TP3 Flask: Organiser votre projet

Rappel TP2 :

tpflask.py

```
from flask import Flask
from datetime import datetime
import requests
import json

app = Flask(__name__)

@app.route("/home")
def home():
    return render_template('home.html', a=somme)

@app.route("/about")
def about():
    return "about page"

def ri3valeur(L):
  L1=[]
  for i in range(0,len(L),3):
      L1.append(L[i])
  return L1
dateLyouma=datetime.today().strftime("%Y-%m-%d")
lyoumaName=datetime.today().strftime("%A")
jours={'Monday':'Lundi','Tuesday':'Mardi', 'Wednesday':'Mercredi',
'Thursday':'jeudi', 'Friday':'Vendredi', 'Saturday':'samedi','Sunday':'Dimanche'}
Lyouma=jours[lyoumaName]
url="https://api.open-meteo.com/v1/forecast?latitude=31,51&longitude=-
9,77&hourly=temperature_2m&hourly=windspeed_10m&hourly=cloud_cover&hourly=precipita
tion&start_date="+dateLyouma+"&end_date="+dateLyouma
response=requests.get(url)
response=requests.get(url).content.decode('utf-8')
data = json.loads(response)

daytemperature=data[0]["hourly"]["temperature_2m"]
windsliste=data[0]["hourly"]["windspeed_10m"]
cloudcover=data[0]["hourly"]["cloud_cover"]
precipitation=data[0]["hourly"]["precipitation"]

listetemperature=ri3valeur(daytemperature)
listewind=ri3valeur(windsliste)
listcloud=ri3valeur(cloudcover)
listprecipitation=ri3valeur(precipitation)
```

serveur.py

```
from tpflask import app
app.run(debug=True)
```

templates/home.html

```
{% extends "layout.html"%}
{% block content%}
    <h1>Meteo ESSAOUIRA</h1>
    <h1>{{Lyouma}}</h1>
    <div class="row">
    {% for i in range(nbval) %}
    <div class="col">
    <h3> {{cloud[i]}} </h3>
    </div>
    {%endfor%}
</div>
{% endblock content%}
```

templates/layout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Projet Flask</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body>
    {% block content%}
    {% endblock %}
</body>
</html>
```

Notre objectif: home.html

-créez un fichier getdata.py dans le répertoire du projet

-placer le code de récupération des donnée dans ce fichier

-modifier le non du fichier tpflask par routes.py

-ajouter `import getdata` dans le fichier routes.py

-Modifier l'URL de l'API pour plus de lisibilité.

```
url="https://api.open-meteo.com/v1/forecast?"
url=url+"latitude=31,51&longitude=-9,77"
url=url+"&hourly=temperature_2m"
url=url+"&hourly=windspeed_10m"
url=url+"&hourly=cloud_cover"
url=url+"&hourly=precipitation"
url=url+"&start_date="+dateLyouma
url=url+"&end_date="+dateLyouma
```

-dans le fichier getData.py créez une liste dataMeteo de huit dictionnaires, chaque dictionnaire comporte les informations à afficher :

```
[{"temps":"00","temperature":"16","precipitation":"0", "cloud":"0", "wind":"25-52",
"image":"mag1.jpg" },  ]
```

Dans ce cas `dataMeteo[0]` comporteras les mesures a afficher de la premier heure, `dataMeteo[1]` les mesures de la deuxième heure ..ect.

Pour les valeurs du dictionnaire il faut créer une fonction pour construire la liste des dictionnaires, il faut aussi créer une fonction pour le choix de l'image adéquat en fonction de l'heure (jour, nuit) de cloud cover et de la précipitation.

NB pour décider sur l'image de la lune ou de soleil vous pouvez récupérer l'heure du lever et du coucher du soleil de l'API avec l'ajout de `daily=sunrise` et `daily= sunset` dans l'URL.

Exemple de la fonction a développer dans les conditions il elif , il faut ajouter les autres conditions.

```
def getImages(Listecloud,dic):
   for c in Listecloud:
      if c<20:
         dic["image"]="sun.jpg"
      elif c<60:
         dic["image"]="sunetcloud.jpg"
      else :
         dic["image"]="cloud.jpg"
   return dic
```

Modifiez le return de la fonction home()

```
      return render_template('home.html', dataMeteo=dataMeteo, ....)
```

Améliorez la presentation du page home.html avec les classes CSS du Bootstrap.

Fin du TP et du projet API Météo