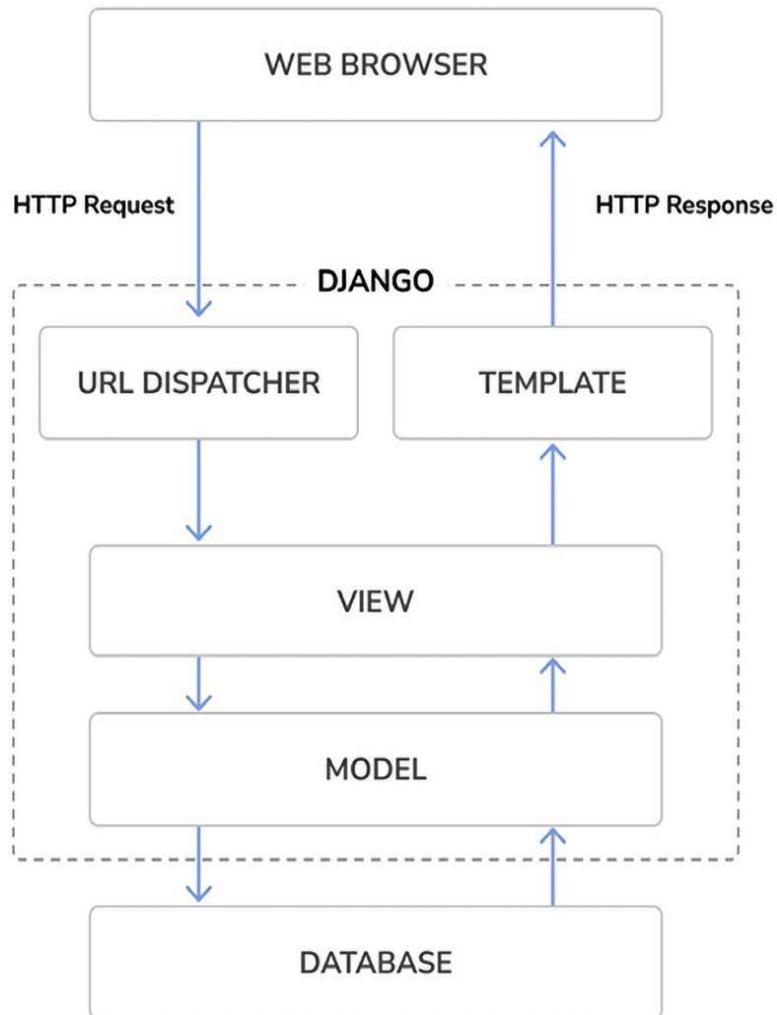


## Partie 1 : Questions de cours (10 points)

### 1. Étiquetage du schéma de l'architecture Django (4 points)



### 2. Cycle de traitement d'une requête HTTP par Django (6 points)

#### Étapes du traitement Django :

##### Le navigateur envoie une requête HTTP

→ L'utilisateur saisit une URL ; le navigateur envoie une requête au serveur Django.

##### Le routeur (URL-dispatcher) analyse l'URL

→ Django consulte les fichiers `urls.py` pour trouver la vue associée à l'URL demandée.

##### La vue (View) est appelée

→ Si l'URL correspond à une vue, Django exécute la fonction ou la classe correspondante.

##### La vue interagit avec le modèle (Model)

→ La vue interroge ou modifie les données via les modèles (ORM).

##### La vue appelle un template (si nécessaire)

→ Django génère une page HTML avec les données retournées par le modèle.

##### Django renvoie une réponse HTTP

→ Le contenu HTML est renvoyé au navigateur sous forme de réponse HTTP.

## Partie 2 : Projet Django - Application Sportify (10 points)

### 3. Fichier cfg/urls.py (1 point)

```
# cfg/urls.py
from django.contrib import admin
from django.urls import path, include

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

### 4. Fichier sportify/urls.py (2 points)

```
# sportify/urls.py
from django.urls import path
from .views import register

urlpatterns = [
    path('register/<str:tag>/<str:email>', register, name='register'),
]
```

5. Fichier sportify/models.py (7 points)

```
# sportify/models.py
from django.db import models

class Athlete(models.Model):
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    phone = models.CharField(max_length=15)

class Activity(models.Model):
    tag = models.CharField(max_length=50, unique=True)
    title = models.CharField(max_length=200)
    date = models.DateField()
    location = models.CharField(max_length=200)
    description = models.TextField()
    capacity = models.PositiveIntegerField()

class Registration(models.Model):
    athlete = models.ForeignKey(Athlete, on_delete=models.CASCADE)
    activity = models.ForeignKey(Activity, on_delete=models.CASCADE)
    reg_date = models.DateField()
```

**Attention à ces erreurs !**

**sc différent de x  
t différent de T  
Défaut d'indentation => -1**

**+2 Points pour tout le monde !**