**django**

**Environnement virtual**

**Python -m venv env**

**Activer l’environnement**

**.\env\scripts\activate**

**pip install -r requirements.txt**

**Installer django**

**Pip install django**

**Creer projet django**

django-admin startproject Nomduprojet

Pour voir tous le contenu du dossier = ls

**Rentrer dans le dossier du projet**

Cd + nom du projet

**Creer application**

python manage.py startapp nom de l’appli

Declarer l’application dans setting

Fichier setting dans la variable dans installed app

‘ Nom de l’app.apps. Nom de l’app+ config’,

Charger le statics

STATIC\_URL='static/'

STATICFILES\_DIRS=[

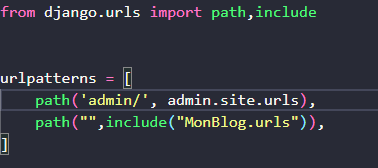
BASE\_DIR/"static"

]

Creer un fichier urls.py dans l’app

Copier le cotenue de urls du projet dans le nouveau fichier urls

Dans le dossier app, le fichier urls qu’on a creer supprimer la ligne import admin et le contenue urlspartens



Creez un dossier templates dans le dossier de l’app

Creer index.html

Declarer un variable “STATICFILES\_DIRS”

Creer une fonction dans views.py

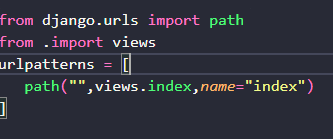
Def index(*request)*

Datas ={}

Return (request,index.html,datas)

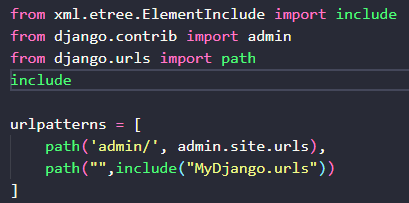
Entrez dans le fichier views

Dossier de l’app le fichier urls.py



Dossier du projet le fichier urls.py

Include



**Lancer** **le serveur**

python manage.py runserver

**Creer user admin**

python .\manage.py createsuperuser

**Faire la migration**

python .\manage.py makemigrations

python manage.py migrate

**repository Github**

**git config --global user.username "Cedric-Assemien"**

**git config --global user.email  "**[**emmanuel.assemien2001@gmail.com**](mailto:emmanuel.assemien2001@gmail.com)**"**

git init –b main

git add .

git commit -m "first commit"

git branch -M main

git remote add origin <https://github.com/Cedric-Assemien/Assemien-Projet-Django.git>

git push -u origin main

**flask python**

**environnement et lancer flask**

* set FLASK\_APP=server.py
* $env:FLASK\_APP = "server.py"
* flask run

Lancer test unitaire

python -m unittest Tests\Test\_unitaire\Test\_unit\_server.py -v

**-v ajoute plus de detail**

coverage run -m unittest

Locust

Temps de reaction du site

**coverage report -m**

**voir le pourcentage de test**

**tu est un expert en test unitaire avec le framework unittest . tu dois ecrire une fonction pour testé toutes les fonctions de ton code pour detecter tous les bugs et verifier tout le bon fonctionnement de toutes les fonction de ton application, si la fonction effectue ceux pourquoi elle as été créé. la premiere fonction est :**