

| | | | |
|---|-------------------------|------------|-------------|
| Master 1 RSD - Technologie des Applications Web Examen (Copie Principale) 20 Janvier 2025 Durée : 60 Minutes | Nom & Prénom | Gr. | Note |
| | | . . . | |

Partie 1 : QCM, cochez la bonne réponse ! (+ 1 : Juste ; -1 : Fausse ; 0 : Pas de réponse)

- 1. Quelle est la définition exacte du Web ?**

 - Un protocole pour transférer des données sur Internet.
 - Un système hypermédia permettant d'accéder aux ressources du réseau Internet.
 - Un langage utilisé pour structurer des pages Web.
 - Une application dédiée au partage de fichiers.
- 2. Quelle caractéristique distingue une ressource JSON d'une ressource XML ?**

 - JSON est orienté documents, tandis que XML est orienté données.
 - XML est plus rapide pour échanger des données.
 - JSON est plus léger et orienté données.
 - JSON ne supporte pas les hiérarchies complexes.
- 3. Que signifie 'batteries included' dans le contexte de Django ?**

 - L'intégration d'outils tiers préinstallés.
 - La disponibilité de nombreuses fonctionnalités prêtes à l'emploi.
 - La possibilité de personnaliser tous les composants.
 - La compatibilité avec tous les frameworks front end.
- 4. Dans Django, quel fichier est utilisé pour configurer la base de données ?**

 - views.py
 - models.py
 - settings.py
 - urls.py
- 5. Dans une requête HTTP POST, où se trouvent les données envoyées au serveur ?**

 - Dans l'en-tête de la requête.
 - Dans le corps de la requête.
 - Dans l'URL de la requête.
 - Les données ne sont pas envoyées au serveur.
- 6. Que fait la commande python manage.py makemigrations ?**

 - Crée un modèle de base de données.
 - Applique les migrations à la base de données.
 - Génère les fichiers de migration pour les modifications des modèles.
 - Supprime toutes les migrations précédentes.
- 7. Quel est le rôle de csrf_token (Cross-Site Request Forgery Token) dans les formulaires Django ?**

 - Empêcher les soumissions multiples d'un formulaire.
 - Assurer que le formulaire est envoyé depuis une source de confiance.
 - Valider automatiquement les données envoyées dans le formulaire.
 - Authentifier les utilisateurs envoyant le formulaire.
- 8. Comment est-ce que Django organise les fichiers statiques comme les images et les fichiers CSS ?**

 - En utilisant le dossier /templates.
 - En les plaçant directement dans le dossier /views.
 - En utilisant un dossier nommé /static dans chaque application.
 - En stockant tous les fichiers dans /settings.

9. **Comment activer l'administration Django pour un modèle ?**

- Ajouter le modèle dans `views.py`.
- L'enregistrer dans `admin.py` avec `admin.site.register()`.
- Le configurer dans `urls.py`.
- Ajouter un décorateur `@admin.register()` dans `models.py`.

10. **Comment définir une relation un-à-plusieurs dans Django ?**

- Utiliser un champ `ForeignKey`.
- Utiliser un champ `ManyToManyField`.
- Utiliser un champ `OneToOneField`.
- Définir manuellement les relations dans la base de données.

Partie 2 : Utilisation du Framework Django 5

1. Créez le projet `cfg` dans le dossier de l'environnement virtuel Python (0,5 pt)

```
(venv) PS E:\cultura> django-admin startproject cfg .
```

2. Créez l'application `cultura` (0,5 pt)

```
(venv) PS E:\cultura> django-admin startapp cultura
```

3. Complétez le fichier `cfg/urls.py` pour inclure le fichier `urls.py` de `cultura` dans le projet (0,5 pt)

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('cultura/', include('cultura.urls')),
]
```

4. Complétez le fichier `cultura/urls.py` pour inclure la vue `book` (0,5 pt)

```
# Format: http://127.0.0.1/cultura/book/tag/email/seats (http://127.0.0.1/cultura/book/eac25/lamis.f@mail.com/2)
from django.urls import path
from .views import book_event
urlpatterns = [
    path("book/<str:tag>/<str:email>/<int:seats>/", book_event, name="book"),
]
```

5. Complétez les modèles de l'application (0,5 pt x 13 = 6,5 pts)

```
# culura/models.py

from django.db import models

class Participant(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 Event(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()

    seats_number = models.PositiveIntegerField()

class Booking(models.Model):

    user = models.ForeignKey(Participant, on_delete=models.CASCADE)

    event = models.ForeignKey(Event, on_delete=models.CASCADE)

    seats = models.PositiveIntegerField()
```

S'apercevoir que User n'existe pas !

6. Complétez la vue `book_event` qui permet la réservation (0,5 pt x 3 = 1,5 pts)

```
from django.http import JsonResponse
from .models import Participant, Event, Booking

def book_event(request, tag, email, seats):
    try:
        # Retrieve the event
        event = Event.objects.get(tag=tag)
        # Retrieve the participant
        participant = Participant.objects.get(email=email)
        # Create a booking using the custom Manager
        booking = Booking.objects.create(participant=participant, event=event, seats=seats)
        # Return a success message
        return JsonResponse({
            'status': 'success',
            'message': f'Booking successful for {seats} seats at the event {event.title}.'})
    except Exception as e:
        # Handle any unexpected exceptions and return a generic error
        return JsonResponse({
            'status': 'error',
            'message': f'Error while booking: {str(e)}'})
```