BUILDING A VIDEO STREAMING WEBSITE WITH DJANGO

REQUIRED TOOLS

- Python
- Django Documentation
- Code Editor
- Terminal like Git Bash or VScode

PROJECT STATGES:

ROOT FOLDER AND VIRTUAL ENVIRONMENT

The first thing we need to do is create a new folder, name is anything you like – **musicsteaming.** Then within the same folder, we need to install a virtual environment.

*Before installing Django, it's recommended to **install Virtualenv** that creates new isolated environments to isolates your Python files on a per-project basis. This will ensure that any changes made to your website won't affect other websites you're developing.*

In your terminal, switch to your **musicstreaming** folder and then install a virtual environment with the command:

```
pip install virtualenv
python -m venv virtualfoldernname
```

If you check the **musicstreaming** folder, you will see a folder named virtual or whatever you name it in the root folder.

In order to activate the virtual environment, use the command:

```
virtualfoldername/Scripts/activate
```

Once you do this, you'll see virtual above or beside your address in the command prompt. In order to turn it back off, simply use the command `deactivate.`

INSTALLING DJANGO AND CREATING A PROJECT

While you're still in the virtual environment, install Django with the command:

```
pip install django
```

You can use `pip freeze` to check the version of Django that was installed.

The next thing to do now is to start the project by creating a subfolder in your root directory. You can do this with the command:

```
django-admin startproject nameofproject
```

If you check your root folder, you will see a subfolder named nameofproject and has a file named **manage.py** and a folder named **nameofproject.**

Now we need to cd into the folder with the command in the terminal:

```
cd nameofproject
```

To view our project on the browser, we use the command:

```
python manage.py runserver
```

and you get the response:

```
Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

?[31m

You have 18 unapplied migration(s). Your project may not work properly until you apply the
migrations for app(s): admin, auth, contenttypes, sessions.?[0m

?[31mRun 'python manage.py migrate' to apply them.?[0m

July 21, 2021 - 20:24:24

Django version 3.2.5, using settings 'veedeeyo.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.
```

Press CTRL+C to break. Visit http://127.0.0.1:8000/ on your browser from the result you got on the terminal. You'll be welcomed with a Django landing page with several texts and "**The install worked successfully! Congratulations!**"

➢ MIGRATING DATABASE

The next thing to do after installing Django and creating the project folder is to migrate to Django database. Django has a database and admin area, so we have to migrate to it in order to connect our application.

We can do this with the command:

```
python manage.py migrate
```

We can then create our app name:

```
python manage.py startapp nameofapp
```

The **nameofapp** folder will be successfully created in the **nameofproject** directory.

If you **ls** in the Terminal, you'll have: **nameofapp,** a folder named **nameofproject, db.sqlite3** and **manage.py**

EDITING THE SETTINGS FILE IN SETTINGS.PY IN NAMEOFPROJECT FOLDER

In order to tell Django that we have a new app, we have to include the app name in the settings.py file inside the **nameofproject** directory.

Do this by opening settings.py in a Code Editor, then scroll till you see:

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

Add **nameofapp** in single quote on the next line and end it with a comma e.g 'nameofapp',

Next, we need a url for our application. One way to do this is by duplicating the **urls.py** file in the **nameofproject** directory and moving it to **nameofapp** directory. We will then edit the one in the **nameofapp** directory to:

```
from django.urls import path
```

```
urlpatterns = [

]
```

We need to let django know that we now have a new urls file in the app folder i.e '**nameofapp**', so we include the following code inside the original '**nameofproject**' urls file

```
            from django.contrib import admin

            from django.urls import path, include (the only thing you added on this line is 'include')


            urlpatterns = [

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

                  path('', include('nameofapp.urls')), (then you added this line too - '' means root)

            ]
```

Next

We need a **templates** directory within the **nameofapp** folder to hold our app files we will be adding manually. This is the folder that will hold our HTML files (incase we are building a webpage)

Create an index.html file, add an HTML tag e.g <h1>My Django Site</h1>. In order to be able to view this in a browser, we need a url for it; this is done by adding the page information to our **nameofapp** urls.py file:

```
from django.urls import path

from . import views     // . here means from this directory, import views


urlpatterns = [

  path('', views.index, name="index"),

]
```

Then we need to create a **view** for the index.html by editing the **views.py** file in the **nameofapp** directory.

```
from django.shortcuts import render


# Create your views here.


def index(request):
    return render (request, 'index.html', {})
```

ADDING FILES MANUALLY

First, you need to add your static command to the **settings.py** file in the **nameofproject** folder. Do this by scrolling down to the bottom of the **settings.py** page and add the following after the STATIC_URL line

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'
```

```
 # Added manually
STATICFILES_DIRS = [
    BASE_DIR / "static",
]
```

This helps tell Django how your file structure is eg c/users/learning/videostraming

BASE_DIR has already been declared at the beginning of the file – check between lines 1-20 of **settings.py**

Also, we need to create a **static** directory in order to put all our static files (all files that hardly change eg css, js, images etc but not index or any other html file). The **static** folder sits inside the **MAIN nameofproject** directory. Then within the

static directory, create another folder and name it **videosapp.** This is the folder that will hold our static files and folders. Then put your index.html file in your **templates** directory.

ADDING STATIC ADDRESS

{% load static %} – put this in your index.html or any html file right befor !<DOCTYPE>

{% static 'videosapp/**vendor/owl-carousel/owl.carousel.css'** %}"> - Edit all links to this **vendor/owl-carousel/owl.carousel.css'** is your custom address, i.e. where the css file is located. All https links remains same