Master 1 RSD

Technologie des Applications Web

Interrogation Ecrite (03/12/2024)

Durée : 60 Minutes

Nom & Prénom	Gr.	Note
Corrigé Type		

Partie 1 : Questions à Choix Multiples (+ 1 : Juste ; -1 : Fausse ; 0 : Pas de réponse)

Cochez la bonne réponse !

1. Quelle est la définition du Web?

- ☐ Une application multimédia hors ligne
- ☒ Un système hypermédia permettant d'accéder aux ressources du réseau Internet
- ☐ Un logiciel permettant de surfer sur Internet
- ☐ Une plateforme dédiée au partage de vidéos

2. Qu'est-ce qu'une URL ?

- ☐ Un fichier stocké dans une base de données
- ☒ Une adresse précisant la localisation d'une ressource sur Internet
- ☐ Un code de programmation HTML
- ☐ Une méthode pour créer des graphiques dynamiques

3. Quelle est une caractéristique clé du Web 3.0 ?

- ☐ L'absence de JavaScript
- ☐ Une centralisation des données
- ☒ Une décentralisation et une domination de l'IA et de l'IoT
- ☐ Un retour aux pages statiques

4. Quels sont les éléments essentiels d'une requête HTTP ?

- ☐ Une adresse IP, une méthode, une liste d'en-têtes et une signature numérique
- ☒ Une URL, une méthode, une liste d'en-têtes, et un corps (facultatif).
- ☐ Une URL, une méthode, un code d'état, et une clé API.
- ☐ Une URL, un protocole de cryptage, et une liste de fichiers à télécharger.

5. Comment fonctionne un moteur de rendu dans un navigateur ?

- ☐ Il exécute les requêtes SQL pour afficher les données dans le navigateur.
- ☒ Il interprète le code HTML, CSS, et JavaScript pour afficher les pages Web et permettre les interactions.
- ☐ Il crypte les fichiers avant de les transmettre au backend pour validation.
- ☐ Il gère les connexions réseau pour transférer les fichiers entre les serveurs.

Partie 2 : Création d'une application Django bmi (Body Index Mass)

(Notation: 5 points)

Nous supposons que vous travaillez sous Windows 10 et que vous utilisez PyCharm Community Edition. Vous avez créé un projet PyCharm nommé body_mass_index, ce qui signifie qu'un environnement virtuel a été automatiquement généré par PyCharm dans le dossier du projet. L'objectif est de compléter les commandes et les fichiers nécessaires à la réalisation de l'application dans les endroits indiqués par une ligne discontinue dans la suite du document.

L'application devra répondre à des requêtes HTTP comme par exemple `http://127.0.0.1/bmi/70/172`, où `70` est le poids en kilogrammes et `172` est la taille en centimètres. Elle calculera le `bmi (Body Mass Index)` à l'aide de la formule suivante : $bmi = weight / (height^2)$. Le poids doit être en Kg et la taille en mètre.

L'application retournera les données (`weight, height)` et le résultat (`bmi`) dans un fichier `JSON` ainsi que la `category défnie par` :

category	condition
"Underweight"	bmi < 18.5
"Normal weight"	18.5 ≤ bmi < 25
"Overweight"	25 ≤ bmi < 30,
"Obesity"	bmi ≥ 30

1. **Installez Django :**

```
(.venv) PS D:\body_mass_index> pip install django
```
(0.50 pt)

2. **Créez le projet** `cfg` (configuration) :

```
(.venv) PS D:\body_mass_index> django-admin.exe startproject cfg .
```
(0.50 pt)

3. **Créez l'application** `bmi`

```
(.venv) PS D:\body_mass_index> py.exe .\manage.py startapp bmi
```
(0.50 pt)

ou bien

```
(.venv) PS D:\body_mass_index> django-admin.exe startapp bmi
```

4. Considérez `cfg/urls.py` pour complétez `bmi/urls.py`

```
# file cfg/urls.py
"""
URL configuration for cfg project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

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


# file bmi/urls.py
from django.urls import path

from . import views

urlpatterns = [

    path('bmi/<int:weight>/<int:height>', views.calculate_bmi, name='calculate_bmi'),      (0.50 pt)
]
```

5. Complétez le fichier `bmi/views.py`

```
# file bmi/views.py

from django.http import JsonResponse

def calculate_bmi(request, weight, height):

    # Convert height from cm to meters (0.50 pt)

    height = height / 100

    # bmi calculation  (0.50 pt)

    try:
        bmi = weight / (height ** 2)   # bmi = weight/(height* height)

    except ZeroDivisionError:

        return JsonResponse({"error": "Height cannot be zero."})

    # Interpretation of bmi (category value determination) (1.50 pt)

    if bmi < 18.5:
        category = "Underweight"
    elif 18.5 <= bmi < 25:
        category = "Normal weight"
    elif 25 <= bmi < 30:
        category = "Overweight"
    else:
        category = "Obesity"

    return JsonResponse({"weight": weight, "height": height, "bmi": bmi, "category": category})
```

```
# Les éléments qui pénalisent sont :
# ==================================
# - la non création du projet dans le dossier 'D:\body_mass_index'
# - L'utilisation des majuscules (noms/code). En plus, je l'ai précisé dans l'examen !
# - Le non-respect de l'indentation
# - les symboles bizarres  ==  ,  ^  , …
```