**How to use static files within the Django project.**

[[biswajit panda](https://medium.com/@biswajitpanda973?source=post_page-----eea9cafa8038--------------------------------)](https://medium.com/@biswajitpanda973?source=post_page-----eea9cafa8038--------------------------------)

[biswajit panda](https://medium.com/@biswajitpanda973?source=post_page-----eea9cafa8038--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F2cd894ffbe14&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40biswajitpanda973%2Fhow-to-use-static-files-within-the-django-project-eea9cafa8038&user=biswajit+panda&userId=2cd894ffbe14&source=post_page-2cd894ffbe14----eea9cafa8038---------------------post_header-----------)

3 min read

·

Dec 11, 2023

Using static files within a Django project is an essential aspect of web development, allowing you to manage and serve assets such as CSS, JavaScript, and images. Here’s a step-by-step guide on how to use static files in a Django project:

for more details about folder structure read this [article](https://medium.com/@biswajitpanda973/render-your-first-html-page-with-django-3e96721b180d)

<https://medium.com/@biswajitpanda973/render-your-first-html-page-with-django-3e96721b180d>

folder structure :

myproject   
│  
├── manage.py  
│  
└── myproject (inner folder)  
 ├── \_\_init\_\_.py  
 ├── asgi.py  
 ├── settings.py  
 ├── urls.py  
 └── wsgi.py  
│  
└── myapp (Django app)  
 ├── \_\_init\_\_.py  
 ├── admin.py  
 ├── apps.py  
 ├── migrations  
 │ └── (migration files)  
 ├── models.py  
 ├── tests.py  
 └── views.py  
│  
└── templates (folder for HTML templates)  
 └── myapp  
│  
└── static  
 └── myapp  
 └── images

Django doesn’t automatically create templates and static folders when you start a new project or app. You typically create these folders based on your project's needs.

In a Django project, the static folder is used to store static files such as stylesheets (CSS), JavaScript files, images, and other files that are served directly to the client without processing by the server. These files remain constant and don't change dynamically based on user requests. The primary purpose of the static folder is to serve these static assets efficiently.

**Configuring Static Files**

**1. Static Files Configuration**

Open your settings.py file (located in the myproject directory) and update the STATICFILES\_DIRS:

# myproject/settings.py  
  
import os  
  
BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
  
STATIC\_URL = '/static/'  
STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, "static"),  
]

Now, your static files, including images, will be organized within the static directory at the base level.

**2. Adding Image Files**

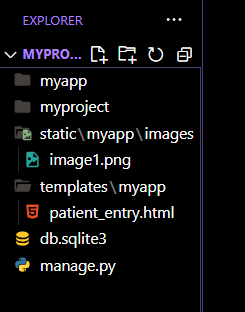
Place your image files inside the static/myapp/images directory. For example, if you have an image named example.jpg, the path would be static/myapp/images/example.jpg.

**Reference Images in Templates**

In your HTML templates, use the {% load static %} template tag to load static files, including images:

<!-- templates/myapp/patient\_entry.html -->  
  
{% load static %}  
  
<img src="{% static 'myapp/images/example.jpg' %}" alt="Example Image">

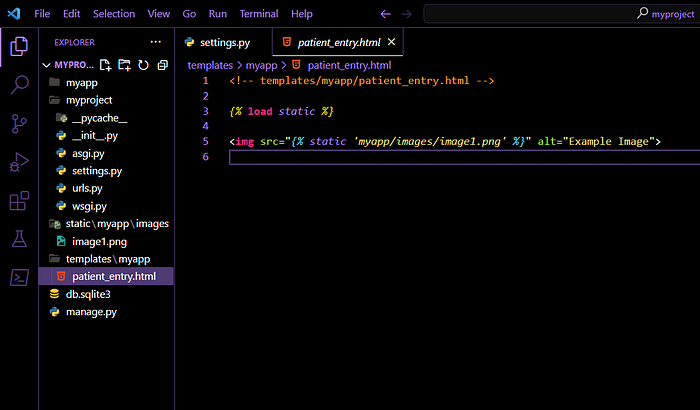
so let me show mine once :

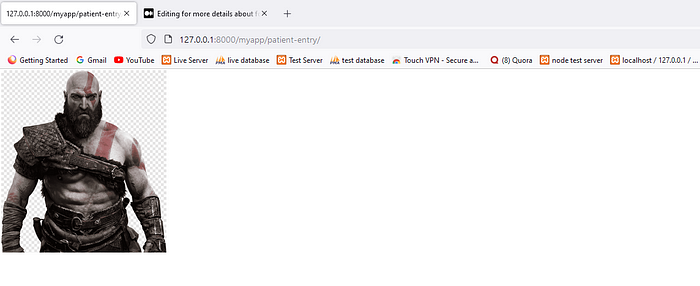


this is my folder structure



changes i made in settings.py



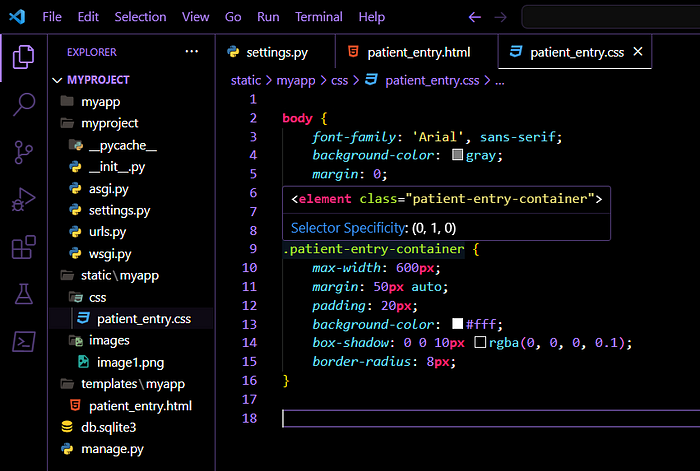


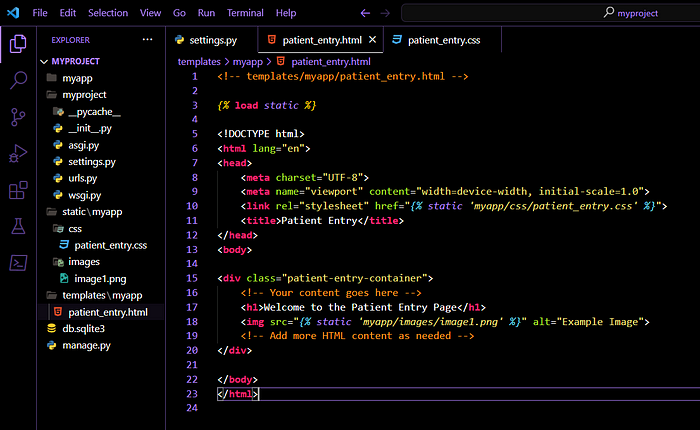
final output

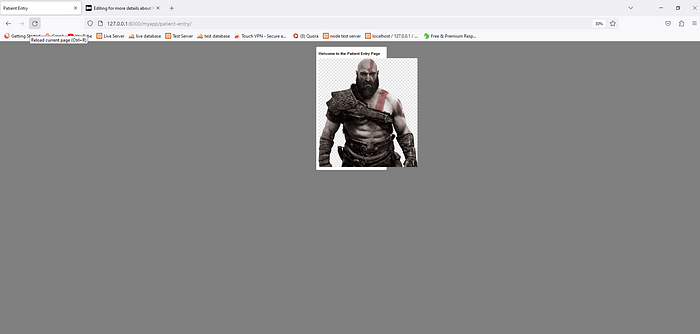
**We can also create a CSS file for our**patient\_entry.html**template and load it in the template**

Create a CSS file for the patient entry template. Assuming your CSS file is named patient\_entry.css, create it in the myapp/static/myapp/css/ directory:

body {  
 font-family: 'Arial', sans-serif;  
 background-color: gray;  
 margin: 0;  
 padding: 0;  
}  
  
.patient-entry-container {  
 max-width: 600px;  
 margin: 50px auto;  
 padding: 20px;  
 background-color: #fff;  
 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
 border-radius: 8px;  
}







output

If you’d like to include JavaScript in your Django project, you can create a static file for it.