_name(self): *# new* response = self.client.get(reverse("about"))  
 self.assertEqual(response.status\_code, 200)



Shell

(.venv) > python manage.py test

1. Update pages/tests.py:

*# pages/tests.py*from django.test import SimpleTestCase  
from django.urls import reverse  
  
  
class HomepageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/")  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_url\_available\_by\_name(self): *# new* response = self.client.get(reverse("home"))  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_template\_name\_correct(self): *# new* response = self.client.get(reverse("home"))  
 self.assertTemplateUsed(response, "pages/home.html")  
  
 def test\_template\_content(self): *# new* response = self.client.get(reverse("home"))  
 self.assertContains(response, "<h1>Homepage</h1>")  
  
  
class AboutpageTests(SimpleTestCase):  
 def test\_url\_exists\_at\_correct\_location(self):  
 response = self.client.get("/about/")  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_url\_available\_by\_name(self): *# new* response = self.client.get(reverse("about"))  
 self.assertEqual(response.status\_code, 200)  
  
 def test\_template\_name\_correct(self): *# new* response = self.client.get(reverse("about"))  
 self.assertTemplateUsed(response, "pages/about.html")  
  
 def test\_template\_content(self): *# new* response = self.client.get(reverse("about"))  
 self.assertContains(response, "<h1>About page</h1>")