

Topic: Project Setup, Modern Admin Panel Part 1

Speaker: Udemy Instructor Destiny | **Notebook:** Health Management System Using Django



1. Create a new folder and a new environment. Activate it.

```
$ python -m venv venv
$ source venv/scripts/activate
$ pip freeze
$ pip install django
```

2. Create a new project: in this new folder.

```
$ django-admin startproject HMS_main.
```

3. Run the server:

```
$ python manage.py runserver
```

<http://localhost:8000/>

4. Open now a VS Code with this folder or type:

```
$ code .
```

* we have to use `PIP INSTALL DJANGO` again in VS Code's terminal.

```

Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ python -m venv venv

Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ venv\scripts\activate
bash: venvscriptsactivate: command not found

Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ source venv/scripts/activate
(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ pip install django
Collecting django
  Downloading Django-5.1.2-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django)
  Using cached tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading Django-5.1.2-py3-none-any.whl (8.3 MB)
----- 8.3/8.3 MB 19.7 MB/s eta 0:00:00
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.1-py3-none-any.whl (44 kB)
Using cached tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.2 sqlparse-0.5.1 tzdata-2024.2
(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ pip freeze
asgiref==3.8.1
Django==5.1.2
sqlparse==0.5.1
tzdata==2024.2
(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ django admin startproject HMS_main .
bash: django: command not found
(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ django-admin startproject HMS_main .
(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ python manage.py runserver
Watching for file changes with StatReloader
[12/Oct/2024 19:34:46] "GET / HTTP/1.1" 200 12068
Performing system checks...

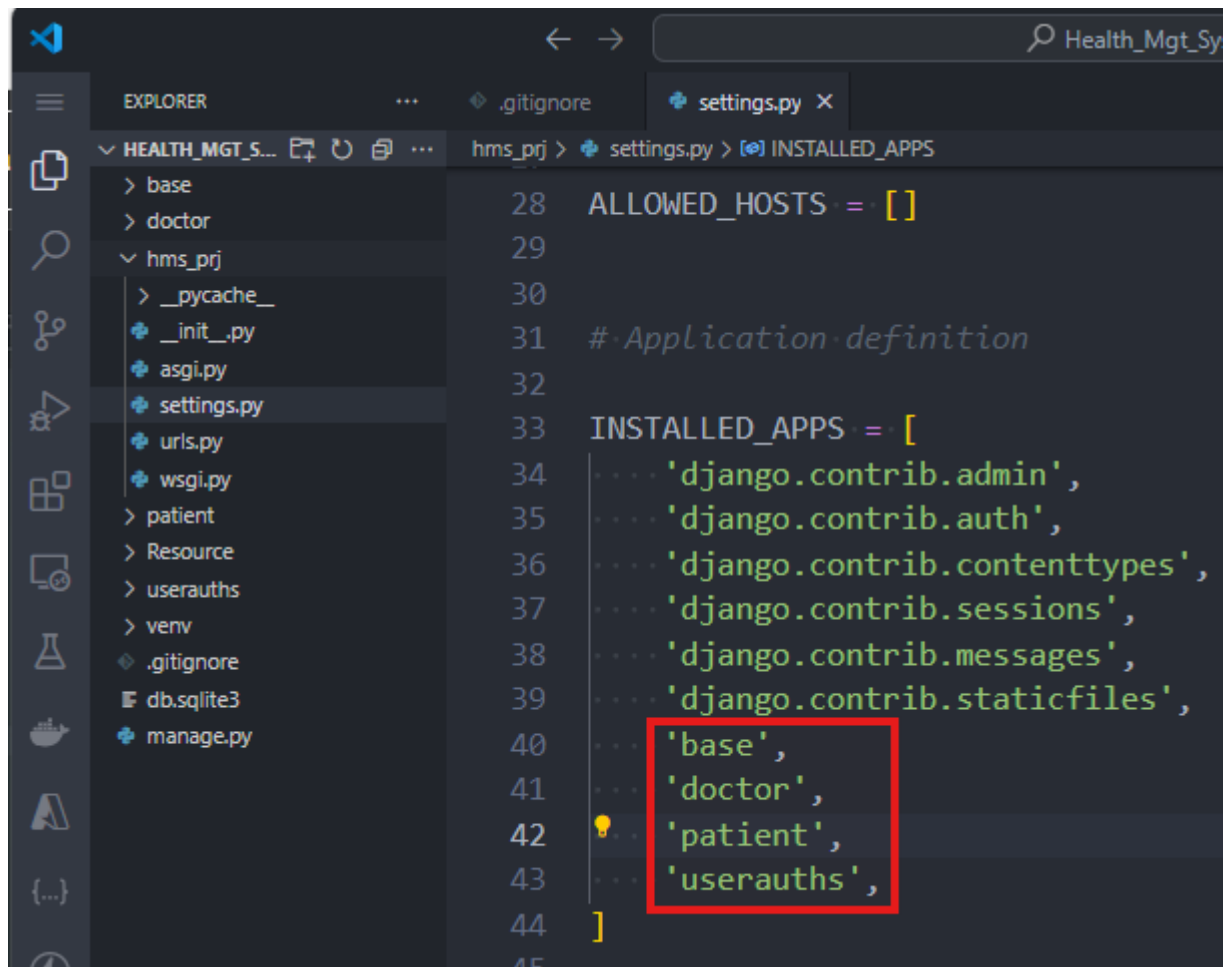
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you
  apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 12, 2024 - 19:33:35
Django version 5.1.2, using settings 'HMS_main.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

(venv)
Rosilie@DELL MINGW64 ~/OneDrive/Desktop/LEARNING DJANGO PROJECTS/Health_Mgt_System
$ code .|

```

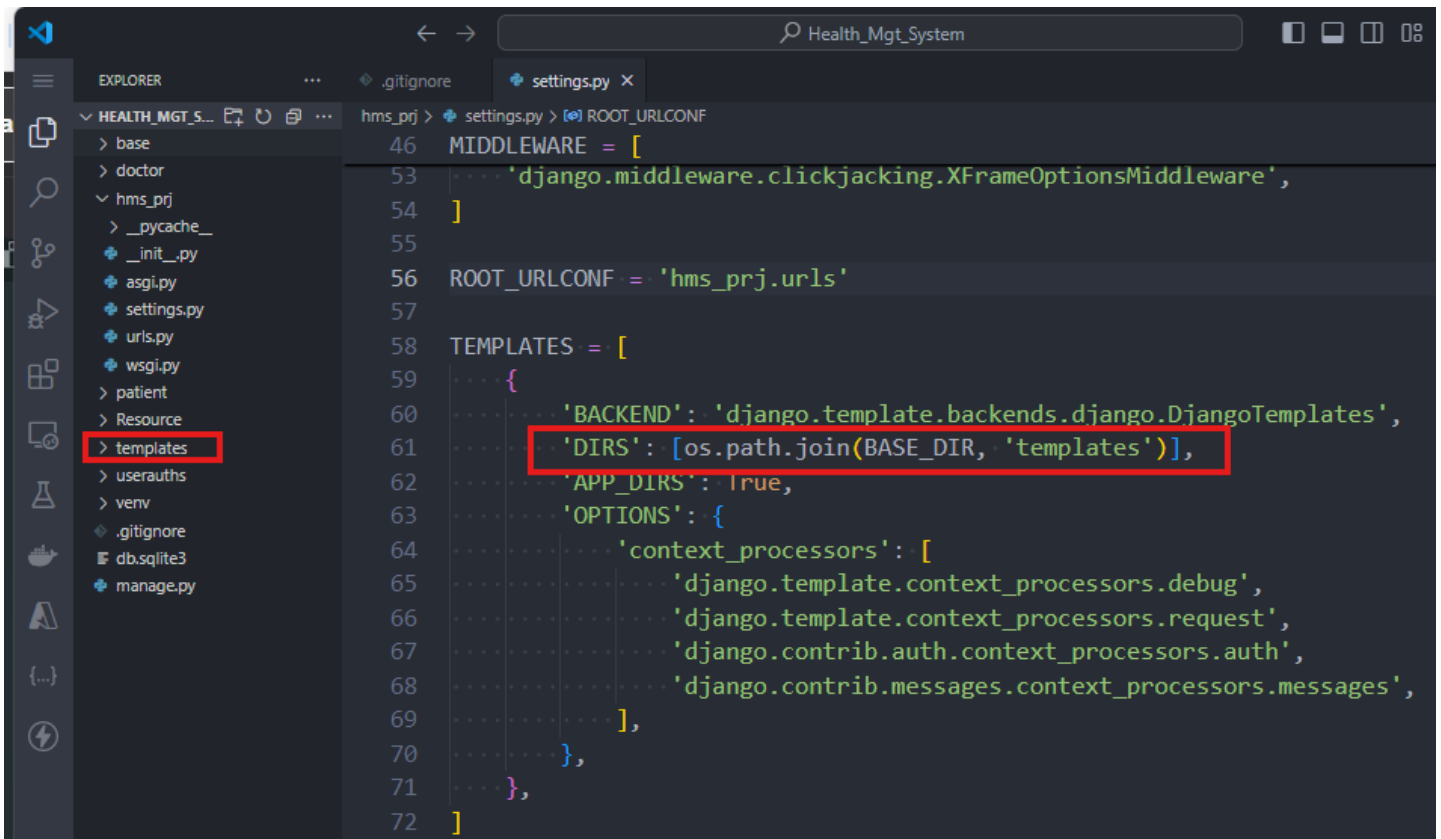
```
$ python manage.py startapp base  
$ python manage.py startapp doctor  
$ python manage.py startapp patient  
$ python manage.py startapp userauths
```



The screenshot shows the Visual Studio Code editor with a Django project named 'Health_Mgt_Sy'. The Explorer panel on the left shows the project structure, including folders for 'base', 'doctor', 'patient', 'Resource', 'userauths', and 'venv', as well as files like 'manage.py', 'db.sqlite3', and '.gitignore'. The main editor window displays the 'settings.py' file. The 'ALLOWED_HOSTS' variable is set to an empty list. The 'INSTALLED_APPS' variable is a list of Django app labels, including 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles', and a red box highlights the newly added apps: 'base', 'doctor', 'patient', and 'userauths'.

```
28 ALLOWED_HOSTS = []  
29  
30  
31 # Application definition  
32  
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'base',  
41     'doctor',  
42     'patient',  
43     'userauths',  
44 ]  
45
```

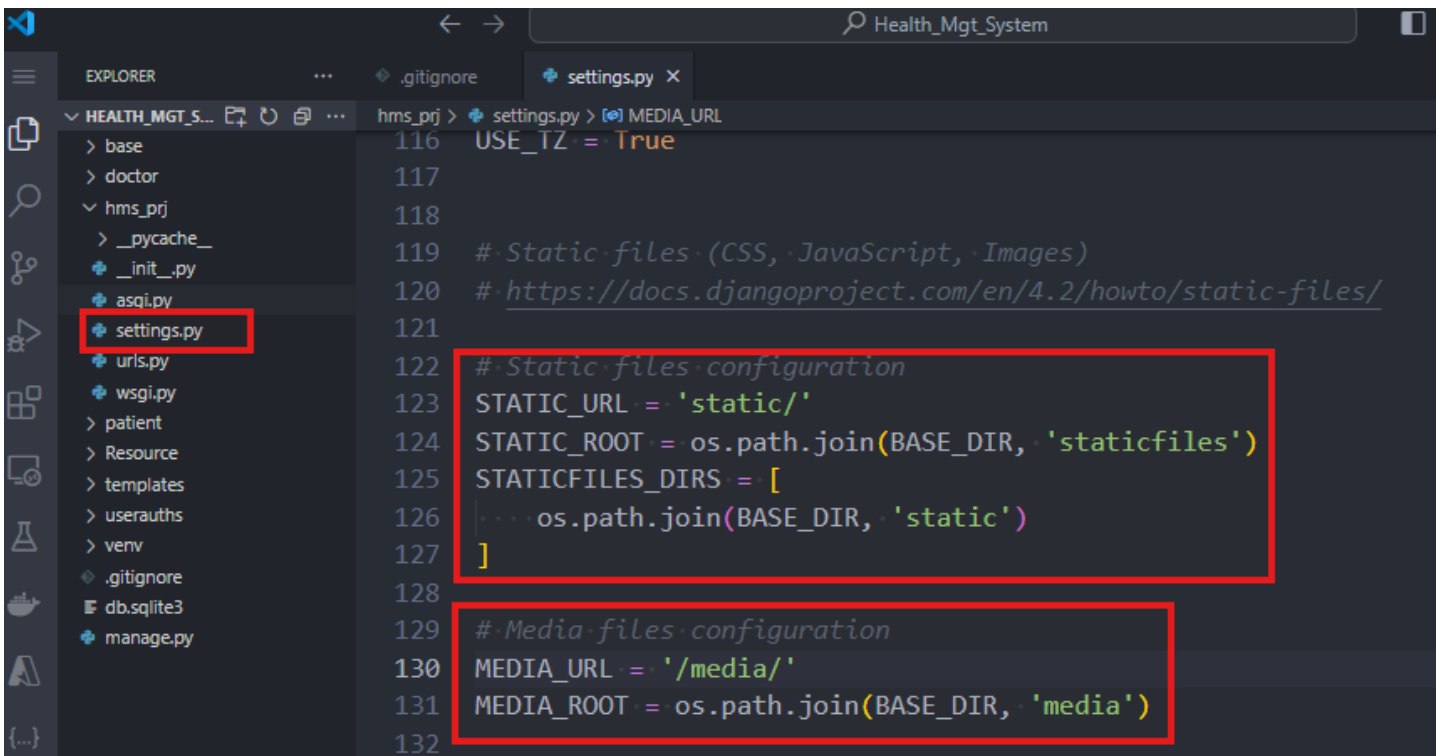
6. Create a new folder, TEMPLATES and update our SETTINGS.PY.



This screenshot shows the Django settings.py file in a code editor. The left sidebar displays the project structure with 'templates' highlighted under the 'Resource' directory. The main editor shows the following configuration:

```
46 MIDDLEWARE = [  
53     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
54 ]  
55  
56 ROOT_URLCONF = 'hms_prj.urls'  
57  
58 TEMPLATES = [  
59     {  
60         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
61         'DIRS': [os.path.join(BASE_DIR, 'templates')],  
62         'APP_DIRS': True,  
63         'OPTIONS': {  
64             'context_processors': [  
65                 'django.template.context_processors.debug',  
66                 'django.template.context_processors.request',  
67                 'django.contrib.auth.context_processors.auth',  
68                 'django.contrib.messages.context_processors.messages',  
69             ],  
70         },  
71     },  
72 ]
```

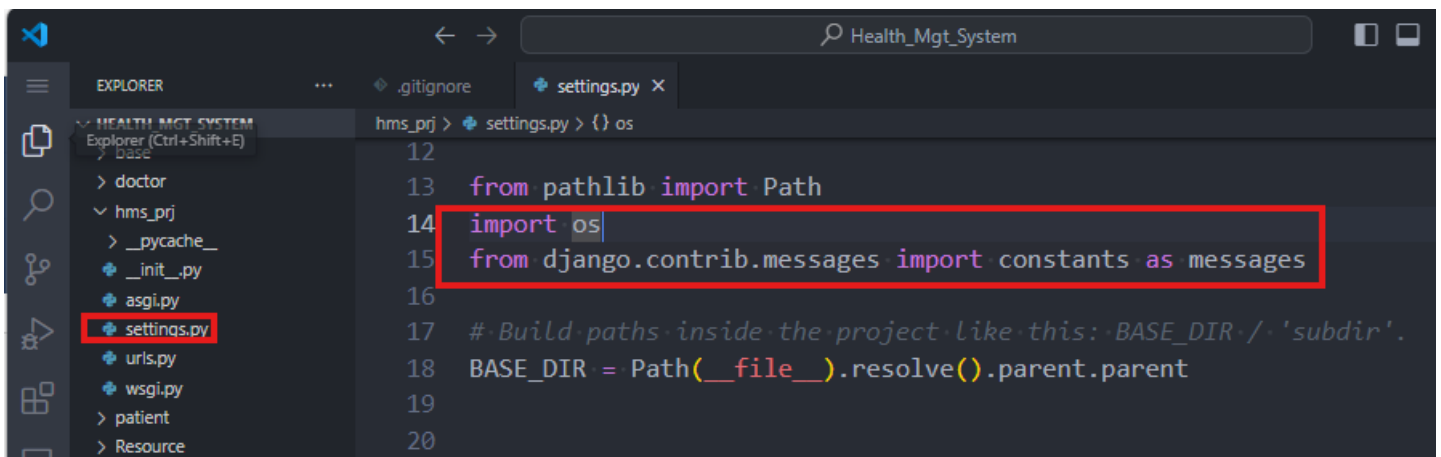
7. Set up the STATIC AND MEDIA configurations.



This screenshot shows the Django settings.py file with static and media configurations. The left sidebar highlights 'settings.py' under the 'Resource' directory. The main editor shows the following configuration:

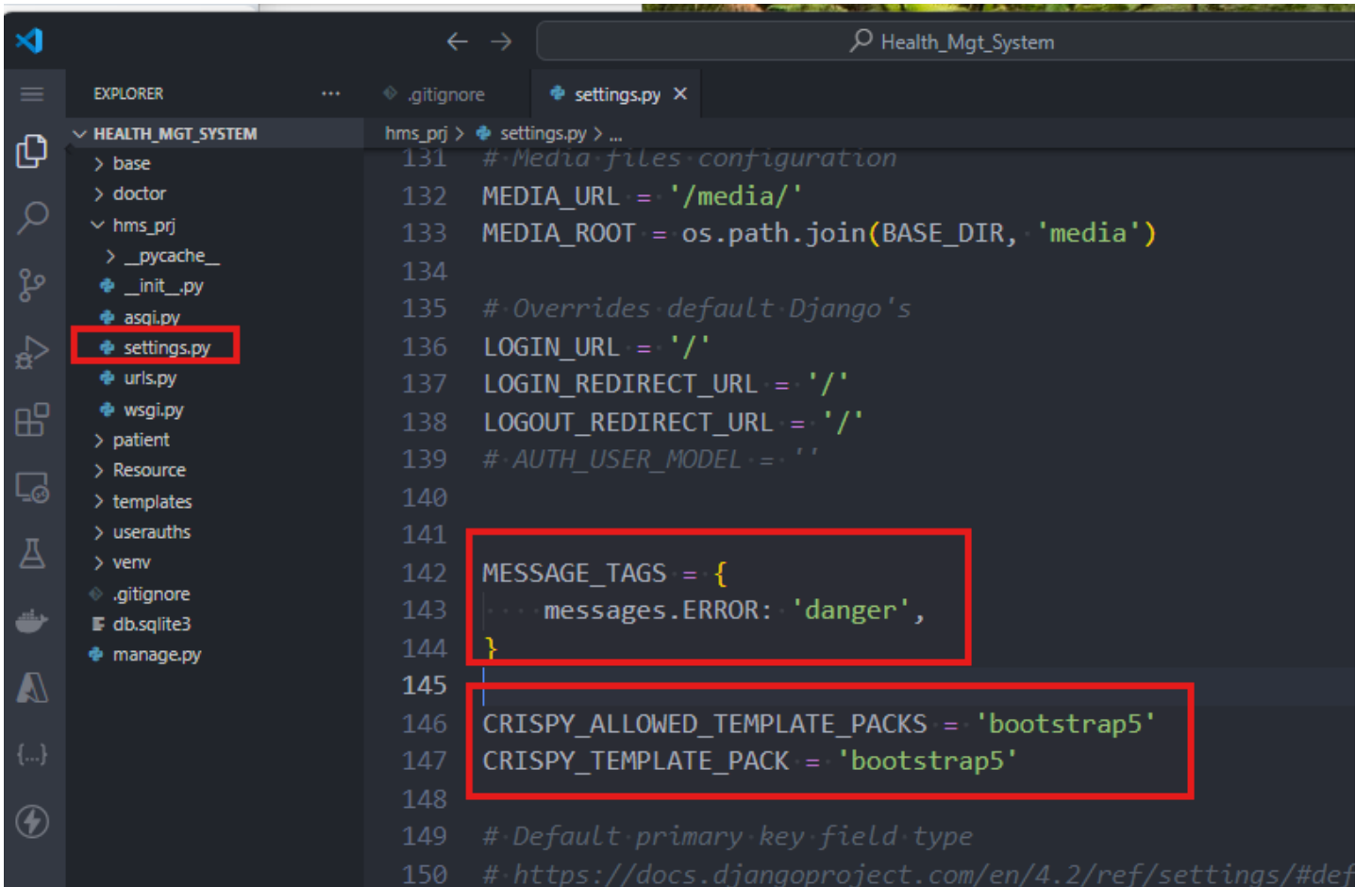
```
116 USE_TZ = True  
117  
118  
119 # Static files (CSS, JavaScript, Images)  
120 # https://docs.djangoproject.com/en/4.2/howto/static-files/  
121  
122 # Static files configuration  
123 STATIC_URL = 'static/'  
124 STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')  
125 STATICFILES_DIRS = [  
126     os.path.join(BASE_DIR, 'static')  
127 ]  
128  
129 # Media files configuration  
130 MEDIA_URL = '/media/'  
131 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
132
```

8. Update SETTINGS.PY to override default login/logout and user models:



```
12
13 from pathlib import Path
14 import os
15 from django.contrib.messages import constants as messages
16
17 # Build paths inside the project like this: BASE_DIR = Path(__file__).resolve().parent.parent
18 BASE_DIR = Path(__file__).resolve().parent.parent
19
20
```

9. For messages and forms configuration, update our SETTINGS.PY:



```
131 # Media files configuration
132 MEDIA_URL = '/media/'
133 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
134
135 # Overrides default Django's
136 LOGIN_URL = '/'
137 LOGIN_REDIRECT_URL = '/'
138 LOGOUT_REDIRECT_URL = '/'
139 # AUTH_USER_MODEL =
140
141
142 MESSAGE_TAGS = {
143     messages.ERROR: 'danger',
144 }
145
146 CRISPY_ALLOWED_TEMPLATE_PACKS = 'bootstrap5'
147 CRISPY_TEMPLATE_PACK = 'bootstrap5'
148
149 # Default primary key field type
150 # https://docs.djangoproject.com/en/4.2/ref/settings/#def
```

10. Update our project URLS.PY to include the path to our static and media files:

```
11 1. Add an import: from other_app.views import Home
12 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14 1. Import the include() function: from django.urls import include, path
15 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path
19 from django.conf import settings
20 from django.conf.urls.static import static
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24 ]
25
26 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
27 urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
28
```

11. To access our ADMIN PANEL, run the migrations to see the default model.

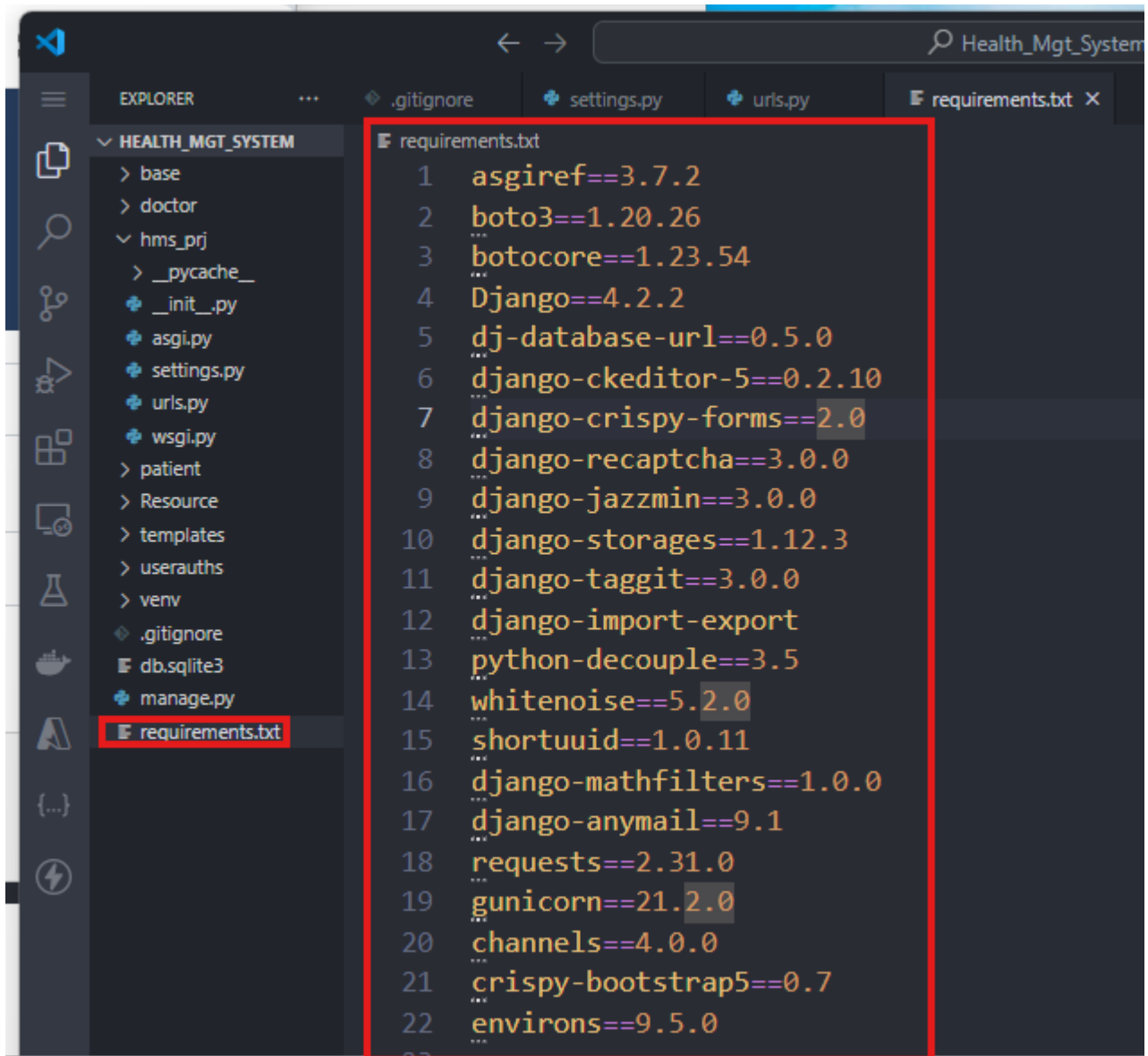
```
$ python manage.py migrate
```

12. Create your superuser

```
$ python manage.py createsuperuser
```

13. Install the packages from the RESOURCE folder. This will install the packages used by the instructor.

```
$ pip install -r requirements.txt
```



The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar displays a project structure for 'HEALTH_MGT_SYSTEM'. The files listed include 'base', 'doctor', 'hms_prj' (containing '__pycache__', '__init__.py', 'asgi.py', 'settings.py', 'urls.py', 'wsgi.py'), 'patient', 'Resource', 'templates', 'userauths', 'venv', and configuration files like '.gitignore', 'db.sqlite3', and 'manage.py'. The 'requirements.txt' file is highlighted in red in the sidebar. The main editor window shows the contents of 'requirements.txt', which is also highlighted with a red border. The file contains 22 lines of dependencies, with line 7, 'django-crispy-forms==2.0', highlighted with a grey background.

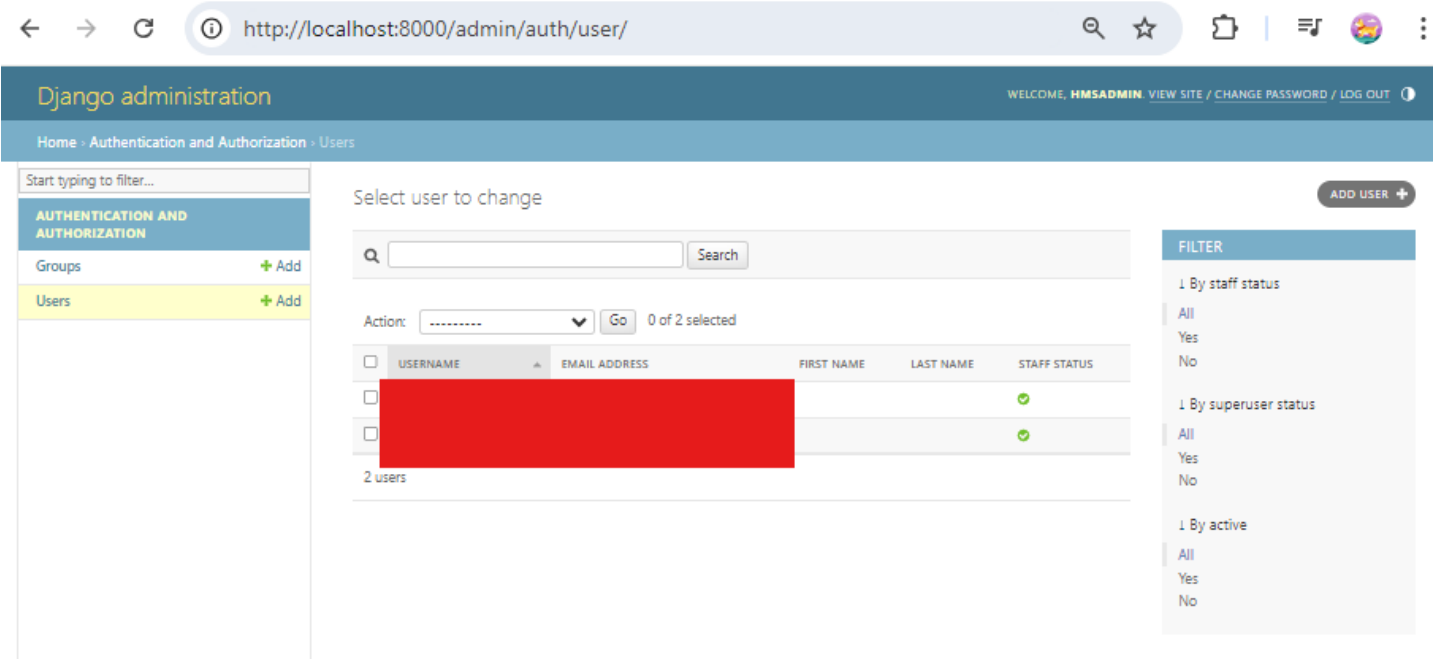
```
1 asgiref==3.7.2
2 boto3==1.20.26
3 botocore==1.23.54
4 Django==4.2.2
5 dj-database-url==0.5.0
6 django-ckeditor-5==0.2.10
7 django-crispy-forms==2.0
8 django-recaptcha==3.0.0
9 django-jazzmin==3.0.0
10 django-storages==1.12.3
11 django-taggit==3.0.0
12 django-import-export
13 python-decouple==3.5
14 whitenoise==5.2.0
15 shortuuid==1.0.11
16 django-mathfilters==1.0.0
17 django-anymail==9.1
18 requests==2.31.0
19 gunicorn==21.2.0
20 channels==4.0.0
21 crispy-bootstrap5==0.7
22 environs==9.5.0
```

14. Instead of using default DJANGO ADMIN DASHBOARD, we use the package [DJANGO JAZZMIN](#) for a more customizable and modern look.

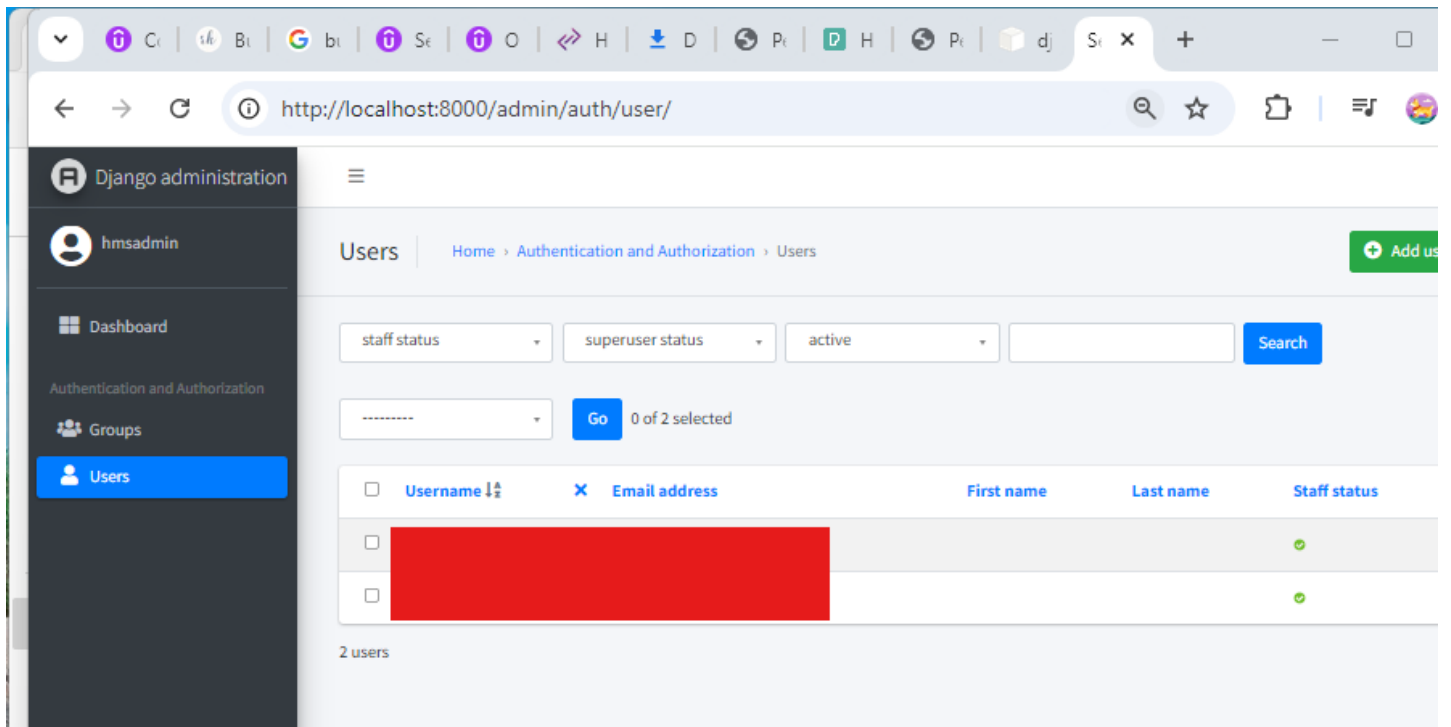
Include JAZZMIN in your INSTALLED_APPS in SETTINGS.PY

```
27 # SECURITY WARNING: don't run with debug turned on in production
28 DEBUG = True
29
30 ALLOWED_HOSTS = []
31
32
33 # Application definition
34
35 INSTALLED_APPS = [
36     'jazzmin',
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'base',
44     'doctor',
45     'patient',
46     'userauths',
47 ]
```

BEFORE JAZZMIN:



AFTER JAZZMIN:



15. To add a more customizable look to the admin panel, add in the `SETTINGS.PY` the `JAZZMIN_SETTINGS`. [Documentation here.](#)

```
154
155
156 # Jazzmin Configuration
JAZZMIN_SETTINGS = {
157     ... # title of the window (Will default to current_admin_site.site_title)
158     "site_title": "RNL HealthCare | Health Management System",
159     ... # Title on the login screen (19 chars max) (defaults to current_admin_site.site_header)
160     "site_header": "Health Management System",
161     ... # Title on the brand (19 chars max) (defaults to current_admin_site.site_header)
162     "site_brand": "RNL HealthCare | Health Management System",
163     ... # Welcome text on the login screen
164     "welcome_sign": "Welcome to the RNL HealthCare, Login Now",
165     ... # Copyright on the footer
166     "copyright": "RNL Web Solutions",
167     ... # List of apps (and/or models) to base side menu ordering off of
168     "order_with_respect_to": ["base", ],
169     ... # Custom icons for side menu apps/models See https://fontawesome.com
170     ... # for the full list of 5.13.0 free icon classes
171     "icons": {
172         "auth": "fas fa-users-cog",
173         "auth.user": "fas fa-user",
174         "auth.Group": "fas fa-users",
175     },
176     "show_ui_builder": True,
177 }
178
179 JAZZMIN_UI_TWEAKS = {
180     "navbar_small_text": False,
181     "footer_small_text": False,
```

16. Reload your page and in the admin dashboard, you will see the UI builder, you need to copy this code and add this as JAZZMIN_TWEAKS. COPY THIS CODE INTO YOUR SETTINGS.PY.

