WebApp with django backend for beginners

In this article we gonna learn how to link web application with Django framework and will gonna be using (Intellij Pycharm) for building our backend (Django). We in the first should get our website ready to link it with our backend (Django).

After getting things up and running (our website) we gonna install Django by opening a CMD (or Terminal if using Linux) and you should have installed pip (the package installer for Python) type the following command:

```
pip install django
```

After installing Django we now gonna create our Django project (make sure to point to the location that you want it to be) and type the following command:

```
django-admin startproject mycvsite
```

Now we have our project created successfully, know we gonna make some changes to our setting.py file in (DATABASES) section, first step we gonna add our MySQL configuration with the following script(make sure to put your own information):

```
DATABASES = {
    'default': {
        'ENGINE':'django.db.backends.mysql',
        'NAME':'here put your db name',
        'USER':'your username',
        'PASSWORD':'your password',
        'HOST':'the host (localhost for now)',
        'PORT':'if exist put it here',
    }
}
```

Next step create static folder (were our CSS/SCSS/JS) then, we gonna add STATIC and STATICFILES file path so we put the following script in the end of the file:

```
mkdir static
```

```
STATIC_URL = '/static/'
MEDIA_URL = '/images/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static')
]
```

Next we create app by typing the following script:

```
python3 manage.py startapp base
```

Add the app name to the setting.py file in the (INSTALLED_APPS) section by typing this script:

```
INSTALLED_APPS = [
    'base',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Now we need create two files inside each other first file is (templates) the second file is (base) and inside it we gonna add our Web App pages by typing the following command:

```
mkdir templates
  cd templates
    mkdir base
```

We gonna now add some functions in the views.py file that renders our templates, view it holds functions that takes requests and anything from (http response) to (template). Now we gonna add the following script:

```
from django.shortcuts import render

def home(request):
    return render(request, 'base/index.html')
```

Now we create file with the name of urls.py for our URLs routing system and insert the following command and script:

```
nano urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('', views.home)
]
```

And adding now our base app URLs to the main urls.py which gonna handle requests comes and decides where it should goes in our homepage add the following script to the file:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('base.urls')),
]
```

Before we start the last step we should make sure that our pages have in the root (in the top of the html page) the following script to make sure the page understand where files is stored in:

```
{% load static %}
```

And make sure to add it in every place that you called your files in, like the following script:

```
<li>
  <a href="https://github.com/iamabdulrazak">
    <img class="social" src="{% static 'images/github.png' %}"> GitHub
  </a>
</li>
```

Know we create superuser for our admin panel and will be ask you some questions, by typing the following command:

```
python3 manage.py createsuperuser
```

We are now have every thing set in the right way, now we migrate all the changes to the database so we can make sure everything in the right place.

Hint: Django builds for us by default admin panel and setting for us also WSGI file.

```
python3 manage.py migrate
```

Finally, now everything is good and in the right place so now we gonna check the application if it's working by running the server with the following command (check in the http://127.0.0.1:8000/):

```
python3 manage.py runserver
```